

/ANALYSIS OF MODEL REFERENCED ADAPTIVE
CONTROL APPLIED TO ROBOTIC DEVICES/

by

DAVID JAMES McCONNELL

B. S., Kansas State University, 1982

A THESIS

submitted in partial fulfillment of the

requirements of the degree

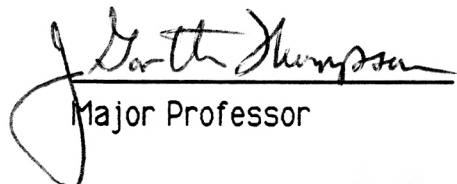
MASTER OF SCIENCE

Department of Mechanical Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1985

Approved by:


Major Professor

LD
2668
.T4
1985
M336

A11202 645027

TABLE OF CONTENTS

Chapter		Page
	INTRODUCTION	1
I	DYNAMIC MODEL	4
	Development of Dynamic Equations	4
	State Variable Equation Development	14
	Calculation of Moments of Inertia of Links	19
II	MODEL REFERENCE ADAPTIVE CONTROL DEVELOPMENT	28
	Proportional Integral Control with Derivative Feedback	28
	PID Control	32
	Proportional Control with Derivative Feedback	52
	Three Axis Control	59
III	OBSERVATIONS AND CONCLUSIONS	63
	BIBLIOGRAPHY	71
	APPENDIX	72

LIST OF TABLES

Table		Page
1	Physical Parameters of Robot Links	22
2	a And b Values of Robot Links	23
3	Link Parameters for Calculating Moments of Inertia	27
4	Standard Test Move Parameters	42

LIST OF FIGURES

Figure		Page
1	Simplified Model of Robot	5
2	Top and Side View of Model	6
3	Model of Robot Link	19
4	Coordinate Systems and Centers of Mass	24
5	Proportional Integral Control with Derivative Feedback	29
6	Modified Control with Second Derivative Feedback Added	32
7	PID Control	32
8	Control System Schematic	39
9	Ideal System Response vs Model Response	45
10	System Response vs Model Response with Increased q Values	46
11	System Gain Adjustments with Increased q Values	46
12	System Response vs Model Response using Robot Dynamic Equations	48
13	System Gain Adjustments using Robot Dynamic Equations	49
14	Proportional Control with Derivative Feedback	51

Figure		Page
15	Ideal System Response vs Model Response	54
16	System Gain Adjustments with Ideal System	55
17	Ideal System Response vs Model Response	56
18	System Gain Adjustments using Ideal System	56
19	System Response vs Model Response using Robot Dynamic Equations	57
20	System Gain Adjustments using Robot Dynamic Equations	57
21	System Response with Fixed Gains on All Axes	60
22	System Response with Model Referenced Adaptive Control on All Axes	60
23	Torso Response with Model Referenced Adaptive Control on Torso Axis and Fixed Gains on All Other Axes	61

INTRODUCTION

The type of control utilized by a robotic manipulator is vital to its performance and usefulness. With proper control, a robot is capable of performing intricate tasks with great accuracy and precision. Without good control, the robot is only able to perform with marginal accuracy and little dependability.

In order for a control system to perform well, it must be able to deal with the changing configurations and loads that the robot experiences. The control system must be able to "adapt" itself to every situation the robot encounters. Without adaptability only under certain conditions will the robot be able to perform adequately. With adaptability the robot can perform well over the entire range of its motion and load carrying capabilities.

This paper investigates a model referenced adaptive control technique developed by Donalson and Leondes [1] and applies it to an International Robomation Intelligence (IRI) Model M-50 robotic manipulator. Dubowsky and DesForges [2] have previously made application of model referenced

adaptive control to robotic manipulators and the work of this paper follows their work on the subject.

Model referenced adaptive control is a method of adjusting the gains of a control system so that a desired response is achieved, regardless of the changes that occur in the physical system being controlled. These changes in the robot are due mainly to inertial properties. By selecting an index of performance, which is a function of an error signal and its derivatives, model referenced adaptive control adjusts the gains of the control system so as to move the index of performance toward zero at a rate determined by a steepest descent trajectory.

The analysis of the model referenced adaptive control is done by means of a computer simulation. Computer simulations allow for numerous tests to be carried out, without ever physically implementing the control system. As a preliminary study, the simulation provides the control system designer with data concerning the system response, ranges of gain adjustment, gain adjustment rates, and how well different control configurations work with the adaptive algorithm.

The simulation consists of 3 basic parts; the mathematical model of the robot, the model of the control system, and the adaptive algorithm. The

mathematical model of the robot consists of the dynamic equations describing the motion of the robot and determining the physical properties of the robot links. The model of the control system consists of the equations that emulate the control, and the adaptive algorithm consists of the equations derived for adjusting the gains of the control system.

Chapter 1 gives the derivation of the dynamic equations of the robot and the calculation of the physical properties of the links. Chapter 2 investigates the use of model referenced adaptive control with various control systems, and discusses their performance. And Chapter 3 contains some observations and conclusions about model reference adaptive control.

I. DYNAMIC MODEL

In this Chapter, the mathematical model describing the robot is developed. The mathematical model is used by the computer to determine the displacement, velocity, and acceleration of the links of the robot, given the torque applied at each joint.

The equations describing the IRI robot are nonlinear, highly coupled equations. In order to simplify their derivation and solution, the following assumptions are made.

1. The wrist roll and wrist pitch axes of the robot are locked.
2. The links of the robot are symmetric with respect to their principle axes of inertia.
3. The links are made of solid, homogenous material and there are no concentrated masses within the links.

Assumption 1 reduces the robot from 5 degrees of freedom to 3 degrees of freedom and allows the wrist and gripper to be considered part of the forearm. Assumptions 2 and 3 simplify the calculation of the moments of inertia of the links. Applying these assumptions results in the

three degree of freedom model of the robot shown in figure 1.

To develop the dynamic equations for the model shown in figure 1, Lagrange's Equation will be used. Defining K as the kinetic energy of the system, P as the potential energy of the system, and L as the difference between K and P , then the torque at joint i of the model, T_i , is given as:

$$1.1 \quad T_i = \frac{d}{dt} \left[\frac{\partial L}{\partial \dot{\theta}_i} \right] - \frac{\partial L}{\partial \theta_i} \quad \text{for } i = 1, 2, \text{ or } 3$$

where $L = K - P$

θ_i = angular displacement of link

and $\dot{\theta}_i$ = angular velocity of link i

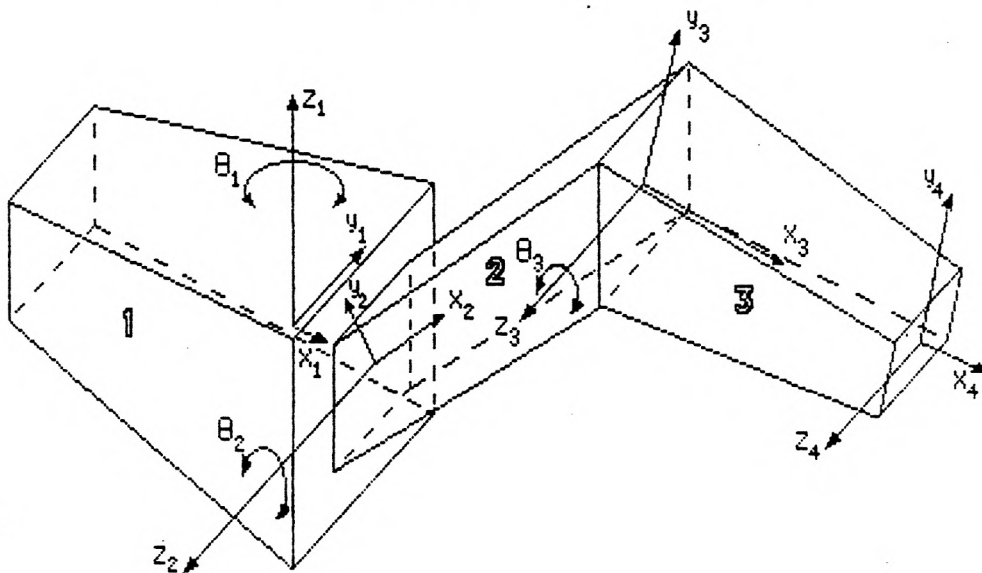


FIGURE 1. Simplified Model of Robot

The following definitions are made for the derivation of the kinetic and potential energy of the system (see Figure 2).

1. $\{X_0, Y_0, Z_0\}$ is the base coordinate system. Base Coordinates will be in upper case.
2. $\{x_i, y_i, z_i\}$ is the link coordinate system of link i .
3. θ_i is the angular displacement of link i about the z_i axis.
4. R_i is the vector to the center of mass of link i with respect to the base coordinate system.

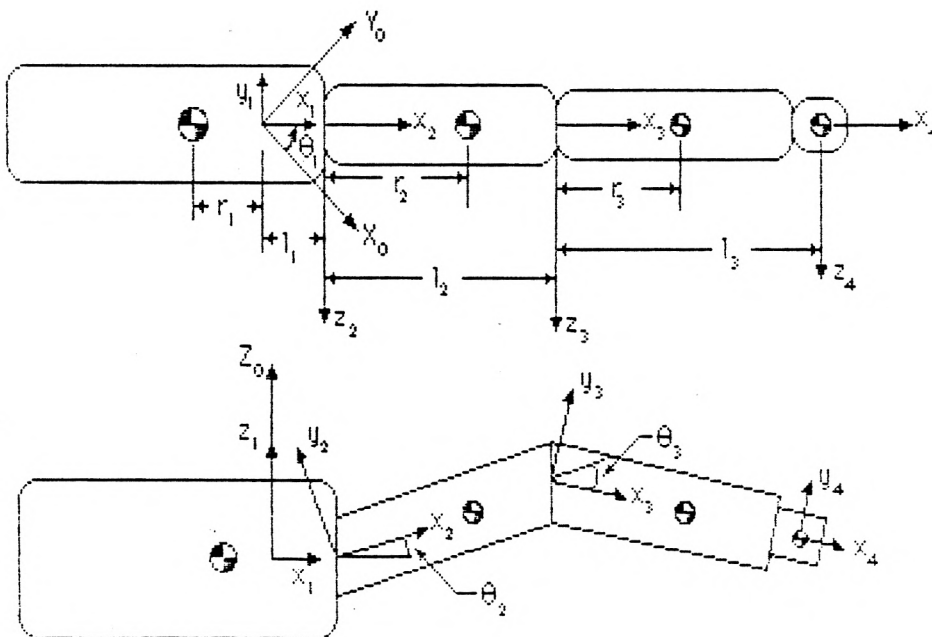


FIGURE 2. Top and Side View of Model

5. ω_i is the angular velocity vector of link i with respect to the center of mass of link i .
6. m_i is the mass of link i in slugs.
7. I_i is the inertial matrix of link i and is defined as:

$$I_i = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \quad \text{where } \begin{aligned} I_{xx} &= \int (y_i^2 + z_i^2) dm & I_{xy} &= \int x_i y_i dm \\ I_{yy} &= \int (x_i^2 + z_i^2) dm & I_{xz} &= \int x_i z_i dm \\ I_{zz} &= \int (x_i^2 + y_i^2) dm & I_{yz} &= \int y_i z_i dm \end{aligned}$$

8. l_1 is the distance from the origin of link 1 to the origin of link 2.
9. l_2 is the distance from the origin of link 2 to the origin of link 3.
10. l_3 is the distance from the origin of link 3 to the centroid of the payload located at the origin of coordinate system 4.
11. r_i is the distance to the centroid of link i with respect to the i th coordinate system. (The centroid of link i is assumed to lie on the x_i axis.)
12. \mathbf{x}_i , \mathbf{y}_i , and \mathbf{z}_i are unit vectors in the x_i , y_i , and z_i directions.

The above definitions will now be used to derive the terms necessary

for the Lagrange Equation. Given $\dot{\mathbf{R}}_i$, $\boldsymbol{\omega}_i$, m_i , and \mathbf{I}_i , the total kinetic energy of the manipulator is defined as:

$$1.2 \quad K = \frac{1}{2} m_1(\dot{\mathbf{R}}_1 \cdot \dot{\mathbf{R}}_1) + \frac{1}{2} m_2(\dot{\mathbf{R}}_2 \cdot \dot{\mathbf{R}}_2) + \frac{1}{2} m_3(\dot{\mathbf{R}}_3 \cdot \dot{\mathbf{R}}_3) \\ + \frac{1}{2} m_4(\dot{\mathbf{R}}_4 \cdot \dot{\mathbf{R}}_4) + \frac{1}{2} \boldsymbol{\omega}_1^t \cdot \mathbf{I}_1 \cdot \boldsymbol{\omega}_1 + \frac{1}{2} \boldsymbol{\omega}_2^t \cdot \mathbf{I}_2 \cdot \boldsymbol{\omega}_2 \\ + \frac{1}{2} \boldsymbol{\omega}_3^t \cdot \mathbf{I}_3 \cdot \boldsymbol{\omega}_3 + \frac{1}{2} \boldsymbol{\omega}_4^t \cdot \mathbf{I}_4 \cdot \boldsymbol{\omega}_4$$

The total potential energy of the manipulator is given as:

$$1.3 \quad P = m_1 g \mathbf{R}_{1Z_0} + m_2 g \mathbf{R}_{2Z_0} + m_3 g \mathbf{R}_{3Z_0} + m_4 g \mathbf{R}_{4Z_0}$$

where \mathbf{R}_{iZ_0} is the component of the \mathbf{R}_i vector in the Z_0 direction.

Determination of the \mathbf{R}_i 's

$$1.4 \quad \mathbf{R}_1 = r_1 \mathbf{x}_1 = r_1 \cos(\theta_1) \mathbf{x}_0 + r_1 \sin(\theta_1) \mathbf{y}_0 + 0 \mathbf{z}_0$$

$$1.5 \quad \mathbf{R}_2 = l_1 \mathbf{x}_1 + r_2 \mathbf{x}_2 \\ = [l_1 \cos(\theta_1) + r_2 \cos(\theta_1) \cos(\theta_2)] \mathbf{x}_0 + \\ [l_1 \sin(\theta_1) + r_2 \sin(\theta_1) \cos(\theta_2)] \mathbf{y}_0 + [r_2 \sin(\theta_2)] \mathbf{z}_0$$

$$1.6 \quad \mathbf{R}_3 = l_1 \mathbf{x}_1 + l_2 \mathbf{x}_2 + r_3 \mathbf{x}_3 \\ = [l_1 \cos(\theta_1) + l_2 \cos(\theta_1) \cos(\theta_2) + r_3 \cos(\theta_1) \cos(\theta_2 + \theta_3)] \mathbf{x}_0 \\ + [l_1 \sin(\theta_1) + l_2 \sin(\theta_1) \cos(\theta_2) + r_3 \sin(\theta_1) \cos(\theta_2 + \theta_3)] \mathbf{y}_0 \\ + [l_2 \sin(\theta_2) + r_3 \sin(\theta_2 + \theta_3)] \mathbf{z}_0$$

$$\begin{aligned}
1.7 \quad \mathbf{R}_4 &= l_1 \mathbf{x}_1 + l_2 \mathbf{x}_2 + l_3 \mathbf{x}_3 \\
&= [l_1 \cos(\theta_1) + l_2 \cos(\theta_1) \cos(\theta_2) + l_3 \cos(\theta_1) \cos(\theta_2 + \theta_3)] \mathbf{x}_0 \\
&+ [l_1 \sin(\theta_1) + l_2 \sin(\theta_1) \cos(\theta_2) + l_3 \sin(\theta_1) \cos(\theta_2 + \theta_3)] \mathbf{y}_0 \\
&+ [l_2 \sin(\theta_2) + l_3 \sin(\theta_2 + \theta_3)] \mathbf{z}_0
\end{aligned}$$

Taking the derivative with respect to time of each of the \mathbf{R} vectors gives:

$$1.8 \quad \dot{\mathbf{R}}_1 = [(-r_1 \sin(\theta_1)) \dot{\theta}_1] \mathbf{x}_0 + [(r_1 \cos(\theta_1)) \dot{\theta}_1] \mathbf{y}_0$$

$$\begin{aligned}
1.9 \quad \dot{\mathbf{R}}_2 &= [(-l_1 \sin(\theta_1) - r_2 \sin(\theta_1) \cos(\theta_2)) \dot{\theta}_1 + (-r_2 \cos(\theta_1) \sin(\theta_2)) \dot{\theta}_2] \mathbf{x}_0 \\
&+ [(l_1 \cos(\theta_1) + r_2 \cos(\theta_1) \cos(\theta_2)) \dot{\theta}_1 + (-r_2 \sin(\theta_1) \sin(\theta_2)) \dot{\theta}_2] \mathbf{y}_0 \\
&+ [(r_2 \cos(\theta_2)) \dot{\theta}_2] \mathbf{z}_0
\end{aligned}$$

$$\begin{aligned}
1.10 \quad \dot{\mathbf{R}}_3 &= [(-l_1 \sin(\theta_1) - l_2 \sin(\theta_1) \cos(\theta_2) - r_3 \sin(\theta_1) \cos(\theta_2 + \theta_3)) \dot{\theta}_1 \\
&+ (-l_2 \cos(\theta_1) \sin(\theta_2) - r_3 \cos(\theta_1) \sin(\theta_2 + \theta_3)) \dot{\theta}_2 \\
&+ (-r_3 \cos(\theta_1) \sin(\theta_2 + \theta_3)) \dot{\theta}_3] \mathbf{x}_0 \\
&+ [(l_1 \cos(\theta_1) + l_2 \cos(\theta_1) \cos(\theta_2) + r_3 \cos(\theta_1) \cos(\theta_2 + \theta_3)) \dot{\theta}_1 \\
&+ (-l_2 \sin(\theta_1) \sin(\theta_2) - r_3 \sin(\theta_1) \sin(\theta_2 + \theta_3)) \dot{\theta}_2 \\
&+ (-r_3 \sin(\theta_1) \sin(\theta_2 + \theta_3)) \dot{\theta}_3] \mathbf{y}_0 \\
&+ [(l_2 \cos(\theta_2) + r_3 \cos(\theta_2 + \theta_3)) \dot{\theta}_2 + (r_3 \cos(\theta_2 + \theta_3)) \dot{\theta}_3] \mathbf{z}_0
\end{aligned}$$

$$\begin{aligned}
1.11 \quad \dot{\mathbf{R}}_4 &= [(-l_1 \sin(\theta_1) - l_2 \sin(\theta_1) \cos(\theta_2) - l_3 \sin(\theta_1) \cos(\theta_2 + \theta_3)) \dot{\theta}_1 \\
&+ (-l_2 \cos(\theta_1) \sin(\theta_2) - l_3 \cos(\theta_1) \sin(\theta_2 + \theta_3)) \dot{\theta}_2
\end{aligned}$$

$$\begin{aligned}
& + (-l_3 \cos(\theta_1) \sin(\theta_2 + \theta_3)) \dot{\theta}_3] \mathbf{x}_0 \\
& + [(l_1 \cos(\theta_1) + l_2 \cos(\theta_1) \cos(\theta_2) + l_3 \cos(\theta_1) \cos(\theta_2 + \theta_3)) \dot{\theta}_1 \\
& + (-l_2 \sin(\theta_1) \sin(\theta_2) - l_3 \sin(\theta_1) \sin(\theta_2 + \theta_3)) \dot{\theta}_2 \\
& + (-l_3 \sin(\theta_1) \sin(\theta_2 + \theta_3)) \dot{\theta}_3] \mathbf{y}_0 \\
& + [(l_2 \cos(\theta_2) + l_3 \cos(\theta_2 + \theta_3)) \dot{\theta}_2 + (l_3 \cos(\theta_2 + \theta_3)) \dot{\theta}_3] \mathbf{z}_0
\end{aligned}$$

Now with the $\dot{\mathbf{R}}_i$ terms of the kinetic energy equation derived, the $\boldsymbol{\omega}_i$ terms are now derived.

$$1.12 \quad \boldsymbol{\omega}_1^t = [0 \ 0 \ \dot{\theta}_1]$$

$$1.13 \quad \boldsymbol{\omega}_2^t = [\dot{\theta}_1 \sin(\theta_2) \ \dot{\theta}_1 \cos(\theta_2) \ \dot{\theta}_2]$$

$$1.14 \quad \boldsymbol{\omega}_3^t = [\dot{\theta}_1 \sin(\theta_2 + \theta_3) \ \dot{\theta}_1 \cos(\theta_2 + \theta_3) \ (\dot{\theta}_2 + \dot{\theta}_3)]$$

$$1.15 \quad \boldsymbol{\omega}_4^t = [\dot{\theta}_1 \sin(\theta_2 + \theta_3) \ \dot{\theta}_1 \cos(\theta_2 + \theta_3) \ (\dot{\theta}_2 + \dot{\theta}_3)]$$

All the terms of the kinetic and potential energy equations are now given. Carrying out the dot products of the $\dot{\mathbf{R}}_i$ terms and multiplying the $\boldsymbol{\omega}_i$ vectors with their respective inertial matrices gives the expression for kinetic energy.

$$\begin{aligned}
1.16 \quad K = & \frac{1}{2} m_1 r_1^2 \dot{\theta}_1^2 + \frac{1}{2} m_2 [(l_1 + r_2 \cos(\theta_2))^2 \dot{\theta}_1^2 + r_2^2 \dot{\theta}_2^2] \\
& + \frac{1}{2} m_3 [(l_1 + l_2 \cos(\theta_2) + r_3 \cos(\theta_2 + \theta_3))^2 \dot{\theta}_1^2 \\
& + (l_2^2 + 2l_2 r_3 \cos(\theta_3) + r_3^2) \dot{\theta}_2^2 + r_3^2 \dot{\theta}_3^2
\end{aligned}$$

$$\begin{aligned}
& + (2r_3^2 + 2l_2r_3\cos(\theta_3))\dot{\theta}_2\dot{\theta}_3] \\
& + 1/2m_4 [(l_1 + l_2\cos(\theta_2) + l_3\cos(\theta_2+\theta_3))^2\dot{\theta}_1^2 \\
& + (l_2^2 + 2l_2l_3\cos(\theta_3) + l_3^2)\dot{\theta}_2^2 + l_3^2\dot{\theta}_3^2 \\
& + (2l_3^2 + 2l_2l_3\cos(\theta_3))\dot{\theta}_2\dot{\theta}_3] \\
& + 1/2\dot{\theta}_1^2 I_{ZZ}^1 + 1/2[\dot{\theta}_1^2 I_{XX}^2 \sin^2(\theta_2) + \dot{\theta}_1^2 I_{YY}^2 \cos^2(\theta_2) \\
& + \dot{\theta}_2^2 I_{ZZ}^2] \\
& + 1/2[\dot{\theta}_1^2 (I_{XX}^3 + I_{XX}^4) \sin^2(\theta_2+\theta_3) \\
& + \dot{\theta}_1^2 (I_{YY}^3 + I_{YY}^4) \cos^2(\theta_2+\theta_3) + (\dot{\theta}_2+\dot{\theta}_3)^2 (I_{ZZ}^3 + I_{ZZ}^4)]
\end{aligned}$$

Note that because of assumptions 2 and 3 made earlier, all of the off diagonal terms of the inertial matrices are zero, and therefore do not appear in the expression for kinetic energy.

Taking the Z_0 components of the \mathbf{R}_i vectors gives the expression for potential energy.

$$1.17 \quad P = m_2gr_2\sin(\theta_2) + (m_3+m_4)gl_2\sin(\theta_2) + (m_3r_3+m_4l_3)g\sin(\theta_2+\theta_3)$$

Now taking the difference of K and P gives the expression for the Lagrangian, L.

$$1.18 \quad L = 1/2m_1r_1^2\dot{\theta}_1^2 + 1/2m_2[(l_1 + r_2\cos(\theta_2))^2\dot{\theta}_1^2 + r_2^2\dot{\theta}_2^2]$$

$$\begin{aligned}
& + \frac{1}{2}m_3 [(l_1 + l_2\cos(\theta_2) + r_3\cos(\theta_2+\theta_3))^2\dot{\theta}_1^2 \\
& + (l_2^2 + 2l_2r_3\cos(\theta_3) + r_3^2)\dot{\theta}_2^2 + r_3^2\dot{\theta}_3^2 \\
& + (2r_3^2 + 2l_2r_3\cos(\theta_3))\dot{\theta}_2\dot{\theta}_3] \\
& + \frac{1}{2}m_4 [(l_1 + l_2\cos(\theta_2) + l_3\cos(\theta_2+\theta_3))^2\dot{\theta}_1^2 \\
& + (l_2^2 + 2l_2l_3\cos(\theta_3) + l_3^2)\dot{\theta}_2^2 + l_3^2\dot{\theta}_3^2 \\
& + (2l_3^2 + 2l_2l_3\cos(\theta_3))\dot{\theta}_2\dot{\theta}_3] \\
& + \frac{1}{2}\dot{\theta}_1^2 I_{ZZ}^1 + \frac{1}{2}[\dot{\theta}_1^2 I_{XX}^2 \sin^2(\theta_2) + \dot{\theta}_1^2 I_{YY}^2 \cos^2(\theta_2) \\
& + \dot{\theta}_2^2 I_{ZZ}^2] + \frac{1}{2}[\dot{\theta}_1^2 (I_{XX}^3 + I_{XX}^4 \sin^2(\theta_2+\theta_3) \\
& + \dot{\theta}_1^2 (I_{YY}^3 + I_{YY}^4) \cos^2(\theta_2+\theta_3) + (\dot{\theta}_2+\dot{\theta}_3)^2 (I_{ZZ}^3 + I_{ZZ}^4)] \\
& - m_2 g r_2 \sin(\theta_2) - (m_3+m_4) g l_2 \sin(\theta_2) - (m_3 r_3 + m_4 l_3) g \sin(\theta_2+\theta_3)
\end{aligned}$$

By Lagrange's equations, the torques at joints 1, 2 and 3 are given as:

$$1.19 \quad T_1 = \frac{d}{dt} \left[\frac{\partial L}{\partial \dot{\theta}_1} \right] - \frac{\partial L}{\partial \theta_1}$$

$$1.20 \quad T_2 = \frac{d}{dt} \left[\frac{\partial L}{\partial \dot{\theta}_2} \right] - \frac{\partial L}{\partial \theta_2}$$

$$1.21 \quad T_3 = \frac{d}{dt} \left[\frac{\partial L}{\partial \dot{\theta}_3} \right] - \frac{\partial L}{\partial \theta_3}$$

Performing these operations give the expressions for torque.

$$1.22 \quad T_1 = [m_1 r_1^2 + m_2 (l_1 + r_2 \cos(\theta_2))^2 + m_3 (l_1 + l_2 \cos(\theta_2))$$

$$\begin{aligned}
& + r_3 \cos(\theta_2 + \theta_3))^2 + m_4(l_1 + l_2 \cos(\theta_2) + l_3 \cos(\theta_2 + \theta_3))^2 \\
& + I_{ZZ}^1 + I_{XX}^2 \sin^2(\theta_2) + I_{YY}^2 \cos^2(\theta_2) + (I_{XX}^3 + I_{XX}^4) \sin^2(\theta_2 + \theta_3) \\
& + (I_{YY}^3 + I_{YY}^4) \cos^2(\theta_2 + \theta_3)] \ddot{\theta}_1 \\
& + [-2 r_2 m_2 \sin(\theta_2)(l_1 + r_2 \cos(\theta_2)) - 2 m_3(l_2 \sin(\theta_2) \\
& + r_3 \sin(\theta_2 + \theta_3))(l_1 + l_2 \cos(\theta_2) + r_3 \cos(\theta_2 + \theta_3)) \\
& - 2 m_4(l_2 \sin(\theta_2) + l_3 \sin(\theta_2 + \theta_3))(l_1 + l_2 \cos(\theta_2) + l_3 \cos(\theta_2 + \theta_3)) \\
& + 2 \sin(\theta_2) \cos(\theta_2)(I_{XX}^2 - I_{YY}^2) \\
& + 2 \sin(\theta_2 + \theta_3) \cos(\theta_2 + \theta_3)(I_{XX}^3 + I_{XX}^4 - I_{YY}^3 - I_{YY}^4)] \dot{\theta}_1 \dot{\theta}_2 \\
& + [-2 m_3 r_3 \sin(\theta_2 + \theta_3)(l_1 + l_2 \cos(\theta_2) + r_3 \cos(\theta_2 + \theta_3)) \\
& - 2 m_4 l_3 \sin(\theta_2 + \theta_3)(l_1 + l_2 \cos(\theta_2) + l_3 \cos(\theta_2 + \theta_3)) \\
& + 2 \cos(\theta_2 + \theta_3) \sin(\theta_2 + \theta_3)(I_{XX}^3 + I_{XX}^4 - I_{YY}^3 - I_{YY}^4)] \dot{\theta}_1 \dot{\theta}_3
\end{aligned}$$

$$\begin{aligned}
1.23 \quad T_2 = & [m_2 r_2^2 + m_3(l_2^2 + 2 l_2 r_3 \cos(\theta_3) + r_3^2) \\
& + m_4(l_2^2 + 2 l_2 l_3 \cos(\theta_3) + l_3^2) + I_{ZZ}^2 + I_{ZZ}^3 + I_{ZZ}^4] \ddot{\theta}_2 \\
& + [m_3(r_3^2 + l_2 r_3 \cos(\theta_3)) + m_4(l_3^2 + l_2 l_3 \cos(\theta_3)) + I_{ZZ}^3 \\
& + I_{ZZ}^4] \ddot{\theta}_3 + [m_2 r_2 \sin(\theta_2)(l_1 + r_2 \cos(\theta_2)) + m_3(l_2 \sin(\theta_2) \\
& + r_3 \sin(\theta_2 + \theta_3))(l_1 + l_2 \cos(\theta_2) + r_3 \cos(\theta_2 + \theta_3)) + m_4(l_2 \sin(\theta_2)
\end{aligned}$$

$$\begin{aligned}
& + l_3 \sin(\theta_2 + \theta_3)(l_1 + l_2 \cos(\theta_2) + l_3 \cos(\theta_2 + \theta_3)) \\
& + (I_{yy}^2 - I_{xx}^2) \sin(\theta_2) \cos(\theta_2) \\
& + (I_{yy}^3 + I_{yy}^4 - I_{xx}^3 - I_{xx}^4) \sin(\theta_2 + \theta_3) \cos(\theta_2 + \theta_3) \dot{\theta}_1^2 \\
& + [-2l_2 \sin(\theta_3)(m_3 r_3 + m_4 l_3)] \dot{\theta}_2 \dot{\theta}_3 \\
& + [-l_2 \sin(\theta_3)(m_3 r_3 + m_4 l_3)] \dot{\theta}_3^2 \\
& + [m_2 g r_2 \cos(\theta_2) + (m_3 + m_4) g l_2 \cos(\theta_2)] \\
& + (m_3 r_3 + m_4 l_3) g \cos(\theta_2 + \theta_3)
\end{aligned}$$

$$\begin{aligned}
1.24 \quad T_3 = & [m_3(r_3^2 + l_2 r_3 \cos(\theta_3)) + m_4(l_3^2 + l_2 l_3 \cos(\theta_3)) + I_{zz}^3 + I_{zz}^4] \ddot{\theta}_2 \\
& + [m_3 r_3^2 + m_4 l_3^2 + I_{zz}^3 + I_{zz}^4] \ddot{\theta}_3 \\
& + [m_3 r_3 \sin(\theta_2 + \theta_3)(l_1 + l_2 \cos(\theta_2) + r_3 \cos(\theta_2 + \theta_3)) \\
& + m_4 l_3 \sin(\theta_2 + \theta_3)(l_1 + l_2 \cos(\theta_2) + l_3 \cos(\theta_2 + \theta_3)) \\
& + (I_{yy}^3 + I_{yy}^4 - I_{xx}^3 - I_{xx}^4) \sin(\theta_2 + \theta_3) \cos(\theta_2 + \theta_3)] \dot{\theta}_1^2 \\
& + [m_3 l_2 r_3 \sin(\theta_3) + m_4 l_2 l_3 \sin(\theta_3)] \dot{\theta}_2^2 \\
& + [(m_3 r_3 + m_4 l_3) g \cos(\theta_2 + \theta_3)]
\end{aligned}$$

In order to implement the dynamic equations of the robot in the computer, they are put in state variable form. Following the form used by Dubowsky and DesForges [2], the differential equations are written in the

form:

$$1.25 \quad \begin{bmatrix} \underline{m}_{11} & \underline{m}_{12} & \underline{m}_{13} \\ \underline{m}_{21} & \underline{m}_{22} & \underline{m}_{23} \\ \underline{m}_{31} & \underline{m}_{32} & \underline{m}_{33} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix} + \begin{bmatrix} g_{11}\dot{\theta}_1 & g_{12}\dot{\theta}_1 & g_{13}\dot{\theta}_1 \\ g_{21}\dot{\theta}_2 & g_{22}\dot{\theta}_2 & g_{23}\dot{\theta}_2 \\ g_{31}\dot{\theta}_3 & g_{32}\dot{\theta}_3 & g_{33}\dot{\theta}_3 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} - \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix}$$

where Q_i represents gravitational and other forces not included in the left hand side of the equation. Note that the elements of the \mathbf{M} matrix are underlined so as to distinguish them from the m 's representing the link masses. The state variables for the system are defined as.

$$1.26 \quad \begin{aligned} y(1) &= \theta_1 & y(4) &= \dot{\theta}_1 & \dot{y}(1) &= y(4) \\ y(2) &= \theta_2 & y(5) &= \dot{\theta}_2 & \dot{y}(2) &= y(5) \\ y(3) &= \theta_3 & y(6) &= \dot{\theta}_3 & \dot{y}(3) &= y(6) \end{aligned}$$

Writing equation 1.25 using the state variables given in equation

1.26 gives:

$$1.27 \quad \begin{bmatrix} \dot{y}(4) \\ \dot{y}(5) \\ \dot{y}(6) \end{bmatrix} = \begin{bmatrix} \underline{m}_{11} & \underline{m}_{12} & \underline{m}_{13} \\ \underline{m}_{21} & \underline{m}_{22} & \underline{m}_{23} \\ \underline{m}_{31} & \underline{m}_{23} & \underline{m}_{33} \end{bmatrix}^{-1} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} - \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix} - \begin{bmatrix} g_{11}y(4) & g_{12}y(4) & g_{13}y(4) \\ g_{21}y(5) & g_{22}y(5) & g_{23}y(5) \\ g_{31}y(6) & g_{32}y(6) & g_{33}y(6) \end{bmatrix} \begin{bmatrix} y(4) \\ y(5) \\ y(6) \end{bmatrix}$$

Equations 1.26 and 1.27 represent the equations of motion for the manipulator.

$$1.28 \quad \dot{y}(1) = y(4)$$

$$\dot{y}(2) = y(5)$$

$$\dot{y}(3) = y(6)$$

$$\begin{aligned} \dot{y}(4) & \\ \dot{y}(5) &= \mathbf{M}^{-1} \begin{bmatrix} T_1 & Q_1 & G_1 \\ T_2 & Q_2 & G_2 \\ T_3 & Q_3 & G_3 \end{bmatrix} \\ \dot{y}(6) & \end{aligned}$$

Simplifying the form of the matrix equation, the last two matrices are multiplied together to form a new column matrix \mathbf{G} . The state equations in 1.28 represent these changes. The expressions for the elements of the \mathbf{M} , \mathbf{Q} , and \mathbf{G} matrices are obtained from the equations of the torques given earlier.

$$\begin{aligned} 1.29a \quad \underline{m}_{11} &= m_1 r_1^2 + m_2 (l_1 + r_2 \cos(\theta_2))^2 + m_3 (l_1 + l_2 \cos(\theta_2) \\ &+ r_3 \cos(\theta_2 + \theta_3))^2 + m_4 (l_1 + l_2 \cos(\theta_2) + l_3 \cos(\theta_2 + \theta_3))^2 + I_{ZZ}^1 \\ &+ I_{xx}^2 \sin^2(\theta_2) + I_{yy}^2 \cos^2(\theta_2) + (I_{yy}^3 + I_{yy}^4) \cos^2(\theta_2 + \theta_3) \\ &+ (I_{xx}^3 + I_{xx}^4) \sin^2(\theta_2 + \theta_3) \end{aligned}$$

$$\begin{aligned} 1.29b \quad \underline{m}_{22} &= m_2 r_2^2 + m_3 (l_2^2 + 2 l_2 r_3 \cos(\theta_3) + r_3^2) \\ &+ m_4 (l_2^2 + 2 l_2 l_3 \cos(\theta_3) + l_3^2) + I_{ZZ}^2 + I_{ZZ}^3 + I_{ZZ}^4 \end{aligned}$$

$$1.29c \quad \underline{m}_{23} = \underline{m}_{32} = m_3 (r_3^2 + l_2 r_3 \cos(\theta_3)) + m_4 (l_3^2 + l_2 l_3 \cos(\theta_3)) + I_{ZZ}^3 + I_{ZZ}^4$$

$$1.29d \quad \underline{m}_{33} = m_3 r_3^2 + m_4 l_3^2 + I_{ZZ}^3 + I_{ZZ}^4$$

$$1.29e \quad G_1 = [-2r_2 m_2 \sin(\theta_2)(l_1 + r_2 \cos(\theta_2)) - 2m_3(l_2 \sin(\theta_2) + r_3 \sin(\theta_2 + \theta_3)) \\ * (l_1 + l_2 \cos(\theta_2) + r_3 \cos(\theta_2 + \theta_3)) - 2m_4(l_2 \sin(\theta_2) + l_3 \sin(\theta_2 + \theta_3)) \\ *(l_1 + l_2 \cos(\theta_2) + l_3 \cos(\theta_2 + \theta_3)) + 2 \sin(\theta_2) \cos(\theta_2) (I_{xx}^2 - I_{yy}^2) \\ + 2 \sin(\theta_2 + \theta_3) \cos(\theta_2 + \theta_3) (I_{xx}^3 + I_{xx}^4 - I_{yy}^3 - I_{yy}^4)] \dot{\theta}_1 \dot{\theta}_2 \\ + [-2 m_3 r_3 \sin(\theta_2 + \theta_3) (l_1 + l_2 \cos(\theta_2) + r_3 \cos(\theta_2 + \theta_3)) \\ - 2 m_4 l_3 \sin(\theta_2 + \theta_3) (l_1 + l_2 \cos(\theta_2) + l_3 \cos(\theta_2 + \theta_3)) \\ + 2 \sin(\theta_2 + \theta_3) \cos(\theta_2 + \theta_3) (I_{xx}^3 + I_{xx}^4 - I_{yy}^3 - I_{yy}^4)] \dot{\theta}_1 \dot{\theta}_3$$

$$1.29f \quad G_2 = [-2l_2 \sin(\theta_3) (m_3 r_3 + m_4 l_3)] \dot{\theta}_2 \dot{\theta}_3$$

$$1.29g \quad G_3 = 0$$

$$1.29h \quad Q_1 = 0$$

$$1.29i \quad Q_2 = [m_2 r_2 \sin(\theta_2) (l_1 + r_2 \cos(\theta_2)) + m_3 (l_2 \sin(\theta_2) + r_3 \sin(\theta_2 + \theta_3)) \\ *(l_1 + l_2 \cos(\theta_2) + r_3 \cos(\theta_2 + \theta_3)) + m_4 (l_2 \sin(\theta_2) + l_3 \sin(\theta_2 + \theta_3)) \\ *(l_1 + l_2 \cos(\theta_2) + l_3 \cos(\theta_2 + \theta_3)) + (I_{yy}^2 - I_{xx}^2) \sin(\theta_2) \cos(\theta_2) \\ + \sin(\theta_2 + \theta_3) \cos(\theta_2 + \theta_3) (I_{yy}^3 + I_{yy}^4 - I_{xx}^3 - I_{xx}^4)] \dot{\theta}_1^2 \\ + [-l_2 \sin(\theta_3) (m_3 r_3 + m_4 l_3)] \dot{\theta}_3^2 \\ + [m_2 g r_2 \cos(\theta_2) + (m_3 + m_4) g l_2 \cos(\theta_2)]$$

$$+ (m_3 r_3 + m_4 l_3) g \cos(\theta_2 + \theta_3)]$$

$$1.29j \quad Q_3 = [m_3 r_3 \sin(\theta_2 + \theta_3) (l_1 + l_2 \cos(\theta_2) + r_3 \cos(\theta_2 + \theta_3)) \\ + m_4 l_3 \sin(\theta_2 + \theta_3) (l_1 + l_2 \cos(\theta_2) + l_3 \cos(\theta_2 + \theta_3)) \\ + \sin(\theta_2 + \theta_3) \cos(\theta_2 + \theta_3) (I_{yy}^3 + I_{yy}^4 - I_{xx}^3 - I_{xx}^4)] \dot{\theta}_1^2 \\ + [m_3 l_2 r_3 \sin(\theta_3) + m_4 l_2 l_3 \sin(\theta_3)] \dot{\theta}_2^2 \\ + [(m_3 r_3 + m_4 l_3) g \cos(\theta_2 + \theta_3)]$$

Symbolically inverting the \mathbf{M} matrix yields:

$$1.30 \quad \mathbf{M}^{-1} = \begin{bmatrix} \frac{1}{m_{11}} & 0 & 0 \\ 0 & \frac{m_{33}}{m_{33}m_{22} - m_{23}^2} & \frac{-m_{23}}{m_{33}m_{22} - m_{23}^2} \\ 0 & \frac{-m_{23}}{m_{33}m_{22} - m_{23}^2} & \frac{m_{22}}{m_{33}m_{22} - m_{23}^2} \end{bmatrix}$$

Finally, the form of the equations used in the computer solution is:

$$1.31 \quad \dot{y}(1) = y(4)$$

$$\dot{y}(2) = y(5)$$

$$\dot{y}(3) = y(6)$$

$$\dot{y}(4) = [T_1 - Q_1 - G_1] / m_{11}$$

$$\dot{y}(5) = [m_{33}(T_2 - Q_2 - G_2) - m_{23}(T_3 - Q_3 - G_3)] / [m_{33}m_{22} - m_{23}^2]$$

$$\dot{y}(6) = [m_{22}(T_3 - Q_3 - G_3) - m_{23}(T_2 - Q_2 - G_2)] / [m_{33}m_{22} - m_{23}^2]$$

Calculation of Moments of Inertia of Links

The actual links of the IRI manipulator are nonhomogeneous nonsymmetrical bodies, but for the purpose of the computer simulation, a simplified model of the links will be used. Figure 3 represents a model link.

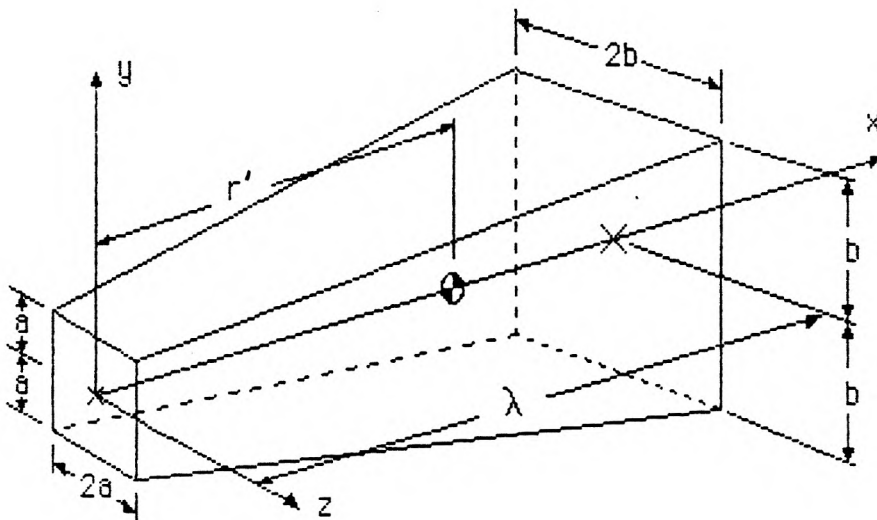


FIGURE 3. Model of Robot Link

The model is a solid homogeneous body that is symmetrical about the x axis. At $x=0$, the cross-section is a square of dimension $2a \times 2a$, and at $x=\lambda$, the cross-section is a square of dimension $2b \times 2b$. The center of mass of the link is located at $x=r'$ on the x axis. The variable r' is used here to represent the distance from the aft end of a link to the center of mass. This is different from r , which is used to represent the distance from the origin of the link coordinate system to the center of mass. For links 2 and 3, r'

will equal r . For link 1, however, r' does not equal r .

To calculate the moments of inertia of each link, the following equations are used.

$$1.32a \quad I_{xx} = \int (y^2 + z^2) dm$$

$$1.32b \quad I_{yy} = \int (x^2 + z^2) dm$$

$$1.32c \quad I_{zz} = \int (x^2 + y^2) dm$$

To perform these integrals, the limits of integration must be determined. If the values of r' and λ were known for each link, then using the formula for \bar{x} , the ratio of a/b could be determined. The equation for \bar{x} is:

$$1.33 \quad r' = \bar{x} = (\int x dm) / (\int dm)$$

Then by knowing the weight of each link, and by selecting a suitable density for the solid, the values of a and b could be determined by:

$$1.34 \quad W / (32.2 \text{ ft/s}^2) = \int dm \quad \text{where } W = \text{weight of link}$$

The proper limits of integration can be defined, once the values of a and b are determined for each link.

From the model of the link given in Figure 3, the integral of $x dm$ is given as:

$$1.35a \quad \int x \, dm = 4\rho \int_0^\lambda x \, dx \int_0^{\frac{b-a}{\lambda}x+a} dy \int_0^{\frac{b-a}{\lambda}x+a} dz$$

$$1.35b \quad = 4\rho \int_0^\lambda x \left[\left(\frac{b-a}{\lambda}x + a \right)^2 dx \right]$$

$$1.35c \quad = \frac{1}{3}\rho\lambda^2[3b^2 + 2ab + a^2]$$

The integral of dm is given as:

$$1.36a \quad \int dm = 4\rho \int_0^\lambda dx \int_0^{\frac{b-a}{\lambda}x+a} dy \int_0^{\frac{b-a}{\lambda}x+a} dz$$

$$1.36b \quad = 4\rho \int_0^\lambda \left[\left(\frac{b-a}{\lambda}x + a \right)^2 dx \right]$$

$$1.36c \quad = \frac{4}{3}\rho\lambda[b^2 + ab + a^2]$$

The location of the center of mass is then given by:

$$\bar{x} = (\int x \, dm) / (\int dm)$$

$$1.37 \quad \bar{x} = [\lambda(3b^2 + 2ab + a^2)] / [4(b^2 + ab + a^2)]$$

and since $\bar{x} = r'$, then

$$1.38 \quad \frac{3b^2 + 2ab + a^2}{b^2 + ab + a^2} = \frac{4r'}{\lambda}$$

$$\text{or } \frac{b}{a} = \frac{4r' - 2\lambda + \sqrt{(2\lambda - 4r')^2 - 4(3\lambda - 4r')(\lambda - 4r')}}{2(3\lambda - 4r')} \quad *$$

*The negative solution of the equation is discarded.

To calculate the ratio of b/a then, the length of the robot links and the location of the center of mass of each link must be known. This information is given in Table 1 and was provided by IRI (with the exception

of the value of r' for link 1 which was estimated).

TABLE 1

Link	λ	r'
1	41.0"	29.0"
2	32.0"	17.5"
3	37.0"	17.25"

Given these physical parameters of the robot links, then the calculated ratios of b/a are:

$$\text{Link 1 } b/a = 6.504$$

$$\text{Link 2 } b/a = 1.330$$

$$\text{Link 3 } b/a = 0.8154$$

The weight of each link is given by the equation

$$1.39 \quad W = 32.2 \int dm$$

The weights of each link have been provided by IRI, however, a value of the volume of a link is necessary for calculating density. Assuming that the links are homogeneous solids with no concentrated masses, then the density can be assumed to be constant over the entire link and is equal to mass divided by volume. Link 2's volume was approximated to be 2.667 ft³ (12" x

12" x 32"), and the weight of link 2 (given by IRI) is 60 lbs. The calculated density of link 2 is:

$$\begin{aligned}
 1.40 \quad \rho &= [60 \text{ lbs}/(32.2 \text{ ft/s}^2)]/2.667 \text{ ft}^3 \\
 &= 0.699 \text{ slugs/ft}^3 \approx 0.7 \text{ slugs/ft}^3
 \end{aligned}$$

All the links of the robot are assumed to have the same density, so the calculation of equation 1.40 will serve as the representative density of all the links. Table 2 shows the results of the solution of the a's and b's using the value of 0.7 slugs/ft³ for ρ .

TABLE 2

Weight (W)	a	b
199 lbs	2.37"	15.40"
60 lbs	5.13"	6.82"
66 lbs	6.43"	5.24"

These values of a and b are used to calculate the limits for the integrals of the moments of inertia given in equations 1.32a, 1.32b, and 1.32c. The general form of the integrals are the same for each link of the robot. Three primary integrals are involved, these being $\int x^2 dm$, $\int y^2 dm$, and $\int z^2 dm$. Due to symmetry about the x axis, though, the integrals of $y^2 dm$ and $z^2 dm$ will be equal. This leaves two expressions to be derived. In

calculating the moments of inertia for the links, the link center of gravity is used for the calculation. The limits of integration for the y and z axes can be expressed as a function of x, given the slope and intersection of the planes, which are the sides of the link solids. These slopes and intersections can be calculated from the values of a and b for each link, given the location of the center of gravity of each link within the link solid. Figure 4 shows the location of the center of gravity of the links of the robot. Given the information in Figure 4, and the values of a and b for each link, the slope and intersection of the sides of the links can be determined. Expressing the limits of integration in the y and z directions as

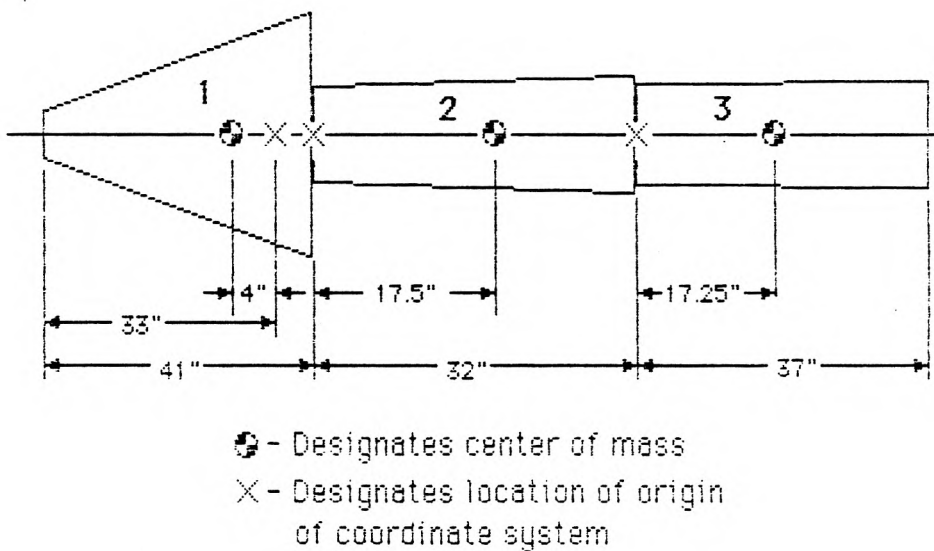


FIGURE 4. Coordinate Systems and Centers of Mass.

functions of x yields the form $-- mx+c$, where m is the slope and c is the point of intersection. Applying this form to the equation for the moment of inertia about the x axis gives:

$$1.41 \quad I_{xx} = \int y^2 dm + \int z^2 dm = 2 \int y^2 dm$$

$$I_{xx} = 2\rho \int_{d_1}^{d_2} dx \int_{-(mx+c)}^{mx+c} y^2 dy \int_{-(mx+c)}^{mx+c} dz$$

where d_1 and d_2 represent the aft and fore intersection of the link solid with the x axis respectively.

Since the links are symmetrical about their x axes, the limits of integration can be changed from $-(mx+c)$ to $mx+c$, to, 0 to $mx+c$, and then multiply the integral by 4. This gives the final expression for inertia about the x axis.

$$1.42 \quad I_{xx} = 8\rho \int_{d_1}^{d_2} dx \int_0^{mx+c} y^2 dy \int_0^{mx+c} dz$$

Performing this integration gives the general form for calculating the moment of inertia about the x axis of each link.

$$1.43 \quad I_{xx} = \frac{8}{3}\rho[(m^4x^5)/5 + m^3x^4c + 2m^2x^3c^2 + 2mx^2c^3 + xc^4] \Big|_{d_1}^{d_2}$$

The calculation of the moments of inertia about the y and z axes follows the same reasoning used previously. Due to symmetry, these inertias are equal, and therefore only one need be derived. From equation

1.32c, the moment of inertia about the z axis is:

$$I_{zz} = \int (x^2 + y^2) dm$$

From equation 1.43, the expression for 2 times the integral of $y^2 dm$ is derived. Dividing this equation by 2 give the expression for the integral of $y^2 dm$. Calculating the integral of $x^2 dm$ gives:

$$\begin{aligned} 1.44 \quad \int x^2 dm &= 4\rho \int_{d_1}^{d_2} x^2 dx \int_0^{mx+c} dy \int_0^{mx+c} dz \\ &= 4\rho \left[\frac{(m^2 x^5)}{5} + \frac{(mcx^4)}{2} + \frac{(c^2 x^3)}{3} \right] \Big|_{d_1}^{d_2} \end{aligned}$$

Combining the expressions for the integrals of $y^2 dm$ and $x^2 dm$ gives the expression for the moment of inertia about the y and z axes of each link.

$$\begin{aligned} 1.45 \quad I_{yy} = I_{zz} &= \frac{4}{3}\rho \left[\frac{(m^4 x^5)}{5} + m^3 x^4 c + 2m^2 x^3 c^2 + 2mx^2 c^3 + xc^4 \right] \Big|_{d_1}^{d_2} \\ &+ 4\rho \left[\frac{(m^2 x^5)}{5} + \frac{(mcx^4)}{2} + \frac{(c^2 x^3)}{3} \right] \Big|_{d_1}^{d_2} \end{aligned}$$

Table 3 contains all the link parameters used in the calculation of the values of the moments of inertia.

TABLE 3

<u>Link</u>	<u>Mass</u>	<u>a(in)</u>	<u>b(in)</u>	<u>d₁(in)</u>	<u>d₂(in)</u>	<u>m</u>	<u>c(in)</u>
1	6.18	2.37	15.40	-29.0	12.0	.318	11.58
2	1.863	5.13	6.82	-17.5	14.5	.0529	6.504
3	2.050	6.43	5.24	-17.25	19.75	-.0321	5.877

Using these values, the respective moments of inertia can be computed using equations 1.43 and 1.45. These values are:

$$I^1_{xx} = 4.084 \quad I^2_{xx} = .3182 \quad I^3_{xx} = .3289$$

$$I^1_{yy} = 5.684 \quad I^2_{yy} = 1.240 \quad I^3_{yy} = 1.771$$

$$I^1_{zz} = 5.684 \quad I^2_{zz} = 1.240 \quad I^3_{zz} = 1.771$$

All inertias are in slug-ft²

This completes the derivation of the dynamic equations of the model robot and the calculation of the moments of inertia of the links. The equations and the values derived in Chapter 1 will be implemented in Chapter 2 in the form of a computer simulation of the robot. In Chapter 2, various control systems will be examined in conjunction with model reference adaptive control.

II. MODEL REFERENCE ADAPTIVE CONTROL DEVELOPMENT

The development of the model reference adaptive control algorithm follows the work done by Donaldson and Leondes [1]. In designing a model reference adaptive control system for the robot the type of control system to be used must be selected. The choice of control used with the model reference adaptive control algorithm has a great deal to do with the way the algorithm performs.

Proportional Integral Control with Derivative Feedback

The IRI robot uses PID control, and in the first efforts to choose a control system, proportional and integral control were incorporated in a feed forward compensator with unity and derivative action in the feedback (see Figure 5). The placement of the derivative action in the feedback gives the desired anticipation without adding another zero to the closed loop transfer function.

Assuming a simplified model of the robot of the form $K_m/s(Js+f)$, where K_m is the motor torque constant in ft-lbs per radian, J is the effective inertia about the axis in slug-ft², and f is the viscous friction

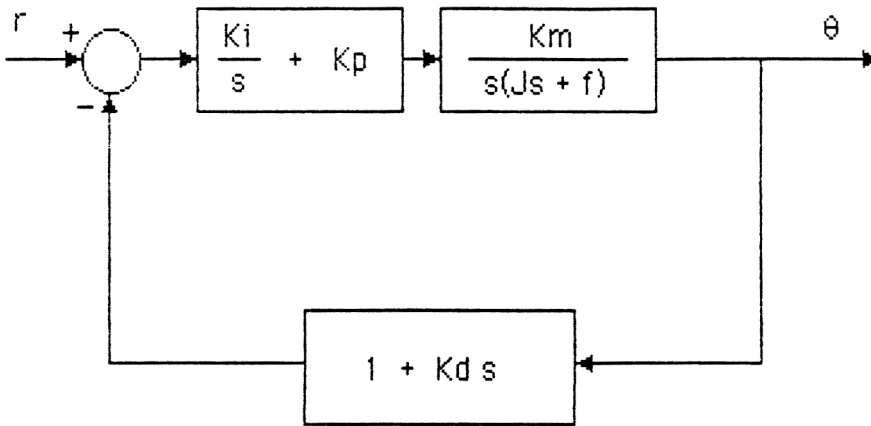


FIGURE 5. Proportional Integral Control with Derivative Feedback

factor in ft-lbs per radian per second, the closed loop transfer function of the control system is:

$$2.1 \quad T(s) = \frac{b_1 s + a_0}{s^3 + a_2 s^2 + a_1 s + a_0}$$

$$\text{where } b_1 = \frac{K_m K_p}{J}$$

$$a_2 = \frac{f + K_m K_p K_d}{J}$$

$$a_1 = \frac{K_m K_p + K_m K_i K_d}{J}$$

$$a_0 = \frac{K_m K_i}{J}$$

and where K_p , K_i , and K_d are the proportional, integral, and derivative gains respectively.

Given the form of the closed loop transfer function of the system, a

reference model of the same form is chosen with constant coefficients. The coefficients of the model are chosen so that the output of the model will have the performance characteristics desired of the robot. The form of the model is:

$$2.2 \quad M(s) = \frac{B_1s + A_0}{s^3 + A_2s^2 + A_1s + A_0}$$

where B_1 , A_0 , A_1 , and A_2 are constant coefficients chosen so that the model responds the way it is desired that the robot respond.

The parameters of the robot model are assumed to vary with time in some unknown manner. The principle cause of the variation is the change in J due to load and orientation. The control system parameters, K_p , K_i , and K_d are adjusted by the model reference adaptive control algorithm so that the values of b_1 , a_2 , a_1 , and a_0 are driven towards the values of the constant coefficients of the reference model, B_1 , A_2 , A_1 , and A_0 , using a steepest descent trajectory.

It is clear that this control system configuration does not produce enough system parameters to independently adjust the coefficients of the closed loop transfer function of the system. There are 4 coefficients to be

adjusted and only 3 system parameters, K_p , K_i , and K_d .

This control system configuration is unfit for the model reference adaptive control algorithm, however, if another system parameter could be added to the control system without increasing the number of coefficients in the closed loop transfer function, then there would be enough system parameters to independently adjust the coefficients of the closed loop transfer function. By adding a second derivative term in the feedback loop, this is accomplished (see Figure 6). The closed loop transfer function is:

$$2.3 \quad T(s) = \frac{b_1 s + a_0}{s^3 + a_2 s^2 + a_1 s + a_0}$$

$$\text{where } b_1 = \frac{K_m K_D}{J + K_m K_p K_a}$$

$$a_2 = \frac{f + K_m K_i K_a + K_m K_d K_D}{J + K_m K_p K_a}$$

$$a_1 = \frac{K_m K_i K_d + K_m K_D}{J + K_m K_p K_a}$$

$$a_0 = \frac{K_m K_i}{J + K_m K_p K_a}$$

and where K_a is the coefficient of the second derivative term

This configuration meets the required number of system parameters needed to independently adjust the coefficients of the closed loop transfer

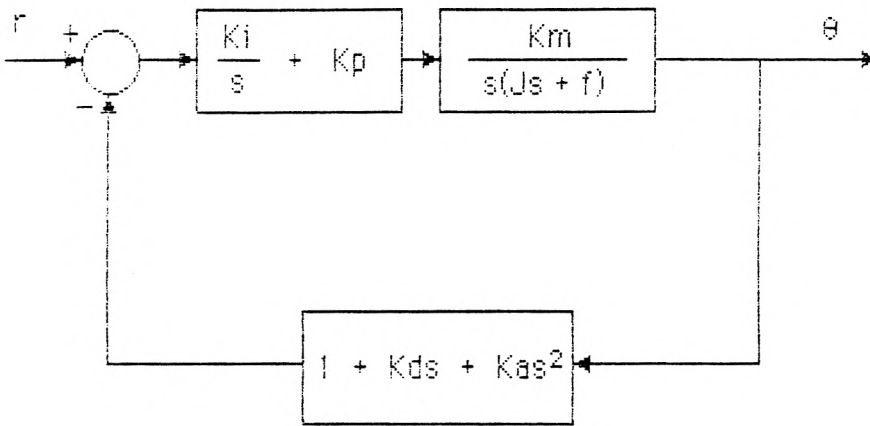


FIGURE 6. Modified Control with Second Derivative Feedback Added.

function. However, relatively complex equations are derived for use in the model reference adaptive control algorithm from this configuration. The equations that result are of a highly coupled nature, and when tested, are unsuccessful in adjusting the gains of the control system.

PID Control

Figure 7 is a diagram of a PID control system, and in this case, the

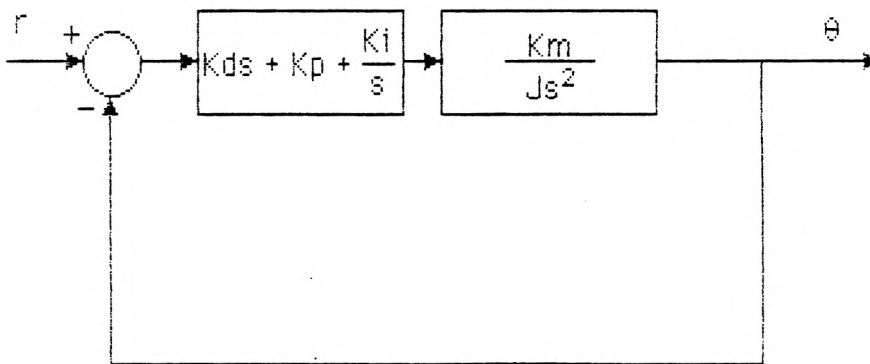


FIGURE 7. PID Control

simplified model of the robot is of the form Km/Js^2 . This form of the

simplified model of the robot neglects any viscous friction terms which might exist in the robot. In neglecting these terms, it is assumed that these forces are small in comparison to the inertial forces acting on the robot and therefore do not significantly affect the dynamic characteristics of the robot arm. Given this control system, the closed loop transfer function is:

$$2.4 \quad T(s) = \frac{a_2 s^2 + a_1 s + a_0}{s^3 + a_2 s^2 + a_1 s + a_0}$$

$$\text{where } a_0 = \frac{K_m K_i}{J}$$

$$a_1 = \frac{K_m K_D}{J}$$

$$a_2 = \frac{K_m K_d}{J}$$

and the model is of the form:

$$2.5 \quad M(s) = \frac{A_2 s^2 + A_1 s + A_0}{s^3 + A_2 s^2 + A_1 s + A_0}$$

where A_0 , A_1 , and A_2 are constant coefficients chosen to give the model the response that is desired of the system.

Following the steps outlined by Donaldson and Leondes [1], the gain adjustment equations are derived. Two initial assumptions are made.

Assumption 1

Km/J Varies slowly compared to the basic time constants of the physical process and the reference model.

Assumption 2

Km/J varies slowly compared to the rates at which the adjusting mechanism adjusts the parameters Kp, Ki, and Kd.

The first step is to define a quadratic error function which is to be minimized by the adjusting algorithm. The error function selected is:

$$2.6 \quad f(e) = \frac{[q_0 e^2 + q_1 \dot{e}^2 + q_2 \ddot{e}^2 + q_3 \ddot{\tilde{e}}^2]}{2}$$

where $e = \theta - y$

q 's are the weighting factors for the errors

θ is the output of the system

and y is the output of the model

A steepest descent method is derived by Donaldson and Leondes in which the rates of adjustments of the coefficients are proportional to the negative of the slopes of the error function in the directions of the coefficients. If the response of the model and system are alike, the slopes of the error function in the direction of the model coefficients are equal to

the negative of the slopes of the error function in the directions of the system coefficients. The slopes of the error function in the directions of the system coefficients are unknown, but the slopes in the directions of the model coefficients can be computed. The adjustment rates are:

$$2.7a \quad \dot{a}_0 = \frac{\partial f(e)}{\partial A_0}$$

$$2.7b \quad \dot{a}_1 = \frac{\partial f(e)}{\partial A_1}$$

$$2.7c \quad \dot{a}_2 = \frac{\partial f(e)}{\partial A_2}$$

With regard to these equations, a third assumption is made.

Assumption 3

The adjusting mechanism will be designed so that it adjusts the parameters K_p , K_i , and K_d at a rate that is rapid when compared with the rate at which $f(e)$ changes due to the effects of the input $r(t)$.

Taking the respective partial derivatives of the error function gives:

$$2.8a \quad \dot{a}_0 = -q_0 e \frac{\partial y}{\partial A_0} - q_1 \dot{e} \frac{\partial \dot{y}}{\partial A_0} - q_2 \ddot{e} \frac{\partial \ddot{y}}{\partial A_0} - q_3 \ddot{\ddot{e}} \frac{\partial \ddot{\ddot{y}}}{\partial A_0}$$

$$2.8b \quad \dot{a}_1 = -q_0 e \frac{\partial y}{\partial A_1} - q_1 \dot{e} \frac{\partial \dot{y}}{\partial A_1} - q_2 \ddot{e} \frac{\partial \ddot{y}}{\partial A_1} - q_3 \ddot{\ddot{e}} \frac{\partial \ddot{\ddot{y}}}{\partial A_1}$$

$$2.8c \quad \dot{a}_2 = -q_0 e \frac{\partial y}{\partial A_2} - q_1 \dot{e} \frac{\partial \dot{y}}{\partial A_2} - q_2 \ddot{e} \frac{\partial \ddot{y}}{\partial A_2} - q_3 \ddot{\ddot{e}} \frac{\partial \ddot{\ddot{y}}}{\partial A_2}$$

The transfer function of the model can be expressed in the form of the differential equation:

$$2.9 \quad \bar{y} + A_2 \ddot{y} + A_1 \dot{y} + A_0 y = A_2 \ddot{r} + A_1 \dot{r} + A_0 r$$

where y is the output of the model and r is the command input to the system and the model.

Taking the partials of the differential equation with respect to A_2 , A_1 , and A_0 yields:

$$2.10a \quad \frac{\partial \bar{y}}{\partial A_0} + A_2 \frac{\partial \ddot{y}}{\partial A_0} + A_1 \frac{\partial \dot{y}}{\partial A_0} + A_0 \frac{\partial y}{\partial A_0} = r - y$$

$$2.10b \quad \frac{\partial \bar{y}}{\partial A_1} + A_2 \frac{\partial \ddot{y}}{\partial A_1} + A_1 \frac{\partial \dot{y}}{\partial A_1} + A_0 \frac{\partial y}{\partial A_1} = \dot{r} - \dot{y}$$

$$2.10c \quad \frac{\partial \bar{y}}{\partial A_2} + A_2 \frac{\partial \ddot{y}}{\partial A_2} + A_1 \frac{\partial \dot{y}}{\partial A_2} + A_0 \frac{\partial y}{\partial A_2} = \ddot{r} - \ddot{y}$$

Defining u as $\partial y / \partial A_0$, v as $\partial y / \partial A_1$, and w as $\partial y / \partial A_2$, and taking the derivatives of u , v , and w with respect to time (assuming the order of differentiation can be interchanged) gives:

$$2.11a \quad u = \frac{\partial y}{\partial A_0} \quad \dot{u} = \frac{\partial \dot{y}}{\partial A_0} \quad \ddot{u} = \frac{\partial \ddot{y}}{\partial A_0} \quad \ddot{\ddot{u}} = \frac{\partial \ddot{\ddot{y}}}{\partial A_0}$$

$$2.11b \quad v = \frac{\partial y}{\partial A_1} \quad \dot{v} = \frac{\partial \dot{y}}{\partial A_1} \quad \ddot{v} = \frac{\partial \ddot{y}}{\partial A_1} \quad \ddot{\ddot{v}} = \frac{\partial \ddot{\ddot{y}}}{\partial A_1}$$

$$2.11c \quad \dot{w} = \frac{\partial \dot{u}}{\partial A_2} \quad \dot{w} = \frac{\partial \dot{v}}{\partial A_2} \quad \ddot{w} = \frac{\partial \ddot{u}}{\partial A_2} \quad \ddot{w} = \frac{\partial \ddot{v}}{\partial A_2}$$

This set of equations is substituted into equations 2.8a, 2.8b, and 2.8c and also into equations 2.10a, 2.10b, and 2.10c. The resulting two sets of equations are:

$$2.12a \quad \dot{a}_0 = -q_0 e u - q_1 \dot{e} \dot{u} - q_2 \ddot{e} \ddot{u} - q_3 \ddot{e} \ddot{u}$$

$$2.12b \quad \dot{a}_1 = -q_0 e v - q_1 \dot{e} \dot{v} - q_2 \ddot{e} \ddot{v} - q_3 \ddot{e} \ddot{v}$$

$$2.12c \quad \dot{a}_2 = -q_0 e w - q_1 \dot{e} \dot{w} - q_2 \ddot{e} \ddot{w} - q_3 \ddot{e} \ddot{w}$$

and

$$2.13a \quad \ddot{u} + A_2 \ddot{u} + A_1 \dot{u} + A_0 u = r - y$$

$$2.13b \quad \ddot{v} + A_2 \ddot{v} + A_1 \dot{v} + A_0 v = \dot{r} - \dot{y}$$

$$2.13c \quad \ddot{w} + A_2 \ddot{w} + A_1 \dot{w} + A_0 w = \ddot{r} - \ddot{y}$$

Previously a_0 , a_1 , and a_2 were defined in terms of the system parameters K_p , K_i , K_d , and the simplified robot model parameters K_m/J .

Taking the derivative with respect to time of these expressions gives:

$$2.14a \quad \dot{a}_0 = K_m \cdot \frac{\dot{K}_i}{J}$$

$$2.14b \quad \dot{a}_1 = K_m \cdot \frac{\dot{K}_p}{J}$$

$$2.14c \quad \dot{a}_2 = K_m \cdot \frac{\dot{K}_d}{J}$$

K_m and J are considered to vary slowly with respect to K_i , K_p , and K_d and are therefore considered constant when taking the derivative with respect to time. Solving for K_m/J in terms of a_0 , a_1 , a_2 , and the system gains and substituting these expressions into the equations 2.14a, 2.14b, and 2.14c yields:

$$2.15a \quad \dot{K}_i = \frac{\dot{a}_0}{a_0} K_i$$

$$2.15b \quad \dot{K}_p = \frac{\dot{a}_1}{a_1} K_p$$

$$2.15c \quad \dot{K}_d = \frac{\dot{a}_2}{a_2} K_d$$

Assuming that the adaptive mechanism adjusts the system gains so that $a_0 \approx A_0$, $a_1 \approx A_1$, and $a_2 \approx A_2$, then the values of a_0 , a_1 , and a_2 can be replaced by the constants A_0 , A_1 , and A_2 . This gives:

$$2.16a \quad \dot{K}_i = \frac{\dot{a}_0}{A_0} K_i$$

$$2.16b \quad \dot{K}_p = \frac{\dot{a}_1}{A_1} K_p$$

$$2.16c \quad \dot{K}_d = \frac{\dot{a}_2}{A_2} K_d$$

All the equations necessary for implementing the model reference adaptive control are derived. In order to implement the system, the

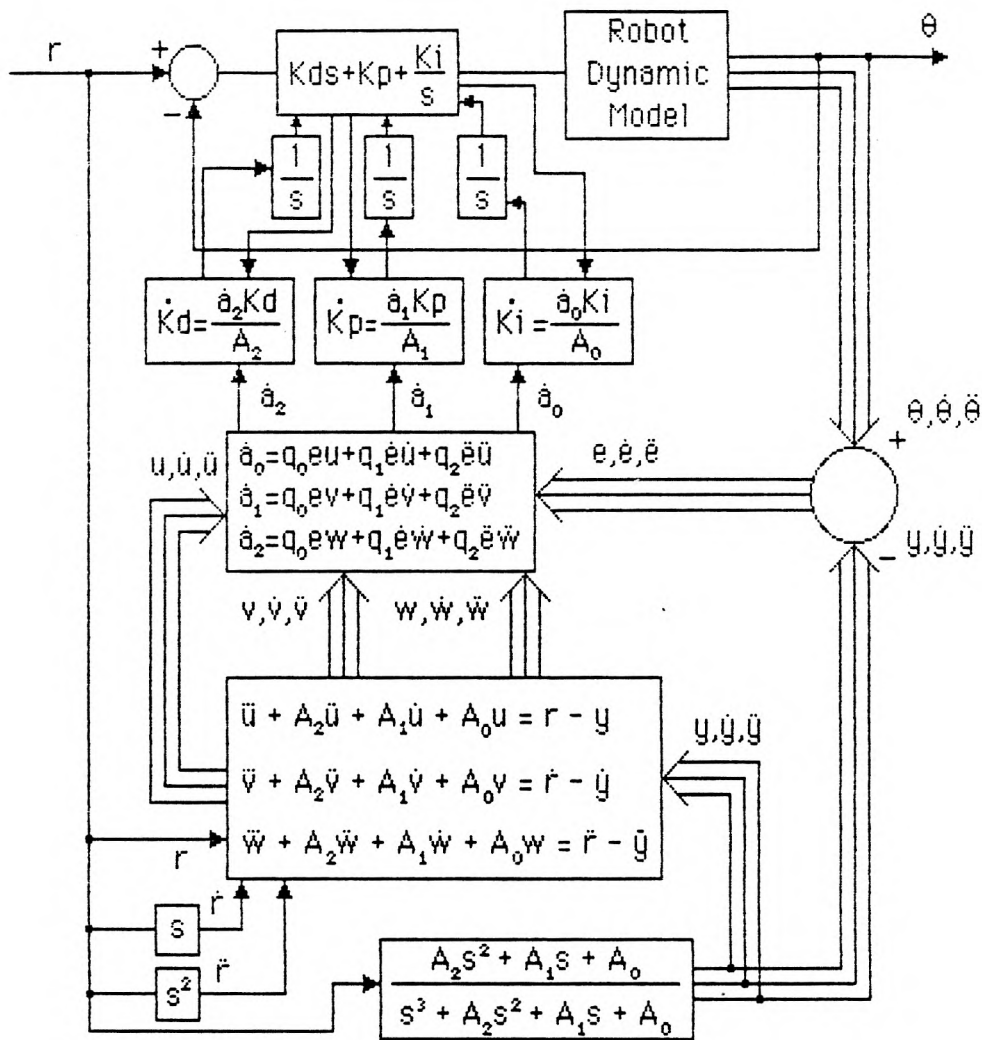


FIGURE 8. Control System Schematic

following procedure is followed (see Figure 8).

1. The command input, r , is fed into the control system and into the model where the model response to the input is calculated.
2. The system response to the input is compared with that of the model, generating the e 's used in the error function.
3. The output from the model along with the derivatives of the input

are used as forcing functions to compute the values of u , v , and w and their derivatives.

4. The values of u , v , and w and their derivatives along with the error signals (e 's) and the error weighting factors (q 's) are used to solve for \dot{a}_0 , \dot{a}_1 , and \dot{a}_2 .
5. Knowing the values of \dot{a}_0 , \dot{a}_1 , and \dot{a}_2 and knowing the present values of K_i , K_p , and K_d , the adjustment rates of the gains are computed.
6. Integrating \dot{K}_i , \dot{K}_p , and \dot{K}_d yields K_i , K_p , and K_d .

The implementation of this in the computer (along with the dynamic equations developed for the robot in the previous chapter) is done to test the equations developed. A model is chosen, thus defining the values of A_0 , A_1 , and A_2 . In choosing the model, a pair of complex conjugate poles are chosen so that a second order system described by them has a damping ratio of 0.9 and a natural frequency of 3 radians per second. A third pole is chosen to be located on the real axis at -10 so as to diminish its effects on the other two dominant poles. The zeros of the model are dictated by the values of A_0 , A_1 , and A_2 chosen in locating the poles. The transfer function of this model is:

$$2.17 \quad M(s) = \frac{15.4s^2 + 63s + 90}{s^3 + 15.4s^2 + 63s + 90}$$

The differential equation describing the model is:

$$2.18 \quad \ddot{y} + 15.4\dot{y} + 63y = 15.4\ddot{r} + 63\dot{r} + 90r$$

In order to calculate the response of the model, the differential equation describing the model is written in state variable form and numerically integrated using a Runge Kutta Gill numerical integration. The form of the state variable equation is:

$$2.19 \quad \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -A_0 & -A_1 & -A_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} A_2 \\ A_1 \\ A_0 - 2A_1A_2 \end{bmatrix} r - \begin{bmatrix} 0 \\ 0 \\ A_2^2 \end{bmatrix} \dot{r}$$

where $A_0 = 90$, $A_1 = 63$, and $A_2 = 15.4$

The output equations are:

$$2.20 \quad y = x_1$$

$$\dot{y} = x_2 + A_2 r$$

$$\ddot{y} = x_3 + A_1 r + A_2 \dot{r}$$

It is noted that there is no output equation given for \ddot{y} . Because the value of \ddot{y} that would be returned by the state equations would be error prone, and due to the fact that it would be difficult to obtain \ddot{x} from the

robot, the error weighting factor q_3 is set to zero. This eliminates the need for both \bar{y} and \bar{x} in the equations used by the adjustment mechanism.

The results of the simulation follow. A standard move sequence is used in all the simulations that follow. The command input, r , given in Table 4 shows the move sequence the robot is commanded to execute. All simulations are for the torso axis of the robot only, the upper-arm and forearm of the robot are extended straight out with gravity offset torques at their respective joints.

TABLE 4

Time (sec) from	to	Input Signal	Learning Signal
0	5	0	Yes
5	6	Ramp, +1 slope	No
6	10	1	No
10	15	1	Yes
15	16	Ramp, -1 slope	No
16	21	0	Yes

Table 4 shows that the robot torso is commanded to move to a displacement of 1 radian at a velocity of 1 radian per second and to hold

there for 4 seconds. After 4 seconds have passed, a payload can be specified for the robot to pick up, after which the robot is given 5 seconds to adjust to the new load. The robot is then commanded to move back to its starting position at a velocity of -1 radian per second. The last column of the table indicates whether or not the robot is being given a "learning signal". A learning signal is a high frequency, low amplitude signal which is added to the command input. The learning signal creates a small perturbation which induces gain adjustment activity in the adjusting mechanism. The learning signal is useful in heightening the adaptive mechanism's sensitivity to changes in the robot's physical status.

To test the performance of the adjustment mechanism, a simulation is made in which the final proper values of the system gains are known. Instead of using the dynamic model of the robot developed in chapter 1, the simplified model of the robot used in the derivation of the adjustment mechanism is used in the simulation. The simplified robot model is K_m/J_s^2 , where K_m is the motor torque constant and J is the inertia. It is assumed that K_m is constant and the inertia about the torso axis will also remain constant (i.e. no payload will be added to the robot arm during the move and the forearm and upper-arm assemblies of the robot will maintain in fixed

positions during the move). By specifying the ratio of K_m/J to be 1, the proper values of K_i , K_p , and K_d correspond to the values of A_0 , A_1 , and A_2 respectively.

Figure 9 represents the response of the ideal system in comparison to the model response when the system gains are preset 10% above their proper values. The standard move was executed with the following parameters: learning signal frequency of 30 hz, amplitude .0035 radians, no payload picked up by the robot, $q_0=10$, $q_1=7$, and $q_2=1.711$. It is the intent of the model reference adaptive control system to cause the response of a system to conform to a desired model response. Figure 9 shows a slight difference in the response of the system and the model at the end of the first ramp up. This, however, is the only noticeable deviation between the system and the model, and therefore it could be said that the model reference adaptive control system did what it was intended to do.

Inspection of the gain adjustment activity, however, reveals that the system gains were not driven to their proper values. The derivative gain, K_d , adjusted to within +8% of its value, the proportional gain, K_p , to within +9.99% of its value, and the integral gain, K_i , did not adjust at all. The fact that there was so little gain adjustment could be because the q values were

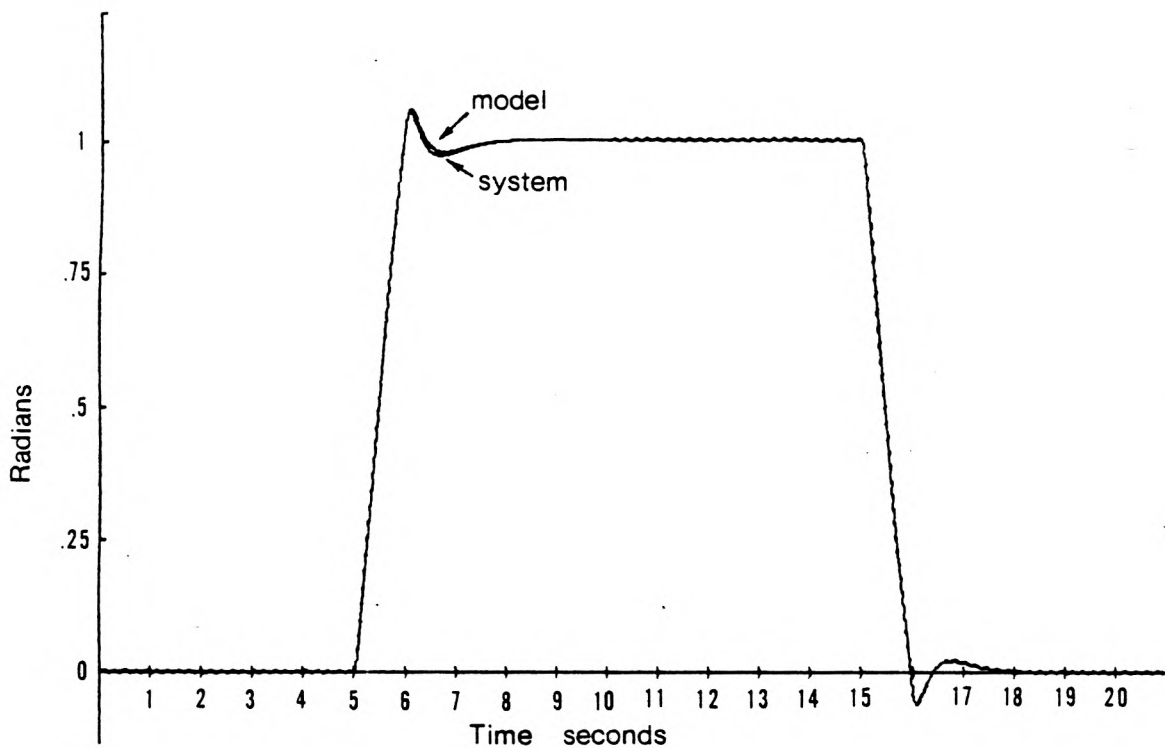


FIGURE 9. Ideal System Response vs Model Response.

too low. Another test was made with $q_0=200$, $q_1=140$, and $q_2=34.22$. Figure 10 is a plot of the response of the system and the model, and Figure 11 is a plot of the gain adjustments. The response of the system in this test is much worse than the previous test. The gains, however, did show more adjustment activity. The derivative gain, K_d , adjusted to within -0.17% of its value, the proportional gain, K_p , to within $+8.8\%$ of its value, and the integral gain, K_i , to within $+9.9\%$ of its value. It appears that the proportional and integral gains are adjusting in the proper direction, but don't adjust rapidly enough to make it to their correct values. The response of the system in this test when compared to Figure 9, though, clearly shows

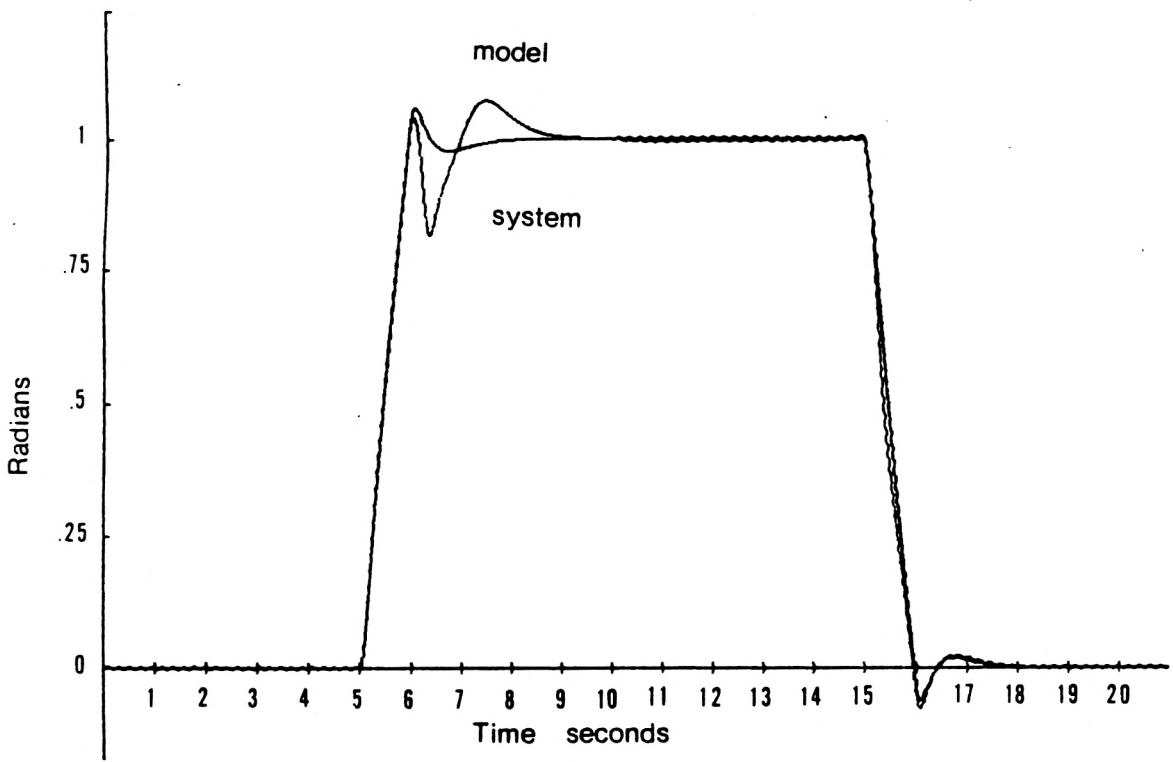


FIGURE 10. System Response vs Model Response with Increased q Values

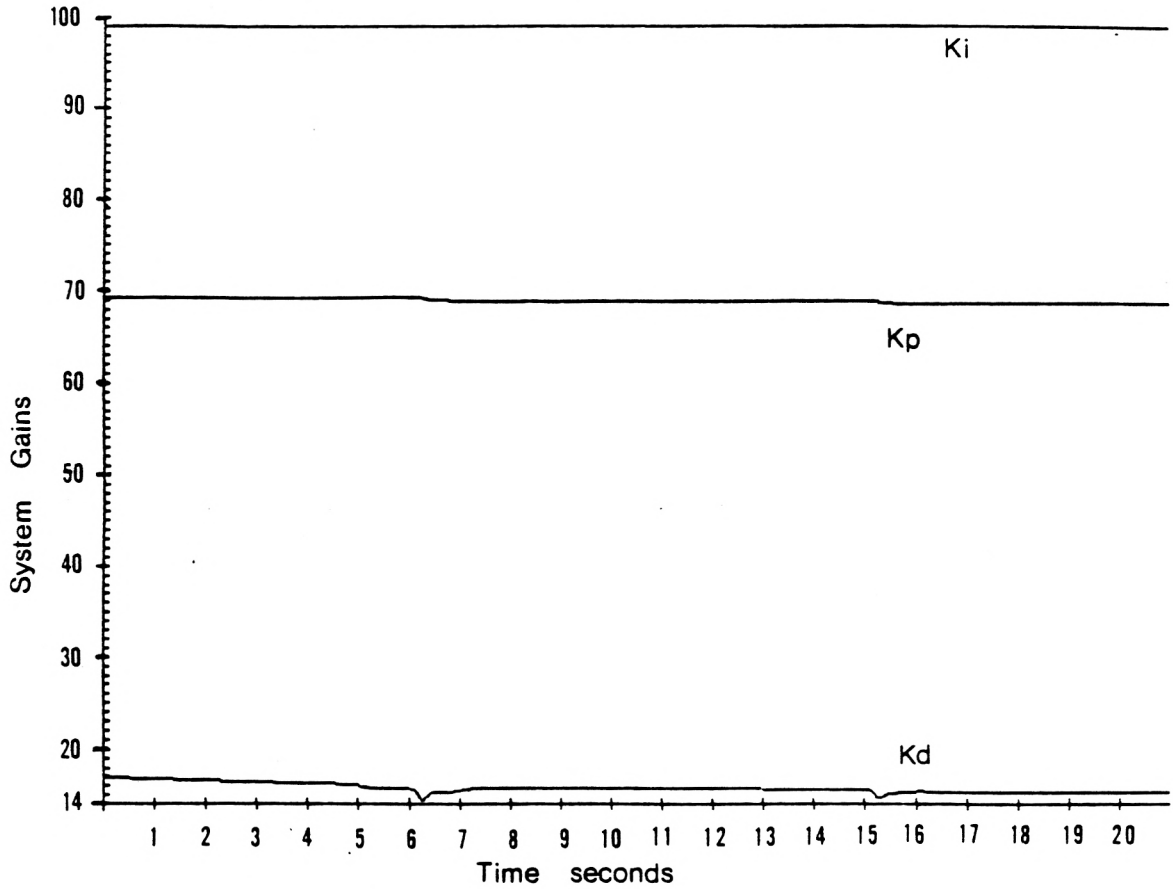


FIGURE 11. System Gain Adjustments with Increased q Values.

that the first q values produced the best system response. Even though the system gains were not driven to their correct values in Figure 9, this does not indicate that the adaptive mechanism failed. On the contrary. The results of the first test showed that the system did respond in the desired manner (i.e. the system response was nearly identical to that of the model).

In examining the adaptive mechanism, the gain adjustments are driven by an error signal generated by the difference between the system and model response. Equation 2.6 shows that the error function is a positive definite function and the Euclidian Space formed by the function $f(e)$ has only one minimum point and has a positive slope at all other points. If the system response at some time is different than that of the model, then the value of $f(e)$ on the $f(e)$ space lies away from the minimum point and the slope of the $f(e)$ space at that point is negative. The gain adjustment equations use an approximation of the slope of the $f(e)$ space at that point to calculate a gain adjustment that is based on a steepest descent trajectory, driving the value of $f(e)$ toward its minimum value of 0.

Therefore, if the system is behaving like the model, then the values of e are small and the value of $f(e)$ is near its minimum. With this in mind, the behavior of the system gains in Figure 9 can be explained by the fact that the

difference between the system response and model response was small, therefore, little gain adjustment occurred.

Figures 12 and 13 represent a test of the gain adaption mechanism using the robot dynamic equations to compute the response of the robot. The standard move is executed with the following parameters: learning signal frequency of 30 hz, amplitude of .0035 radians, a payload of 2 slugs is picked up, $q_0=200$, $q_1=140$, and $q_2=34.22$. The initial values of gains are $K_p=98.19$, $K_i=140.26$, and $K_d=24$ (these values of gain make the closed loop transfer function coefficients directly proportional to those of the model).

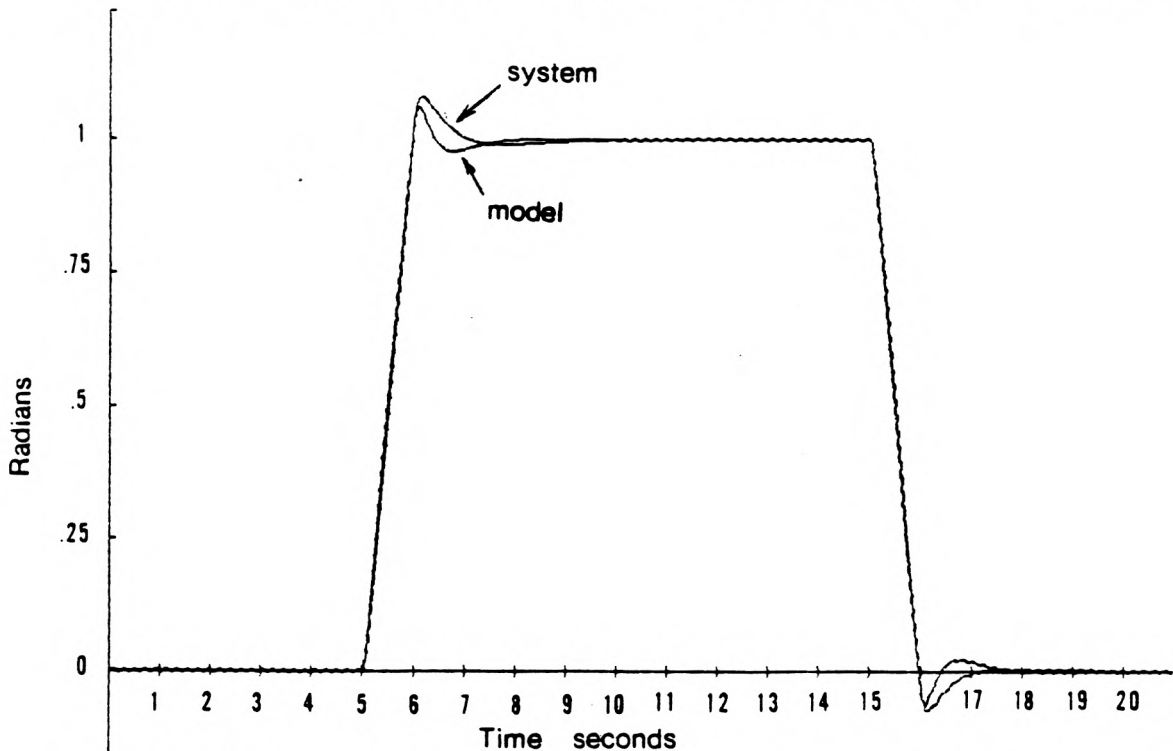


FIGURE 12. System Response vs Model Response using Robot Dynamic Equations.

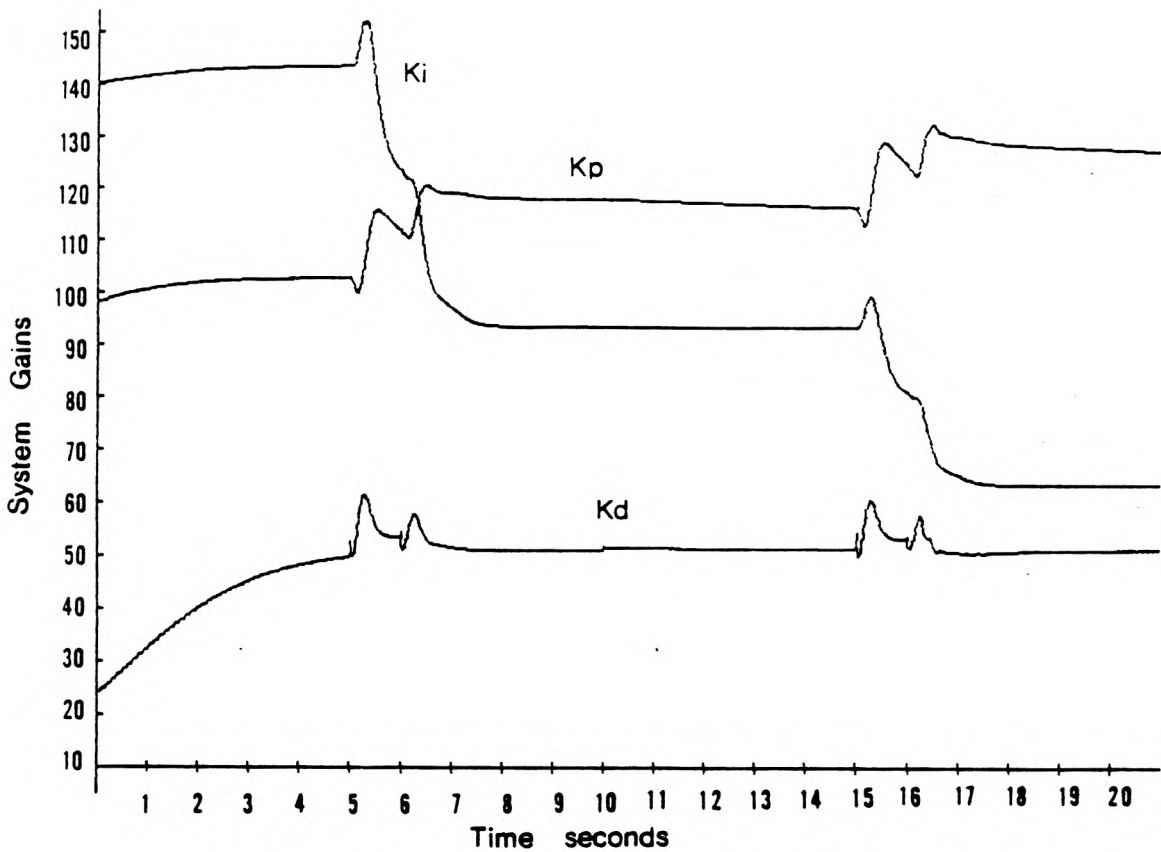


FIGURE 13. System Gain Adjustments using Robot Dynamic Equations.

Figure 12 shows that the robot's response is generally of the same form as that of the model. The robot tends to experience more overshoot than the model but it recovers well and has a good settling time comparable to that of the model. Another note is that the robot shows a consistent response when comparing its motion with and without a payload. This supports the control system's ability to adapt to a changed physical condition and maintain consistent performance. The gains of the system did not adjust in a predictable manner that was expected with the addition of a payload, however, the gains of the system are adjusted so as to minimize

the error function which is a function of the $f(e)$ space.

Based on these results, the PID control system responded in a manner consistent with the expectations of the model reference adaptive control system. Although perfect conformity to the model response was not achieved, the performance was consistent when tested under changing physical conditions.

Proportional Control with Derivative Feedback

Figure 14 is a diagram of the control system using proportional control with derivative feedback. The development of the gain adaption mechanism is exactly the same as for the PID control, with the only difference being that the simplified model of the robot used is $K_m/s(Js+f)$.

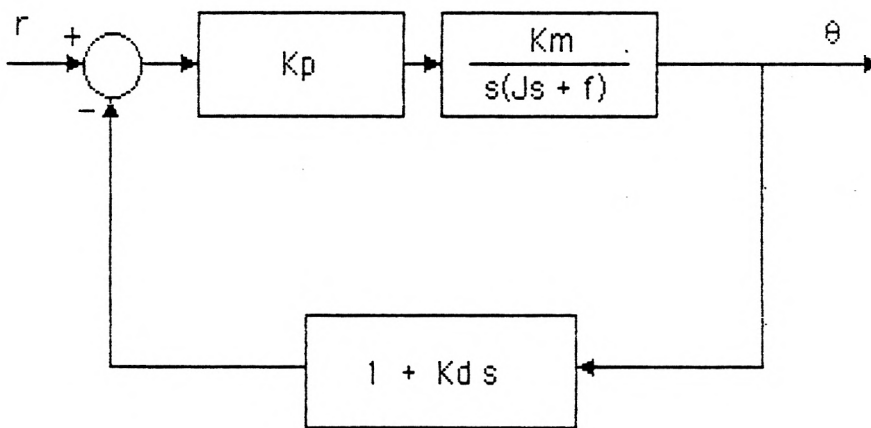


FIGURE 14. Proportional Control with Derivative Feedback.

The closed loop transfer function of the system is:

$$2.21 \quad T(s) = \frac{1}{a_2 s^2 + a_1 s + 1}$$

$$\text{where } a_2 = \frac{J}{K_m K_p}$$

$$a_1 = \frac{f}{K_m K_p} + K_d$$

and where K_p is the proportional gain, K_d is the derivative

feedback gain, f is the viscous friction factor, J is the inertia of the system, and K_m is the motor torque constant.

The transfer function of the model is:

$$2.22 \quad M(s) = \frac{1}{A_2 s^2 + A_1 s + 1}$$

where A_1 and A_2 are constant coefficients chosen to give the model the desired response.

The error equation to be minimized is:

$$2.23 \quad f(e) = \frac{[q_0 e^2 + q_1 \dot{e}^2 + q_2 \ddot{e}^2]}{2}$$

where $e = \theta - y$

θ is the output of the system

and y is the output of the model

The rates at which the coefficients of the system adjust are:

$$2.24a \quad \dot{a}_1 = \frac{\partial f(e)}{\partial A_1} = -q_0 e u - q_1 \dot{e} u - q_2 \ddot{e} u$$

$$2.24b \quad \dot{a}_2 = \frac{\partial f(e)}{\partial A_2} = -q_0 e v - q_1 \dot{e} v - q_2 \ddot{e} v$$

where $u = \frac{\partial y}{\partial A_1}$

and $v = \frac{\partial y}{\partial A_2}$

The equations for calculating u and v and their derivatives are:

$$2.25a \quad A_2 \ddot{u} + A_1 \dot{u} + u = -\ddot{y}$$

$$2.25b \quad A_2 \ddot{v} + A_1 \dot{v} + v = -\ddot{y}$$

Taking the derivatives of the coefficients of the system closed loop transfer function gives:

$$2.26a \quad \dot{a}_2 = \frac{-J \cdot \dot{K}_D}{K_m K_p^2}$$

$$2.26b \quad \dot{a}_1 = \frac{-f \cdot \dot{K}_D}{K_m K_p^2} + \dot{K}_D$$

Solving for the derivatives of the gains yields:

$$2.27a \quad \dot{K}_D = \frac{-\dot{a}_2 K_D}{A_2}$$

$$2.27b \quad \dot{K}_D = \dot{a}_1 - \frac{\dot{a}_1 (A_1 - K_D)}{A_2}$$

The coefficients of the model transfer function are chosen so that the second order system described by them has a natural frequency of 3 radians per second and a damping ratio of 0.9. The model transfer function is:

$$2.28 \quad M(s) = \frac{1}{0.111s^2 + 0.6s + 1}$$

To test the gain adaption equations, an ideal system of the form $K_m/s(Js+f)$ is used to represent the robot. By setting K_m and J equal to 1

and f equal to zero, the proper values of the system parameters, K_p and K_d , can be determined.

Figures 15 and 16 represent a test of an ideal system with the system parameters initially off by +10%. The standard move is executed with the following parameters: learning signal of frequency 30 hz and amplitude of 0.035 radians, no payload was picked up by the robot, $q_0=1$, $q_1=0.6$, and $q_2=0.111$. The response of the model and the system are very close. The model rises quicker than the system on the first ramp, but after that, the responses are identical. The plots of the gain adjustments in Figure 16 shows that both gains are driven to their proper values during the course of

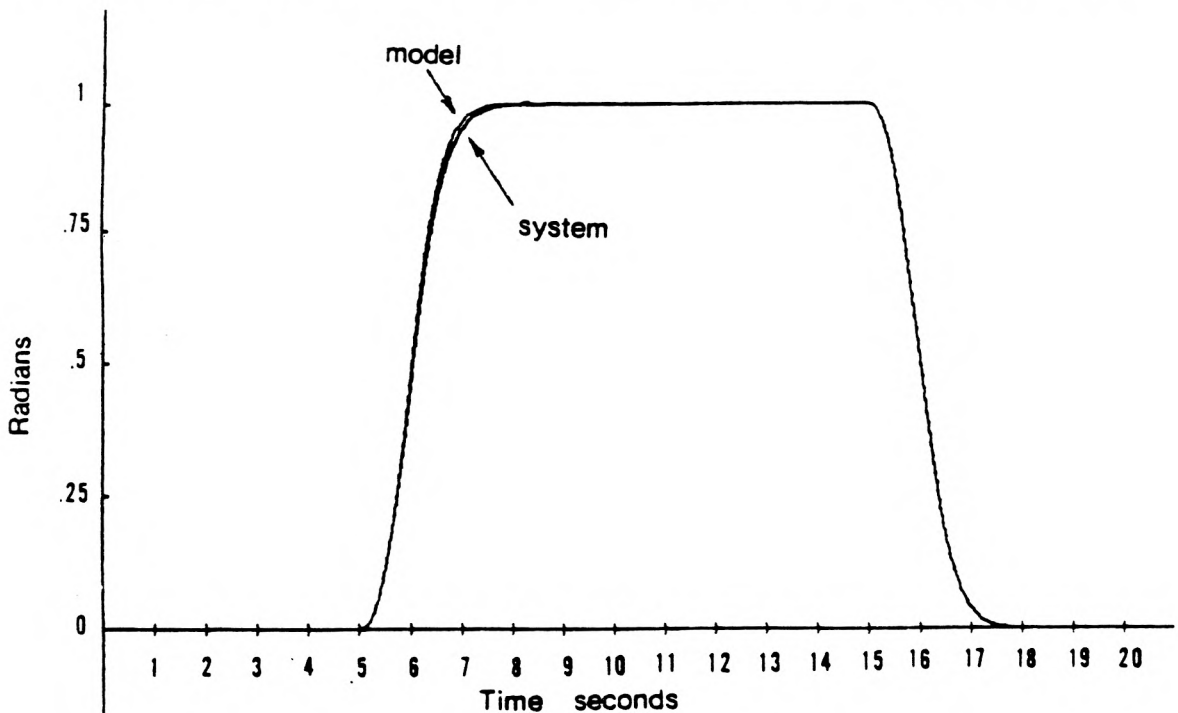


FIGURE 15. Ideal System Response vs Model Response.

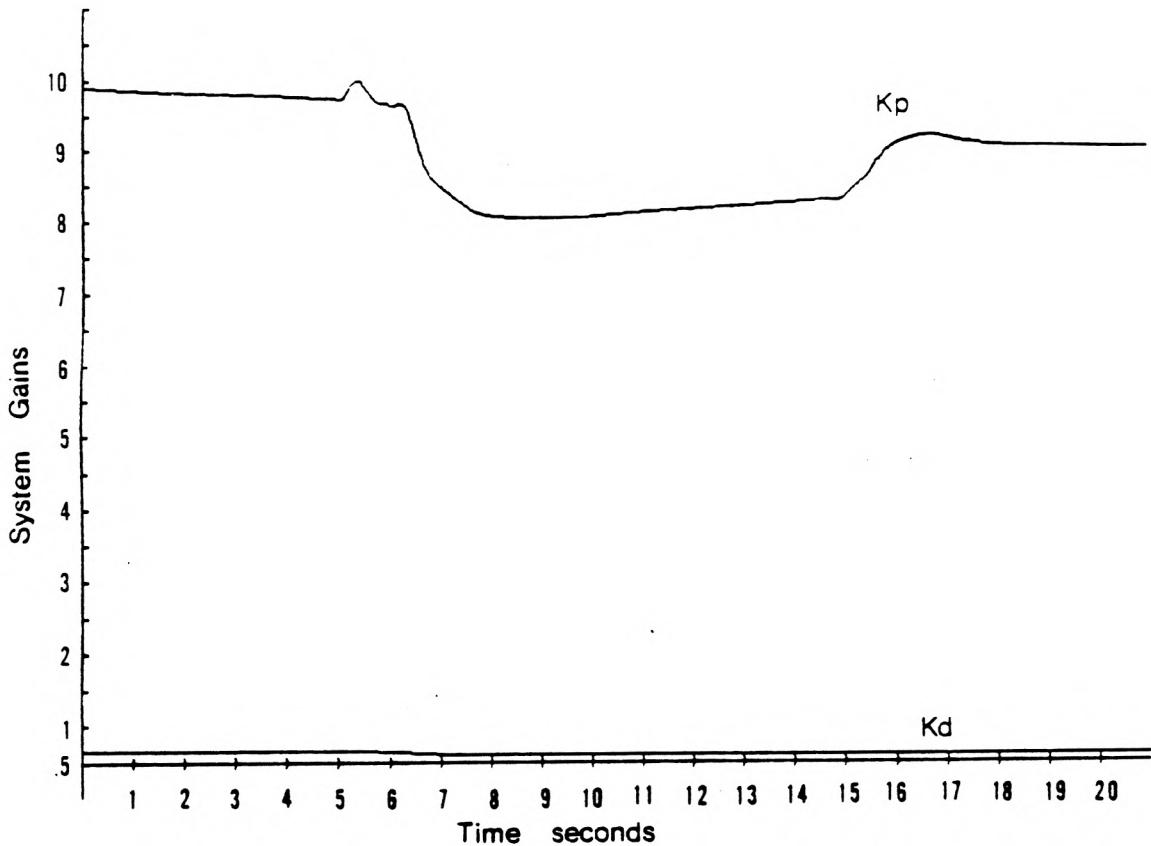


FIGURE 16. System Gain Adjustments with Ideal System.

the move. Another test, with the gains initially off by +70% is shown in Figures 17 and 18. Figure 17 shows that initially the response of the system is greatly different from the model, but on the second ramp down, the responses, again, are nearly identical. The gains in this test did not exactly reach their proper values, but it appears that the adjustment mechanism is driving them toward those values.

Figures 19 and 20 show the results of the test of the gain adaption mechanism using the robot dynamic equations. The initial values of the gains are; $K_p=13.068$, and $K_d=0.6$. The q values are; $q_0=1$, $q_1=0.6$, and

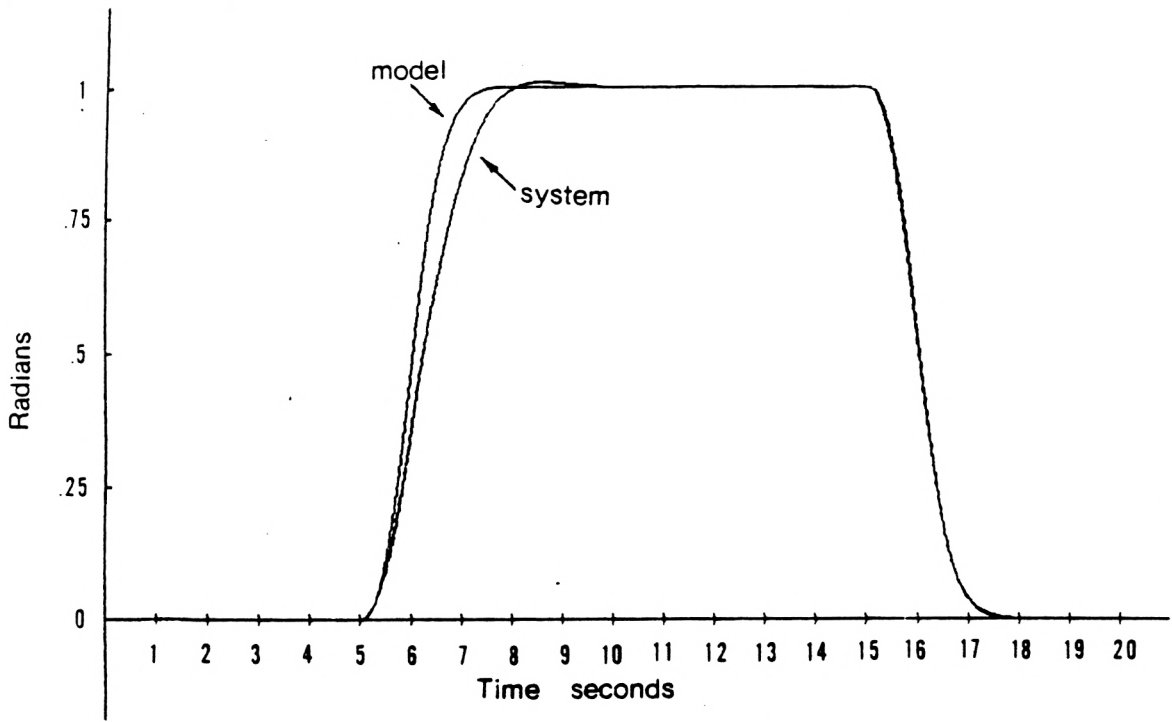


FIGURE 17. Ideal System Response vs Model Response.

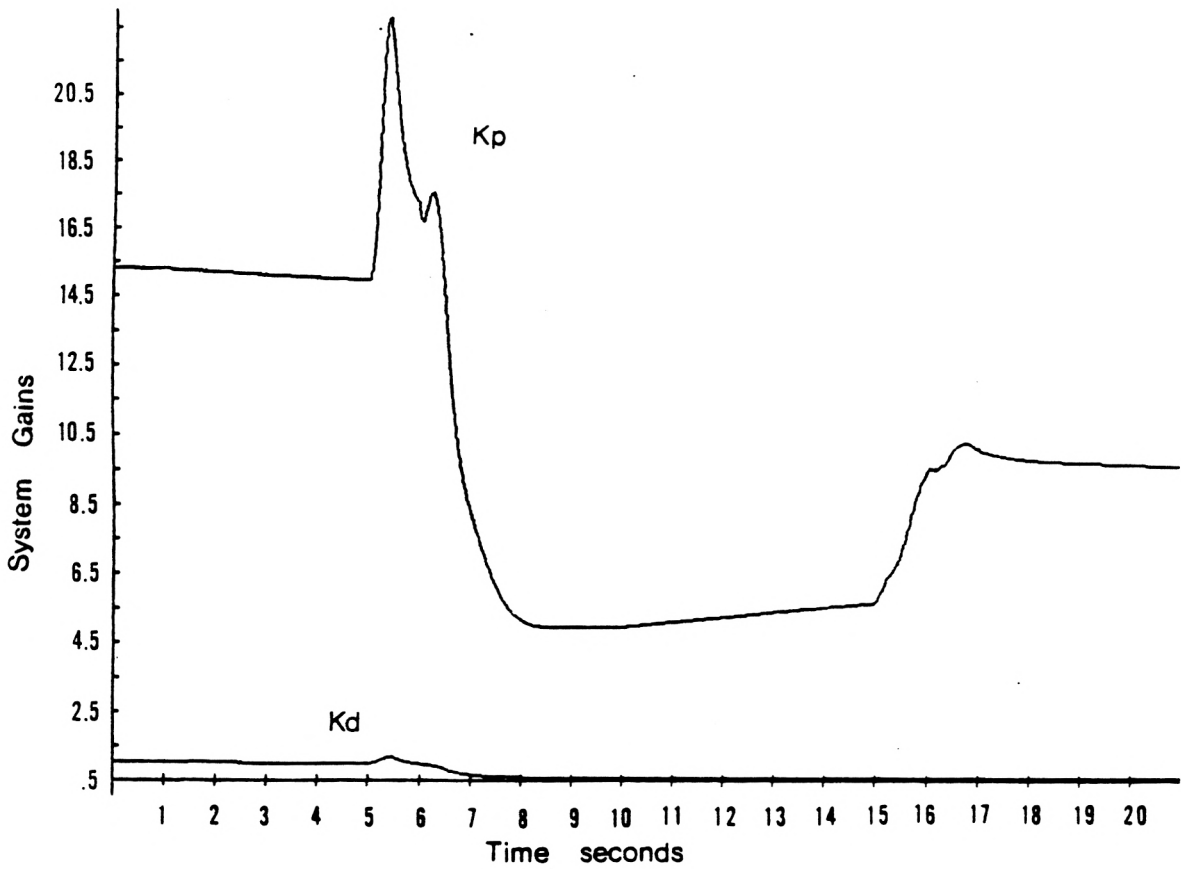


FIGURE 18. System Gain Adjustments using Ideal System.

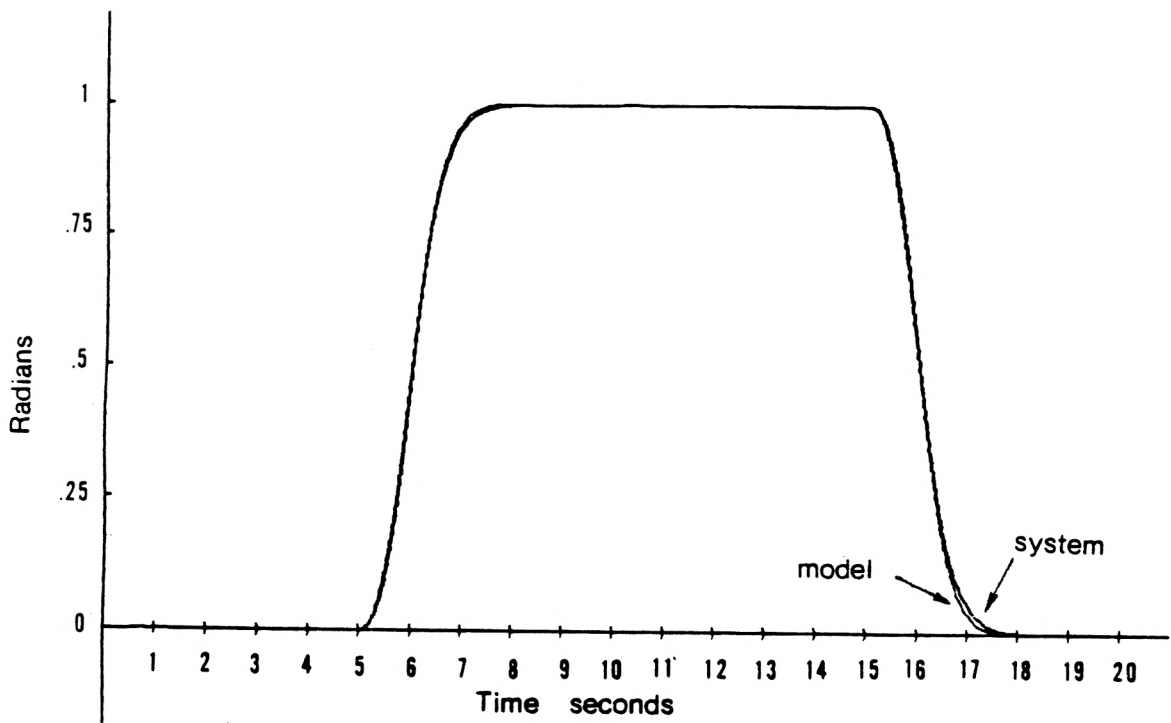


FIGURE 19. System Response vs Model Response using Robot Dynamic Equations.

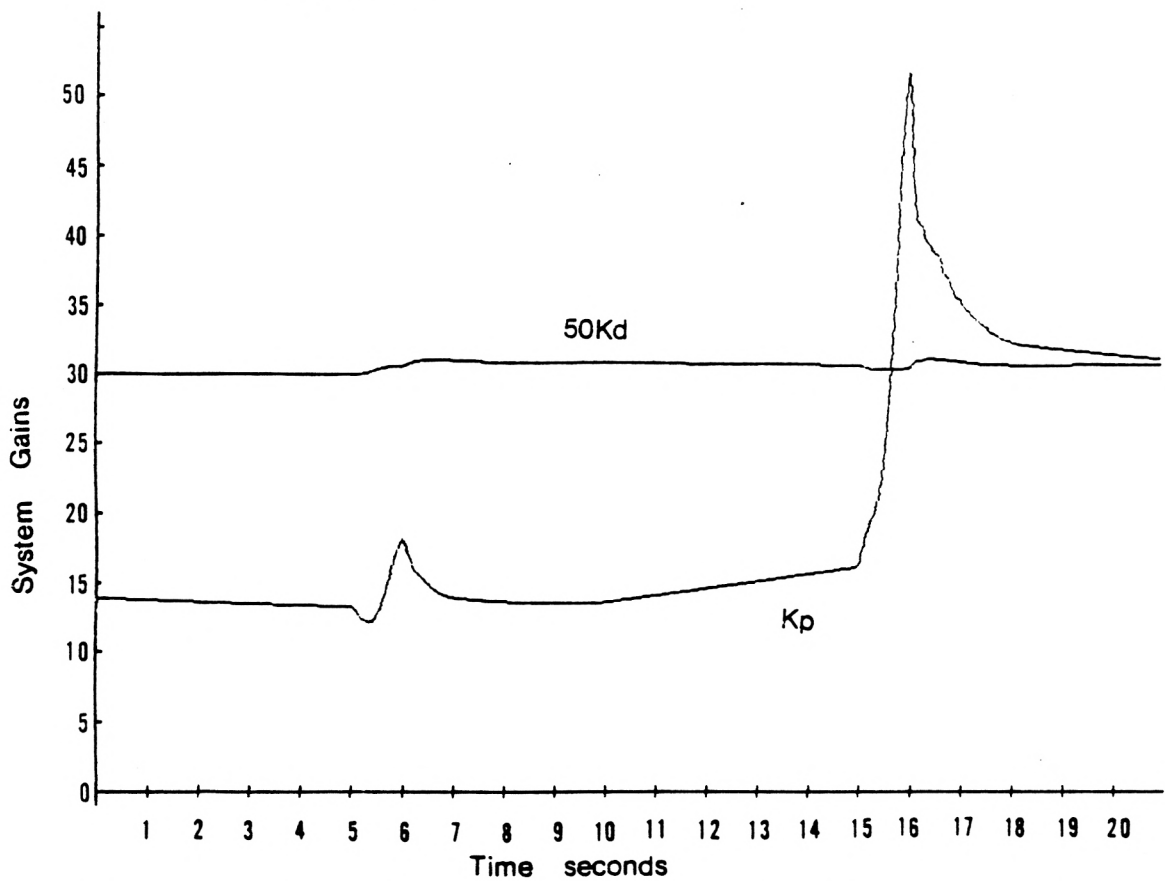


FIGURE 20. System Gain Adjustments using Robot Dynamic Equations.

$q_2=0.111$. At time equals 10 seconds, a payload of 2 slugs is added, causing the gains to adjust in order to maintain proper system response. Figure 19 shows that the response of the system is very close to that of the model, with the exception that the system has a bit more lag than the model after the second ramp.

This concludes the results of the simulations made with the various control configurations on the torso axis. In the next section, some simulations are made with all three axes of the robot being controlled.

Three Axis Control

The three axis simulation of the robot is the final test of the model referenced adaptive control system. Again, a standard move is executed by the robot. The move consists of a tuck, load, and rollout sequence. For the tuck portion of the move, θ_1 , θ_2 , and θ_3 are commanded to simultaneously ramp at a rate of 1 radian/second for 1 second from an initial displacement of 0. This causes the elbow to move up from 0 degrees to 57.3 degrees (1 radian), the shoulder to move up from 0 degrees to 57.3 degrees (1 radian), and the torso to swing from 0 degrees to 57.3 degrees (1 radian). A load of 2 slugs is added to the robot, and then, for the rollout portion of the move, the reverse motion of the tuck is performed.

For the three axis simulation, the proportional with derivative feedback control configuration is used for all three axes. The results are qualitative in nature, and are concerned primarily with the form of the response curves. Gain adjustment data was analyzed, but it is beyond the scope of this paper to adequately evaluate those results.

Figure 21 shows the response of the three axes with gains fixed. The shoulder axis is affected the most by the inertial changes followed by the elbow and then the torso. Figure 22 shows the response of the three axes

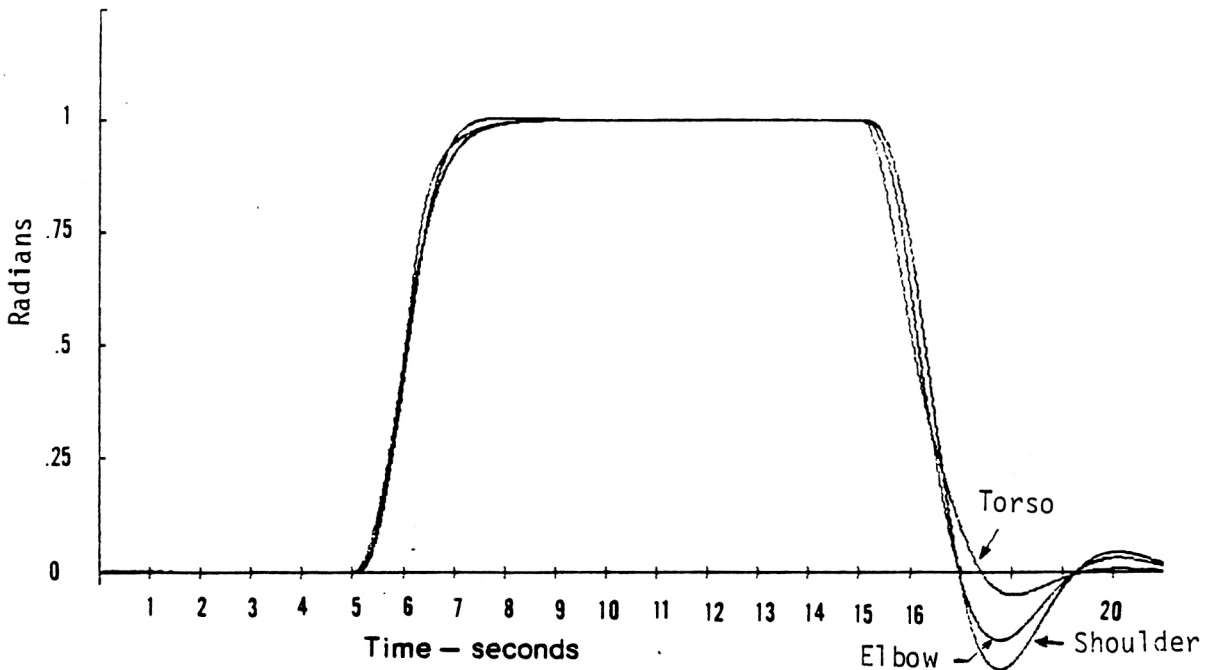


FIGURE 21. System Response with Fixed Gains on All Axes.

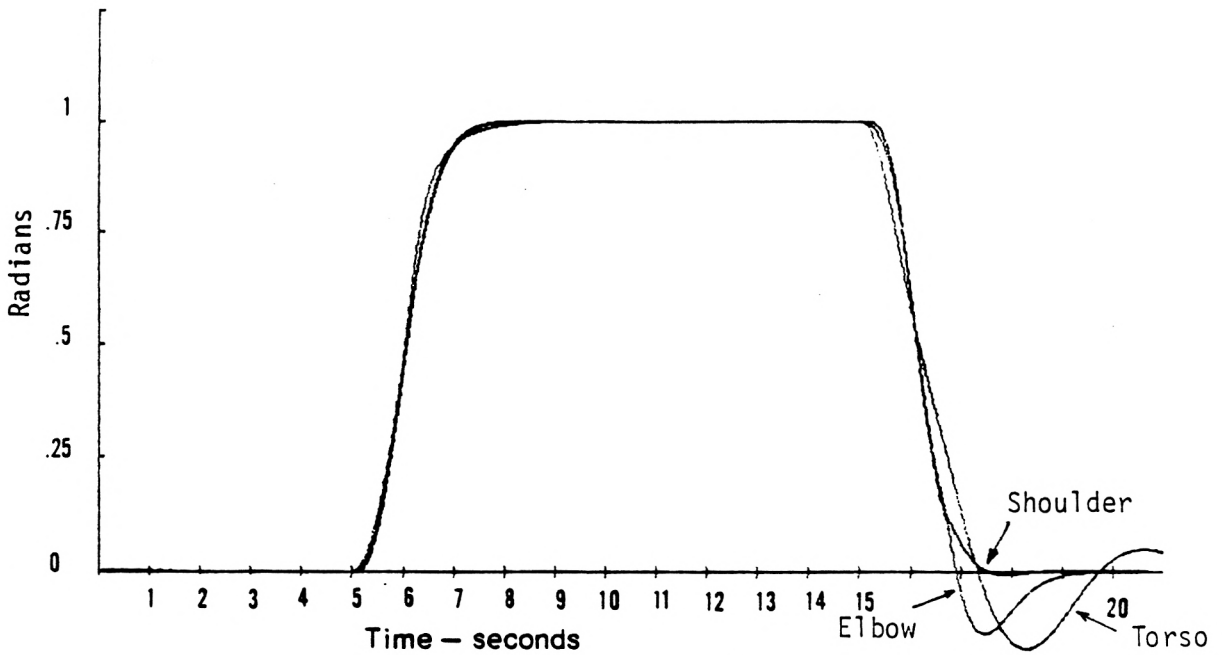


FIGURE 22. System Response with Model Referenced Adaptive Control on All Axes.

with model referenced adaptive control on all three axes. The response of the shoulder axis is greatly improved by the adaptive control, however, the torso's response is greatly degraded when compared to the fixed gain simulation. The elbow's response was somewhat better with adaptive control.

Figure 23 shows the response of the torso compared to that of the model with the gains fixed on the shoulder and elbow, and model referenced adaptive control on the torso. This shows a greatly improved response for

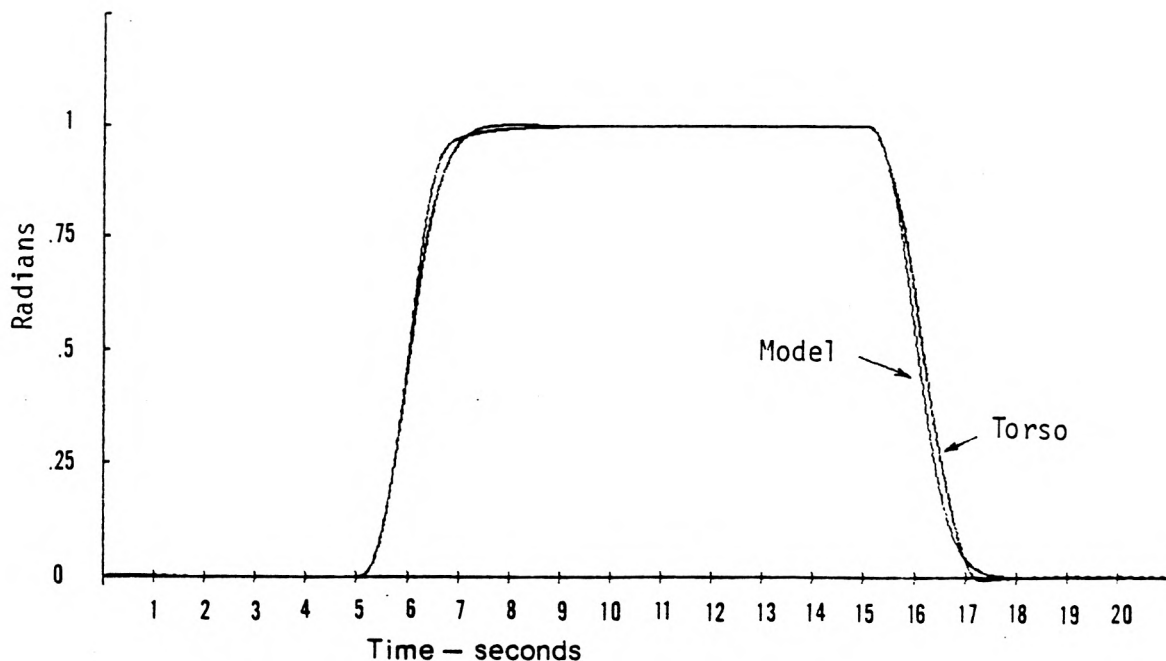


FIGURE 23. Torso Response with Model Referenced Adaptive Control on Torso Axis and Fixed Gains on All Other Axes.

the torso over that of Figure 22.

It was the overall case that when model reference adaptive control is added to all three axes, the adaptive algorithm does not yield good response from all axes. In addition to this, the q value selection becomes much more critical. Numerous q values were tested that gave unstable response from one or more of the axes involved in the simulation. Selecting q values that produced good response in each axis separately might well cause the system on the whole to go unstable.

This concludes the results of the tests of the model referenced adaptive control system. In the next chapter, some observations and conclusions are made about these results with recommendations for further study.

III. OBSERVATIONS AND CONCLUSIONS

The results presented in Chapter 2 represent the best results obtained from the research for each control system. During the course of investigating each of the different systems, several observations were made which were not included in the presentation of the results.

The tests of the different control systems shows that the control system configuration has a great deal to do with the way in which the adaptive mechanism performs. This is not surprising, since the development of the adaptive algorithm is different for each control configuration. Certainly there are some control configurations that are more suited to, and respond better to this method of model reference adaptive control than others.

Mathematically speaking, the shape of the $f(e)$ space determines the behavior of the adjusting mechanism. In considering a comparison between the PID control and the Proportional Control with Derivative Feedback, the latter system's gain adjustments performed much better than those of the PID control system. This, though, was due primarily to the differences in

their $f(e)$ spaces. The $f(e)$ space for the PID control system was apparently much flatter in places than that of the Proportional with Derivative Feedback system.

Another observed effect on the performance of the adjusting algorithm was the manner in which the adjusting equations are derived for each control system. Depending on the control system, there is usually more than one way to derive the adjustment rate equations. In the first attempt to derive gain adjustment equations for the Proportional with Derivative Feedback system, the transfer function was divided through by the coefficient of the second order term, J . This is in contrast to the form of the transfer function presented in Chapter 2. The tests of the gain adaption equations derived from this form of the transfer function were unsuccessful in adapting the gains.

In the development of the gain adjustment equations, a simplified model of the robot is used. The dynamic equations of the robot contain nonlinear terms of the form $\dot{\theta}^2$ and $\dot{\theta}_i \dot{\theta}_j$, but no linear first order terms are contained in these equations. In the case of the Proportional with Derivative Feedback system, the gain adjustment equations are derived using a simplified model of the robot of the form $K_m/(Js^2+fs)$. This form treats

the nonlinear first order terms as being linear for the purpose of developing the gain adjustment equations. If viscous friction is incorporated into the dynamic equations of the robot, the terms representing it are linear first order terms. Adding viscous friction terms to the dynamic equations, though, has no noticeable effect on the performance of the PID control or the Proportional with Derivative Feedback system.

During the testing of the different systems, there was a lot of juggling of the q values used. The process of trying to find a set of q 's that yielded the best response from the adaptive mechanism led to a selection based on a suggestion in reference [1]. There Donalson and Leondes suggest that $q_0=q_2a_2$, and $q_1=q_2a_1$. This makes the q values proportional to the coefficients of the closed loop transfer function. By maintaining these ratios among the q values, the best system performance was obtained.

Experimentation was also done on the amplitude of the learning signal being applied to the system. The Proportional with Derivative Feedback system's response to the learning signal increased significantly with an increase in amplitude. Learning signals with amplitudes as high as 5 degrees were tested. With large learning signals, the gains would adjust very quickly to their proper values. During the testing of large amplitude

learning signals, it was observed that after the first ramp input, when the input signal was 1, the proportional gain adjusted to an incorrect value and stayed there until the next ramp was input that brought the position back to zero. This steady state error of the gain value for a nonzero input was noticed regularly with the proportional and derivative gains of that system.

In conclusion, model reference adaptive control provides a viable means of approaching the complex problem of robotic manipulator control. The adaptive control mechanism contains only linear differential equations that can easily be solved on a real time basis by the control computer. The type of control configuration used must be carefully selected, though, as not all control systems are suited for this particular adaptive method.

Recommendations for Further Study

There are several alternative means of deriving gain adjustment equations for the purpose of model referenced adaptive control. One such method is sensitivity analysis.

Using the proportional with derivative feedback system, the following analysis can be made. The system equation is given as:

$$3.1 \quad T_s(s) = \frac{K_m K_D / J}{s^2 + (K_m K_D / J) s + K_m K_P / J} = \frac{a_0}{s^2 + a_1 s + a_0}$$

The model equation is given as:

$$3.2 \quad T_m(s) = \frac{A_0}{s^2 + A_1s + A_0}$$

And the index of performance is:

$$3.3 \quad I(a_0, a_1) = 1/2 [Q_0 e^2 + Q_1 \dot{e}^2]$$

$$\text{where } e = \theta - y$$

$$\dot{e} = \dot{\theta} - \dot{y}$$

It is desired that the values of a_0 and a_1 that minimize $I(a_0, a_1)$ be found.

Therefore the values where $\frac{\partial I}{\partial a_0} = F_0(a_0, a_1) \gg 0$ and $\frac{\partial I}{\partial a_1} = F_1(a_0, a_1) \gg 0$ must

be found, where:

$$3.4 \quad F_0 = Q_0 e \frac{\partial e}{\partial a_0} + Q_1 \dot{e} \frac{\partial \dot{e}}{\partial a_0}$$

$$3.5 \quad F_1 = Q_0 e \frac{\partial e}{\partial a_1} + Q_1 \dot{e} \frac{\partial \dot{e}}{\partial a_1}$$

Expanding equations 3.4 and 3.5 in a Taylor Series gives:

$$3.6 \quad F_0(a_0 + \delta a_0, a_1 + \delta a_1) = F_0(a_0, a_1) + \frac{\partial F_0}{\partial a_0} \delta a_0 + \frac{\partial F_0}{\partial a_1} \delta a_1 + \text{H.O.T.}$$

$$3.7 \quad F_1(a_0 + \delta a_0, a_1 + \delta a_1) = F_1(a_0, a_1) + \frac{\partial F_1}{\partial a_0} \delta a_0 + \frac{\partial F_1}{\partial a_1} \delta a_1 + \text{H.O.T.}$$

By neglecting the higher order terms (H.O.T) and setting equations 3.6 and 3.7 equal to 0, δa_0 and δa_1 can be solved for.

Let: $G_{00} = \frac{\partial F_0}{\partial a_0}$, $G_{01} = \frac{\partial F_0}{\partial a_1}$, $G_{10} = \frac{\partial F_1}{\partial a_0}$, and $G_{11} = \frac{\partial F_1}{\partial a_1}$

Then:
$$\begin{bmatrix} G_{00} & G_{01} \\ G_{10} & G_{11} \end{bmatrix} \begin{bmatrix} \delta a_0 \\ \delta a_1 \end{bmatrix} = \begin{bmatrix} -F_0 \\ -F_1 \end{bmatrix}$$

Therefore:

3.8
$$\delta a_0 = (F_1 G_{01} - F_0 G_{11}) / (G_{00} G_{11} - G_{10} G_{01})$$

3.9
$$\delta a_1 = (F_0 G_{10} - F_1 G_{00}) / (G_{00} G_{11} - G_{10} G_{01})$$

Let: $\frac{\partial y}{\partial A_0} = u_0$ and $\frac{\partial y}{\partial A_1} = u_1$ then:

3.10
$$F_0 = Q_0 e u_0 + Q_1 \dot{e} \dot{u}_0$$

3.10
$$F_1 = Q_0 e u_1 + Q_1 \dot{e} \dot{u}_1$$

3.11
$$G_{00} = Q_0 e v_{00} + Q_1 \dot{e} \dot{v}_{00} + Q_0 u_0^2 + Q_1 \dot{u}_0^2$$

3.12
$$G_{01} = G_{10} = Q_0 \dot{e} \dot{v}_{01} + Q_1 e v_{01} + Q_0 u_0 u_1 + Q_1 \dot{u}_0 \dot{u}_1$$

3.13
$$G_{11} = Q_0 e v_{11} + Q_1 \dot{e} \dot{v}_{11} + Q_0 u_1^2 + Q_1 \dot{u}_1^2$$

Where $\frac{\partial u_0}{\partial A_0} = v_{00}$, $\frac{\partial u_0}{\partial A_1} = v_{01}$ and $\frac{\partial u_1}{\partial A_1} = v_{11}$

The model equation is:

3.14
$$\ddot{y} + A_1 \dot{y} + A_0 y = A_0 r$$

Differentiating with respect to A_0 and A_1 yields:

3.15
$$\ddot{u}_0 + A_1 \dot{u}_0 + A_0 u_0 = r - y$$

$$3.16 \quad \ddot{u}_1 + A_1 \dot{u}_1 + A_0 u_1 = -\dot{y}$$

and

$$3.17 \quad \ddot{v}_{00} + A_1 \dot{v}_{00} + A_0 v_{00} = -2u_0$$

$$3.18 \quad \ddot{v}_{01} + A_1 \dot{v}_{01} + A_0 v_{01} = -\dot{u}_0 - u_1$$

$$3.19 \quad \ddot{v}_{11} + A_1 \dot{v}_{11} + A_0 v_{11} = -2\dot{u}_1$$

From equation 3.1a $= \frac{K_m K_d}{J}$ and $a_0 = \frac{K_m K_p}{J}$. This gives the expressions

for δa_1 and δa_2 .

$$3.20 \quad \delta a_1 = \frac{K_m}{J} \delta K_d = \frac{a_1}{K_d} \delta K_d \approx \frac{A_1}{K_d} \delta K_d$$

$$3.21 \quad \delta a_0 = \frac{K_m}{J} \delta K_p = \frac{a_0}{K_p} \delta K_p \approx \frac{A_0}{K_p} \delta K_p$$

Now solving for δK_d and δK_p yields:

$$3.22 \quad \delta K_d = \frac{K_d}{A_1} \delta a_1$$

$$3.23 \quad \delta K_p = \frac{K_p}{A_0} \delta a_0$$

This alternative method of deriving the gain sensitivity equations could be used with a model referenced adaptive control system. Further study could be made to determine this methods performance.

Another area of suggested further study is in the area of determining the q values used in the index of performance. Also, along with this,

would be a study of the stability of the system as a function of the q values and an analytical approach to determining the values of the q 's.

BIBLIOGRAPHY

Works Cited

- [1] Donalson, D. D., Leondes, C. T., "A Model Referenced Parameter Tracking Technique for Adaptive Control Systems: Part I -- The Principles of Adaptation and Part II -- Stability Analysis by the Second Method of Lyapunov," Transactions of the IEEE on Applications and Industry, Vol. 82, No. 68, Sep., 1963, 241-262
- [2] Dubowsky, S., DesForges, D. T., "The Application of Model Referenced Adaptive Control to Robotic Manipulators," ASME Journal of Dynamic Systems, Measurement, and Control, Vol. 101, Sep., 1979, 193-200.

Works Consulted

Kuo, B., Digital Control Systems, Holt, Rinehart, and Winston, 1980.

Ogata, K., Modern Control Engineering, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1970.

Paul, R., Robotic Manipulators: Mathematics, Programming, and Control, The MIT Press, Cambridge, Mass., 1983.

Castro, R., "An Experimental Design and Study of a Flexible Manipulator," M S Thesis, University of California, Los Angeles, Calif., 1976.

APPENDIX

The following is a listing of the computer programs used in the analysis of the model reference adaptive control system.

```

DIMENSION DTHETA(6),T(3),THETA(6),Q(6),Y(3),DY(3),U(9),DU(9),
$S(9),P(3)

```

```

-----
C This COMMON statement is needed for the graphics subroutines and for the
C ROBOTEQ subroutine. The top row of variables is used by the graphics
C and the second row is used by the subroutine that contains the dynamic
C equations describing the robot, subroutine ROBOTEQ.
C -----

```

```

COMMON LEFT,RIGHT,BOTTOM,TOP,XREG,YREG,IXREG,IYREG,
M4,IXX4,IYY4,IZZ4,T

```

```

-----
C The REAL and INTEGER statements contain variables used in the graphics
C subroutines, robot dynamic equations subroutine, and the main program.
C The variable LEFT must be defined as real for the graphics subroutines
C to work. M4, IXX4, IYY4, and IZZ4 are used in the robot dynamics equa-
C tions to describe the payload being handled by the robot. KP, KI, KD,
C and KM are the system gains used in the control of the robot. INT is the
C value of the integral of the error signal of the control system and INCR
C is the step size of the Runge Kutta Gill numerical integration that is
C used in this program. FLAG is simply a flag used in selecting output to
C be plotted.
C -----

```

```

REAL LEFT,M4,IXX4,IYY4,IZZ4,KP,KI,KD,KM,INT,INCR

```

```

INTEGER FLAG

```

```

-----
C This is where you specify a payload for the robot. These variables are
C common with the robot equations subroutine, so changing them in the main
C program updates them in the dynamics equations also. M4 is the mass of
C the payload in slugs. I**4 is the second moment of inertia of the pay-
C load about the * axis in slug-ft squared (where * is X, Y, or Z). The
C axes of the payload are defined in the thesis in figure .
C -----

```

```

M4=0.
IXX4=0.
IYY4=0.
IZZ4=0.

```

```

-----
C This loop writes blank lines to the screen so that the graphics screen is
C clear of unwanted material when it plots output.
C -----

```

```

DO 100 I=1,30
  WRITE (9,10)
  I=I+1
100

```

```

-----
C This section sets up the graphics screen for plotting output. GINIT and
C GCLEAR initialize the graphics subroutines and clear the graphics screen.
C WINDOW defines the plotting area of the graph, AXES labels axes on the

```

```

C plotting area, and FRAME frames the graphics screen.
C -----
C
C CALL GINIT
C CALL GCLEAR
C CALL WINDOW (0.,21.,-.25,1.25)
C CALL AXES (0.,0.,1.,.25)
C CALL WINDOW (0.,21.,10.,50.)
C CALL AXES (0.,10.,1.,2.)
C
C -----
C N is the order of the differential equations describing the robot
C dynamics and is used by the RI, Runga Kutta Gill, subroutine that
C integrates the robot dynamic equations. INCR, as stated earlier is the
C step size of numerical integration.
C -----
C
C N=6
C INCR=.01
C
C -----
C The Qn values (where n=0,1,or2) are the weighting factors used in the
C error functions of the gain adjustment sensitivity equations in the
C CONTROL subroutine.
C -----
C
C Q0=200.
C Q1=140.
C Q2=34.22
C
C -----
C This loop determines the number of times the simulation is run, according
C to the FLAG=n,m statement. Each time the simulation is run, FLAG can be
C used to specify a different CALL DRAW statement.
C -----
C
C DO 300 FLAG=0,1
C
C -----
C This is where you specify the initial values of the system gains. KM is
C constant for the rest of the program. KP, KI, and KD are adjusted by
C the CONTROL subroutine according to the system dynamics.
C -----
C
C KM=40.
C KI=140.26
C KP=98.19
C KD=24.
C
C AKP=KP
C AKI=KI
C AKD=KD
C
C AT=0
C AY=0
C -----
C This initializes the following variables. TIME is the time is seconds.

```

C T(1) is the torque applied to the torso of the robot in ft-lbs. RSAVE
 C is used to calculate the derivative of the input signal R. ESAVE is
 C used to calculate the derivative of the error signal E. And INT is the
 C integral of the error signal E.
 C -----
 C

TIME=0.
 T(1)=0.
 RSAVE=0.
 RDSAVE=0.
 ESAVE=0.
 INT=0.

C -----
 C This loop initializes all the joint angles, velocities, and accelerations
 C to zero. If a different initial condition for a joint is desired, then
 C it can be specified after this loop.
 C -----
 C

DO 200 J=1,6
 DTHETA(J)=0.
 THETA(J)=0.
 Q(J)=0.

200

CALL MOVE(0..24.)

C -----
 C This starts the simulation and describes the inputs to the system.
 C -----
 C

150 CONTINUE
 R=.0035*SIN(30.*TIME)
 GO TO 1000

C
 250 CONTINUE
 R=1.*(TIME-5.)
 GO TO 1000

C
 350 CONTINUE
 R=1.
 GO TO 1000

C
 450 CONTINUE
 M4=2.
 IXX4=.3
 IYY4=.3
 IZZ4=.3
 R=1+.0035*SIN(30.*TIME)
 GO TO 1000

C
 550 CONTINUE
 R=1.-1.*(TIME-15.)
 GO TO 1000

C
 650 CONTINUE
 R=.0035*SIN(30.*TIME)

C -----
 C

```

C Calculation of the derivative of the input, R, using a linear approxima-
C tion of the input over the interval INCR.
C -----
1000      RDOT=(R-RSAVE)/INCR
          RSAVE=R
          RDDOT=(RDOT-RDSAVE)/INCR
          RDSAVE=RDOT
C -----
C Calculation of the error signal, E. The control uses unity feedback and
C the error is the difference in the torso position, THETA(1), and the
C input, R.
C -----
          E=R-THETA(1)
C -----
C Calculation of the derivative of the error signal, E.
C -----
          EDOT=(E-ESAVE)/INCR
          ESAVE=E
C -----
C Calculation of the integral of the error signal, E, using a linear
C approximation of error over the interval INCR.
C -----
          INT=INT+E*INCR
C -----
C Calculation of the torque applied to the torso using PID control in the
C feed-forward path.
C -----
          T(1)=KM*(KP*E+KI*INT+KD*EDOT)
C -----
C Call to subroutines that give the response of the math model of the
C robot to the calculated torque.
C -----
          CALL RI(N,TIME,INCR,THETA,DTHETA,Q)
C -----
C Call to subroutines that calculate the response of the model to the input
C and that contain the gain adjustment algorithm for the control system.
C -----
          CALL CONTROL(THETA,DTHETA,R,KP,KI,KD,Y,DY,INCR,TIME.
$          Q0,Q1,Q2,RDOT,U,DU,S,P,RDDOT)
          WRITE (9,35) TIME,KP,KI,KD
35          FORMAT (10X,F5.2,3F20.3)
C -----
C Plotting of output.
C -----
          IF (FLAG.LT.1) GO TO 4000
C
          CALL MOVE(TIME-INCR,AT)
          CALL DRAW(TIME,THETA(1))
          CALL MOVE(TIME-INCR,AY)
          CALL DRAW(TIME,Y(1))
C          CALL MOVE(TIME-INCR,AKP)
C          CALL DRAW(TIME,KP)
C          CALL MOVE(TIME-INCR,AKI)
C          CALL DRAW(TIME,KI)
C          CALL MOVE(TIME-INCR,AKD)
C          CALL DRAW(TIME,KD)
C          AT=THETA(1)

```

AY=Y(1)

C
C
C
C
C
C
C
C
C
C
4000
C

AKP=KP
AKI=KI
AKD=KD

Increment time.

TIME=TIME+INCR

IF (TIME.LT.5.) GO TO 150
IF (TIME.LT.6.) GO TO 250
IF (TIME.LT.10.) GO TO 350
IF (TIME.LT.15.) GO TO 450
IF (TIME.LT.16.) GO TO 550
IF (TIME.LT.21.) GO TO 650

C
C
C
C
C
C
C

When the simulation is completed, FLAG is incremented for the next
pass through. If only a single output is desired on the plot, then the
DO 300 statement will be for FLAG=0 and the program will terminate
after dumping the plot to the printer.

C
C
300
C
C

CONTINUE

CALL DUMPGR(1)

C
10
END

FORMAT(' ')
END

```

SUBROUTINE CONTROL(THETA,DTHETA,R,KP,KI,KD,Y,DY,INCR,TIME,
$ Q0,Q1,Q2,RDOT,U,DU,S,P,RDDOT)
DIMENSION THETA(6),DTHETA(6),U(9),DU(9),Y(3),DY(3),S(9),P(3)
REAL KP,KI,KD,KPD,KID,KDD,P,S,INCR,J
    
```

C
C
C -----
C Initialization of parameters used in the calculation of the U-values and
C the model response to the input, R.
C -----

```

IF (TIME.GT.0.) GO TO 500
DO 40 I=1,9
U(I)=0.
DU(I)=0.
40 S(I)=0.
DO 50 I=1,3
Y(I)=0.
DY(I)=0.
50 P(I)=0.
500 CONTINUE
    
```

C
C -----
C Constants of model transfer function.
C -----
A2=15.4
A1=63.
A0=90.

C
C -----
C Call to subroutine that calculates model's response to input, R.
C -----
NEQ=3
CALL MI(NEQ,INCR,Y,DY,P,R,RDOT)

C
C -----
C Call to subroutine that calculates U-values used in determining values
C for rate of gain adjustments.
C -----
NEQ=9
CALL UI(NEQ,INCR,U,DU,S,R,Y,RDOT,RDDOT)

C
C -----
C Calculation of model displacement, velocity, and acceleration in terms of
C state variables as defined by state equations.
C -----
YDISP=Y(1)
YVEL=(Y(2)+A2*R)
YACCL=Y(3)+A1*R+A2*RDOT

C
C -----
C Error equations used in error functions.
C -----
E0=THETA(1)-YDISP
E1=THETA(4)-YVEL
E2=DTHETA(4)-YACCL

C
C -----
C Calculation of rate of change of gains and calculation of new values of

C gains.

C -----
C $A2D = -(Q0 \cdot E0 + Q1 \cdot E1 + Q2 \cdot E2) \cdot (Q0 \cdot U(7) + Q1 \cdot U(8) + Q2 \cdot U(9))$
C $A1D = -(Q0 \cdot E0 + Q1 \cdot E1 + Q2 \cdot E2) \cdot (Q0 \cdot U(4) + Q1 \cdot U(5) + Q2 \cdot U(6))$
C $A0D = -(Q0 \cdot E0 + Q1 \cdot E1 + Q2 \cdot E2) \cdot (Q0 \cdot U(1) + Q1 \cdot U(2) + Q2 \cdot U(3))$
C $A0D = -1000. \cdot (-Q0 \cdot E0 \cdot U(1) - Q1 \cdot E1 \cdot U(2) - Q2 \cdot E2 \cdot U(3))$
C $A1D = 100. \cdot (-Q0 \cdot E0 \cdot U(4) - Q1 \cdot E1 \cdot U(5) - Q2 \cdot E2 \cdot U(6))$
C $A2D = 5. \cdot (-Q0 \cdot E0 \cdot U(7) - Q1 \cdot E1 \cdot U(8) - Q2 \cdot E2 \cdot U(9))$

C
C $KDD = A2D \cdot KD / A2$
C $KPD = A1D \cdot KP / A1$
C $KID = A0D \cdot KI / A0$

C
C $KP = KP + KPD \cdot INCR$
C $KI = KI + KID \cdot INCR$
C $KD = KD + KDD \cdot INCR$

C
C RETURN
C END

```
SUBROUTINE RI(NEQ,X,H,Y,DY,Q)
DIMENSION A(2)
DIMENSION Y(NEQ),DY(NEQ),Q(NEQ)
A(1)=.2928932188134524
A(2)=1.707106781186547
H2=H/2.0
CALL ROBOTEQ(NEQ,X,Y,DY)
DO 13 I=1,NEQ
B=H2*DY(I)-Q(I)
Y(I)=Y(I)+B
13 Q(I)=Q(I)+3.0*B-H2*DY(I)
DO 20 J=1,2
CALL ROBOTEQ(NEQ,X,Y,DY)
DO 20 I=1,NEQ
B=A(J)*(H*DY(I)-Q(I))
Y(I)=Y(I)+B
20 Q(I)=Q(I)+3.0*B-A(J)*H*DY(I)
CALL ROBOTEQ(NEQ,X,Y,DY)
DO 26 I=1,NEQ
B=.1666666666666666*(H*DY(I)-2.0*Q(I))
Y(I)=Y(I)+B
26 Q(I)=Q(I)+3.0*B-H2*DY(I)
RETURN
END
```

```

SUBROUTINE ROBOTEQ(N,X,Y,DY)
C
DIMENSION Y(N),DY(N),M(3,3),G(3),Q(3),T(3)
C-----
C These are the common variables needed for the graphics routines
C-----
COMMON LEFT,RIGHT,BOTTOM,TOP,XREG,YREG,IXREG,IYREG,
$ M4,IXX4,IYY4,IZZ4,T
C
REAL L1,L2,L3,M1,M2,M3,M4,IXX1,IYY1,IZZ1,IXX2,IYY2,IZZ2,
$ IXX3,IYY3,IZZ3,IXX4,IYY4,IZZ4,MR1,MR2,MR3,M
C-----
C Calculation of repeated expressions in state variable equations
C-----
L1=.6667
L2=2.667
L3=3.25
R1=-.3333
R2=1.458
R3=1.4375
M1=6.18
M2=1.863
M3=2.050
IXX1=23.37
IYY1=33.75
IZZ1=33.75
IXX2=.3182
IYY2=5.361
IZZ2=5.361
IXX3=.3118
IYY3=6.804
IZZ3=6.804
C
S1=SIN(Y(1))
C1=COS(Y(1))
S2=SIN(Y(2))
C2=COS(Y(2))
S3=SIN(Y(3))
C3=COS(Y(3))
S23=SIN(Y(2)+Y(3))
C23=COS(Y(2)+Y(3))
MR1=M3+M4
MR2=M3*R3+M4*L3
MR3=M3*R3**2+M4*L3**2
C-----
C Calculation of elements of the M matrix
C-----
M(1,1)=M1*R1**2+M2*(L1+R2*C2)**2+M3*(L1+L2*C2+R3*C23)**2
$ +M4*(L1+L2*C2+L3*C23)**2
$ +(IYY3+IYY4)*C23**2+IZZ1+IXX2*S2**2+IYY2*C2**2
$ +(IXX3+IXX4)*S23**2
C
M(2,2)=M2*R2**2+MR1*L2**2+2.*MR2*L2*C3+MR3+IZZ2+IZZ3+IZZ4
C
M(2,3)=MR2*L2*C3+MR3+IZZ3+IZZ4
C

```

```

M(3,3)=MR3+IZZ3+IZZ4
C
C -----
C Calculation of elements of G matrix
C -----
C
G(1)=(-2.*R2*M2*S2*(L1+R2*C2)-2.*M3*(L2*S2+R3*S23)*
$ (L1+L2*C2+R3*C23)-2.*M4*(L2*S2+L3*S23)*(L1+L2*C2+L3*C23)
$ +2.*S2*C2*(IXX2-IYY2)+2.*S23*C23*(IXX3+IXX4-IYY3-IYY4))
$ *Y(4)*Y(5)+(-2.*M3*R3*S23*(L1+L2*C2+R3*C23)-2.*
$ M4*L3*S23*(L1+L2*C2+L3*C23)+2.*C23*S23*
$ (IXX3+IXX4-IYY3-IYY4))*Y(4)*Y(6)
C
G(2)=-2.*L2*S3*(M3*R3+M4*L3)*Y(5)*Y(6)
C
G(3)=0
C
C -----
C Calculation of elements of Q matrix
C -----
C
Q(1)=0.
C
Q(2)=(M2*R2*S2*(L1+R2*C2)+M3*(L2*S2+R3*S23)*
$ (L1+L2*C2+R3*C23)+M4*(L2*S2+L3*S23)*(L1+L2*C2+L3*C23)
$ +(IYY2-IXX2)*S2*C2+(IYY3+IYY4-IXX3-IXX4)*S23*C23)*
$ Y(4)**2
C
$ -(MR2*L2*S3)*Y(6)**2
C
$ +32.2*(M2*R2*C2+MR1*L2*C2+MR2*C23)
C
Q(3)=(M3*R3*S23*(L1+L2*C2+L3*C23)+M4*L3*S23*
$ (L1+L2*C2+R3*C23)+(IYY3+IYY4-IXX3-IXX4)
$ *S23*C23)*Y(4)**2
C
$ +MR2*L2*S3*Y(5)**2
C
$ +32.2*MR2*C23
C
T(2)=32.2*(M2*R2*C2+MR1*L2*C2+MR2*C23)
C
T(3)=32.2*MR2*C23
C
C -----
C This is where you put the equations for your state variables.
C DY(n) is the expression for y-dot(n).
C -----
C
DY(1)=Y(4)
DY(2)=Y(5)
DY(3)=Y(6)
C
DY(4)=(T(1)-Q(1)-G(1))/M(1,1)
C
DY(5)=(M(3,3)*(T(2)-Q(2)-G(2))-M(2,3)*(T(3)-Q(3)))
$ /(M(3,3)*M(2,2)-M(2,3)**2)
C
DY(6)=(M(2,2)*(T(3)-Q(3))-M(2,3)*(T(2)-Q(2)-G(2)))
$ /(M(3,3)*M(2,2)-M(2,3)**2)

```

C
C

RETURN
END

```
C *****
C
C File TSOM stands for torso model equation subroutine. This subroutine
C is called by the Runga Kutta Gill subroutine that calculates the output
C values of the model system used for the torso of the robot. The name
C of the Runga Kutta Gill subroutine used with the torso is TSOMI, which
C stands for torso model integration subroutine.
C
C The state equations defined in this subroutine represent the differential
C equation describing the third order model used in the control of the
C torso axis of the robot.
C *****
C
C SUBROUTINE MODEL(X,DX,R,RDOT)
C DIMENSION X(3),DX(3)
C
C A2=15.4
C A1=63.
C A0=90.
C
C -----
C State equations describing system model
C -----
C
C DX(1)=X(2)+A2*R
C DX(2)=X(3)+A1*R
C DX(3)=-A0*X(1)-A1*X(2)-A2*X(3)+(A0-A1*A2-A1*A2)*R-A2*A2*RDOT
C
C ***** The state variables are defined as:
C ***** Y(t)=X1
C ***** Ydot(t)=X2+A2*R
C ***** Ydoubledot(t)=X3+A1*R+A2*RDOT
C
C RETURN
C END
```

SUBROUTINE UVALUE(U,DU,R,RDOT,Y,RDDOT)
DIMENSION Y(3),U(9),DU(9)

C

A2=15.4
A1=63.
A0=90.

C

C

C

C

C

The state equations describing the U's are now given

DU(1)=U(2)
DU(2)=U(3)
DU(3)=-A0*U(1)-A1*U(2)-A2*U(3)-Y(1)+R
DU(4)=U(5)
DU(5)=U(6)
DU(6)=-A0*U(4)-A1*U(5)-A2*U(6)-(Y(2)+A2*R)+RDOT
DU(7)=U(8)
DU(8)=U(9)
DU(9)=-A0*U(7)-A1*U(8)-A2*U(9)-(Y(3)+A1*R+A2*RDOT)+RDDOT

C

RETURN
END

DIMENSION DTHETA(6),T(3),THETA(6),Q(6),Y(2),DY(2),U(4),DU(4),
 \$S(4),P(2)

 This COMMON statement is needed for the graphics subroutines and for the
 ROBOTEQ subroutine. The top row of variables is used by the graphics
 and the second row is used by the subroutine that contains the dynamic
 equations describing the robot, subroutine ROBOTEQ.

COMMON LEFT,RIGHT,BOTTOM,TOP,XREG,YREG,IXREG,IYREG,
 \$M4,IXX4,IYY4,IZZ4,T

 The REAL and INTEGER statements contain variables used in the graphics
 subroutines, robot dynamic equations subroutine, and the main program.
 The variable LEFT must be defined as real for the graphics subroutines
 to work. M4, IXX4, IYY4, and IZZ4 are used in the robot dynamics equa-
 tions to describe the payload being handled by the robot. KP, KI, KD,
 and KM are the system gains used in the control of the robot. INT is the
 value of the integral of the error signal of the control system and INCR
 is the step size of the Runga Kutta Gill numerical integration that is
 used in this program. FLAG is simply a flag used in selecting output to
 be plotted.

REAL LEFT,M4,IXX4,IYY4,IZZ4,KP,KD,KPD,KDD,KM,INCR

INTEGER FLAG

 This loop writes blank lines to the screen so that the graphics screen is
 clear of unwanted material when it plots output.

DO 100 I=1,30
 WRITE (9,10)
 I=I+1

 This section sets up the graphics screen for plotting output. GINIT and
 GCLEAR initialize the graphics subroutines and clear the graphics screen.
 WINDOW defines the plotting area of the graph, AXES labels axes on the
 plotting area, and FRAME frames the graphics screen.

CALL GINIT
 CALL GCLEAR
 CALL WINDOW (0.,21.,-.25,1.25)
 CALL AXES (0.,0.,1.,.25)
 CALL WINDOW (0.,21.,-.1,60.)
 CALL AXES (0.,0.,1.,5.)

 N is the order of the differential equations describing the robot
 dynamics and is used by the RI, Runga Kutta Gill, subroutine that

C integrates the robot dynamic equations. INCR, as stated earlier is the
C step size of numerical integration.
C -----
C

N1=6
N2=2
N3=4
INCR=.05

C -----
C The Qn values (where n=0,1,or2) are the weighting factors used in the
C error functions of the gain adjustment sensitivity equations in the
C CONTROL subroutine.
C -----
C

Q0=1.
Q1=.6
Q2=.11111111
A1=.6
A2=.11111111

C -----
C This loop determines the number of times the simulation is run, according
C to the FLAG=n,m statement. Each time the simulation is run, FLAG can be
C used to specify a different CALL DRAW statement.
C -----
C

DO 300 FLAG=0,1

C -----
C This is where you specify a payload for the robot. These variables are
C common with the robot equations subroutine, so changing them in the main
C program updates them in the dynamics equations also. M4 is the mass of
C the payload in slugs. I**4 is the second moment of inertia of the pay-
C load about the * axis in slug-ft squared (where * is X, Y, or Z). The
C axes of the payload are defined in the thesis in figure .
C -----
C

M4=0.
IXX4=0.
IYY4=0.
IZZ4=0.

C -----
C This is where you specify the initial values of the system gains. KM is
C constant for the rest of the program. KP, KI, and KD are adjusted by
C the CONTROL subroutine according to the system dynamics.
C -----
C

KM=40.
KP=13.068
KD=.6

C
C AKP=KP
C AKD=KD
C AT=0
C AY=0

```

C
C -----
C This initializes the following variables. TIME is the time in seconds.
C T(1) is the torque applied to the torso of the robot in ft-lbs. RSAVE
C is used to calculate the derivative of the input signal R. ESAVE is
C used to calculate the derivative of the error signal E. And INT is the
C integral of the error signal E.
C -----
C
C TIME=0.
C T(1)=0.
C
C -----
C This loop initializes all the joint angles, velocities, and accelerations
C to zero. If a different initial condition for a joint is desired, then
C it can be specified after this loop.
C -----
C
C DO 200 J=1,6
C   DTHETA(J)=0.
C   THETA(J)=0.
200  Q(J)=0.
C DO 40 I=1,4
C   U(I)=0.
C   DU(I)=0.
40   S(I)=0.
C DO 50 I=1,2
C   Y(I)=0.
C   DY(I)=0.
50   P(I)=0.
C
C CALL MOVE(0.,0.)
C
C -----
C This starts the simulation and describes the inputs to the system.
C -----
150 CONTINUE
C   R=.0175*SIN(30.*TIME)
C   GO TO 1000
C
C 250 CONTINUE
C   R=1.*(TIME-5.)
C   GO TO 1000
C
C 350 CONTINUE
C   R=1.
C   GO TO 1000
C
C 450 CONTINUE
C   M4=2.
C   IXX4=.3
C   IYY4=.3
C   IZZ4=.3
C   R=1.+0.0175*SIN(30.*TIME)
C   GO TO 1000
C
C 550 CONTINUE

```

R=1.-1.*(TIME-15.)
GO TO 1000

C
650

CONTINUE
R=.0175*SIN(30.*TIME)

C
C
C
C

Calculation of the torque applied to the torso.

1000

T(1)=KM*(R-THETA(1)-THETA(4))*KD)*KP

C
C
C
C
C

Call to subroutines that give the response of the math model of the robot, the response of the model reference to the input, and the U-values used in the gain sensitivity equations.

CALL RI(N1,TIME,INCR,THETA,DTHETA,Q)
CALL DI(N2,INCR,Y,DY,P,R)
CALL VI(N3,INCR,U,DU,S,R,Y,DY)

C
C
C

Calculation of gain adjustments

EO=THETA(1)-Y(1)
E1=THETA(4)-Y(2)
E2=DTHETA(4)-DY(2)
A1D=-Q0*EO*U(1)-Q1*E1*U(2)-Q2*E2*DU(2)
A2D=-Q0*EO*U(3)-Q1*E1*U(4)-Q2*E2*DU(4)
KDD=A1D-A2D*(A1-KD)/A2
KPD=-KP*A2D/A2
KP=KP+KPD*INCR
KD=KD+KDD*INCR

C

WRITE (9,25) TIME,KP,KD
FORMAT(F5.2,2F20.3)
IF (FLAG.LT.1) GO TO 6000
CALL MOVE(TIME-INCR,AT)
CALL DRAW(TIME,THETA(1))
CALL MOVE(TIME-INCR,AY)
CALL DRAW(TIME,Y(1))

25

C
C
C
C
C

CALL MOVE(TIME-INCR,AKP)
CALL DRAW(TIME,KP)
CALL MOVE(TIME-INCR,50.*AKD)
CALL DRAW(TIME,50.*KD)

C
C
C
C

AT=THETA(1)
AY=Y(1)

AKP=KP
AKD=KD

C

6000

TIME=TIME+INCR

C

IF (TIME.LT.5.) GO TO 150
IF (TIME.LT.6.) GO TO 250
IF (TIME.LT.10.) GO TO 350
IF (TIME.LT.15.) GO TO 450
IF (TIME.LT.16.) GO TO 550

PAGE 5 LIST VER 081282 4 5/10/85 19:09:47 SYS:0005..MRAC.SA

IF (TIME.LT.21.) GO TO 650

C

300 CONTINUE

C

CALL DUMPGR(1)

C

10 FORMAT(' ')

20 FORMAT(F5.2,3F20.3)

END

SUBROUTINE VALUE(U,DU,R,Y,DY)
DIMENSION Y(2),DY(2),U(4),DU(4)

C

A1=.6
A2=.11111111

C

DU(1)=U(2)
DU(2)=(-Y(2)-A1*U(2)-U(1))/A2
DU(3)=U(4)
DU(4)=(-DY(2)-A1*U(4)-U(3))/A2

C

RETURN
END

```

DIMENSION Y(3),DY(3),THETA(3),DTHETA(3),U(9),DU(9),P(3),Q(3),S(9)
C
COMMON LEFT,RIGHT,BOTTOM,TOP,XREG,YREG,IXREG,IYREG
C
REAL LEFT,KP,KI,KD,INCR,P,KPD,KID,KDD,J
C
INTEGER FLAG
C
DO 100 I=1,30
  WRITE (9,10)
100  I=I+1
C
CALL GINIT
CALL GCLEAR
C
CALL WINDOW (0.,21.,-.25,1.25)
C
CALL AXES (0.,0.,1.,.25)
CALL WINDOW (0.,21.,14.,17.)
CALL AXES (0.,14.,1.,.2)
C
CALL MOVE (5.,0.)
C
CALL DRAW (6.,1.)
C
CALL DRAW (15.,1.)
C
CALL DRAW (16.,0.)
C
N1=3
N2=3
N3=9
INCR=.01
C
Q0=200.
Q1=140.
Q2=34.22
A2=15.4
A1=63.
A0=90.
C
DO 300 FLAG=0
C
KI=99.
KP=69.3
KD=16.94
C
DO 200 I=1,3
  THETA(I)=0.
  DTHETA(I)=0.
200  Q(I)=0.
  DO 40 I=1,9
    U(I)=0.
    DU(I)=0.
40  S(I)=0.
  DO 50 I=1,3
    Y(I)=0.
    DY(I)=0.
50  P(I)=0.
C
TIME=0.
RDSAVE=0.
RSAVE=0.

```

```

J=1.
CALL MOVE(0.,16.)
C
150 CONTINUE
R=.0035*SIN(30.*TIME)
GO TO 1000
C
250 CONTINUE
R=1.*(TIME-5.)
GO TO 1000
C
350 CONTINUE
R=1.
GO TO 1000
C
450 CONTINUE
R=1.+0.0035*SIN(30.*TIME)
GO TO 1000
C
550 CONTINUE
R=1.-1.*(TIME-15.)
GO TO 1000
C
650 CONTINUE
R=0.+0.0035*SIN(30.*TIME)
C
1000 RDDOT=(R-RSAVE)/INCR
      RSAVE=R
      RDDOT=(RDDOT-RDSAVE)/INCR
      RDSAVE=RDDOT
      CALL TI(N1,INCR,THETA,DTHETA,Q,R,RDOT,KP,KI,KD,J)
      CALL MI(N2,INCR,Y,DY,P,R,RDOT)
      CALL UI(N3,INCR,U,DU,S,R,Y,RDOT,RDDOT)
      WRITE (9,20) TIME,KP,KI,KD
20    FORMAT (10X,F5.2,3F10.3)
      YDISP=Y(1)
      YVEL=(Y(2)+A2*R)
      YACCL=Y(3)+A1*R+A2*RDOT
      EO=THETA(1)-YDISP
      E1=THETA(2)+KD*R/J-YVEL
      E2=THETA(3)+KP*R/J+KD/J*RDOT-YACCL
      AOD=-QO*EO*U(1)-Q1*E1*U(2)-Q2*E2*U(3)
      A1D=-QO*EO*U(4)-Q1*E1*U(5)-Q2*E2*U(6)
      A2D=-QO*EO*U(7)-Q1*E1*U(8)-Q2*E2*U(9)
      KDD=A2D*KD/A2
      KPD=A1D*KP/A1
      KID=AOD*KI/AO
      KP=KP+KPD*INCR
      KI=KI+KID*INCR
      KD=KD+KDD*INCR
      IF (FLAG.GE.1) GO TO 2000
      CALL DRAW (TIME,KD)
      GO TO 6000
2000 IF (FLAG.GE.2) GO TO 3000
      CALL DRAW (TIME,Y(1))
      GO TO 6000
3000 IF (FLAG.GE.3) GO TO 4000

```

```
CALL DRAW (TIME,U(7))
GO TO 6000
4000 IF (FLAG.GE.4) GO TO 6000
CALL DRAW (TIME,KP)
GO TO 6000
5000 CALL DRAW (TIME,KI)
6000 TIME=TIME+INCR
C
IF (TIME.LT.5.) GO TO 150
IF (TIME.LT.6.) GO TO 250
IF (TIME.LT.10.) GO TO 350
IF (TIME.LT.15.) GO TO 450
IF (TIME.LT.16.) GO TO 550
IF (TIME.LT.21.) GO TO 650
C
300 CONTINUE
C
CALL DUMPGR(1)
C
10 FORMAT(' ')
END
```


SUBROUTINE TEST(X,DX,R,RDOT,KP,KI,KD,J)
DIMENSION X(3),DX(3)
REAL KP,KI,KD,J

C

A2=KD/J
A1=KP/J
A0=KI/J

C

DX(1)=X(2)+A2*R
DX(2)=X(3)+A1*R
DX(3)=-A0*X(1)-A1*X(2)-A2*X(3)+(A0-A1*A2-A1*A2)*R-A2*A2*RDOT

C

RETURN
END

SUBROUTINE D(X,DX,R)
DIMENSION X(2),DX(2)

C

A1=.6
A2=.11111111

C

DX(1)=X(2)
DX(2)=(R-A1*X(2)-X(1))/A2

C

RETURN
END

```
DIMENSION Y(2),DY(2),THETA(2),DTHETA(2),U(4),DU(4),P(2),Q(2),S(4)
C
COMMON LEFT,RIGHT,BOTTOM,TOP,XREG,YREG,IXREG,IYREG
C
REAL LEFT,KP,KD,INCR,P,KPD,KDD,J
C
INTEGER FLAG
C
DO 100 I=1,30
  WRITE (9,10)
100  I=I+1
C
CALL GINIT
CALL GCLEAR
CALL WINDOW (0.,21.,-.25,1.25)
CALL AXES (0.,0.,1.,.25)
C
CALL WINDOW (0.,21.,.5,1.3)
C
CALL AXES (0.,.5,1.,.1)
C
N1=2
N2=2
N3=4
INCR=.05
C
Q0=1.
Q1=.6
Q2=.11111111
A1=.6
A2=.11111111
C
DO 300 FLAG=0,1
C
KP=9.9
KD=.66
C
DO 200 I=1,2
  THETA(I)=0.
  DTHETA(I)=0.
200  Q(I)=0.
  DO 40 I=1,4
    U(I)=0.
    DU(I)=0.
40  S(I)=0.
  DO 50 I=1,2
    Y(I)=0.
    DY(I)=0.
50  P(I)=0.
C
TIME=0.
J=1.
CALL MOVE(0.,0.)
C
150 CONTINUE
R=.0175*SIN(30.*TIME)
GO TO 1000
C
250 CONTINUE
```

```

R=1.*(TIME-5.)
GO TO 1000
C
350 CONTINUE
R=1.
GO TO 1000
C
450 CONTINUE
R=1+.0175*SIN(30.*TIME)
GO TO 1000
C
550 CONTINUE
R=1.-1.*(TIME-15.)
GO TO 1000
C
650 CONTINUE
R=.0175*SIN(30.*TIME)
C
1000 CALL SI(N1,INCR,THETA,DTHETA,Q,R,KP,KD,J)
CALL DI(N2,INCR,Y,DY,P,R)
CALL VI(N3,INCR,U,DU,S,R,Y,DY)
EO=THETA(1)-Y(1)
E1=THETA(2)-Y(2)
E2=DTHETA(2)-DY(2)
A1D=-Q0*EO*U(1)-Q1*E1*U(2)-Q2*E2*DU(2)
A2D=-Q0*EO*U(3)-Q1*E1*U(4)-Q2*E2*DU(4)
KDD=A1D+A2D*(A1-KD)/A2
KPD=-KP*A2D/A2
KP=KP+KPD*INCR
KD=KD+KDD*INCR
WRITE(9,20)TIME,KP,KD
WRITE(9,10)
IF (FLAG.GE.1) GO TO 2000
CALL DRAW (TIME,THETA(1))
GO TO 6000
2000 IF (FLAG.GE.2) GO TO 3000
CALL DRAW (TIME,Y(1))
GO TO 6000
3000 IF (FLAG.GE.3) GO TO 4000
CALL DRAW (TIME,EO*U(1))
GO TO 6000
4000 IF (FLAG.GE.4) GO TO 5000
CALL DRAW (TIME,KP)
GO TO 6000
5000 CALL DRAW (TIME,KI)
6000 TIME=TIME+INCR
C
IF (TIME.LT.5.) GO TO 150
IF (TIME.LT.6.) GO TO 250
IF (TIME.LT.10.) GO TO 350
IF (TIME.LT.15.) GO TO 450
IF (TIME.LT.16.) GO TO 550
IF (TIME.LT.21.) GO TO 650
C
300 CONTINUE
C
CALL DUMPGR(1)

```

```
C
10  FORMAT(' ')
20  FORMAT(F5.2,3F20.3)
    END
```

```
SUBROUTINE SYS(X,DX,R,KP,KD,J)
DIMENSION X(2),DX(2)
REAL KP,KD,J
```

C

```
A1=KD
A2=J/KP
```

C

```
DX(1)=X(2)
DX(2)=(R-A1*X(2)-X(1))/A2
```

C

```
RETURN
END
```

DIMENSION DTHETA(6),T(3),THETA(6),Q(6),Y(2),DY(2),U(4),DU(4),
 \$S(4),P(2)

C
 C
 C
 C
 C
 C
 C
 C
 C
 C
 C

 Program TUKROL.SA adds proportional plus derivative feedback control
 to the shoulder and elbow axes.

 This COMMON statement is needed for the graphics subroutines and for the
 ROBOTEQ subroutine. The top row of variables is used by the graphics
 and the second row is used by the subroutine that contains the dynamic
 equations describing the robot, subroutine ROBOTEQ.

COMMON LEFT,RIGHT,BOTTOM,TOP,XREG,YREG,IXREG,IYREG,
 \$M4,IXX4,IYY4,IZZ4,T

C
 C
 C
 C
 C
 C
 C
 C
 C
 C
 C
 C

 The REAL and INTEGER statements contain variables used in the graphics
 subroutines, robot dynamic equations subroutine, and the main program.
 The variable LEFT must be defined as real for the graphics subroutines
 to work. M4, IXX4, IYY4, and IZZ4 are used in the robot dynamics equa-
 tions to describe the payload being handled by the robot. KP, KD,
 and KM are the system gains used in the control of the robot. INCR
 is the step size of the Runga Kutta Gill numerical integration that is
 used in this program. FLAG is simply a flag used in selecting output to
 be plotted.

REAL LEFT,M4,IXX4,IYY4,IZZ4,KP,KD,KPD,KDD,KM,INCR,KMS,
 &KME,KPS,KPE,KDS,KDE

INTEGER FLAG

C
 C
 C
 C
 C
 C
 C
 C
 C

 This loop writes blank lines to the screen so that the graphics screen is
 clear of unwanted material when it plots output.

```
DO 100 I=1,30
  WRITE (9,10)
  I=I+1
```

100

C
 C
 C
 C
 C
 C
 C
 C

 This section sets up the graphics screen for plotting output. GINIT and
 GCLEAR initialize the graphics subroutines and clear the graphics screen.
 WINDOW defines the plotting area of the graph, AXES labels axes on the
 plotting area, and FRAME frames the graphics screen.

```
CALL GINIT
CALL GCLEAR
CALL WINDOW (0.,21.,-.50,1.25)
CALL AXES (0.,0.,1.,.25)
CALL WINDOW (0.,21.,-.1,100.)
```

C CALL AXES (0.,0.,1.,5.)

C -----
C N is the order of the differential equations describing the robot
C dynamics and is used by the RI, Runga Kutta Gill, subroutine that
C integrates the robot dynamic equations. INCR, as stated earlier is the
C step size of numerical integration.
C -----

C N1=6
C N2=2
C N3=4
C INCR=.01

C -----
C The Qn values (where n=0,1,or2) are the weighting factors used in the
C error functions of the gain adjustment sensitivity equations in the
C CONTROL subroutine.
C -----

C QOT=1.
C Q1T=.15
C Q2T=.01
C QOS=1.
C Q1S=.6
C Q2S=.005
C QOE=1.
C Q1E=.6
C Q2E=.01
C A1=.6
C A2=.11111111

C -----
C This loop determines the number of times the simulation is run, according
C to the FLAG=n,m statement. Each time the simulation is run, FLAG can be
C used to specify a different CALL DRAW statement.
C -----

C DO 300 FLAG=0,0

C -----
C This is where you specify a payload for the robot. These variables are
C common with the robot equations subroutine, so changing them in the main
C program updates them in the dynamics equations also. M4 is the mass of
C the payload in slugs. I**4 is the second moment of inertia of the pay-
C load about the * axis in slug-ft squared (where * is X, Y, or Z). The
C axes of the payload are defined in the thesis in figure
C -----

C M4=0.
C IXX4=0.
C IYY4=0.
C IZZ4=0.

C -----
C This is where you specify the initial values of the system gains. KM is
C constant for the rest of the program. KP, and KD are adjusted by


```

C -----
150 CONTINUE
    R=.0175*SIN(30.*TIME)
    RS=0.
    RE=0.
    GO TO 1000

```

```

C
250 CONTINUE
    R=1.*(TIME-5.)
    RS=R
    RE=R
    GO TO 1000

```

```

C
350 CONTINUE
    R=1.
    RS=R
    RE=R
    GO TO 1000

```

```

C
450 CONTINUE
    M4=2.
    IXX4=.3
    IYY4=.3
    IZZ4=.3
    R=1+.0175*SIN(30.*TIME)
    RS=1.
    RE=1.
    GO TO 1000

```

```

C
550 CONTINUE
    R=1.-1.*(TIME-15.)
    RS=R
    RE=R
    GO TO 1000

```

```

C
650 CONTINUE
    R=.0175*SIN(30.*TIME)
    RS=.0
    RE=.0

```

```

C -----
C Calculation of the torque applied to the torso, shoulder, and elbow.
C -----
1000 T(1)=KM*KP*(R-THETA(1)-KD*THETA(4))
      T(2)=KMS*KPS*(RS-THETA(2)-KDS*THETA(5))
      T(3)=KME*KPE*(RE-THETA(3)-KDE*THETA(6))

```

```

C -----
C Call to subroutines that give the response of the math model of the
C robot, the response of the model reference to the input, and the U-values
C used in the gain sensitivity equations.
C -----

```

```

      CALL RITR(N1,TIME,INCR,THETA,DTHETA,Q)
      CALL DI(N2,INCR,Y,DY,P,R)
      CALL VI(N3,INCR,U,DU,S,R,Y,DY)

```

```

C -----
C Calculation of gain adjustments for the torso.
C -----

```

```

EO=THETA(1)-Y(1)
E1=THETA(4)-Y(2)
E2=DTHETA(4)-DY(2)
A1D=-QOT*EO*U(1)-Q1T*E1*U(2)-Q2T*E2*DU(2)
A2D=-QOT*EO*U(3)-Q1T*E1*U(4)-Q2T*E2*DU(4)
KDD=A1D-A2D*(A1-KD)/A2
KPD=-KP*A2D/A2
KP=KP+KPD*INCR
KD=KD+KDD*INCR

```

C
C
C
C
C

Calculation of gain adjustments for the shoulder

```

ESO=THETA(2)-Y(1)
ES1=THETA(5)-Y(2)
ES2=DTHETA(5)-DY(2)
AS1D=-QOS*ESO*U(1)-Q1S*ES1*U(2)-Q2S*ES2*DU(2)
AS2D=-QOS*ESO*U(3)-Q1S*ES1*U(4)-Q2S*ES2*DU(4)
DSDD=AS1D-AS2D*(A1-KDS)/A2
DSPD=-KPS*AS2D/A2
KPS=KPS+DSPD*INCR
KDS=KDS+DSDD*INCR

```

C
C
C
C
C

Calculation of the gain adjustments for the elbow.

```

EEO=THETA(3)-Y(1)
EE1=THETA(6)-Y(2)
EE2=DTHETA(6)-DY(2)
AE1D=-QOE*EEO*U(1)-Q1E*EE1*U(2)-Q2E*EE2*DU(2)
AE2D=-QOE*EEO*U(3)-Q1E*EE1*U(4)-Q2E*EE2*DU(4)
DEDD=AE1D-AE2D*(A1-KDE)/A2
DEPD=-KPE*AE2D/A2
KPE=KPE+DEPD*INCR
KDE=KDE+DEDD*INCR

```

C
C
C
C

```

WRITE (9,25) TIME,KD,KDS,KDE,KP,KPS,KPE
FORMAT(F5.2,6F10.3)
IF (FLAG.LT.1) GO TO 6000
CALL MOVE(TIME-INCR,AT1)
CALL DRAW(TIME,THETA(1))
CALL MOVE(TIME-INCR,AY)
CALL DRAW(TIME,Y(1))
CALL MOVE(TIME-INCR,AT3)
CALL DRAW(TIME,THETA(3))

CALL MOVE(TIME-INCR,AKP)
CALL DRAW(TIME,KP)
CALL MOVE(TIME-INCR,AKPS)
CALL DRAW(TIME,KPS)
CALL MOVE(TIME-INCR,AKPE)
CALL DRAW(TIME,KPE)

```

25
C
C
C
C
C
C
C
C
C
C

```

AT1=THETA(1)
AT2=THETA(2)
AT3=THETA(3)
AY=Y(1)

```

C

AKP=KP
AKD=KD
AKPS=KPS
AKPE=KPE

C

6000

TIME=TIME+INCR

C

IF (TIME.LT.5.) GO TO 150
IF (TIME.LT.6.) GO TO 250
IF (TIME.LT.10.) GO TO 350
IF (TIME.LT.15.) GO TO 450
IF (TIME.LT.16.) GO TO 550
IF (TIME.LT.21.) GO TO 650

C

300

CONTINUE

C

CALL DUMPGR(1)

C

10

FORMAT(' ')

20

FORMAT(F5.2,3F20.3)

END

ACKNOWLEDGEMENTS

I would like to express my appreciation to the following persons who contributed in a major way to this thesis: Dr Garth Thompson for his guidance and perseverance; Dr. Chi L. Huang for his assistance with the dynamic equations; David Boyd for the graphics routines he wrote for the computer; Doug Folken for his help after I left Kansas; Motorola for their graciousness in letting me work on my thesis at work; Department of Mechanical Engineering, Kansas State University for financial support; my parents for their continual support and encouragement; and finally, to my wife Kris who inspired me towards this degree.

VITA

David J. McConnell

Candidate for the Degree of
Master of Science

Thesis: ANALYSIS OF MODEL REFERENCED ADAPTIVE CONTROL APPLIED
TO ROBOTIC DEVICES

Major Field:
Mechanical Engineering

Biographical:

Personal Data: Born in Joplin, Missouri, September 12, 1959, the son of Jack D. and Jane M. McConnell.

Education: Graduated from Liberal High School, Liberal, Kansas, 1978; received Bachelor of Science degree from Kansas State University, with a major in Mechanical Engineering, in December, 1982; received the Master of Science degree from Kansas State University, with a major in Mechanical Engineering, in July, 1985.

Professional experience: Accepted employment with Motorola in Phoenix, Arizona in July, 1984, in the Equipment Engineering Department of the Discrete Manufacturing Facility of the Semiconductor Sector, and is presently employed there.

ANALYSIS OF MODEL REFERENCED ADAPTIVE
CONTROL APPLIED TO ROBOTIC DEVICES

by

DAVID JAMES McCONNELL

B. S., Kansas State University, 1982

AN ABSTRACT OF A THESIS

submitted in partial fulfillment of the

requirements of the degree

MASTER OF SCIENCE

Department of Mechanical Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1985

ABSTRACT

Model referenced adaptive control is a method of adjusting the gains of the closed loop transfer function of a control system, so that as the physical system being controlled changes, uniform performance can be maintained.

This paper is an analysis of model referenced adaptive control applied to an International Robomation Intelligence M50 robot. A computer simulation of the control is performed for the analysis.

Several control system configurations are analyzed for their use with the model referenced control algorithm. Tests are made for proper gain adjustment and system response for various systems.