

NATIONAL INSTITUTE FOR HIGHER EDUCATION, DUBLIN

School of Electronic Engineering.

Control Technology Research Unit.

Thesis Submitted for the Degree of  
Master of Engineering.

*Modelling and Simulation of a Puma 560 Manipulator  
for Control System Appraisal.*

by:

George P. Anderson BE

for:

Dr. John Ringwood.

September 1988.

I declare that the research herein was completed  
by the undersigned.

Signed George Anderson Date 22-Sept-88

## ABSTRACT

## **ABSTRACT**

### **Modelling and Simulation of a PUMA 560 Manipulator for Control System Appraisal.**

**Presented for award of Degree of Master of Engineering  
by George Patrick Anderson.**

The PUMA 560 Industrial Manipulator is presently controlled using a gain-scheduled PID control strategy. The implementation of adaptive control techniques can improve manipulator performance, [20]. One of the principal drawbacks with adaptive control is the difficulty in guaranteeing robustness over the complete operating range. This is particularly important when the plant is expensive and potentially self-damaging. An accurate, tuned, dynamic model, which includes the system's main dynamics is a great asset to the control engineer. A complete model for the Puma 560 has not previously been developed.

The research presented in this thesis derives a tuned and fully tested third order dynamic model for the first three links of the PUMA 560. A mathematical method of specifying the manipulator link structures and inter-relationships is discussed. By treating the manipulator as a connected chain of rigid bodies the Euler-Lagrange method is used to formulate second order dynamic equations and the servo-motor actuators and friction dynamics are incorporated into the model. The resulting third order model is simulated and tested in open and closed loop conditions. The model is then tuned, to provide a better representation of the actual manipulator dynamics, by comparing actual and simulated joint movements for the same actuator voltage input. PID control was implemented in software on each of the model joints independently. Simulated positional joint control resulted in sustained oscillations in joint positions indicating that simple constant gain PID controllers, without dynamic decoupling, were unable to produce satisfactory performance. The PUMA 560 controller board PID parameters were identified for two operating regions of joint 1. These values produced better dynamic performance when implemented in the simulated controller.

## **ACKNOWLEDGEMENTS**

The continued help and encouragement from my project supervisor, Dr. John Ringwood is greatly appreciated. Thanks to all the staff of the Electronic Engineering Dept. at The N.I.H.E. Thanks also to Pat Cox, Seamus O'Driscoll, Niall Walsh, Dave O'Riordan and all the Postgrads for their support. Special thanks to Breda M<sup>c</sup> Manus for typing the thesis.

## CONTENTS

## CONTENTS

<b>CHAPTER 1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation for this Research.	2
1.2	Preview of Thesis.	5
<b>CHAPTER 2</b>	<b>Structural Description of a Robot Manipulator.</b>	<b>6</b>
2.1	Link Specification and Dimensions.	6
2.2	Specification of Coordinate frames and their Inter-relationships.	10
2.3	PUMA 560 Link Dimensions and T Transformations.	13
2.4	Validation of Kinematic Equations.	16
2.5	Describing Link Mass Position.	16
2.5.1	Coordinate frame differential movements.	17
2.5.2	The Delta Operator.	18
2.5.3	Calculating Differential changes in Coordinate frames.	21
2.6	Summary.	23
<b>CHAPTER 3</b>	<b>Derivation of Second Order Dynamic Equations for an Elbow-type Manipulator.</b>	<b>24</b>
3.1	The Generalised Formulation of Dynamics.	25
3.1.1	The Lagrangian and Euler-Lagrange Equation.	26
3.2	Forming the Lagrangian for a Manipulator.	27
3.2.1	Kinetic Energy.	27
3.2.2	Potential Energy.	32
3.2.3	The General Manipulator Lagrangian.	34

3.3	General Manipulator Dynamic Equations.	35
3.3.1	Analysis of Self-Inertia Terms.	36
3.3.2	Analysis of Coupling Inertia Terms.	36
3.3.3	Analysis of Coupling and Coriolis Terms.	37
3.3.4	Analysis of Gravity Loading Terms.	38
3.3.5	Significance of Coefficient Terms in Control Environment.	38
3.4	Summary	39

<b>CHAPTER 4</b>	<b>Second Order Dynamic Equations for the PUMA 560 Manipulator.</b>	<b>40</b>
------------------	---	-----------

4.1	General Second Order Dynamic Equations for a Manipulator.	40
4.2	Matrix Representation of Dynamic Equations.	41
4.3	Calculation of Coefficient Terms.	43
4.3.1	Calculation of Effective and Coupling Inertia Terms.	43
4.3.2	Calculation of Coriolis and Centripetal Terms for Joint One.	44
4.3.3	Calculation of Coriolis and Centripetal Terms for Joints Two and Three.	45
4.3.4	Calculation of Gravity Loading Terms.	46
4.4	Link Parameter Estimations.	47
4.5	Numerical Evaluation of Coefficient Terms.	53
4.5.1	Numerical Evaluation of Effective and Coupling Inertias.	53
4.5.2	Numerical Evaluation of Centripetal and Coriolis Terms for Link One.	53
4.5.3	Numerical Evaluation of Centripetal Terms for Link Two.	54
4.5.4	Numerical Evaluation of Gravity Terms for the first Three Links.	54
4.6	Summary.	55



<b>CHAPTER 5</b>	<b>Third Order Dynamic Equations for the PUMA 560.</b>	<b>56</b>
5.1	Servo-motor Equations.	56
5.2	Third Order Dynamic Equations.	58
5.3	Matrix Representation of Third Order Dynamic Equations.	60
5.4	Summary.	63
<b>CHAPTER 6</b>	<b>PUMA 560 Robot Arm Simulator.</b>	<b>64</b>
6.1	Simulator Facilities.	64
6.2	Numerical Integration Methods.	65
6.2.1	Implementation of 4 <sup>th</sup> Order Runge-Kutta Integration.	68
6.3	Matrix Inversion Routine.	70
6.4	Simulation Program Structure.	71
6.5	Summary.	74
<b>CHAPTER 7</b>	<b>Open and Closed Loop Model Performance</b>	<b>76</b>
7.1	PUMA 560 Hardware Configuration.	78
7.1.1	Acquiring Manipulator Input / Output Data.	79
7.1.2	Acquiring Controller Board Input / Output Data.	79
7.2	Motor Output Torque Tests.	81
7.3	Estimating Friction Coefficients using Gravity Tests.	83
7.4	Estimating Radii of Gyration using Self Inertial Tests.	85
7.4.1	Estimation of $k_{3ZZ}^2$ .	85
7.4.2	Estimation of $k_{2ZZ}^2$ .	88
7.4.3	Estimation of $k_{3XX}^2$ , $k_{3YY}^2$ and $k_{1ZZ}^2$ .	90
7.4.4	Estimating the Remaining Radii of Gyration.	90

7.5	Open Loop Path Tracking Tests.	92
7.6	Considerations for Closed Loop Tests on Simulator.	92
7.6.1	A Review of PID Control.	96
7.6.2	Ziegler Nicholls Tuning of Controller Parameters.	99
7.7	PID Control of Robot Dynamic Model.	99
7.7.1	PID Control of Joint 1.	99
7.7.2	PID Control of Joint 2.	100
7.7.3	PID Control of Joint 3.	102
7.7.4	Multi-joint Control of Dynamic Model.	102
7.8	PUMA 560 Controller Parameter Identification.	102
7.9	Simulation Results of Model Control using measured PID parameters.	107
7.10	Summary.	108
CHAPTER 8	Conclusions	109

## BIBLIOGRAPHY

APPENDIX A     Analysis of Pseudo Inertial Matrix  $J_1$ .

APPENDIX B     Investigation of 4<sup>th</sup> Order Runge-Kutta Numerical Integration on a system with a Driving Input.

# CHAPTER 1

## INTRODUCTION

Robot manipulators are becoming increasingly important in the manufacturing industry. They present a means of lowering production time and costs, improving precision and increasing profits by around the clock operation. There are presently about 80,000 robotic units used in production throughout the world and the figure increases annually. The production of robot manipulators has itself become a large industry. In America alone there are more than 10 robot manufacturers. The manipulator market has become both lucrative and competitive [1]. Robot manufacturers must take advantage of the more recent cheaper computer power available to enhance manipulator performance.

Manipulator arm construction has changed little over the decades and is unlikely to do so. The mechanical design necessary to achieve dexterity results in a system with complex dynamic properties. Since the physical structure of manipulators cannot be improved upon by any large extent, the main performance improvements must be made in robot control systems.

Until recently industrial robot designers used simple PID gain scheduled controllers. In this type of control different PID gains are automatically switched as the manipulator moves into different zones within the robot's range. If the payload changes, however, this will not be sensed by the gain schedule controller. This has meant that controllers are often detuned, to cater for heaviest load cases, with the result that the manipulator operates below peak performance in much of the work space. Sophisticated control algorithms can improve the systems performance but have seldom been used in the past for two main reasons [2]:

- a) because of the economics of supplying sufficient computing power, and
- b) because of imprecise dynamic models.

If an accurate dynamic model and sufficient computational power are available, the control algorithm can predict how to control the manipulator's actuators when the robot is in motion. Such a controller must compensate for the complicated effects of inertial coupling, centripetal, coriolis, gravity and friction forces at each of the manipulator joints. This will enable the robot to follow desired

trajectories through space with smaller tracking errors, or to follow the previous paths at increased velocities. These desirable attributes both increase productivity and broaden the possible spectrum of applications.

Over the last decade, the literature has abounded with variations of old and new control strategies [2] [3] [4]. In order to obtain the optimum performance from a manipulator an accurate system dynamic model must be known. An alternative method of dealing with poor knowledge of system behaviour is through the use of adaptive control techniques. Adaptive control methods have been given much attention in recent years and have found applications in the power industries, the nuclear industry and aircraft industries to name but a few. As computational power becomes cheaper it becomes feasible to employ adaptive control on smaller systems.

Adaptive controllers can be viewed as being composed of two parts.

- (i) An identification portion which identifies parameters of the plant, and
- (ii) A control law portion which implements a control law based on the identified parameters.

The principles and different methods of adaptive control have been dealt with by many authors [2], [3], [5]. As the complexity of the controlled system increases it becomes increasingly difficult to provide guaranteed stability. Discrepancies due to unmodelled dynamics or external disturbances may cause the controller to become unstable. One of the main problems associated with adaptive control techniques is to guarantee robustness for the full operating range.

### 1.1 Motivation for this research

The EEC, seeing the need for improved controller performances, sponsored a three year project, (under the Stimulation program), to develop self-tuning and adaptive control techniques for robotic and fermentation systems. The project goals were to

- a) develop improved controller techniques,
- b) fully test these techniques and

c) implement the techniques on industrial systems.

The NIHE Dublin are equipped with both a Puma 560 industrial manipulator and a 40 Litre fermentation plant which are but available for controller implementation.

The Puma 560 is an elbow type manipulator with six revolute joints. Three relatively large links, which have a likeness to a human torso, upper arm and forearm, determine end effector position. The positions of these three links are changed using revolute joints which are often referred to as the waist, shoulder and elbow joints. At the end of the forearm a "wrist" is formed using three articulated joints. These three joints determine the end effector orientation.

For the control of an elbow type manipulator the dynamics of the first three joints present the largest problem. Coupling between the joints is considerable because the links are large. This results in highly coupled and nonlinear dynamics. Developing a correct dynamic model for a multi-degree of freedom manipulator is a difficult but necessary task if adaptive control techniques are to be implemented successfully on the manipulator. The need for an accurate dynamic model is twofold:

- a) an accurate model provides insight to the control problem and
- b) it enables the control engineer to fully test a controller before attempts are made to implement them.

Of the two, the latter provides the greater motivation. It is particularly important to ensure that the controller is completely tested prior to implementation when the plant is expensive and potentially self damaging. Other important advantages exist, for example, it is much easier to test a controller on a software model than to implement it on an actual system.

Although recent research has presented means of modelling manipulators as connected chains of rigid bodies many parameters which appear in the model such as effects of friction, gearing backlash, flexibilities and actuator dynamics remain unknown. Such models may suffice for preliminary control strategy appraisal, however, they do not satisfy the control engineer how wishes to fully test a novel control strategy before implementation.

Although many researchers [6], [7], [8] have written extensively on the derivation of manipulator dynamic models the presented models ignore many of the factors that contribute considerably to the plant dynamics. Programs have been written to derive symbolic second order dynamic equations for manipulators. These packages typically model the manipulators as a chain of rigid bodies, resulting in second order coupled differential equations. The coefficients of these equations are trigonometrical functions of the link masses, inertias, centres of masses, dimensions and joint positions, velocities and accelerations. Accurate values for these parameters are seldom known. If the parameters are known, the second order model often ignores actuator dynamics, friction forces and the dynamics of the gearing systems. These unmodeled dynamics will effect positioning accuracy.

Another problem with presented manipulator models is that they are never fully tested or tuned to fit the system dynamics. Testing and tuning would remove many system uncertainties. An accurately tuned dynamic model is invaluable to the control engineer when testing a control technique for a complex system. If the accuracy of the model is guaranteed, the controller can be fully tested before being implemented on the actual system.

This thesis develops a fully tested third order dynamic model for the first three links of the Puma 560 industrial robot. Only the dynamics of the first three links is studied since they present the most challenging control and modelling problem. The wrist links are dimensionally small and are not subjected to the large coupling effects, experienced by the positioning links, due to their small inertias. The dynamic model has taken the effects of:

- 1) gravity forces,
- 2) coupling and inertial forces,
- 3) centripetal and coriolis forces,
- 4) friction forces, and
- 5) actuator dynamics

into account. The model has been tested in both open and closed loop situations. The link parameters which constitute an integral part of the dynamic equation coefficient terms have been tuned to give a better representation of the manipulator dynamics.

A complete tuned third order dynamic model for first 3 links of a Puma 560 industrial manipulator has not been presented in the literature before.

## 1.2 Preview of Thesis

The research presented in this thesis is organised as follows.

Chapter 2 details how the geometric structure of a manipulator may be mathematically specified. This includes a presentation of kinematic equations and analysis for differential joint rotations.

Chapter 3 gives a detailed derivation of second order dynamic equations for a general elbow type manipulator. Special attention is given to physical interpretation of the dynamic coefficient terms.

Chapter 4 uses the procedure given in chapter 3 to generate second order dynamic equations for the Puma 560 industrial manipulator. Arguments for the choice of link mechanical parameters are given and nominal values are chosen.

Chapter 5 incorporates actuator dynamics and friction forces into the second order equations resulting in third order dynamic equations. These are written in matrix form for clarity and ease of simulation.

The simulation main characteristics, numerical integration technique and associated considerations are treated in Chapter 6,

Chapter 7 concentrates on fully testing the Puma 560 model using open and closed loop validation data, recorded, from the industrial manipulator. The dynamic model is tuned to remove response discrepancies.

Conclusions are given in Chapter 9.

## CHAPTER 2

### STRUCTURAL DESCRIPTION OF A ROBOT MANIPULATOR

Before any analysis can be undertaken it is necessary to accurately describe robot manipulator structure and freedom of movement. Kinematics is described as the study and description of possible motions of material bodies. Kinematic equations provide us with robot link positions given the various joint angles. Indeed the kinematic equations supply the position and orientation of all link's with respect to either the base coordinates or with respect to any of the other links. These equations are comprised of a set of homogeneous transformation matrices which relate the position of the links to each other and to the base coordinates. This is important when formulating the dynamic equations.

As mentioned in the introduction a manipulator may be considered to be a chain of rigid bodies, with articulated joints. The link shape is defined (from a kinematic view point) as the relationship between the joints at each end of the link. When the joint angles are specified, and link dimensions and shapes are known it is possible to determine the position of each link. This is the basis for deriving the kinematic equations.

This chapter outlines how the mechanical structure of a manipulator may be mathematically described. Firstly the link dimensions are specified, and a coordinate frame is embedded in each link. Then using homogeneous transformations, the relationship between these frames is defined, the particulars of the Puma 560 are presented and finally the delta operator is presented as a means of describing the effects of differential joint movements.

#### 2.1 Link Specifications and Dimensions

Figure 2.1 shows the physical structure of a typical elbow type industrial robot, in this case the Puma 560. The manipulator has six degrees of freedom. The positions of the first three links determine the end effector position, while the last three links specify tool orientation. For convenience the first three joints are referred to as the waist, shoulder and elbow joints and the final three joints collectively form the wrist. In order to specify the tool position and orientation



knowledge is required of the link sizes and joint angles.

In the geometric sense, a link may be considered to be a rigid structure, supporting one or two joint axes. When specifying the link dimensions it is therefore necessary, to give the relationship between the joint axes. Two groups of manipulator joint exist, revolute joints and prismatic joints. Figure 2.2 (a) shows a typical revolute joint while a prismatic joint is sketched in Figure 2.2 (b). The properties of prismatic joints are not examined here since analysis of an elbow type manipulator, with only revolute joints, is considered in this thesis.

See Figure 2.3 for a diagram of a typical manipulator link. Any rotation about joint axis  $n$ , causes link  $n+1$  to rotate relative to link  $n$ . All parts in the link will scribe out an arc in free space, with the joint axis  $n$  as it's centre. Rotation of link  $n+1$  about joint axis  $n+1$  will similarly cause all parts of link  $n+1$  to scribe a different arc. This time axis  $n+1$  serves as the arc centre. The shortest distance between the two axes is perpendicular to both and is referred to as the common normal between the axes. The length of the common normal, the *Link Length*,  $l_n$ , is one of the values used to describe the link. The other dimension is referred to as the *link twist*  $\alpha_n$ , and denotes the angle between the axes when projected onto a plane perpendicular to the common normal.  $\alpha_n$  refers to the amount the link axes have been twisted out of parallel.

Thus any link can be described by two dimensions  $l_n$  and  $\alpha_n$ . It remains to specify the relationships between consecutive links. Any link  $n$ , except for the end links, will have two common normals associated with it. One for the lower and one for the higher link. The distance along the axis between the two normals is called the distance between the links,  $d_n$ . The angle between the links  $q_n$  is measured in a plane normal to the axis.

The four parameters explained above are listed below.

- $l_n$  = link length
- = shortest distance between axes, found on common normal
  
- $\alpha_n$  = link twist
- = angle between axis measured in plane perpendicular to  
          common normal

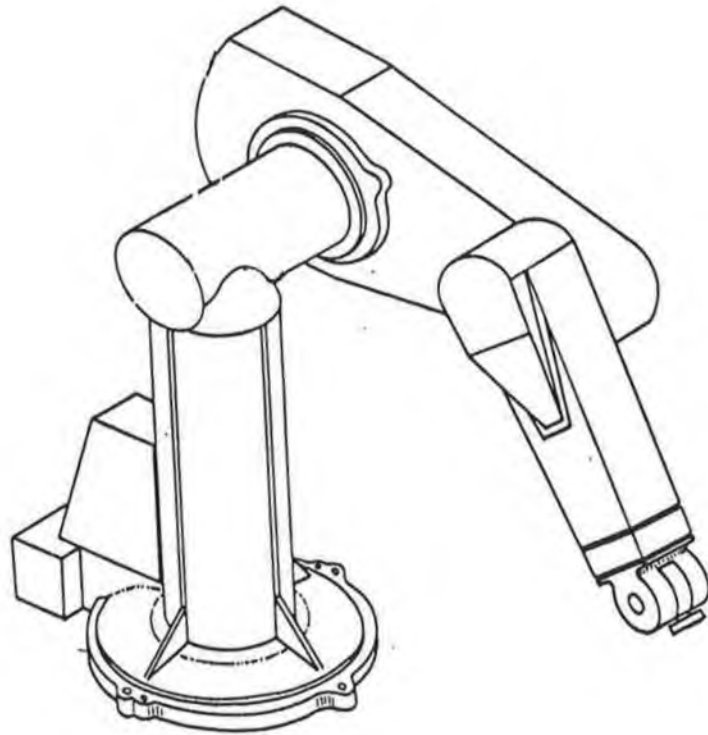


Figure 2.1 Physical Structure of an Elbow Type Manipulator



Figure 2.2(a) Typical revolute Joint

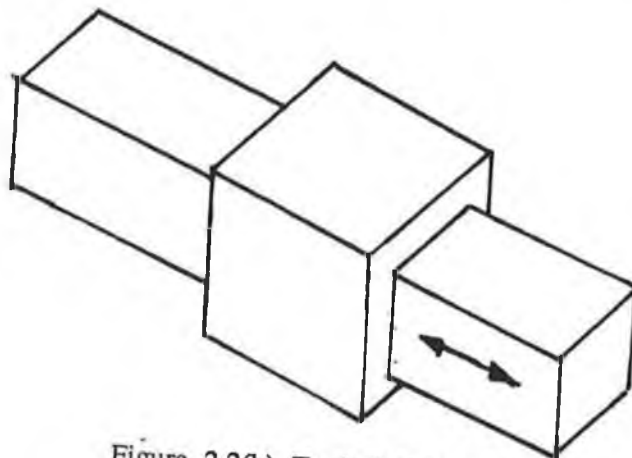


Figure 2.2(b) Typical Prismatic Joint

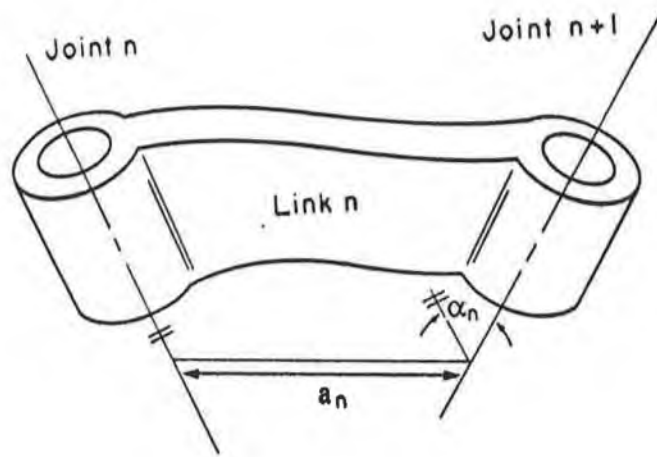


Figure 2.3 Typical Manipulator Link

$d_n$  = distance between two links,  
= distance measured along axis of joint  $n$  between common normals intersecting this axis.

$q_n$  = angle between links  
= angles measured between common normal on plane perpendicular to axis  $n$ .

Specification of the four values above for each of the six manipulator links allows us to determine the link and end effector positions. With the current information, however, this would be a tedious task. To simplify the procedure it is convenient to assign a coordinate frame to each link, and then to form a set of homogeneous transformations which describe the relative position and orientation of each link to that of the previous one, [6], [7].

## 2.2 Specification of Coordinate Frames and Their Interrelationship

After specifying the structure of the manipulator links, it is necessary to assign a coordinate frame to each link. It is important that the coordinate frames are chosen so that the movement of the manipulator links can be easily be described. Therefore we need to use some logical criterion when selecting frame origins and axes. Coordinate frames are chosen as suggested by Denavit & Hartenberg [10]

The origin of the coordinate frame for each link  $n$  is chosen to be the intersection of the common normal between the axes of joints  $n$  and  $n+1$ , and the axis of the  $n+1$  joint. Figure 2.4 shows how common normals can be constructed for joint axes that are not parallel. This choice of origin ensures that, firstly the frame origin will be positioned at a rotating axis, and secondly that the position of the common normals is always known.

The axes of the coordinate frames are chosen as follows [6]. The  $z$  axis of coordinate frame  $n$  is taken to be the axis of rotation of joint  $n$ , while the  $x$  axis is chosen as the normal directed from joint  $n$  to joint  $n+1$ . The  $y$  axis completes the conventional orthogonal set of right handed axis. A typical frame has been assigned in Figure 2.4.

Once the links parameters have been specified, and each link is given a

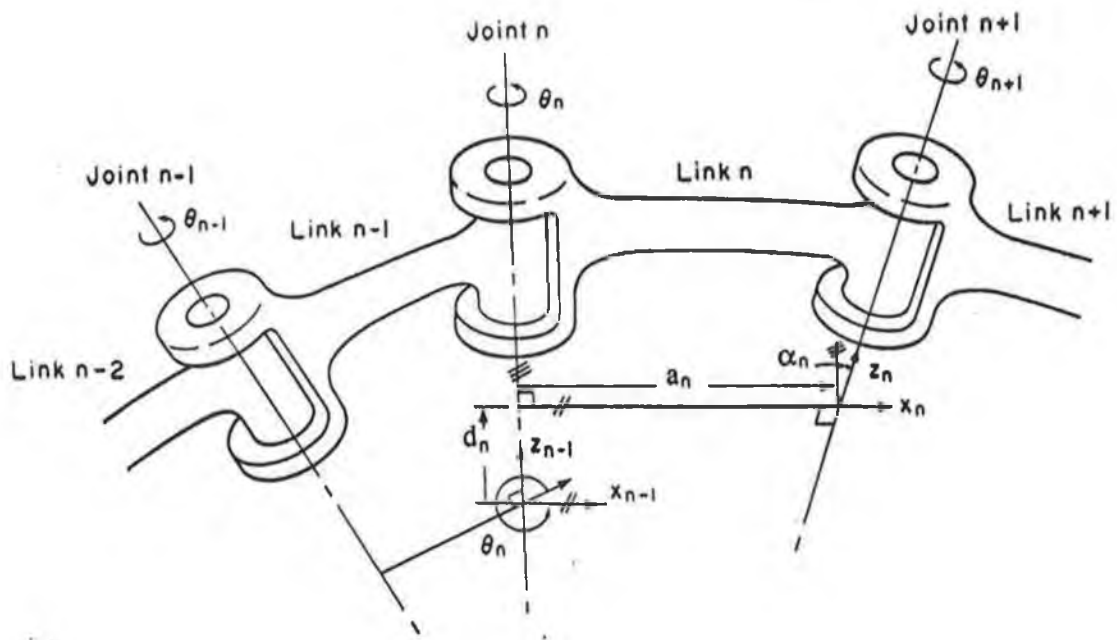


Figure 2.4 Construction of Common Normals Between axes and a Typical Coordinate Frame

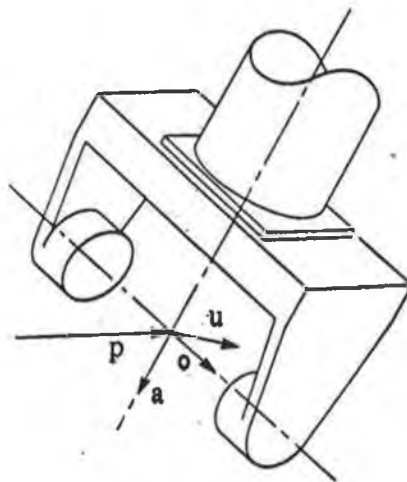


Figure 2.5  $u$ ,  $o$ ,  $a$ , &  $p$  Vectors

coordinate frame, the next step is to develop a method of describing the interrelationship between the coordinate frames. This makes it possible to specify the position of each manipulator link at any time. One method of achieving this is to create homogeneous transformations which are conveniently performed using matrix algebra. In this context, the transformation matrices are known as T matrices ( $\in \mathbf{R}^{4 \times 4}$ ). These are chosen to transform the origin of a reference coordinate frame to the origin of the present reference frame, as suggested by Denavit & Hartenberg [10].

Each of the T matrices has two frames associated with it. That of the reference frame, and that of the frame in question. The reference frame is denoted by a preceding superscript and a preceding subscript denotes the frame in question. For example  ${}^aT_b$  describes the transformation required to superimpose coordinate frame "a" on to coordinate frame "b".

The homogeneous transformation required to perform this manoeuvre on two consecutive revolute joints may be considered as the composite result of four simpler movements. Explicitly, rotating the coordinate frame an angle  $q_n$  about the  $z_{n-1}$  axis, next translating the origin of the frame a distance  $d_n$  along the same axis, translating the origin along the common normal a length  $l_n$  and lastly by rotating the coordinate frame the twist angle,  $\alpha_n$ , about the common normal. The two translations may be treated as a single translation. These movements ensure that both coordinate frames are superimposed and in essence describes the position of the present coordinate frame with respect to the previous. The element values in each of the T matrices depend on link sizes, shapes and manipulator joint angles.

The general form of a T matrix is

$${}^{n-1}T_n = \text{Rot}(z, \Theta_n) \text{Trans}(0,0,d_n) \text{Trans}(a_n,0,0) \text{Rot}(x, \alpha_n)$$

where

- $z$  = previous joint axis
- $d_n$  = distance between links n and n-1
- $a_n$  = link length
- $x$  = common normal between axes n and n-1
- $\alpha_n$  = link twist
- $\Theta_n$  = joint n angle

$${}^{n-1}T_n = \begin{bmatrix} \cos \theta_n & -\sin \theta_n \cos \alpha_n & \sin \theta_n \sin \alpha_n & a \cos \theta_n \\ \sin \theta_n & \cos \theta_n \cos \alpha_n & -\cos \theta_n \sin \alpha_n & a \sin \theta_n \\ 0 & \sin \alpha_n & \cos \alpha_n & d \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The elements of  ${}^{n-1}T_n$  are trigometrical functions of the joint angle.

${}^0T_1$  describes the position and orientation of the first link with respect to the base coordinates. Similarly  ${}^1T_2$  gives the second link position and orientation w.r.t. link one's coordinate frame. The product of consecutive T matrices, allows us to describe link positions w.r.t. any coordinate frames.  ${}^0T_2 = {}^0T_1 {}^1T_2$  describes the position and orientation of link two w.r.t. the base origin. In general  ${}^0T_i = {}^0T_1 {}^1T_2 \dots {}^{i-1}T_i$  gives position and orientation of the  $i^{\text{th}}$  link w.r.t. the base.

For a six degree of freedom manipulator, the position and orientation of the end effector w.r.t. the base coordinate frame, is contained in the product of the first six T matrices, and has the form:-

$${}^0T_6 = {}^0T_1 {}^1T_2 \dots {}^5T_6 = \begin{bmatrix} u_x & o_x & a_x & p_x \\ u_y & o_y & a_y & p_y \\ u_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Column vectors  $u$ ,  $o$ ,  $a$ , and  $p$ , are sketched in Figure 2.5. Vector  $p$  gives the end effector position w.r.t. the base, while vectors  $o$ ,  $a$ , and  $u$  give tool orientation.

### 2.3 Puma 560 Link dimensions and T transformations

Figure 2.6 shows how the coordinate frames have been chosen for the Puma 560. The link specifications are given in Table 2.1. Figure 2.7 shows a line sketch of the link structure when the manipulator is in the same pose as Figure 2.5 ( in bold ), and when it has moved upwards ( in a dotted line ). It is seen how the robot shape is accurately specified by these simple distances and angles. Using this information, and the procedure given above, a set of T

matrices for the Puma 560 can be determined. Various authors have chosen subtly different coordinate frames. This does not, however, cause difficulties once the homogeneous transformations are correctly chosen. A typical set of T matrices for the Puma 560 is given as:

$${}^0T_1 = \begin{bmatrix} C_1 & 0 & -S_1 & 0 \\ S_1 & 0 & C_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1T_2 = \begin{bmatrix} C_2 & -S_2 & 0 & a_2 C_2 \\ S_2 & C_2 & 0 & a_2 S_2 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2T_3 = \begin{bmatrix} C_3 & 0 & S_3 & a_3 C_3 \\ S_3 & 0 & -C_3 & a_3 S_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^3T_4 = \begin{bmatrix} C_4 & 0 & -S_4 & 0 \\ S_4 & 0 & C_4 & 0 \\ 0 & -1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^4T_5 = \begin{bmatrix} C_5 & 0 & S_5 & 0 \\ S_5 & 0 & -C_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



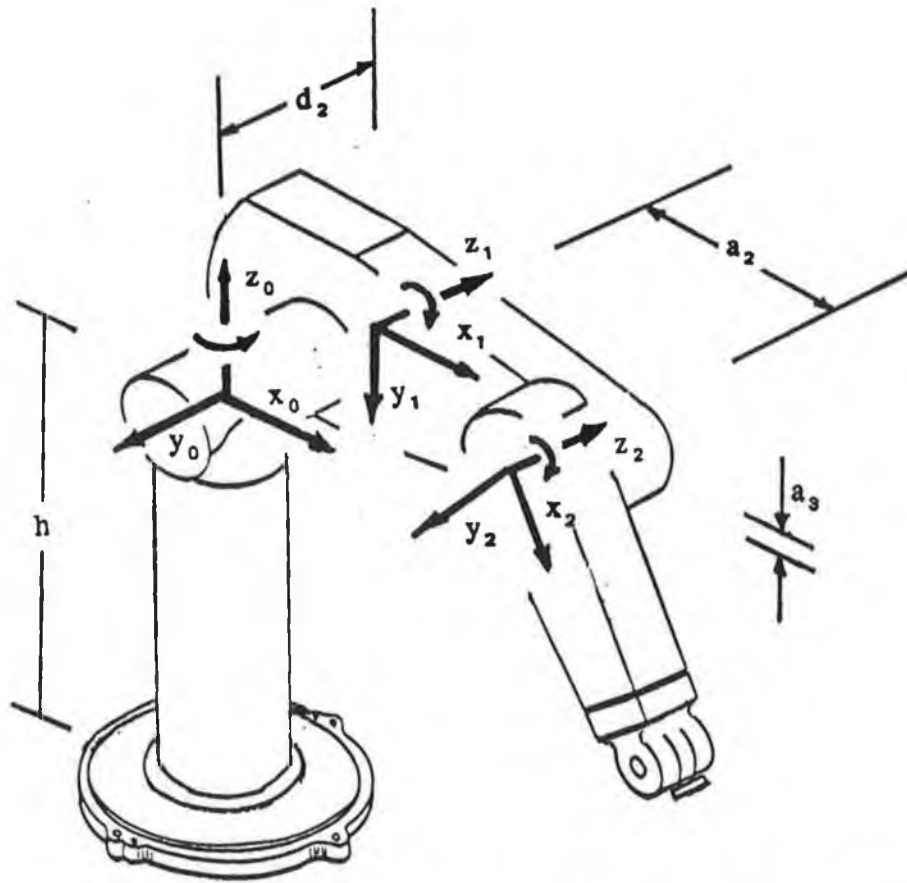


Figure 2.6 Coordinate Frames for the first Three Links of the Puma 560

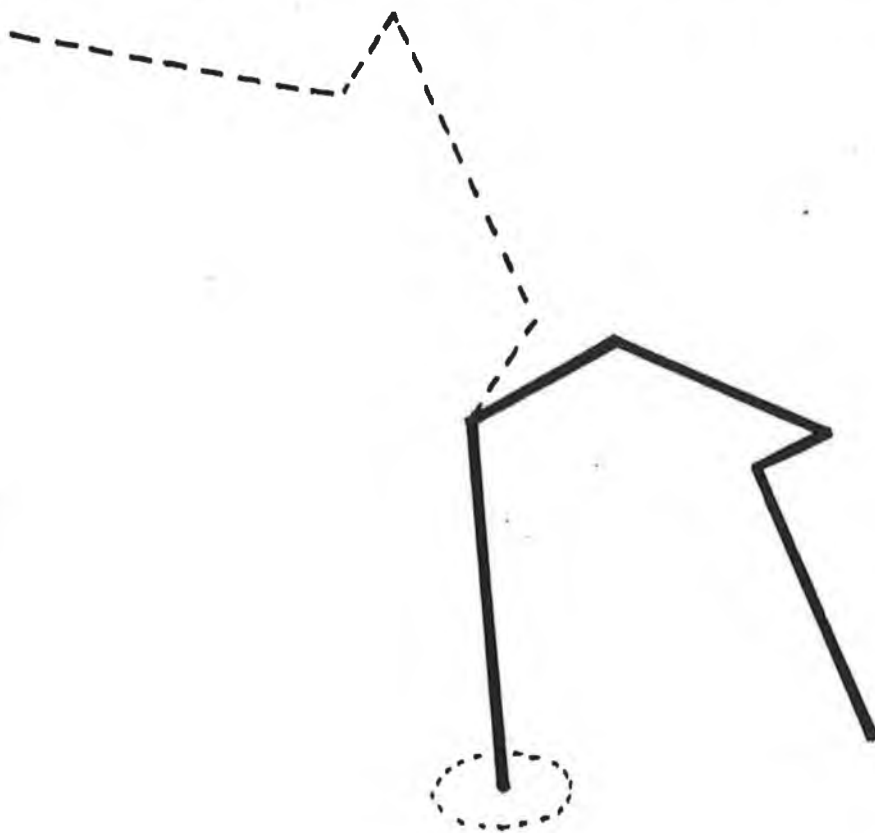


Figure 2.7 Line sketch of the Puma 560

$${}^5_6T = \begin{bmatrix} C_6 & -S_6 & 0 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where  $C_i = \cos(q_i)$

$S_i = \sin(q_i)$

## 2.4 Validation of Kinematic Equations

The set of T matrices for the Puma 560 listed above, were programmed on a MicroVax minicomputer. The program outputs the positions of each of the links, in terms of the base reference frame, for a set of manipulator angles. Test data was obtained by commanding the Puma Industrial robot to move to a precision reference point (with specified joint angles) and reading the positions of the end effector in cartesian coordinates. These operations were achieved using VAL, the Puma programming language at the monitor level.

The kinematic equations derived were found to be an accurate representation of the relationship between the joint angles and the position of the end effector. such accuracy is important for two reasons, (a) it is necessary for the robot supervisory system to know where each of the links are at any time , and (b) setpoints for the controllers are derived from the kinematic equations (an open loop calculation ). Furthermore, the accuracy of a robot simulation depends heavily on the kinematic equations, since they form the the basis (and influence the structure of ) the dynamic equations.

## 2.5 Describing link mass positions

The six Puma 560 transformation matrices given in Section 2.3 allow us to calculate the position and orientation of the robot links relative to any coordinate frame. This is achieved by multiplying together the appropriate homogeneous transformation matrices. When formulating the dynamic equations for the manipulator it is necessary to determine the effect that differential changes in any of the robot joints will have on the position of the coordinate frames. The remainder of this chapter examines how point masses in a link may be described

w.r.t. different coordinate frames, and how the effects of differential positional changes can be mathematically described in an effective manner.

Figure 2.8 shows how the position of a point mass in a link  $j$ , can be specified with respect to different coordinate frames. In this diagram coordinate frame  $i$  and  $j$  positions are sketched relative to the base frame. These two frames are not necessarily consecutive frames. The following passage explains how the position of a point mass  $p$ , located in link  $j$  is given with respect to different coordinate frames. Point  $p$  is located at  ${}^i r_j$  w.r.t. coordinate frame  $i$ .

$${}^i r_j = ({}^i x_j \quad {}^i y_j \quad {}^i z_j \quad 1)$$

The preceding superscript shows that this vector  $r$  is taken w.r.t. coordinate frame  $i$ . The subscript shows that the point mass is in link  $j$ . The final element in  ${}^i r_j = ({}^i x_j \quad {}^i y_j \quad {}^i z_j \quad 1)$  is a scaling factor. All vectors have this fourth element in order to be compatible with the Denavit-Hartenberg [10] four parameter representation. When referring to vectors taken w.r.t. the base, the preceding superscript of 0 is often omitted. Thus  $r_j$  is the vector from the origin of base coordinate frame to the point mass in link  $i$ .

The same point is located at  ${}^j r_j$  w.r.t. coordinate frame  $j$ . The vector from the base frame to the point mass is  $r$  and is a function of the joint angles. Not surprisingly  $r_j$ ,  ${}^j r_j$  and  ${}^i r_j$  can be expressed in terms of each other as shown below

$$r_j = {}^0 T^i {}^i r_j = {}^0 T^j {}^j r_j = {}^0 T^j r_j$$

It is important to be able to express link point mass positions w.r.t. any coordinate frame. This is necessary when calculating the various forces exerted at a joint due to the position velocity and acceleration of all the manipulator links.

### 2.5.1 Coordinate frame differential movements

This section considers how the position and orientation of the link coordinate frames are effected by differential movements of a joint. Any change in position of a joint  $i$  will only affect the positions of the outer links  $j$  ( $j \geq i$ ). A change in the position and orientation of the outer coordinate frames is

equivalent to a translation followed by a rotation. The task is to accurately specify the correct translation and rotation for each coordinate frame. Details of the procedure to achieve this is outlined below.

### 2.5.2 The delta operator

Any differential movement of joint  $i$ , will cause the coordinate frames, represented by the transformation matrix  ${}^0_jT$  ( $j \geq i$ ) to change. We can consider the changes in any of these coordinate frames as being the result of a differential translation, followed by a differential rotation. The delta operator is a  $4 \times 4$  matrix, derived in this section as a means of describing the effects that differential joint angle changes have on other links.

Figure 2.9 represents two coordinate frames, that of the base, and that of an arbitrary link  $j$ . Although the base reference frame is oblivious to changes in any of the robot joint angles, movement of joint  $i$  ( $i < j$ ) has caused the origin of the coordinate frame  $j$  to move from position  $P_j$  to  $P'_j$ . The orientation of frame  $j$  has also changed. These changes in position and orientation can be precisely described by the translation  $(dx, dy, dz)$  and the rotation  $(k, d\Theta)$ , where:

- Translation  $(d_x, d_y, d_z)$  is the transformation representing a translation of  $(d_x, d_y, d_z)$  in base coordinates.

- Rotation  $(k, d\Theta)$  is a transformation representing a differential rotation  $d\Theta$  about a vector  $k$ , where  $k$  is described in base coordinates.

Given a coordinate frame  ${}^0_pT$ , we can express  ${}^0_pT + d{}^0_pT$  as:-

$${}^0_pT + d{}^0_pT = \text{Translation}(d_x, d_y, d_z) \text{Rotation}(k, d\Theta) {}^0_pT$$

It follows that

$$d{}^0_pT = ( \text{Translation}(d_x, d_y, d_z) \text{Rotation}(k, d\Theta) - I ) {}^0_pT$$

where  $I$  is the identity matrix.

It is not necessary to express the translation vector,  $(d_x, d_y, d_z)$ , in base coordinates. We could have described it with respect to coordinate frame  ${}^0_jT$ .

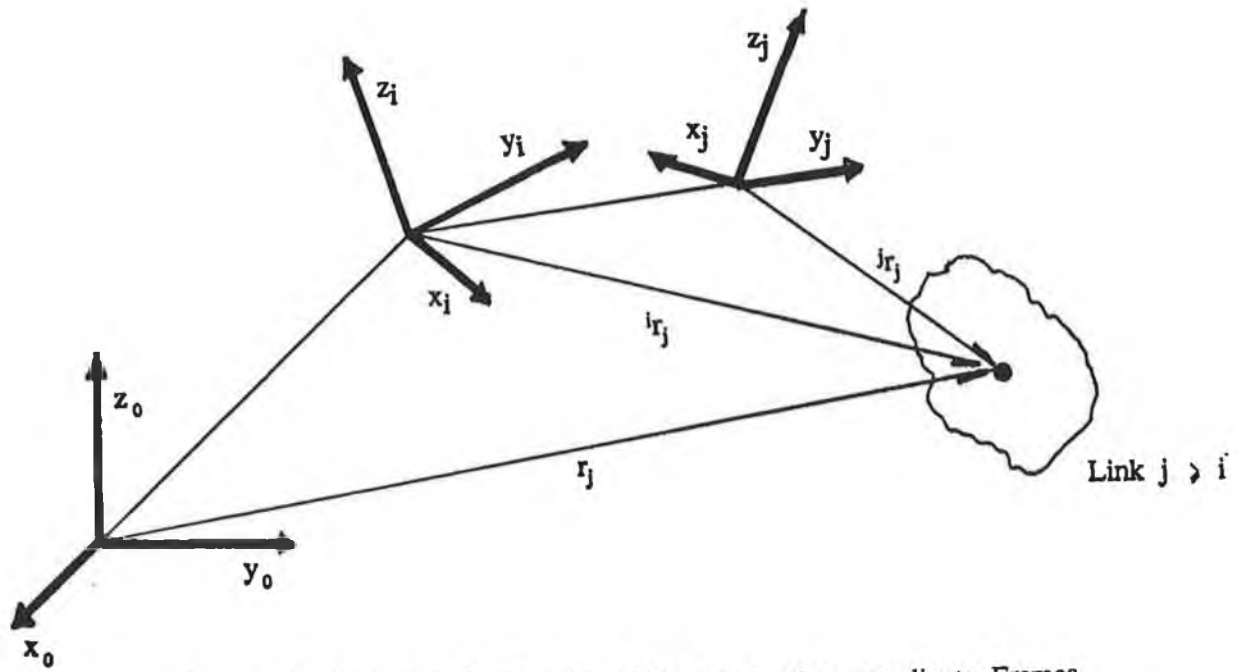


Figure 2.8 Position Of point in a link  $j$  wrt other coordinate Frames

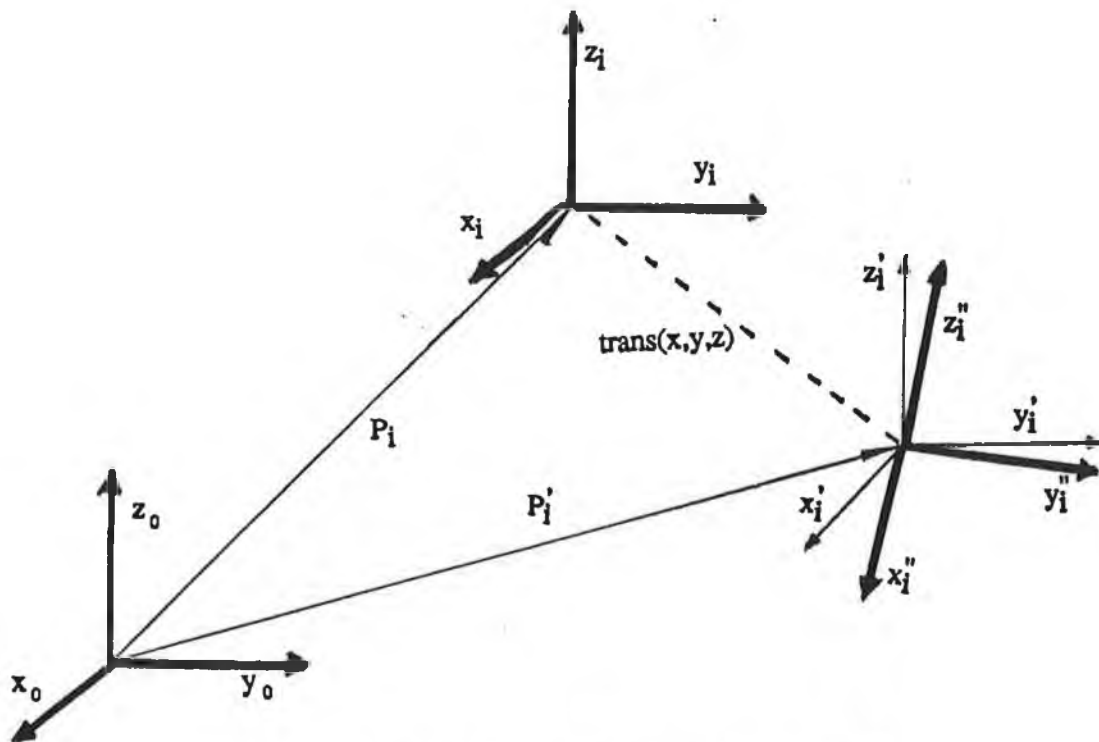


Figure 2.9 Changes in  ${}^0T$  due to changes in  $q_i$

Similarly the vector  $k$  could be written in terms of coordinate frame  ${}^0T$ . If this is the case, then different translation and rotation transformation matrices are obtained.

A new product for  ${}^0T + d_p{}^0T$  results:-

$${}^0T + d_p{}^0T = ( \text{Translation}(d_x, d_y, d_z) \text{Rotation}(k, d\Theta) ) {}^0T$$

and

$$d_p{}^0T = ( \text{Translation}(d_x, d_y, d_z) \text{Rotation}(k, d\Theta) - I ) {}^0T$$

where the two transformation matrices,  $\text{Translation}(d_x, d_y, d_z)$  and  $\text{Rotation}(k, d\Theta)$ , are chosen w.r.t. frame  ${}^0T$ .

${}^p{}^0T + d_p{}^0T$  is the new (position and orientation) of coordinate frame  $p$ .

The bold print signifies that the translation and rotation are described wrt  ${}^0T$ .

In both the above expressions for  $d_p{}^0T$  we find one common subexpression ie.

$$(\text{translation}(d_x, d_y, d_z) \text{Rotation}(k, d\Theta) - I)$$

The elements in this matrix transformation are dependant upon both the translation and rotation being performed, and the reference coordinate frame. We denote this transformation by the delta symbol,  $\Delta$ , and henceforth refer to it as the delta operator. It is necessary to specify what reference frame the delta operator is described with respect to, and what variable is causing the change. To accomodate this, the following notation is used. A preceding superscript is used to denote the reference frame, and a following subscript gives the differential variable causing the change. Thus  ${}^j\Delta_i$  describes the transformation for a differential change, described in the  ${}^jT$  reference frame, due to a change in joint  $i$ .

Because the  $T$  transformations were chosen so that the joint variable  $q_i$  rotates about the  $z_{i-1}$  axis we find that  ${}^{i-1}\Delta_i$  accomodates the differential change in the position of link  $i$  when  $q_i$  moves. The transformation for a differential translation along the vector  $(dx \ dy \ dz)$

$$\text{Trans}(dx \ dy \ dz) = \begin{bmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The transformation for a differential rotation of angle  $dq$  about an axis  $k = (k_x \ k_y \ k_z)$  equals

$$\text{Rot}(k, dq) = \begin{bmatrix} 1 & -k_z dq & k_y dq & 0 \\ k_z dq & 1 & -k_x dq & 0 \\ -k_y dq & -k_x dq & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Thus

$${}^{i-1}\Delta_i = \begin{bmatrix} 0 & -k_z dq & k_y dq & dx \\ k_z dq & 0 & -k_x dq & dy \\ -k_y dq & -k_x dq & 0 & dz \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

### 2.5.3 Calculating differential changes in coordinate frames

For any manipulator, differential changes in the position and orientation of the end effector are due to differential changes in the joint coordinates  $dq_i$ . For a manipulator with six degrees of freedom,  ${}^0T$  describes the position and orientation of the end effector w.r.t. the robot base coordinates for a manipulator with 6 DOF. In order to keep this section as general as possible analysis is done for any transformation frame  ${}^0T$  of the manipulator. For a manipulator with  $p$  degrees of freedom (DOF), the differential change in  ${}^0T$ , due to a differential change in joint  $i$ , may be represented by the matrix product

$$d{}^0T = {}^0T \dots {}^{i-2}T \ {}^{i-1}\Delta_i \ {}^{i-1}T \dots {}^{p-1}T$$

ie. a differential change in joint angle  $i$  (which can be described by a change in  ${}^{i-1}T$ ) causes a differential change,  $d{}^0T$ , in the end effector position and orientation.  ${}^{i-1}\Delta_i$  causes all coordinate frames  ${}^jT$  for  $j \geq i$  to change.  ${}^{i-1}\Delta_i$  is easier to determine than  ${}^p\Delta_i$  so  ${}^{i-1}\Delta_i$  is used when calculating differential

movements throughout the analysis.

The values in the delta operator matrices may be calculated in general form, however, since Puma 560 has revolute joints the delta operator may be simplified. The z axes of the coordinate frames were chosen to coincide with the joint axes. This means that the delta operators  ${}^{i-1}\Delta_i$  will have no translational components, ie.  $(dx \ dy \ dz) = (000)$  and that the differential rotation can only have a z component, ie.  $(k_x \ k_y \ k_z) = (001)$

Therefore

$$\begin{aligned} {}^{i-1}\Delta_i &= \begin{bmatrix} 0 & -dq & 0 & 0 \\ dq & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} dq \\ &= Q \, dq \end{aligned} \tag{2.1}$$

where

$$Q = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.2}$$

$$\begin{aligned} \text{Now } d_p^0T &= {}^0T \dots {}^{i-2}T \, {}^{i-1}\Delta_i \, {}^{i-1}T \dots {}^pT \\ &= {}^0T \dots {}^{i-2}T \, Q \, dq_i \, {}^{i-1}T \dots {}^pT \\ &= {}^0T \dots {}^{i-2}T \, Q \, {}^{i-1}T \dots {}^pT \, dq_i \\ \frac{d_p^0T}{dq_i} &= {}^0T \dots {}^{i-2}T \, Q \, {}^{i-1}T \dots {}^pT \end{aligned} \tag{2.3}$$



This in essence provides us with a means of finding the rate of change of a coordinate frame w.r.t. a joint angle. Naturally this is invaluable for calculating the velocities and accelerations of the links due to joint movements. Second derivatives may also be calculated using the Q matrix. For example

$$\frac{\delta^2 \mathbf{p}^T}{\delta q_i \delta q_j} = {}^0T \dots {}^{i-1}T \mathbf{Q} \quad {}^{i-1}T \dots {}^{j-1}T \mathbf{Q} \quad {}^{j-1}T \dots \mathbf{p}^T \quad ( 2.4 )$$

In all the derivative expressions the Q matrix premultiplies the coordinate frame transformation which is primarily causing the movement.

## 2.6 Summary

This chapter has shown how the kinematic equations for any manipulator can be formed using carefully chosen coordinate frames and a set of homogeneous transformations. The procedure was as follows :

- i) Specify the shape and size of each link
- ii) Embed a coordinate frame in each link
- iii) Formulate a homogeneous transformation to superimpose coordinate frame n-1 onto frame n.

The relative position and orientation on the links are related by the product of the transformation matrices. The manipulator links were specified in terms of their link length and twist angle. The link dimensions of the Puma 560 were given as an example. The criteria for forming the homogeneous transformations relating the various link coordinate frames was detailed and used to generate transformation matrices for the Puma 560. The remainder of the chapter examined how differential manipulator movements could be mathematically described using the delta operator matrix. This matrix provided a means of determining the effect that rotating joint angle n has on the outer link positions. This in turn provided expressions for link velocities caused by moving links.

## CHAPTER 3

### DERIVATION OF SECOND ORDER DYNAMIC EQUATIONS FOR AN ELBOW TYPE MANIPULATOR

This chapter concerns itself with the derivation of dynamic equations for an elbow type manipulator. The dynamic equations relate forces and torques to the positions, velocities and accelerations of the robot links. The physical structure of any manipulator influences the forces that exist at the manipulator joints. Although two main types of manipulator joints exist, revolute and prismatic, only the revolute type is examined here since the Puma 560 has no prismatic joints. The actuator dynamics are not considered at this stage, resulting in force inputs to the joints.

The robot represents a highly nonlinear, multivariable system. A mathematical model describing the nonlinearities and crosscoupling is necessary, firstly to provide insight into the control problem, and secondly for simulation purposes. The dynamic equations have been developed in general form using the systematic approach suggested by Paul [5], by firstly forming the Lagrangian for the complete manipulator, and then substituting this into the Lagrange-Euler equation.

For clarity, the different coefficient elements in the general dynamic equation are grouped together and renamed. The significance of each of these groups is discussed in detail, to give a greater understanding of the dynamic equations' behaviour. It becomes apparent how link shape and construction influence the dynamic equation complexity.

Control system design for high-performance manipulators require thorough understanding of robot dynamics. We often find, however, that control engineering applications ignore the physical interpretations of manipulator dynamics. A global understanding of a plant dynamics is important when choosing a control strategy. A special effort is made in this chapter to qualify the effects that different mechanical parameters have on a manipulator's dynamic equations.

### 3.1 The Generalised Formulation of Dynamics

Research into the motion of material bodies, has been carried out over many centuries. The work has developed into a large area known as analytical mechanics, or dynamics. Goldstein [13] gives a good introduction to the mechanics of moving particles, and explains how the theories may be extended to the encompass systems of many particles. He gives details of how infinitesimal rotations of a body, may be represented using matrix transformations, Goldstein [pp.164-183], and analyses the forces experienced as a result of these movements.

When deriving a model for a complex system, such as a manipulator it is necessary to follow a very systematic formulation procedure in order to avoid unnecessary errors. For a structured system, like a manipulator, it is convenient to make use of two well established fields of classical mechanics.

Kinematics : The study and description of possible motions of material bodies.

and

Dynamics : The study of the laws which determine among all possible motions which motion will actually take place in any given case.

Treating a system, such as a manipulator, as an assembly of point masses in a structured set of coordinate frames, makes it possible to generate the plant dynamic equations. The derivation given in this chapter uses this type of approach.

Several approaches for modelling arm dynamics have been taken by various researchers. The Euler-Lagrangian [5], recursive Lagrangian [11], and Newton-Euler [6] are popular methods. The closed form dynamic models, based on the first two Lagrange derivations, yield equations suitable for both direct and inverse dynamic problems. This compact method produces equations-of-motion that are appropriate for modelling and control applications. The Newton-Euler formulation consists of a set of backwards and forwards equations that recursively calculate the complete dynamics. These are significantly more efficient than Euler-Lagrange, and more suitable for real-time control [5]. However they require explicit representations of the internal forces in the system, and the

equation coefficients are difficult to interpret.

The basic distinction between the Euler-Lagrange and Newton Euler methods is that while the Euler-Lagrangian formulation produces equations that are easily understood (ie the equation parameters are distinctly separate) but computationally inefficient, the Newton-Euler approach gives more compact equations with parameters that are less separable.

The Euler-Lagrangian formulation was chosen in preference to the other methods because a) it generates dynamic equations in a systematic manner, reducing the probability of errors and b) a non-recursive formulation was considered more suitable for simulation and analysis because of the explicit nature of the equation solutions.

### 3.1.1 The Lagrangian and Euler-Lagrange Equation

The dynamics of any manipulator can be described by a set of 2nd order differential equations. These equations relate the torque experienced on each joint to the positions, velocities, and accelerations of the robot links. To simplify many of the expressions we will make use of the summation convention first used by Einstein: Whenever an index occurs two or more times in a term, it is implied, without any further symbols, that the terms will be summed over all possible values of the index. Throughout this thesis, the summation convention will be used in equations, unless otherwise stated. Where convenient, or to remove ambiguity, the summation sign  $\Sigma$  may be occasionally displayed explicitly, eg when certain values are excluded from the summation.

The Lagrangian is defined to be the difference of the kinetic and potential energies of the system :-

$$L = K - P$$

where      L = Lagrangian  
              K = Kinetic energy  
              P = Potential energy

The dynamic equations are formulated from the Euler - Lagrangian equation :-

$$F_i = \frac{d}{dt} \frac{\delta L}{\delta \dot{q}_i} - \frac{\delta L}{\delta q_i}$$

where  $q_i$  are the system variables, (joint angles in the Puma robot case) and where  $F_i$  is the force experienced on the system variables, (torques in the case of the Puma), as detailed in Goldstein [13]. Once the the potential and kinetic energies have been expressed in an appropriate coordinate frames we are assured that the forces  $F_i$  for each joint can be calculated.

### 3.2 Forming the Lagrangian for a Manipulator

Section 3.1 presented the Euler-Lagrange equation. This will now be used to generate dynamic equations for a mechanical system. The first step in producing the Lagrangian is to derive expressions for the kinetic and potential energies for the manipulator. The kinetic and potential energies are formed in general terms, using the structural information held in the T transformation matrices presented in Chapter 2, and by treating each of the robot links as a collection of distributed point masses. This method caters for the nonuniform density of the link by defining separate inertias, radii of gyration and masses for each link. The significance of these parameters is detailed later in this chapter. When expressions for the kinetic and potential energies are found the Lagrangian is formed as their difference. By substituting the Lagrangian into the Euler-Lagrangian equation, a system of dynamic equations is produced.

#### 3.2.1 Kinetic energy

In order to calculate the kinetic energy of a manipulator system, it is necessary to calculate the velocity of all point masses in each link, at any point in time. The link velocities can be determined once the structure of the manipulator and the angular velocities of the manipulator joints are known. The manipulator structural information is partially held in the T matrices. Section 2.6 details were given of how point masses in a link may be specified w.r.t. any coordinate frame, while Section 2.7 explained how link velocities are obtained. Once the velocity of the links are known, values of link masses, radii of gyration, and inertias are used to calculate the manipulator kinetic energy .

Kinetic energy	$k = 1/2 mv^2$
where	$v = \text{velocity}$
	$m = \text{mass}$

The velocity w.r.t. the base frame of a point mass situated at position  $r$  is,

$$\left[ \frac{dr}{dt} \right] = \dot{r}$$

where  $\dot{r}$  equals the rate of change of position  $r$  w.r.t. time.

We need to extend this equation to incorporate all masses in the various manipulator links. The velocity of a point volume in a link is found by summing the velocity contributions of all the robot joints ( $j \leq n$ ). For example, the velocity of a point mass in link 4 may be found by

$$\begin{aligned} \frac{dr}{dt} &= \frac{\delta ( {}^0T_4 r_4 )}{\delta t} \\ &= \left[ \begin{array}{cc} \frac{\delta {}^0T}{\delta q_j} & \frac{\delta q_j}{\delta t} \end{array} \right] {}^4r_4 \\ &= \left[ \begin{array}{cccc} \frac{\delta {}^0T}{\delta q_1} \frac{\delta q_1}{\delta t} + \frac{\delta {}^0T}{\delta q_2} \frac{\delta q_2}{\delta t} + \frac{\delta {}^0T}{\delta q_3} \frac{\delta q_3}{\delta t} + \frac{\delta {}^0T}{\delta q_4} \frac{\delta q_4}{\delta t} & & & \end{array} \right] {}^4r_4 \\ &= \left[ \begin{array}{c} \sum_{j=1}^4 \frac{\delta {}^0T}{\delta q_j} q_j \end{array} \right] {}^4r_4 \end{aligned}$$

In general terms velocity of a point mass at  ${}^i r_i$  described w.r.t. the base is

$$\frac{dr_i}{dt} = \left[ \sum_{j=1}^i \frac{\delta j^T}{\delta q_j} q_j \right] {}^i r_i \quad (3.1)$$

The mass position vector,  ${}^i r_i = ( {}^i x_i \ {}^i y_i \ {}^i z_i \ 1 )$ , is the vector from the origin of link  $i$ , to the point mass in the link. Velocity squared equals the dot product

$$\left[ \frac{dr_i}{dt} \right]^2 = {}^i r_i \cdot {}^i r_i$$

In matrix form this may be represented as

$$\left[ \frac{dr_i}{dt} \right]^2 = \text{trace} [ {}^i r_i {}^i r_i^T ] \quad (3.2)$$

${}^i r_i^T$  represents the transpose of the mass position vector. By substitution

$$\left[ \frac{dr_i}{dt} \right]^2 = \text{trace} \left[ \sum_{j=1}^i \frac{\delta T_i}{\delta q_j} q_j \ {}^i r_i \left[ \sum_{k=1}^i \frac{\delta T_i}{\delta q_k} q_k \ {}^i r_i \right]^T \right] \quad (3.3)$$

Rearranging

$$\left[ \frac{dr_i}{dt} \right]^2 = \text{trace} \left[ \sum_{j=1}^i \sum_{k=1}^i \frac{\delta T_i}{\delta q_j} \frac{\delta T_i}{\delta q_k} {}^i r_i \ {}^i r_i^T q_j q_k \right] \quad (3.4)$$

The kinetic energy of this differential mass is simply:-

$$dK_i = \frac{1}{2} \text{trace} \left[ \sum_{j=1}^i \sum_{k=1}^i \frac{\delta T_i}{\delta q_j} ( {}^i r_i \ {}^i r_i^T ) \frac{\delta T_i}{\delta q_k} q_j q_k \right] dm \quad (3.5)$$

$$= \frac{1}{2} \text{trace} \left[ \sum_{j=1}^i \sum_{k=1}^i \frac{\delta T_i}{\delta q_j} ( {}^i r_i \ dm \ {}^i r_i ) \frac{\delta T_i}{\delta q_k} q_j q_k \right] \quad (3.6)$$

In order to calculate the kinetic energy of a link, it is necessary to sum the kinetic energy of all link particles, over the volume of the link. We have already derived an expression for the square of the velocity of any point mass in a link. The total kinetic energy of the link  $i$  equals the integral of the differential kinetic energies over the complete volume of the link.

$$K_i = \int_{\text{link } i} d K_i$$

$$= \frac{1}{2} \text{trace} \left[ \sum_{j=1}^i \sum_{k=1}^i \frac{\delta T_i}{\delta q_j} \int_{\text{link } i} ({}^i r_i \quad {}^i r_i^T) dm \frac{\delta T_i}{\delta q_k} q_j q_k \right]$$

but  ${}^i r_i = ({}^i x_i \quad {}^i y_i \quad {}^i z_i \quad 1)$  thus

$$J_i = \int_{\text{link } i} ({}^i r_i) ({}^i r_i^T) dm \quad (3.7)$$

$$\int_{\text{link}} ({}^i r_i) ({}^i r_i^T) dm = \int_{\text{link}} \begin{bmatrix} {}^i x_i^2 & {}^i x_i {}^i y_i & {}^i x_i {}^i z_i & {}^i x_i \\ {}^i x_i {}^i y_i & {}^i y_i^2 & {}^i y_i {}^i z_i & {}^i y_i \\ {}^i x_i {}^i z_i & {}^i y_i {}^i z_i & {}^i z_i^2 & {}^i z_i \\ {}^i x_i & {}^i y_i & {}^i z_i & 1 \end{bmatrix} dm$$

$$K_i = \int_{\text{link } i} d K_i = \frac{1}{2} \text{trace} \left[ \sum_{j=1}^i \sum_{k=1}^i \frac{\delta T_i^T}{\delta q_j} J_i \frac{\delta T_i^T}{\delta q_k} q_j q_k \right] \quad (3.8)$$

The matrix  $J_i$  given in equation (3.8) is often referred to as the pseudo inertial matrix for link  $i$ . It has a similar format to the inertial tensor matrix, which is well documented in literature, Goldstein [13]. Appendix A gives details of generation and physical interpretation of  $J_i$ .

$$J_i = \begin{bmatrix} \frac{-m_i k_{pxx}^2 + m_i k_{pyy}^2 + m_i k_{pzz}^2}{2} & 0 & 0 & m_i \bar{x}_i \\ 0 & \frac{m_i k_{pxx}^2 - m_i k_{pyy}^2 + m_i k_{pzz}^2}{2} & 0 & m_i \bar{y}_i \\ 0 & 0 & \frac{m_i k_{pxx}^2 + m_i k_{pyy}^2 - m_i k_{pzz}^2}{2} & m_i \bar{z}_i \\ m_i \bar{x}_i & m_i \bar{y}_i & m_i \bar{z}_i & m_i \end{bmatrix} \quad (3.9)$$



The elements of the pseudo inertial matrix,  $J_i$ , are functions of the radius of gyration of the  $i$ . The radius of gyration is related to the link inertias and the mass by the following equalities:

$$\begin{aligned}
 m_i k_{pxx}^2 &= I_{pxx} \\
 m_i k_{pyy}^2 &= I_{pyy} \\
 m_i k_{pzz}^2 &= I_{pzz} \\
 m_i k_{pxy}^2 &= I_{pxy} \\
 m_i k_{pyz}^2 &= I_{pyz} \\
 m_i k_{pxz}^2 &= I_{pxz}
 \end{aligned} \tag{3.10}$$

Where

- .  $I_{ixx}$  represents the moment of inertia about the x axis of the link  $i$  coordinate frame,
- .  $I_{iyy}$  represents the moment of inertia about the y axis of the link  $i$  coordinate frame,
- .  $I_{izz}$  represents the moment of inertia about the z axis of the link  $i$  coordinate frame,
- .  $I_{ixy}$ ,  $I_{ixz}$ ,  $I_{ixy}$ , represent the cross products of inertia for link  $i$ , about the various coordinate frame axes,
- .  $m_i$  equals the mass of link  $i$ ,
- .  $\bar{x}_i$ ,  $\bar{y}_i$ ,  $\bar{z}_i$  are the x, y, & z components of the link centres of masses w.r.t. link  $i$  coordinate frame,
- .  $m_i \bar{x}_i$ ,  $m_i \bar{y}_i$ ,  $m_i \bar{z}_i$ , are the first moments of link  $i$ .

It is reasonable to assume that the cross products of inertia, are approximately zero, Goldstein [13]. This accounts for the zero elements in the  $J_i$  matrix.

The radius of gyration gives a clear indication of the link mass distribution. The radius of gyration is the radius such that if all the mass of the link were situated a distance  $k_i^2$  from the axis, its moment of inertia would be  $I_i$ .

The total kinetic energy of the manipulator links, equals the sum of the kinetic energies of the individual links. Therefore

$$K = \sum_{i=1}^6 K_i = \frac{1}{2} \sum_{i=1}^6 \text{trace} \left[ \sum_{j=1}^i \sum_{k=1}^i \frac{\delta T_i}{\delta q_j} J_i \frac{\delta T_i}{\delta q_k} q_j q_k \right] \tag{3.11}$$

Actuator kinetic energy must also be taken into account when calculating the total kinetic energy of the manipulator. Because of the gearing ratios at each link joint we find that the actuator kinetic energy is multiplied by a factor of the square of the gear ratio.

$$K_{act\ i} = \frac{1}{2} I a_i \dot{q}_i^2 \quad (3.12)$$

$\dot{q}_i$  = velocity of rotation of joint i.

$I a_i$  = inertia of motor i

The kinetic energy of the complete manipulator can now be written as the sum of the kinetic energies of all six links.

$$K = \frac{1}{2} \sum_{i=1}^6 \sum_{j=1}^6 \sum_{k=1}^6 \text{trace} \left[ \frac{\delta T_i}{\delta q_j} J_i \frac{\delta T_i^T}{\delta q_k} \right] q_j q_k + \frac{1}{2} \sum_{i=1}^6 I a_i \dot{q}_i^2 \quad (3.13)$$

### 3.2.2 Potential energy

The potential energy of a mass, m, a distance h above the reference plane, in a gravity field g is given by:-

$$\text{Potential energy } P = mgh \quad (3.14)$$

The potential energy of each link i will firstly be calculated w.r.t. coordinate frame i. Figure 3.1 shows a typical link and its associated reference frame. Two vectors have been drawn on this diagram.  ${}^i\bar{r}_i$  which describes the position of the link centre of mass w.r.t. frame i, and g which gives the direction in which gravity acts upon the link. As the link moves  ${}^i\bar{r}_i$  remains fixed but g changes

$$\begin{aligned} g^T &= (g_x \ g_y \ g_z \ 1) \\ |g| &= 9.81 \text{ m/s}^2 \end{aligned}$$

Potential energy of any point mass equals the vector dot product of the gravity vector and the position vector, scaled by the mass.

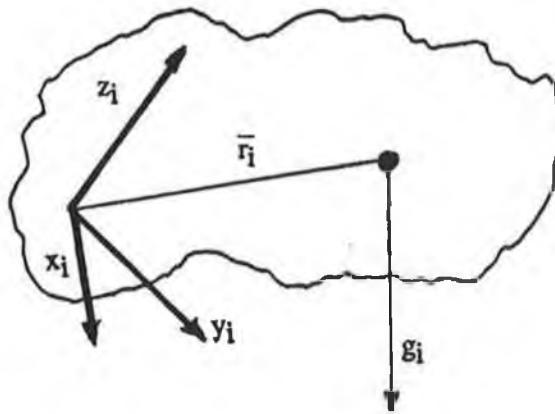


Figure 3.1 Gravity vector and centre of Mass Vector

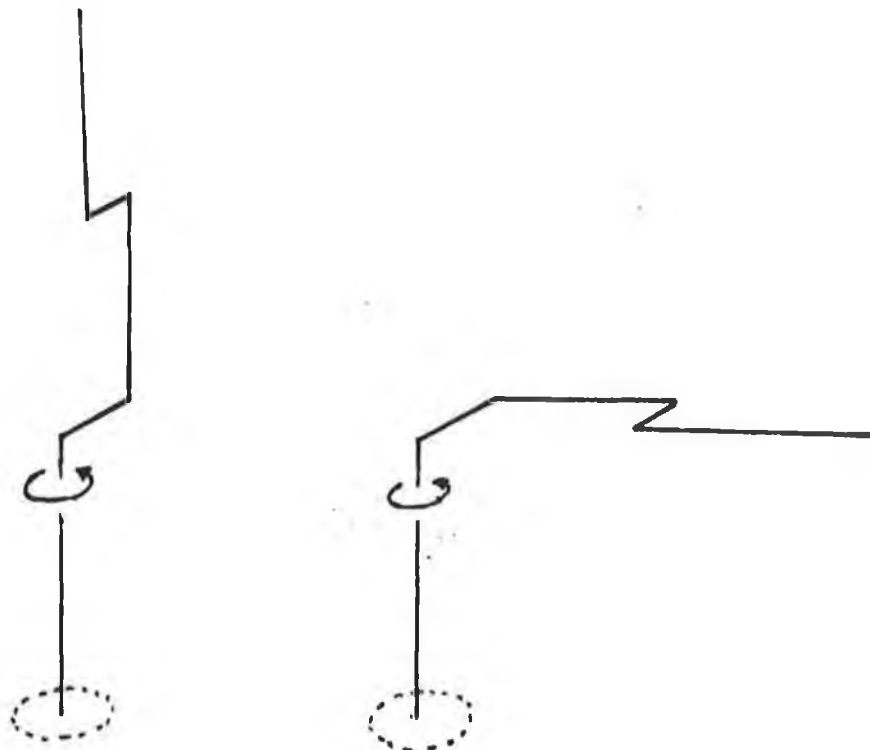


Figure 3.2(a) Manipulator in vertical pose

Figure 3.2(b) manipulator in horizontal pose

$$P = -m \mathbf{g} \cdot \bar{\mathbf{r}} \quad (3.15)$$

The negative sign is caused by the choice of the gravity vector. In order to express the potential energy of link  $i$  w.r.t. other coordinate frames we must transform it with a suitable transformation matrix. To achieve this we use the  $T$  matrices given in section 2.3. Thus the potential energy of link  $i$  w.r.t. the base is

$$P_i = -m_i (\mathbf{g}^T) \cdot ({}^0_i T \mathbf{r}_i)$$

The potential energy ( w.r.t. link  $i$  coordinate frame ) whose centre of mass is at  $\mathbf{r}_i$  is :-

$$P_i = -m_i (\mathbf{g}^T) \cdot (T_i \mathbf{r}_i) \quad (3.16)$$

Total potential energy of the manipulator is :-

$$P = - \sum_{i=1}^N m_i \mathbf{g}^T T_i \mathbf{r}_i \quad (3.17)$$

Where  $N$  equals the number of degrees of freedom of the manipulator.

### 3.2.3 The General Manipulator Lagrangian

Using the expressions given above for kinetic and potential energies the Lagrangian is formed as the difference between kinetic and potential energy.

$$L = K - P$$

$$L = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^i \sum_{k=1}^i \text{trace} \left[ \frac{\delta T_i}{\delta q_j} J_i \frac{\delta T_i^T}{\delta q_k} \right] q_j q_k + \frac{1}{2} \sum_{i=1}^N I a_i q_i^2 + \sum_{i=1}^N m_i \mathbf{g}^T T_i \mathbf{r}_i \quad (3.18)$$

We can now generate dynamic equations in general terms for any  $N$  degree of freedom manipulator by substituting the Lagrangian into the Euler-Lagrange equation:-

$$F_i = \frac{d}{dt} \frac{\delta L}{\delta \dot{q}_i} - \frac{\delta L}{\delta q_i} \quad (3.19)$$

giving

$$F_i = \sum_{j=i}^N \sum_{k=1}^j \text{trace} \left[ \frac{\delta T_j}{\delta q_k} J_j \frac{\delta T_j^T}{\delta q_k} \right] q_k + I a_i q_i$$

$$+ \sum_{j=i}^N \sum_{k=1}^j \sum_{m=1}^j \text{trace} \left[ \frac{\delta^2 T_j}{\delta q_k \delta q_m} J_j \frac{\delta T_j^T}{\delta q_i} \right] q_m q_k - \sum_{j=i}^N m_j g^T \frac{\delta T_j}{\delta q_i} j \bar{r}_j$$

where the summation convention is used throughout the analysis.

### 3.3 General Manipulator Dynamic Equation

The dynamic equations given in equation (3.17) relate the forces experienced at each to the manipulator joints to the present positions, velocities and accelerations of the links. In the present form it is difficult to extract physical meaning from the coefficients. The equations may be rewritten, grouping and renaming similar terms for clarity as:

$$F_i = \sum_{j=1}^N D_{ij} \dot{q}_j + I a_i q_i + \sum_{j=1}^N \sum_{k=1}^N C_{ijk} q_j \dot{q}_k + G_i \quad (3.21)$$

$q_i, \dot{q}_i, \ddot{q}_i$  = system variables, joint  $i$  position, velocity, and acceleration,

$F_i$  = torque acting on joint  $i$ ,

$D_{ii}, D_{ij}$  = effective and coupling inertias for joint  $i$ ,

$I a_i$  = actuator inertia of joint  $i$ ,

$C_{ijj}, C_{ijk}$  = centripetal and coriolis forces for joint  $i$ ,

$G_i$  = gravity loading for joint  $i$ ,

The coefficient terms  $D_{ij}, I_i, C_{ijk}$  and  $G_i$  are highly nonlinear functions of the robot states (position velocity and accelerations). Terms which multiply similar states have been grouped together.

### 3.3.1 Analysis of Self Inertial Terms

The inertial coefficients reflect the accelerational coupling between the robot links.

$$D_{ii} = \sum_{p=1}^N \text{trace} \left[ \frac{\delta T_p}{\delta q_i} J_p \frac{\delta T_j^T}{\delta q_i} \right] \quad (3.22)$$

$D_{ii}$  is called the self-inertia or effective inertia at joint  $i$ . Acceleration at joint  $i$  will cause a torque equal to  $D_{ii} \ddot{q}_i$  at the joint. As with all the dynamic equation coefficients  $D_{ii}$  terms depend upon all the states of the manipulator. For example Figure 3.2 shows two possible configurations for the Puma 560 robot. In Figure 3.2(a) the robot arm is vertical. The waist joint will find it easy to accelerate when in this position ie. the effective inertia at joint 1 ( $D_{1,1}$ ) is relatively small. In Figure 3.2(b) however acceleration is more difficult to achieve due to the extra work required to rotate links 2 and 3 over a larger distance. In this case  $D_{1,1}$  is at its maximum value. Obviously  $D_{ii}$  is highly nonlinear due to its dependency on the system state.

### 3.3.2 Analysis of Coupling Inertia Terms

$$D_{ij} = \sum_{p=(\max i, j)}^N \text{trace} \left[ \frac{\delta T_p}{\delta q_j} J_p \frac{\delta T_p^T}{\delta q_i} \right] \quad (3.23)$$

$D_{ij}$  is known as the coupling inertia between joints  $i$  and  $j$ , since it quantifies the effect that acceleration at joint  $i$  (or  $j$ ) will have on the torque experienced at joint  $j$  (or  $i$ ). The torque equals  $D_{ij} \ddot{q}_i$  (or  $D_{ij} \ddot{q}_j$ ). Figure 3.3 shows how acceleration of link 3 causes a torque at joint 2. Again we find that the coefficient term  $D_{ij}$  is nonlinear because of its dependance on manipulator positions.

$D_{ij} = D_{ji}$  because of the Newtons second law of dynamics. ie. for each action there is an equal and opposite reaction.

### 3.3.3

### Analysis of Centripetal and Coriolis Terms

The centripetal and coriolis terms indicate the velocity coupling of the links. Both centripetal and coriolis forces are related to the inertial terms as is apparent by comparing equations (3.22) and (3.24). Both are essentially derivatives of inertial terms. As such, many of the simplifying properties of the inertial terms may be extended to include coriolis and centripetal forces.

$$C_{ijj} = \sum_{p=\max(i,j)}^N \text{trace} \left[ \frac{\delta^2 T_D}{\delta q_j \delta q_j} J_P \frac{\delta T_P^T}{\delta q_i} \right] \quad (3.24)$$

Centripetal force  $C_{ijj} \dot{q}_j$  characterises the effect that the velocity of joint  $j$  has a link  $i$ . For example if an elbow type manipulator was in the position shown in Fig 3.4 and if joint 1 was rotating at a velocity  $\dot{q}_1$ , then link 3 would experience an outward force equal to  $C_{311} \dot{q}_1 \dot{q}_1$ . No joint will experience a centripetal force due to its own rotation thus  $C_{iii} = 0$ . Centripetal forces are unique to rotational joints.

$$C_{ijk} = \sum_{p=\max(i,j,k)}^N \text{trace} \left[ \frac{\delta^2 T_D}{\delta q_j \delta q_k} J_P \frac{\delta T_P^T}{\delta q_i} \right] \quad (3.25)$$

Coriolis coefficient  $C_{ijk}$  gives the effect that coupling of velocities of links  $j$  and  $k$  have a link  $i$ . Thus  $C_{123}$  refers to the effect that the velocities of links 2 and 3 have a joint 1.

Coriolis forces are not exerted on joint  $i$ , by velocities at joints  $j$  and  $k$ , if either  $j$  or  $k$  are less than  $i$ . Two other simplifications of the terms are possible

$$(a) \quad C_{iji} = 0 \quad \text{for all } i \geq j \quad (3.26)$$

$$(b) \quad C_{ijk} = -C_{kji} \quad \text{for } i, k \geq j \quad (3.27)$$

Equation (3.26) is the non-interacting property and states that there is no coriolis force on a link due to coupling of its velocity with the velocity of an inner link. The reflective coupling given in equation (3.27) says that the coriolis coefficient for link  $i$ , due to the coupling of the velocities of links  $j$  and  $k$  ( $j \geq k, i$ ) is equal and opposite to the coriolis coefficient for link  $m$  due to the coupled velocities of links  $j$  and  $i$ .

The coriolis forces account for the difference between forces measured in rotating and non rotating coordinate frames. The difference results from the fact that the energies of a particle moving in the two frames are different [14]. The rotation of the first frame adds to the total energy a component which depends upon the coordinates of the particle, and is proportional to the angular velocity of the rotating frame. Coriolis force is the result of coupling between the velocities of an inner rotational link and an outer link.

### 3.3.4 Analysis of Gravity Loading Terms

$$G_i = \sum_{j=i}^N m_j g^T \frac{\delta T_j}{\delta q_i} j \bar{r}_j \quad (3.28)$$

In the case of an elbow type manipulator, like the Puma robots, joint 2 and 3 always experience a torque due to gravitational forces. Joint 1, however, rotates around an axis that is parallel to the gravity field. Thus

$$g \cdot (z_{i-1} \times {}^i r_i) = 0$$

$$G_1 = 0$$

The gravitational term of a joint specifies the work required to overcome the gravity field. Counterbalances are often employed on the main links to reduce the effects of gravity. By altering the link's mass distribution (and by bringing the links centre of mass closer to the rotational axis) the gravitational coefficient can be reduced.

### 3.3.5 Significance of Coefficient Terms in Control Environment

Gravity loading provides a control problem at all positions and, along with inertial terms influences the manipulator servo and positioning accuracy. The inertial terms have their greatest effect when the joints are accelerating quickly but, even at low accelerations, their effect is relatively large. Actuator inertias are often large especially when magnified through the gearing systems. In the smaller links the actuator inertias often overshadow effective and coupling inertias. Centripetal and coriolis forces are significant at high speeds however their effect is often minimal. It is difficult to ascertain whether some of the



dynamic contributions may be ignored before model simulations have been carried out. For some manipulator structures such as cartesian robot arms, coriolis and centripetal terms are nonexistent. In elbow type manipulators, however, all dynamic coefficients exist.

### 3.4 Summary

This chapter has used the Euler-Lagrangian method for formulating dynamic equations for a N degree of freedom manipulator. The procedure took the following steps:

- i) Kinetic energy for general masses in a link are found.
- ii) The pseudo inertial matrix which specifies the mass distribution of a link was discussed.
- iii) The total kinetic energy of the manipulator was obtained
- iv) The total potential energy for the manipulator are found.
- v) The Lagrangian for the manipulator are formed.
- vi) The terms in the resultant dynamic equation are regrouped.

The transformation matrices provided the framework upon which the manipulator's Lagrangian was formed. The dynamic equation coefficients were collected into groups describing classical mechanical properties of manipulator. A thorough explanation of the physical significance of each of these groups is given, to aid the model validation and tuning.

## CHAPTER 4

### SECOND ORDER DYNAMIC EQUATIONS FOR THE PUMA 560 MANIPULATOR

The coefficient terms of the dynamic equations, which are highly coupled and nonlinear functions of the robot joint angles, link dimensions, masses, inertias etc. need to be evaluated. Only the dynamics of the first three links of the Puma 560 are examined in this thesis. Control of the Puma gripper can be treated as a separate problem. Knowledge of the angular position of the first 3 links gives end effector position, while tool orientation is determined by the wrist joints. The positioning and path tracking problem is dominated by the dynamics of the first three links. Normally, when tracking a given path, the gripper and payload are treated as a lumped mass on link three. This does not introduce significant errors since the last three links are dimensionally small and the payload travels with the third link.

This chapter presents a set of second order differential equations which describe the dynamics of the first three links. Actuator dynamics have not been considered at this stage. Since the robot dynamic equations require accurate estimates of the parameters this chapter also examines how values for link dimensions, masses, inertia, and radii of gyration for the first three links can be accurately estimated.

#### 4.1 General Second Order Dynamic Equations for a Manipulator

The Lagrangian formulation of the second order differential equations for a manipulator with  $n$  degrees of freedom is given from eqn 3.21 as

$$F_i = \sum_{j=1}^N D_{ij} \ddot{q}_j + I a_i \dot{q}_i + \sum_{j=1}^N \sum_{k=1}^N C_{ijk} \dot{q}_j \dot{q}_k + D_i \quad i = 1 \dots n \quad (4.1)$$

where

$q_i, \dot{q}_i, \ddot{q}_i$  = system variables, joint  $i$  position, velocity, and acceleration,

$F_i$  = torque acting on joint  $i$ ,

$D_{ii}$  ,  $D_{ij}$  = effective and coupling inertias for joint  $i$ ,  
 $Ia_i$  = actuator inertia of joint  $i$ ,  
 $C_{ijj}$  ,  $C_{ijk}$  = centripetal and coriolis forces for joint  $i$ ,  
 $G_i$  = gravity loading for joint  $i$ ,

#### 4.2 Matrix representation of dynamic equations

For the purpose of simulation and control it is convenient to describe the robot dynamic equations in matrix form. Although the equations are highly nonlinear and cross-coupled they can be written in a state space type of format, once the following matrices and vectors are defined. The three state vectors are chosen so that the first element in each vector refers to joint angle one, the second to joint angle two, and the third to joint angle three, thus

$\mathbf{q} = (q_1 \ q_2 \ q_3)^T \in \mathbf{R}^3$  contains joint angles,  
 $\dot{\mathbf{q}} = (\dot{q}_1 \ \dot{q}_2 \ \dot{q}_3)^T \in \mathbf{R}^3$  contains joint velocities, and  
 $\ddot{\mathbf{q}} = (\ddot{q}_1 \ \ddot{q}_2 \ \ddot{q}_3)^T \in \mathbf{R}^3$  contains joint accelerations.

The dynamic equation, eqn 4.1, may be written as:-

$$D(\mathbf{q})\ddot{\mathbf{q}} + I_a \dot{\mathbf{q}} + P(\dot{\mathbf{q}}, \mathbf{q}) = \mathbf{F} \quad (4.2)$$

if the following matrices are defined

$$D(\mathbf{q}) = \begin{bmatrix} D_{11} & D_{12} & D_{13} \\ D_{21} & D_{22} & D_{23} \\ D_{31} & D_{32} & D_{33} \end{bmatrix} \in \mathbf{R}^{3 \times 3} \quad (4.3)$$

which contains all the effective and coupling inertial terms.

$$I_a = \begin{bmatrix} Ia_1 & 0 & 0 \\ 0 & Ia_2 & 0 \\ 0 & 0 & Ia_3 \end{bmatrix} \in \mathbf{R}^{3 \times 3} \quad (4.4)$$

$I_a$  contains the reflected motor inertia terms.

$$P(\dot{\mathbf{q}}, \mathbf{q}) = \begin{bmatrix} \dot{q}^T C^1 \dot{q} + G_1 \\ \dot{q}^T C^2 \dot{q} + G_2 \\ \dot{q}^T C^3 \dot{q} + G_3 \end{bmatrix} \in \mathbf{R}^{1 \times 3} \quad (4.5)$$

contains the effect of centripetal, coriolis and gravity forces.

where

$$C^1 = \begin{bmatrix} C_{111} & C_{112} & C_{113} \\ C_{121} & C_{122} & C_{123} \\ C_{131} & C_{132} & C_{133} \end{bmatrix} \in \mathbb{R}^{3 \times 3} \quad (4.6)$$

contains the centripetal and coriolis forces experienced by joint one.

$$C^2 = \begin{bmatrix} C_{211} & C_{212} & C_{213} \\ C_{221} & C_{222} & C_{223} \\ C_{231} & C_{232} & C_{233} \end{bmatrix} \in \mathbb{R}^{3 \times 3} \quad (4.7)$$

contains the centripetal and coriolis forces experienced by joint two.

$$C^3 = \begin{bmatrix} C_{311} & C_{312} & C_{313} \\ C_{321} & C_{322} & C_{323} \\ C_{331} & C_{332} & C_{333} \end{bmatrix} \in \mathbb{R}^{3 \times 3} \quad (4.8)$$

contains the centripetal and coriolis forces experienced by joint three.

The matrices defined above allow us to present the second order dynamic equations in a state space type of format. This proved very convenient when simulating. The elements of these matrices are given from equations (3.28), (3.23) and (3.24) as:

$$G_i = \sum_{p=i}^3 -m_p g^T \frac{\delta T_p}{\delta q_i} P \bar{R}_p$$

$$D_{ij} = \sum_{p=i}^3 \text{trace} \left[ \frac{\delta T_p}{\delta q_j} J_p \frac{\delta T_p}{\delta q_i} \right]$$

$$C_{ijk} = \sum_{p=\max(i,j,k)}^3 \text{trace} \left[ \frac{\delta^2 T_p}{\delta q_j \delta q_k} J_p \frac{\delta T_p}{\delta q_i} \right]$$

The element values in the pseudo inertial matrix,  $J_p$  (see eqn (3.9)), are central to the correct evaluation of both the inertial, centripetal and coriolis parameters.

### 4.3

#### Calculation of the Coefficient terms

Equations 3.28, 3.29 and 3.24 give expressions for each of the second order dynamic terms. In this section these terms are expanded and symbolic expressions for these terms are generated. Section 3.5.3 discussed the physical significance of each of the terms and also explained how several of the terms are equivalent. These relationships are as listed below

$$D_{ij} = D_{ji} \quad (3.25)$$

$$C_{ijk} = C_{ikj} \quad (3.26)$$

$$C_{ijk} = -C_{kji} \quad \text{for all } i, k \geq j, \quad (3.27)$$

$$C_{iji} = 0 \quad \text{for } i \geq j \quad (3.29)$$

#### 4.3.1 Calculation of Effective and Coupling Inertial Terms

All the terms will now be calculated. The elements of the inertial matrix,  $D(q)$ , are calculated using eqn (3.23) as detailed below

$$D_{11} = \sum_{p=1}^3 \text{trace} \left[ \begin{array}{cc} \frac{\delta T_p}{\delta q_1} & J_p \\ & \frac{\delta T_p^T}{\delta q_1} \end{array} \right]$$

Using eqn (2.1) to relate the  $\Delta$  and  $Q$  operators and eqn (2.3) to obtain the derivatives of the homogeneous transformations expanding gives

$$\begin{aligned} D_{11} = & \text{trace} ( \Delta \quad {}^0T \quad J_1 \quad {}^0T^T \quad \Delta^T ) \\ & + \text{trace} ( \Delta \quad {}^0T \quad {}^1T \quad J_2 \quad {}^0T^T \quad {}^1T^T \quad \Delta^T ) \\ & + \text{trace} ( \Delta \quad {}^0T \quad {}^1T \quad {}^2T \quad J_3 \quad {}^0T^T \quad {}^1T^T \quad {}^2T^T \quad \Delta^T ) \end{aligned}$$

Substituting for the Puma 560 homogeneous transformation matrices, given in Section 2.3, and pseudo inertial matrices as presented in eqn 3.9, the expression for the  $D_{11}$  term is:

$$\begin{aligned} D_{11} = & m_1 k_{1zz}^2 \\ & + m_2 (k_{2xx}^2 S_2^2 + k_{2yy}^2 C_2^2 + a_2^2 C_2^2 + 2a_2 \bar{x}_2 C_2^2) \\ & + m_3 [ k_{3xx}^2 S_{23}^2 + k_{3zz}^2 C_{23}^2 + d_3 + a_2^2 C_2^2 + a_3^2 C_{23}^2 + 2a_2 a_3 C_2 C_{23} \\ & + 2\bar{x}_3 (a_2 C_2 C_{23} + a_3 C_{23}^2) + 2\bar{y}_3 d_3 + 2\bar{z}_3 (a_3 C_{23} S_{23} + a_2 C_2 S_{23}) ] \\ & \dots (4.9) \end{aligned}$$

$$\begin{aligned}
\text{where } S_i &= \sin(q_i) \\
C_i &= \cos(q_i) \\
S_{ij} &= \sin(q_i + q_j) \\
C_{ij} &= \cos(q_i + q_j)
\end{aligned}$$

All other terms were calculated in a similar manner using equations (2.3) and (2.4) to obtain derivatives of the T transformation matrices.

$$\begin{aligned}
D_{12} &= m_2 a_2 \bar{z}_2 S_2 \\
&\quad + m_3 [ (d_3 \bar{x}_3 + a_3 \bar{y}_3 + a_3 d_3) S_{23} \\
&\quad + (a_2 \bar{y}_3 + a_2 d_3) S_2 - d_3 \bar{z}_3 C_{23} ] \quad (4.10)
\end{aligned}$$

$$D_{13} = m_3 [ (\bar{x}_3 d_3 + a_3 \bar{y}_3 + a_3 d_3) S_{23} - \bar{z}_3 d_3 C_{23} ] \quad (4.11)$$

$$D_{21} = D_{12} \quad (4.12)$$

$$\begin{aligned}
D_{22} &= m_2 (k_{2ZZ}^2 + a_2 + 2a_2 \bar{x}_2) \\
&\quad + m_3 [ (2a_2 a_3 + 2a_2 \bar{x}_3) C_3 + 2a_2 \bar{z}_3 S_3 + k_{3ZZ}^2 + a_2^2 + a_3 + 2a_3 \bar{x}_3 ] \\
&\quad \dots (4.13)
\end{aligned}$$

$$D_{23} = m_3 [ (a_2 \bar{x}_3 + a_2 a_3) C_3 + a_2 \bar{z}_3 S_3 + 2a_3 \bar{x}_3 + a_3^2 + k_{3ZZ}^2 ] \quad \dots (4.14)$$

$$D_{31} = D_{31} \quad (4.15)$$

$$D_{32} = D_{23} \quad (4.16)$$

$$D_{33} = m_3 (k_{3ZZ}^2 + a_3^2 + 2a_3 \bar{x}_3) \quad (4.17)$$

#### 4.3.2 Calculation of the Coriolis and Centripetal Terms for joint one

The elements in the centripetal and coriolis force matrix,  $C^1$ , affecting link one, are calculated using eqn 4.5 as detailed below.

$$C_{111} = 0 \quad (4.18)$$

$$\begin{aligned}
C_{112} = & m_2[(k_{2xx}^2 - k_{2yy}^2 - a_2^2 - 2a_2\bar{x}_2)C_2S_2 - a_2\bar{y}_2C_{22}] \\
& + m_3[k_{3xx}^2(C_2S_2 + S_2C_3 - 2S_2S_3S_{23} + 2S_2S_3S_{23} - C_2S_2 - C_3S_3) \\
& + \bar{x}_3(-2a_2C_2S_{23} + 4a_3S_2S_3S_{23} + a_2S_3 - 2a_3C_2S_2 - 2a_3C_3S_3) \\
& + \bar{z}_3(a_2C_2C_{23} - a_2S_2S_{23} + 2a_3C_{23}^2 - a_3) + a_2a_3S_{23} \\
& - 2a_2a_3C_2S_{23} - a_2^2C_2S_2 + 2a_3^2S_2S_3S_{23} \\
& - a_3^2(C_2S_2 + C_2S_3) ] \quad (4.19)
\end{aligned}$$

$$\begin{aligned}
C_{113} = & m_3[ k_{3xx}^2(C_2S_2 + C_3S_3 - 2S_2S_3S_{23}) \\
& + k_{3yy}^2(2S_2S_3S_{23} - C_2S_2 - S_3C_3) \\
& + \bar{x}_3(4a_3S_2S_3S_{23} - 2a_3C_2S_2 - 2a_3C_3S_3 - a_2C_2S_{23}) \\
& + \bar{z}_3(2a_3C_{23}^2 + a_2C_2C_{23} - a_3) + 2a_3^2S_2S_3S_{23} - a_2a_3C_2S_{23} \\
& - a_3^2C_2S_2 - a_3^2C_3S_3 ] \quad (4.20)
\end{aligned}$$

$$C_{121} = C_{112} \quad (4.21)$$

$$C_{112} = m_2a_2\bar{z}_2C_2 + m_3[d_3\bar{z}_3S_{23} + (d_3\bar{x}_3 + a_3\bar{y}_3 + a_3d_3)C_{23}] \quad \dots (4.22)$$

$$C_{123} = m_3[d_3\bar{z}_3S_{23} + (d_3\bar{x}_3 + a_3\bar{y}_3 + a_3d_3)C_{23}] \quad (4.23)$$

$$C_{131} = C_{113} \quad (4.24)$$

$$C_{132} = C_{123} \quad (4.25)$$

$$C_{133} = m_3[d_3\bar{z}_3S_{23} + (d_3\bar{x}_3 + a_3\bar{y}_3 + a_3d_3)C_{23}] \quad (4.26)$$

### 4.3.3 Calculation of Centripetal and Coriolis Forces for Joints Two and Three

The elements in the centripetal and coriolis force matrix,  $C^2$ , for link two, are calculated using eqn 4.5, and are detailed below.

$$C_{211} = -C_{112} \quad (4.27)$$

$$C_{212} = 0 \quad (4.28)$$

$$C_{213} = 0 \quad (4.29)$$

$$C_{211} = C_{212} \quad (4.30)$$

$$C_{222} = 0 \quad (4.31)$$

$$C_{223} = m_3 [(-a_2\bar{x}_3 - a_2a_3)S_2 + a_2\bar{z}_3C_3] \quad (4.32)$$

$$C_{231} = C_{213} = 0 \quad (4.33)$$

$$C_{232} = C_{223} \quad (4.34)$$

$$C_{233} = m_3 [(-a_2\bar{x}_3 - a_2a_3)S_3 + a_2\bar{z}_3C_3] \quad (4.35)$$

The elements in the centripetal and coriolis force matrix,  $C^3$ , for link three, are calculated using eqn (4.5) and are detailed below.

$$C_{311} = -C_{133} \quad (4.36)$$

$$C_{312} = -C_{213} \quad (4.37)$$

$$C_{313} = 0 \quad (4.38)$$

$$C_{321} = C_{312} \quad (4.39)$$

$$C_{322} = -C_{223} \quad (4.40)$$

$$C_{323} = 0 \quad (4.41)$$

$$C_{331} = C_{313} \quad (4.42)$$

$$C_{332} = C_{323} \quad (4.43)$$

$$C_{333} = 0 \quad (4.44)$$

#### 4.3.4 Calculation of Gravity Loading Terms

The gravity loading terms for the first three links, calculated using eqn.(3.3) are detailed below.

$$G_1 = 0 \quad (4.45)$$

$$G_2 = -m_2g[(\bar{x}_2 + a_2)C_2 - \bar{y}_3S_2] \\ - m_3g(\bar{x}_3C_{23} + \bar{z}_3S_{23} + a_3C_{23} + a_2C_2) \quad (4.46)$$

$$G_3 = -m_3g(\bar{x}_3C_{23} + \bar{z}_3S_{23} + a_3C_{23}) \quad (4.47)$$



It is apparent that the dynamic terms listed above are highly dependant upon measurements of the manipulator link parameters and variables eg angles. For example the self inertial term for joint one depends on joint angles  $q_2$ ,  $q_3$ , the moments of inertia about the various axes of links 2 and 3, the dimensions of links 2 and 3 and the positions of the centres of masses of links 2 and 3. It is therefore important to have accurate data on the robot's geometrical and mechanical parameter values. Since the manufacturers were unable to provide this information, the data is obtained by analytical and experimental means. The next section discusses these methods.

#### 4.4 Link Parameter Estimations

Disassembly of the Puma 560 robot arm into its individual links makes it possible to measure the link parameters needed to calculate the elements of the dynamic equations, namely link masses, centres of masses, and rotational inertias.

The effect of the motor armature inertias in the manipulator is relatively straightforward to calculate. The motor and drive mechanism at each joint contributes to the inertia about that joint an amount equal to the inertia of the rotating pieces magnified by the gear ratio squared. The reflected motor inertias for each of the joints were thus calculated as follows:- firstly the motor armature inertia was obtained from Unimation (the manufacturers of the Puma robots), the inertia of the gearing connected directly to the armature calculated, and finally this total inertia was scaled up by the square of the gearing ratio of the particular link. This modified motor inertia is referred to as the reflected motor inertia because of the manner in which it is increased by the gearing.

Link masses are easily found by weighing the separate links, however link mechanical parameters need to be estimated. Several approaches have been taken to estimate these parameters, both analytical and experimental. Paul [15] takes a rather simplistic approach by assuming that the link masses are evenly distributed over the surface of the link. We know that this is not actually the case, and so this represents one of the simplest approximations possible, and results in a very dubious set of estimates. Bejczy [16], obtains more accurate estimates by firstly taking detailed measurements of all the link internal components, calculating their individual moments of inertia, and later getting the cumulative effect using the parallel axis theorem, Goldstein [13].

Armstrong [17] obtains his estimates from a set of experimental results. The tests involve the suspension of the individual links from two wires as shown in Fig 4.1. A rotational inertia pendulum was created and a knowledge of

- 1) the link masses
- 2) the location of centre of gravity
- 3) the distance from wire attachment to axis of rotation
- 4) the length of wires and
- 5) the period of rotational oscillation

provides the data necessary to calculate the moments of inertia about the various axes, once some conditions are fulfilled. The link must be carefully made to swing about each axis so that a fundamental mode of oscillation is achieved, and all measurements are in SI units. The formula

$$I = \frac{M g r^2}{w^2 l}$$

is used, where

- I = inertia about the axis of rotation
- M = link mass
- g = force of gravity
- r = distance from each suspension wire to the axis of rotation
- w = oscillation frequency rad/sec
- l = length of supporting wire

Tables 4.1, 4.2, and 4.3 show the values for link parameters by Paul [15], Bejczy [16], and Armstrong [17]. Since the authors have used different reference frames the inertial terms were evaluated about different axes. The parallel axis theorem is used to transform the inertias so that they are related to the coordinate frame axes chosen in Chapter 3. There is a large difference between the estimates given by Paul and either of the other approaches. The two latter estimates differ by no more than 12%. This is quite a good agreement considering that the values were measured using two completely different approaches. Armstrong has included tolerances in his estimations. These vary from  $\pm 1\%$  for link mass measurements to  $\pm 50\%$  for some of the inertia terms. This emphasizes the need to validate parameter estimates.

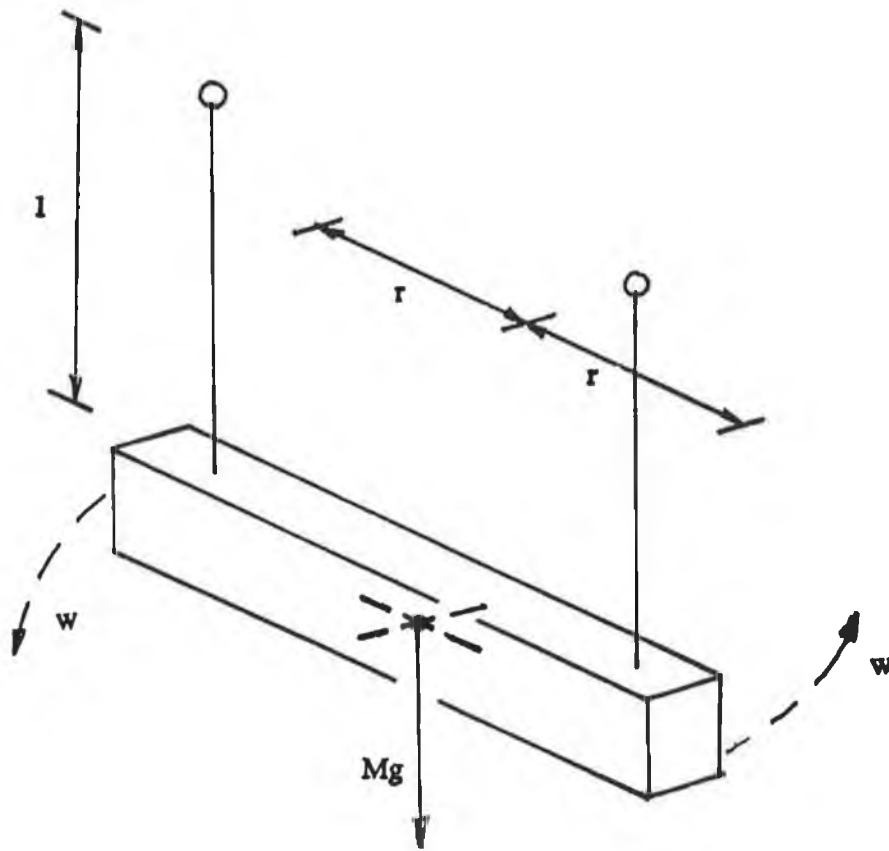


Figure 4.1 Rotational pendulum

TABLE 4.1  
Puma 500 link Mechanical Parameters according to Paul

link	mass	centre of mass			radius of gyration			reflected motor inertia
	$m_i$ (kg)	$x_i$ (m)	$y_i$ (m)	$z_i$ (m)	$k_{i_{xx}}^2$ (m <sup>2</sup> )	$k_{i_{yy}}^2$ (m <sup>2</sup> )	$k_{i_{zz}}^2$ (m <sup>2</sup> )	$Ia_i$ (kg-m <sup>2</sup> )
1	12.89	0.00	0.3278	0.0400	0.1886	0.3660	0.0242	0.7646
2	22.74	-0.3219	0.0059	0.2100	0.2436	0.3900	0.5074	2.4396
3	5.019	0.0234	0.0214	0.0132	0.4581	0.0435	0.0661	0.5836
3*	6.988	0.0187	0.0272	0.1502	0.2083	0.0936	0.0541	0.5663
4	1.27	0.000	0.0803	-0.0048	0.4049	0.0457	0.0092	0.1342
5	0.67	0.000	-0.0202	0.0025	0.0008	0.0377	0.0349	0.0693
6+	0.18	0.000	0.000	0.0042	0.0008	0.0126	0.0004	0.10439

\* Considering the last three links as a load added to link three.

+ Without end effector mechanism.

TABLE 4.2  
Puma 500 link Mechanical Parameters according to Bejczy

link	mass	centre of mass			radius of gyration			reflected motor inertia
	$m_i$ (kg)	$x_i$ (m)	$y_i$ (m)	$z_i$ (m)	$k_{i_{xx}}^2$ (m <sup>2</sup> )	$k_{i_{yy}}^2$ (m <sup>2</sup> )	$k_{i_{zz}}^2$ (m <sup>2</sup> )	$Ia_i$ (kg-m <sup>2</sup> )
1	12.96	0.00	0.3088	0.0389	0.1816	0.0152	0.1811	0.7766
2	22.37	-0.3289	0.005	0.2038	0.0596	0.1930	0.1514	2.3616
3	5.01	0.0204	0.0137	0.0037	0.0151	0.0155	0.0021	0.5827
3*	6.97	0.0136	0.0092	0.1522	0.0783	0.0786	0.0021	0.5827
4	1.18	0.000	0.0863	-0.0029	0.0119	0.0029	0.0118	0.1060
5	0.62	0.000	-0.0102	0.0013	0.0009	0.0009	0.0009	0.0949
6+	0.16	0.000	0.000	0.0029	0.0008	0.0008	0.0004	0.1078

\* Considering the last three links as a load added to link three.

+ Without end effector mechanism.

TABLE 4.3  
Puma 500 link Mechanical Parameters according to Armstrong

link	mass	centre of mass			radius of gyration			reflected motor inertia
	$m_i$ (kg)	$x_i$ (m)	$y_i$ (m)	$z_i$ (m)	$k_{ixx}^2$ (m <sup>2</sup> )	$k_{iyy}^2$ (m <sup>2</sup> )	$k_{izz}^2$ (m <sup>2</sup> )	$Ia_i$ (kg-m <sup>2</sup> )
1	12.89	0.00	0.3138	0.0390	0.1936	0.0160	0.1771	0.7856
2	22.74	-0.3389	0.0049	0.2148	0.0636	0.1870	0.1654	2.3726
3	5.019	0.0224	0.0144	0.0127	0.0221	0.0175	0.0021	0.5837
3*	6.988	0.0147	0.0172	0.1462	0.0883	0.0786	0.0021	0.5977
4	1.27	0.000	0.0793	-0.0028	0.0129	0.0119	0.0180	0.1190
5	0.67	0.000	-0.0172	0.0029	0.0008	0.0008	0.0009	0.0957
6+	0.18	0.000	0.000	0.0032	0.0008	0.0007	0.0004	0.1055

\* Considering the last three links as a load added to link three.

+ Without end effector mechanism.

Theoretically both of the latter methods could provide exact results if experimental errors are eliminated. It is difficult to establish how critical these values are to the model responses before simulation tests have been run. In order to obtain initial link parameter estimates for substitution into the above parameter formulae the average of Armstrongs and Bejczy estimates were used. Table 4.4 presents these values. Numerical estimates of the dynamic coefficient terms using these values are listed below.

#### 4.5 Numerical Evaluation of the Coefficient terms

Substitution of the numerical values for the parameters into the above equations and simplifying gives the dynamic equation coefficient terms in the following numerical form. Note that the reflected motor inertias have been incorporated into the effective inertial terms.

##### 4.5.1 Numerical evaluation of effective and coupling inertia terms

$$D_{11} = 0.7766 + 1.7209 + 2.1007C_2^2 + 0.5323S_2^2 - 0.0330C_2C_{23} - 0.0405C_{23}S_{23} + 0.9161C_2S_{23}$$

$$D_{22} = 2.3616 + 3.0574 + 0.9161S_3 - 0.0331C_3$$

$$D_{33} = 0.5827 + 0.5468$$

Where the first term of  $D_{ii}$  is the reflected motor inertia of the  $i^{\text{th}}$  joint.

$$D_{12} = 2.4492S_2 - 0.007S_{23} - 0.1596C_{23}$$

$$D_{13} = -0.007S_{23} - 0.1596C_{23}$$

$$D_{23} = 0.5468 + 0.4581S_3 - 0.0165C_3$$

##### 4.5.2 Numerical Evaluation of Centripetal & Coriolis Terms for Link One

$$C_{123} = 0.0203 - 1.5685C_2S_2 + 0.5322C_3S_3 - 1.0643S_2S_3S_{23}$$

$$+ 0.0331C_2S_{23} + 0.04581C_2C_{23} - 0.4581S_2S_{23} - 0.0483C_{22} \\ - 0.0405C_{23}^2 - 0.0166S_3$$

$$C_{113} = 0.0203 + 0.5322C_2S_2 + 0.5322C_3S_3 \\ - 1.0643S_2S_3S_{23} + 0.0166C_2S_{23} + 0.4581C_2C_{23} - 0.0405C_{23}^2 \\ - 0.0166S_3$$

$$C_{122} = 1.9686C_2 + 0.1596S_{23} - 0.007C_{23}$$

$$C_{123} = 0.1596S_{23} - 0.007C_{23}$$

$$C_{133} = 0.1596S_{23} - 0.007C_{23}$$

#### 4.5.3 Numerical Evaluation of Centripetal & Coriolis Terms for Link Two

$$C_{213} = 0$$

$$C_{223} = 0.4581C_3 + 0.0165S_3$$

$$C_{233} = C_{223}$$

#### 4.5.4 Numerical Evaluation of Gravity Terms for the First Three Links

$$G_1 = 0$$

$$G_2 = - 52.1060C_2 + 1.0972S_2 + 0.3761C_{23} - 10.4068S_{23}$$

$$G_3 = 0.3761C_{23} - 10.4068S_{23}$$

where the unit of  $D_{ij}$ ,  $D_{ij}$ ,  $C_{ijk}$  is  $\text{kg} \cdot \text{m}^1$  and the unit of  $D_i$  is  $\text{kg} \cdot \text{m}^1 \cdot \text{s}^{-1}$

The set of coefficients given above represent the nominal values used in the simulations.



#### 4.6 Summary

This chapter has outlined how each of the second order dynamic equations coefficient term are symbolically calculated using equations (3.28), (3.23), and (3.24). The derivatives in these equations were obtained using the Q matrix derived in Section 2.5.2. The complete symbolic expressions for all coefficient terms were presented. For ease of simulation the dynamic equations were written in matrix form. It is convenient to store similar groups of terms in arrays when simulating the equations. Since the parameters comprising these expressions were unavailable from the manipulator manufacturers, their estimated values (presented by Paul, Armstrong and Bejczy) are examined and nominal values are selected. Numerical expressions for the coefficient terms were presented for use in a nominal second order dynamic model.

The formulation of dynamics described above has assumed, (1) that the links are completely rigid, (2) that the servomotors do not provide any gyroscopic effects, (3) that there is no backlash in any of the gearing, and that friction is not present, but most importantly (4) that the actuator dynamics have been omitted. The first assumption is accurate as a direct result of the manipulator construction. The motor armatures rotate at relatively low speeds and have low masses, thus gyroscopic effects are negligible. Backlash can be removed by careful adjustment of the joint gearing. Friction is a significant force and as such it is incorporated into the model in the next chapter. Actuator dynamics are particularly important in any manipulator. The dynamics of the permanent magnet d.c. servomotors cannot be ignored and are incorporated in the next chapter. It is therefore apparent that the dynamic model presented so far is not complete.

## CHAPTER 5

### THIRD ORDER DYNAMIC EQUATIONS FOR THE PUMA 560

This chapter extends the second order dynamic model, presented in Chapter 4, to incorporate the actuator dynamics and friction forces. Often the friction forces present in manipulators are quite large. For large manipulators friction forces can comprise up to 23% of torque required, Craig [7]. Friction in the Puma 560 is modelled as Coulomb friction. Permanent magnet d.c. servomotors are used to actuate the Puma 560 joints.

The motor dynamics, for the first three links, are incorporated into the manipulator system equations, using knowledge of the gearing ratios at each joint, and the equivalent circuit model of the motors. Modelling the motors as first order systems results in a third order set of differential equations describing the complete Puma 560 dynamics.

The resulting third order equations have voltage, rather than torque, inputs. Ideally the equations represent the exact model of the manipulator dynamics. In reality, no set of equations can ever exactly specify the dynamics of a plant, however the third order set of equations represents the main dynamics of the manipulator. In the case of the Puma 560 manipulator the consequences of unmodelled link elasticity, for example, are negligible compared with effect of possible inaccuracies introduced by nonexact measurement of link mechanical parameters. We assume the influence of the unmodelled dynamics is small, and that in the control situation, their effects can be modeled as noise disturbances, and adequately handled by the controller.

Once the third order model has been derived it is written in a matrix format for clarity and ease of simulation. The matrix and vector elements in the resulting "state" equations, change throughout time.

#### 5.1 Servomotor Equations

The Puma 560 robot arm is driven by electric permanent magnet direct current servomotors. There are two sizes of motor employed in the robot, 100 Watt motors for the first three links, and 50 Watt motors to drive the three wrist

links. Since we are concerned with the dynamics of the first three joints, which affect the end effector path tracking problem, only the larger motors will be considered here. Figure 5.1 shows a simple equivalent circuit model for a permanent magnet D.C. motor.

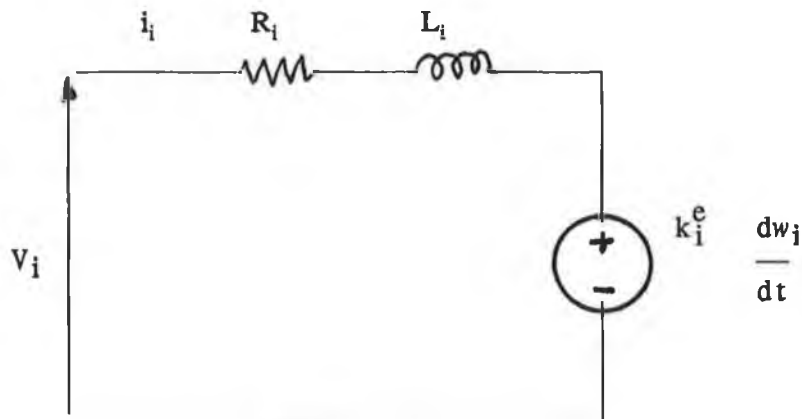


Figure 5.1

where

$w_i$  = motor position

$R_i$  = armature resistance

$L_i$  = armature inductance

$i_i$  = armature current

$k_i^e$  = voltage constant

$k_i^t$  = torque constant

$v_i$  = armature voltage

the  $i$  subscript denoting the motor number.

The voltage equation around the armature loop is given by

$$V_i = R_i i_i + L_i \frac{di_i}{dt} + k_i^e \frac{dw_i}{dt} \quad (5.1)$$

Torque produced by the motor is proportional to armature current

$$F_i = k_i^t i_i \quad (5.2)$$

where  $F_i$  is the torque experienced at joint  $i$ .

Specifications of the 100 Watt servomotors were obtained directly from the manufacturers and are known to be.

$$R = 1.6 \quad \text{Ohms}$$

$$L = 0.0048 \quad \text{Henries}$$

$$k_f = 0.26 \quad \text{volt/rad/sec}$$

$$k_t = 0.2611 \quad \text{kg.m/amp}$$

Each link is driven by a motor through a set of gears. Figure 5.2 shows the gearing systems for the first three robot joints. If we let  $q_i$  denote the  $i^{\text{th}}$  link position, then the joint position is given by  $w_i$  where

$$w_i = N_i q_i \quad (5.3)$$

where  $N_i$  is the gear ratio of the  $i^{\text{th}}$  link. For the Puma 560 the gear ratios have been calculated as

$$N_1 = 62.55$$

$$N_2 = 107.81$$

$$N_3 = 53.63$$

## 5.2 Third Order Dynamic Equations

From Equation (4.1) the dynamic behaviour of the first three links of the Puma 560 robot are governed by a set of second order differential equations as:

$$F_i = \sum_{j=1}^3 D_{ij} \ddot{q}_j + I a_i \ddot{q}_i + \sum_{j=1}^3 \sum_{k=1}^3 D_{ijk} \dot{q}_j \dot{q}_k + D_i \quad i = 1..3$$

$q_i, \dot{q}_i, \ddot{q}_i$  = system variables, joint  $i$  position, velocity, and acceleration

$F_i$  = torque acting on joint  $i$ .

$D_{ii}, D_{ij}$  = effective and coupling inertias for joint  $i$

$I a_i$  = actuator inertia of joint  $i$

$C_{ijj}, C_{ijk}$  = centripetal and coriolis forces for joint  $i$

$G_i$  = gravity loading for joint  $i$

The friction experienced at each of the joints is a function of the link position and velocity. In this model the friction is modelled as coulomb friction, and is easily introduced into the second order model. The friction experienced at joint  $i$  will be denoted by

friction at joint  $i = H_i \dot{q}_i$

The second order equation is now written as

$$\dot{F}_i = \sum_{j=1}^3 D_{ij} \dot{q}_j + I a_i \ddot{q}_i + \sum_{j=1}^3 \sum_{k=1}^3 D_{ijk} \dot{q}_j \dot{q}_k + H_i \dot{q}_i + D_i \quad (5.4)$$

Eqn (5.4) describes torques at various links in terms of joint positions, velocities and accelerations. In reality, torque is not applied directly to the joints, but rather voltage is applied to the actuator motors. For the purposes of simulation and control we require system equations with voltage rather than torque inputs. This is achieved by combining equations (5.1) and (5.4).

Firstly differentiate eqn (5.4), giving

$$\begin{aligned} \dot{F}_i = & \sum_{j=1}^3 (D_{ij} \ddot{q}_j + \dot{D}_{ij} \dot{q}_j) + I a_i \ddot{q}_i \\ & + \sum_{j=1}^3 \sum_{k=1}^3 (D_{ijk} \ddot{q}_j \dot{q}_k + D_{ijk} \dot{q}_j \ddot{q}_k + \dot{D}_{ijk} \dot{q}_j \dot{q}_k) + H_i \ddot{q}_i + \dot{D}_i \end{aligned} \quad (5.5)$$

Substituting eqn (5.5) and eqn (5.4) into eqn (5.1) gives

$$\begin{aligned} V_i = & \frac{R_i}{k_i^T} \left[ \sum_{j=1}^3 D_{ij} \ddot{q}_j + I a_i \ddot{q}_i + \sum_{j=1}^3 \sum_{k=1}^3 D_{ijk} \dot{q}_j \dot{q}_k + D_i + H_i \dot{q}_i \right] \\ & + \frac{L_i}{k_i^T} \left[ \sum_{j=1}^3 (D_{ij} \ddot{q}_j + \dot{D}_{ij} \dot{q}_j) + I a_i \ddot{q}_i \right] \\ & + \frac{L_i}{k_i^T} \left[ \sum_{j=1}^3 \sum_{k=1}^3 (D_{ijk} \dot{q}_j \dot{q}_k + D_{ijk} \dot{q}_j \ddot{q}_k + \dot{D}_{ijk} \dot{q}_j \dot{q}_k) + \dot{D}_i + H_i \dot{q}_i \right] \\ & + k_i^e N_i \dot{q}_i \end{aligned} \quad (5.6)$$

Rearranging the terms gives

$$V_i = \frac{L_i}{k_i^T} \left[ \sum_{j=1}^3 D_{ij} \ddot{q}_j + I a_i \ddot{q}_i + 2 \sum_{j=1}^3 \sum_{k=1}^3 D_{ijk} \dot{q}_j \dot{q}_k + \sum_{j=1}^3 \sum_{k=1}^3 D_{ijk} \dot{q}_j \ddot{q}_k \right]$$

... cont

$$\begin{aligned}
& + \frac{R_i}{k_i^T} \left[ \sum_{j=1}^3 D_{ij} \ddot{q}_j + I a_i \dot{q}_i + H_i \dot{q}_i \right] + \frac{L_i}{k_i^T} \sum_{j=1}^3 \dot{D}_{ij} \dot{q}_j \\
& + \frac{L_i}{k_i^T} \sum_{j=1}^3 \sum_{k=1}^3 \dot{D}_{ijk} \dot{q}_j \dot{q}_k \\
& + \frac{R_i}{k_i^T} \sum_{j=1}^3 \sum_{k=1}^3 D_{ijk} \dot{q}_j \dot{q}_k \\
& + k_i^e N_i q_i \\
& + \frac{R_i}{k_i^T} H_i \dot{q}_i \\
& + \frac{L_i}{k_i^T} \dot{D}_i \\
& + \frac{R_i}{k_i^T} D_i
\end{aligned} \tag{5.7}$$

Equation (5.7) gives explicit third order differential equations for the first three robot links. The expression contains derivatives of the coefficient terms of equation (5.4). Although the third order expressions are more complex than the second, it is still possible to write them in a state space type of format.

### 5.3 Matrix Representation of Third Order Dynamic Equations

When simulating the third order equations it is convenient to store similar coefficient terms and states in arrays. The procedure taken is similar to that employed when writing the second order equations in matrix form. New states are introduced to cater for the third order derivatives. In order to write equation (5.6) in matrix form, the following plant states, matrices and vectors are defined:

$$\begin{array}{lll}
x_1 = q_1 & x_2 = \dot{q}_1 & x_3 = \ddot{q}_1 \\
x_4 = q_2 & x_5 = \dot{q}_2 & x_6 = \ddot{q}_2 \\
x_7 = q_3 & x_8 = \dot{q}_3 & x_9 = \ddot{q}_3
\end{array}$$

Equation (5.7) may be written in matrix form as:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u} \quad (5.8)$$

where  $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9) \in \mathbb{R}^9$

$\mathbf{f}(\mathbf{x})$  is a vector whose elements are dependant on the vector  $\mathbf{x}$  and the manipulator parameters,  $\mathbf{f}(\mathbf{x}) \in \mathbb{R}^9$

and  $\mathbf{g}(\mathbf{x})$  is a matrix whose elements are dependant upon the vector  $\mathbf{x}$  and the manipulator parameters,  $\mathbf{g}(\mathbf{x}) \in \mathbb{R}^{3 \times 9}$ . Eqn (5.8) may be written in elemental form as :-

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dots \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dots \\ \dot{x}_7 \\ \dot{x}_8 \\ \dot{x}_9 \end{bmatrix} = \begin{bmatrix} x_4 \\ x_5 \\ x_6 \\ \dots \\ x_7 \\ x_8 \\ x_9 \\ \dots \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \dots \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \dots \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} \quad (5.9)$$

$\bar{D}^{-1} \bar{P}(\mathbf{x})$ 
 $\bar{D}^{-1}$

where

$$\bar{D} = \begin{bmatrix} \frac{L_1}{k_1^l} & 0 & 0 \\ 0 & \frac{L_2}{k_2^l} & 0 \\ 0 & 0 & \frac{L_3}{k_3^l} \end{bmatrix} \quad \bar{D} + \begin{bmatrix} I a_1 & 0 & 0 \\ 0 & I a_2 & 0 \\ 0 & 0 & I a_3 \end{bmatrix} \quad (5.10)$$

$$\begin{aligned}
 \underline{P}(x) = & \begin{bmatrix} 2L_1 & (x_4 x_5 x_6) D^1 & \frac{k_1}{L_1} \\ 2L_2 & (x_4 x_5 x_6) D^2 & \frac{k_2}{L_2} \\ 2L_3 & (x_4 x_5 x_6) D^3 & \frac{k_3}{L_3} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & R_2 & 0 \\ 0 & R_3 & 0 \end{bmatrix} + D + \begin{bmatrix} \frac{k_1}{R_1 I a_1} + H_1 & 0 & 0 \\ 0 & \frac{k_2}{R_2 I a_2} + H_1 & 0 \\ 0 & 0 & \frac{k_3}{R_3 I a_3} + H_1 \end{bmatrix} \\
 & \begin{bmatrix} \frac{k_1}{L_1} x_4 x_5 x_6 (L_1 D^1 + R_1 D^1) \\ \frac{k_2}{L_2} x_4 x_5 x_6 (L_2 D^2 + R_2 D^2) \\ \frac{k_3}{L_3} x_4 x_5 x_6 (L_3 D^3 + R_3 D^3) \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & k_2 N^z + H_1 & 0 \\ 0 & 0 & k_3 N^z + H_1 \end{bmatrix} e \\
 & \begin{bmatrix} x_4 \\ x_5 \\ x_6 \end{bmatrix}
 \end{aligned}
 \tag{5.11}$$



The matrices in the representation given above contain elements that are trigometrical functions of the manipulator joint angles and velocities. In order to assess the validity of the above model, it is necessary to simulate it in software.

#### 5.4 Summary

This chapter has shown how actuator dynamics for the Puma 560 manipulator are incorporated into the basic second order model of a robot. This was achieved as follows:

- i) A first order approximation of the DC servomotors was presented,
- ii) the gearing ratios which couple the motor torque to the joint torque were given,
- iii) the relationship between motor current and torque was presented,
- iv) the derivative of the second order manipulator dynamic equation was found,
- v) the second order dynamic equation and its derivative were substituted into the motor equation, and
- vi) similar terms were grouped together for clarity

In the case of the Puma 560 a first order approximation of the permanent magnet D.C. motor drive dynamics was chosen, resulting in a set of third order differential equations for the manipulator. The third order set of equations have voltage inputs and joint angle outputs. Since the Puma 560 control signals drive the joint motors (via the power amplifiers) the third order model is more suitable for controller appraisal.

The third order equations were written in matrix form, similar to that used in the second order case, with vector and matrix elements as functions of the plant states. Backlash in the gearing, and elasticity have been ignored, assuming their contribution to be negligible. Many of the link mechanical parameters in the model are not exactly determinable, (eg. link inertias), emphasising the need for model validation.

## CHAPTER 6

### PUMA 560 ROBOT ARM SIMULATOR

In order to assess the model's accuracy and to facilitate controller appraisal, the model was simulated in software. The simulation uses a classical fourth order Runge-Kutta technique for numerical integration. Justification for this choice is given in Section 6.3. The main objective of this chapter is to show that the simulation is designed for both accuracy and flexibility. Accuracy with regard to round off and truncation errors, and flexibility for controller appraisal.

In this chapter the simulation facilities are listed, program structure is outlined with the help of a flowchart, the criterion for choice of numerical integration is discussed. The matrix inversion routine used when evaluating the dynamic equations is also studied.

#### 6.1 Simulator Facilities

Care was taken in the design of the simulator to ensure that sufficient facilities are provided to allow full model testing and ease of interfacing to controller programs. To this end the simulator allows detailed examination of the effect that varying particular parameters have on the model dynamics. This proves valuable when tuning the simulation to fit the actual robot dynamics. The model simulation provides facilities that make it suitable for easy interfacing to controller procedures. Simulator inputs are

- (1) Joint constraints,
- (2) Initial positions velocities and accelerations,
- (3) Motor voltages as a function of time, and the output is
- (4) Joint positions, velocities and accelerations sampled synchronously.

The motor inputs are treated as piecewise constant voltages over the sampling periods. A data file containing voltage versus time is read by the simulator at each sample interval. The simulation program was designed so that movement in each of the joints can be determined by one of the following:

- (a) joint i is free to move subject to the manipulator dynamics and a user specified input voltage to joint i,
- (b) joint i maintains constant angular position,
- (c) joint i maintains constant angular velocity,
- (d) joint i maintains constant angular acceleration.

The significance of (a) is that the joint movements are unconstrained and determined completely by the dynamic model, while (b), (c) and (d) correspond to holding selected states constant in the dynamic model. Although the implementation of these facilities increases the complexity of the simulator considerably, they allow the user to focus on specific dynamic properties of the manipulator model. For example, coriolis torques generated at joint 1 due to the relative velocities at joints 1 and 2 may be examined by setting velocities  $\dot{x}_1$  and  $\dot{x}_2$  to be constant. By changing the combinations, and relative magnitudes of the joint velocities and accelerations, all dynamics can be analysed. For a three degree of freedom manipulator with four constraints per joint there are  $4^3$  or 64 possible constraint combinations. Limits are set on the reachable positions of each of the three lower joints as shown in Fig 6.1. Maximum velocities and accelerations are also set. The values of the joint states are specified at run time.

## 6.2 Numerical Integration Methods

Any numerical integration technique involves predicting the system states at the iteration step k, given the states at step k-1 and the present inputs to the system. This section discusses why the Runge-Kutta integration was chosen in preference to the other available methods.

From the eqn (5.8) third order equations are given in state space format as:

$$\dot{x} = f(x) + g(x)v$$

This equation may be equivalently rewritten as

$$\dot{x} = F(x,v,t) \tag{6.1}$$

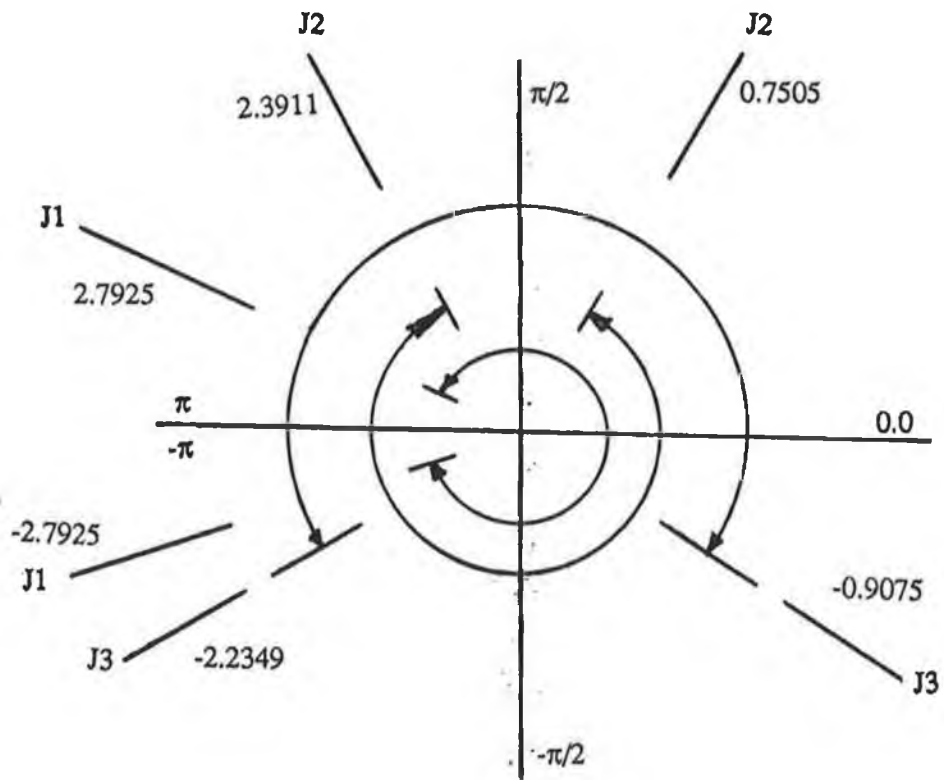


Figure 6.1 Ranges of joints 1, 2 & 3

indicating that the state derivatives are functions of the (i) present states, (ii) the input voltage and (iii) time. Given the initial state values and the input voltage profile, it is possible to compute numerical values of the solution. Numerical solutions are necessary when closed form formulae for the states are unavailable. Closed form formulae would be difficult if not impossible to derive for the Puma 560 third order dynamic equations. A numerical integration method provides a suitable means of obtaining a solution to the equations.

The principle of all numerical integration methods is to estimate the system states at time (t+h) given the states at time (t), and the system equations, where

t = present time  
t+h = present time plus time increment

In each step, computations are done by some formula normally based upon the Taylor series:

$$x(t+h) = x(t) + h \dot{x}(t) + \frac{h^2}{2!} \ddot{x}(t) + \dots \quad (6.2)$$

If h is chosen to be sufficiently small and if sufficient derivative powers of x are taken then the value of x(t+h) can be perfectly found. The simplest method, called the Euler method, only takes the first power of h into account assuming  $h^n$  ( $n > 1$ ) are zero. This is valid only if  $h \ll 1$ . The Euler formula has the form

$$x(t+h) = x(t) + h F(x,v,t) \quad (6.3)$$

The Euler method is of limited practical use due to its large truncation error, per step, of order  $h^2$ . This is exaggerated when a large step length h is used. If it was possible to use a very small step h and if round off errors did not occur in the calculations then the Euler method would satisfy most requirements.

The Euler method was initially used to provide numerical solutions for the Puma 560 third order model. It was found however that the solutions produced were highly dependant on the value of step increment. Even when small values of h were used, the solutions remained dependant upon the sample period. This was

due to both truncation and round off errors.

Truncation errors in Runge-Kutta numerical methods are of the order of  $h^{n+1}$  giving a closer approximation of the system functions. Thus numerical methods which include higher order derivatives of  $F$  have improved accuracy. One of the most widely used numerical integration methods in engineering applications is the fourth order Runge Kutta method.

Since fourth order numerical integration methods incorporate the first four derivatives of the Taylor series the truncation error is of the order of  $h^5$ , which provides increased accuracy and makes larger incremental steps possible. The fourth order classical Runge-Kutta method was selected mainly because of the accuracy and ease of implementation. Runge Kutta algorithms have the following desirable properties

- (1) the integration is self starting,
- (2) the step size can be easily changed between iterations,
- (3) they do not require derivative evaluation,
- (4) they has good stability characteristics,
- (5) they are applicable to nonlinear systems.

### 6.2.1 Implementation of 4<sup>th</sup> Order Runge-Kutta Integration

Popular derivations of the Runge Kutta method do not explicitly include an input to the system. Analysis, however, shows that if the input remains constant between iterations, the Runge-Kutta integration caters for a system described by equation (6.1). Appendix B examines this area.

For an  $n^{\text{th}}$  order set of equations (with inputs) written as:

$$\dot{x}_i = f_i(x_1, x_2 \dots x_n, v_i, t) \quad 1 \leq i \leq n \quad (6.4)$$

the formula for advancing the solution one step is

$$x_{i,r+1} = x_{i,r} + 1/6 (K_{i,1} + 2K_{i,2} + 2K_{i,3} + K_{i,4}) \quad 1 \leq i \leq n \quad (6.5)$$

where

$$x_{i,r+1} = x_i(t_{r+1}) = x_i(t_0 + (r+1)h) \quad (6.6)$$

$$K_{i,1} = hf_i(x_{1,r}, x_{2,r} \dots x_{n,r}, v_{i,r}, t_r) \quad (6.7)$$

$$K_{i,2} = hf_i(x_{1,r}+0.5K_{1,1}, \dots, x_{n,r}+0.5K_{n,1}, v_{i,r+0.5}, t_{r+0.5}) \dots (6.8)$$

$$K_{i,3} = hf_i(x_{1,r}+0.5K_{1,2}, \dots, x_{n,r}+0.5K_{n,2}, v_{i,r+0.5}, t_{r+0.5}) \dots (6.9)$$

$$K_{i,4} = hf_i(x_{1,r}+0.5K_{1,3}, \dots, x_{n,r}+0.5K_{n,3}, v_{i,r+1}, t_{r+1}) \dots (6.10)$$

When performing the computation the quantities  $K_{i,1}$  ( $1 \leq i \leq n$ ) must first be found then  $K_{i,2}$ ,  $K_{i,3}$  and lastly  $K_{i,4}$ . The scheme requires the evaluation of the function  $f_i(x,v,t)$  four times for each state per integration step. In cases where the functions  $f_i$  are complicated the fourth order Runge-Kutta method involves a considerable amount of computation. When simulating the Puma 560 model on the microVax minicomputer using single precision arithmetic with  $h=0.05$ , a ten second simulation takes approximately 9 minutes cpu time. Since the simulator accuracy, rather than speed was the main objective, the computational burden was deemed acceptable. Values of  $h$  between 50mS and 40mS did not improve the simulation solution but increased the execution time. When  $h$  was chosen to be any larger than 0.035 mS the simulation output became inaccurate when the joint angle velocities were high.

In the case of the Puma 560 simulation the mode of joint movement, selected as (a), (b), (c) or (d) in Section 6.1, influence the integration procedure. In each iteration step the Runge-Kutta algorithm firstly integrates the state variable differential equations subject to no joint constraints. Once calculated, the procedure checks whether the joint is constrained, and if so, the constraint consequences override the Runge Kutta values. Joint constraints restrict the values of the elements of the derivative of the state space vector. At present, the simulation does not take advantage of the model simplifications that exist due to joint constraints. Each joint constraint reduces equation (6.3) by one order. The present simulation program integrates the full unconstrained dynamic response at each step and then corrects the state values. Thus simulation speed remains unchanged regardless of joint constraints.

### 6.3 Matrix Inversion Routine

Solving equation (5.9) involves inversion of the  $\bar{D}$  matrix. Two areas of concern exist: (a) the possibility of  $\bar{D}$  becoming singular and (b) the routine must be designed so that round off errors are minimised. An expression for  $\bar{D}$  may be written from (5.10) as:

$$\bar{D} = \begin{bmatrix} \frac{L_1}{k_1^t} & 0 & 0 \\ 0 & \frac{L_2}{k_2^t} & 0 \\ 0 & 0 & \frac{L_3}{k_3^t} \end{bmatrix} \left[ D + \begin{bmatrix} I a_1 & 0 & 0 \\ 0 & I a_2 & 0 \\ 0 & 0 & I a_3 \end{bmatrix} \right] \in \mathbb{R}^{3 \times 3}$$

Using the mnemonics used in the simulation program the expression for  $\bar{D}$  may be written as

$$\bar{D} = \text{LKMAT} [D + \text{IMAT}]$$

LKMAT is a diagonal matrix,  $\in \mathbb{R}^{3 \times 3}$ , whose elements are quotients of the motor equivalent circuit parameters ie. the armature inductance divided by the torque constant for the motor. The first three joints in the Puma 560 are driven by identical motors, having identical equivalent circuit parameters. Thus, in the case of the Puma 560, LKMAT is a scaled version of the identity matrix. IMAT,  $\in \mathbb{R}^{3 \times 3}$ , is simply added to  $\bar{D}$ . IMAT contains the reflected motor inertias. These values are positive and comparatively large. The main source of any problems will stem from the behaviour of the  $\bar{D}$  matrix containing the link inertial terms. Certain properties of the matrix are known [18].

$$D(q) = [d_{ij}]$$

is symmetric:  $d_{ij} = d_{ji}$  for all  $i$  and  $j$

and

Positive definite:  $d_{ii} > 0$  for all  $i$   
 $d_{ii}^2 < d_{ii} d_{jj}$  for all  $i, j$

Since  $D$  is positive definite,  $D + \text{IMAT}$  is also positive definite. LKMAT scales



the sum of D and IMAT by a positive number thus  $\bar{D}$  is positive definite.  $\bar{D}$  is therefore always nonsingular.

The other consideration was to ensure that round off errors are kept as small as possible. In Gauss Jordan Elimination round off errors occur when there are large magnitude differences in the elements of matrix columns. Partial pivoting reduces roundoff errors by taking the largest element in the matrix column as the pivotal coefficient. Gauss Jordan elimination with partial pivoting was found to give sufficiently accurate matrix inverses when tested on matrices with element magnitude differences of 10,000 when using double precision arithmetic on a microVax minicomputer. Using single precision reduced accuracy by about 15%. Double precision has been used in the inversion routine in the simulator.

#### 6.4 Simulation Program Structure

Figure 6.2 shows the flowchart for this program. A brief summary of the simulation programs operation is now presented.

Equations (5.8) and (5.9) give the third order dynamics in a matrix form. The matrices and state vectors in equation (5.9) have been partitioned. States  $x_1$  to  $x_6$  are linearly related to states  $x_4$  to  $x_9$ . However the high order states, representing the third derivatives of joint positions are highly nonlinear and coupled as indicated in equations (5.10) and (5.11). The derivative of the state vector ( $\in \mathbb{R}^9$ ) in eqn (5.9) is redefined as three subvectors ( $\in \mathbb{R}^3$ ) for use in the simulation. This is more convenient when evaluating the dynamic equation using arrays. The matrix D was calculated in software simply as the product and sum of three readily available matrices- namely the inertial matrix D and two constant diagonal matrices. The expression for D is given in eqn (5.10). Calculation of the  $\bar{P}(x)$  vector is more involved and was achieved in the simulation program by rewriting equation (5.11) as shown in equation (6.2).

$$\begin{aligned} \bar{P}(x) = & - (FMAT + BMAT + GMAT + HMAT) x \\ & - (JKMAT + KMAT) \dot{x} \\ & + LMAT \end{aligned} \tag{6.2}$$

These mnemonics are consistent with those used in the simulation program

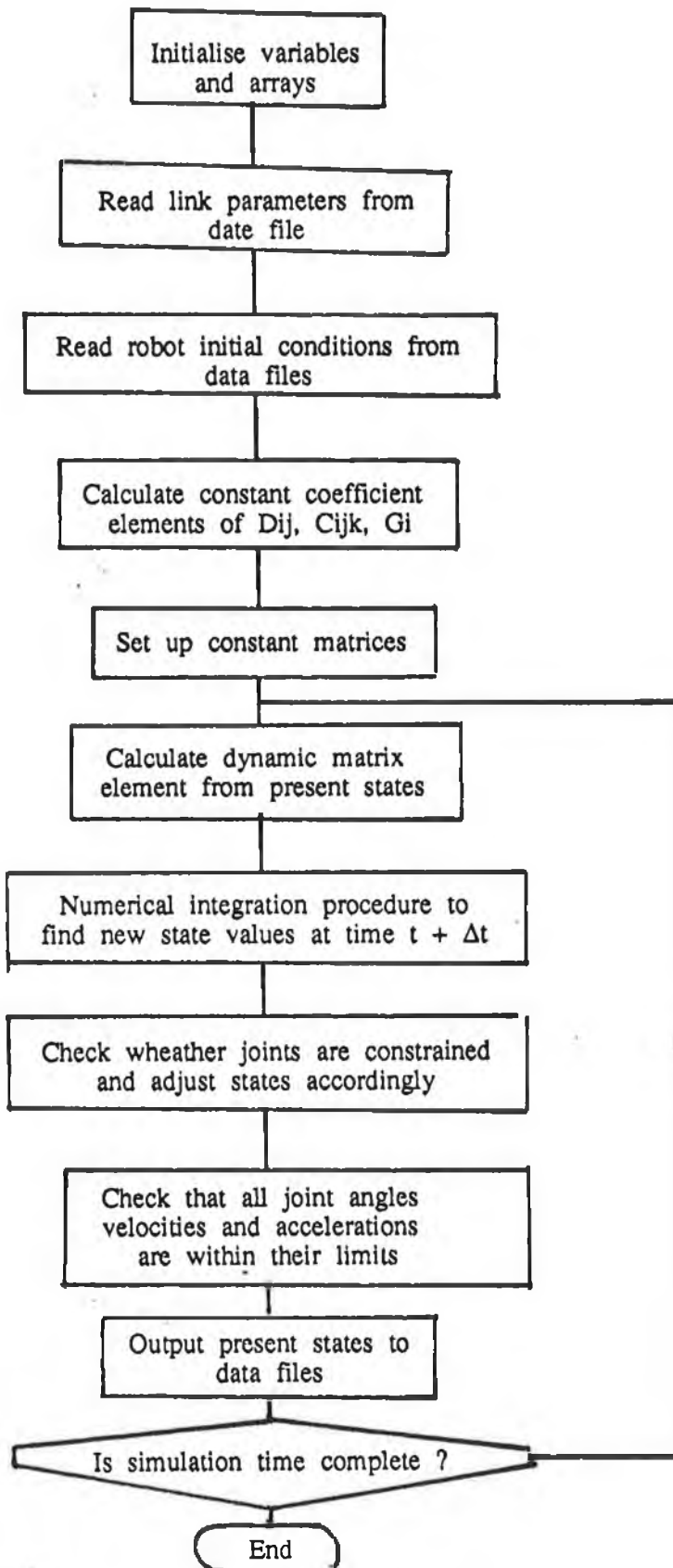


Figure 6.2 Flowchart of simulation program

where

$$\mathbf{x} = (x_1 \ x_2 \ x_3) = (x_7 \ x_8 \ x_9)$$

$$\mathbf{x} = (x_1 \ x_2 \ x_3) = (x_4 \ x_5 \ x_6)$$

By comparing equations (5.9) and (6.1), the matrices have values:

$$\text{FMAT} = \begin{bmatrix} \frac{2L_1}{k_1^t} (x_4 \ x_5 \ x_6) D^1 \\ \frac{2L_2}{k_2^t} (x_4 \ x_5 \ x_6) D^2 \\ \frac{2L_3}{k_3^t} (x_4 \ x_5 \ x_6) D^3 \end{bmatrix}$$

$$\text{BMAT} = \begin{bmatrix} \frac{L_1}{k_1^t} & 0 & 0 \\ 0 & \frac{L_2}{k_2^t} & 0 \\ 0 & 0 & \frac{L_3}{k_3^t} \end{bmatrix} D$$

$$\text{GMAT} = \begin{bmatrix} \frac{R_1}{k_1^t} & 0 & 0 \\ 0 & \frac{R_2}{k_2^t} & 0 \\ 0 & 0 & \frac{R_3}{k_3^t} \end{bmatrix} D$$

$$\text{HMAT} = \begin{bmatrix} \frac{R_1 I a_1}{k_1^t} + H_1 & 0 & 0 \\ 0 & \frac{R_2 I a_2}{k_2^t} + H_2 & 0 \\ 0 & 0 & \frac{R_3 I a_3}{k_3^t} + H_3 \end{bmatrix}$$

$$\text{JKMAT} = \begin{bmatrix} \frac{1}{k_1^t} (x_4 \ x_5 \ x_6) (L_1 D^1 + R_1 D^1) \\ \frac{1}{k_2^t} (x_4 \ x_5 \ x_6) (L_2 D^2 + R_2 D^2) \\ \frac{1}{k_3^t} (x_4 \ x_5 \ x_6) (L_3 D^3 + R_3 D^3) \end{bmatrix}$$

$$\text{KMAT} = \begin{bmatrix} k_1^e N^1 + H_1 & 0 & 0 \\ 0 & k_2^e N^2 + H_2 & 0 \\ 0 & 0 & k_3^e N^3 + H_3 \end{bmatrix}$$

$$\text{LMAT} = \begin{bmatrix} \frac{L_1 D_1}{k_1^t} + \frac{R_1 D_1}{k_1^t} \\ \frac{L_2 D_2}{k_2^t} + \frac{R_2 D_2}{k_2^t} \\ \frac{L_3 D_3}{k_3^t} + \frac{R_3 D_3}{k_3^t} \end{bmatrix}$$

These matrices are evaluated at each integration step and the vector  $\bar{P}(x)$  is found. When the inverse of  $\bar{D}$  has been calculated no variables in equation (5.9) are unknown.

## 6.5 Summary

This chapter outlined the main considerations taken when designing the simulation program for the Puma 560 third order dynamic model. Section 6.1 detailed how conditions can be put on the model joints to facilitate examination of different coupling effects between the links. The conditions override the Runge-Kutta integrated outputs at the end of each iteration. It also enables the control engineer to examine how a controller will respond to different operating conditions.

The fourth order Runge-Kutta numerical integration method was selected for the integration for the following reasons:

- i) small truncation errors, of the order of  $h^5$
- ii) small round off errors since  $h$  may be chosen as a relatively large number  
(  $h = 50\text{mS}$  produced accurate results )
- iii) ease of implementation, and
- iv) it was found to be suitable for a system with a peicewise constant input.

The matrix inversion routine was designed for error minimisation. Gauss Jordon elimination was employed with partial pivoting with accurate results. The matrix  $\bar{D}$  was found to be always nonsingular.

## CHAPTER 7

### OPEN & CLOSED LOOP MODEL PERFORMANCE

This chapter firstly examines the behaviour of the third order dynamic model for the Puma 560 industrial robot which was presented in Chapter 5, and tunes it to ensure that it duplicates the industrial manipulator dynamics. Secondly, a classical PID controller has been implemented on the model joints and performance is evaluated.

The nominal robot link parameters given in Chapter 4 were initially used in the model coefficient terms. Tests on the model performance are chosen to record dynamic effects which relate to different manipulator dynamics. Analysis of these tests allow accurate estimates for link parameters to be made. Selected link parameters are varied to tune the model. Numerical values for some of the link parameters are well known:- namely the position of the centre of mass for each of the links, link masses and motor dynamics. These will remain fixed during all model tuning. Test data is obtained from measurements of the industrial manipulator.

Tests on the dynamic model can be divided into two main groups, open loop and closed loop tests. Simple open loop tests on general model performance may be completed, without reference data from the Puma, when the engineer has an intuitive knowledge of how the manipulator should act. These preliminary tests examine basic model properties and are documented in Sections 7.3 - 7.9. The majority of open loop tests, however, require test data from the actual robot because the manipulator dynamics are highly coupled and difficult to decipher. The dynamic model is tuned by comparing the simulator and actual manipulator responses for similar voltage inputs. Once tuned, the closed loop dynamic model performance is accessed.

Proportional Integral Differential (PID) control is employed on each of the model joints. Each controller is tuned using Ziegler Nichols ultimate sensitivity method. Both single joint and multi-joint control of the dynamic model are examined. The results obtained from the closed loop tests provide insight regarding the complexity of the control problem. It is found that multi-joint control of the dynamic model using single loop positional PID controllers ( without gain scheduling ) produces poor results.

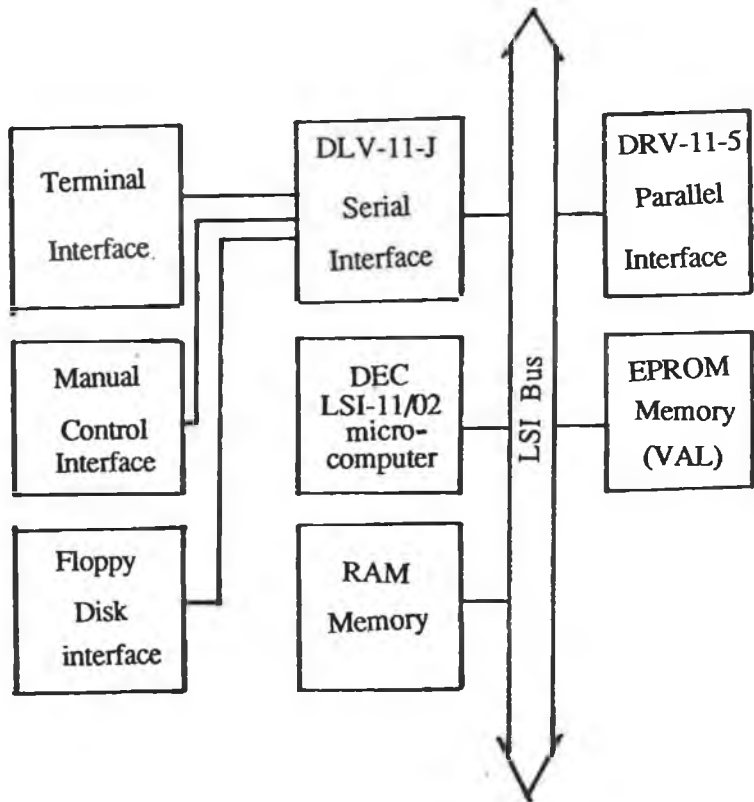
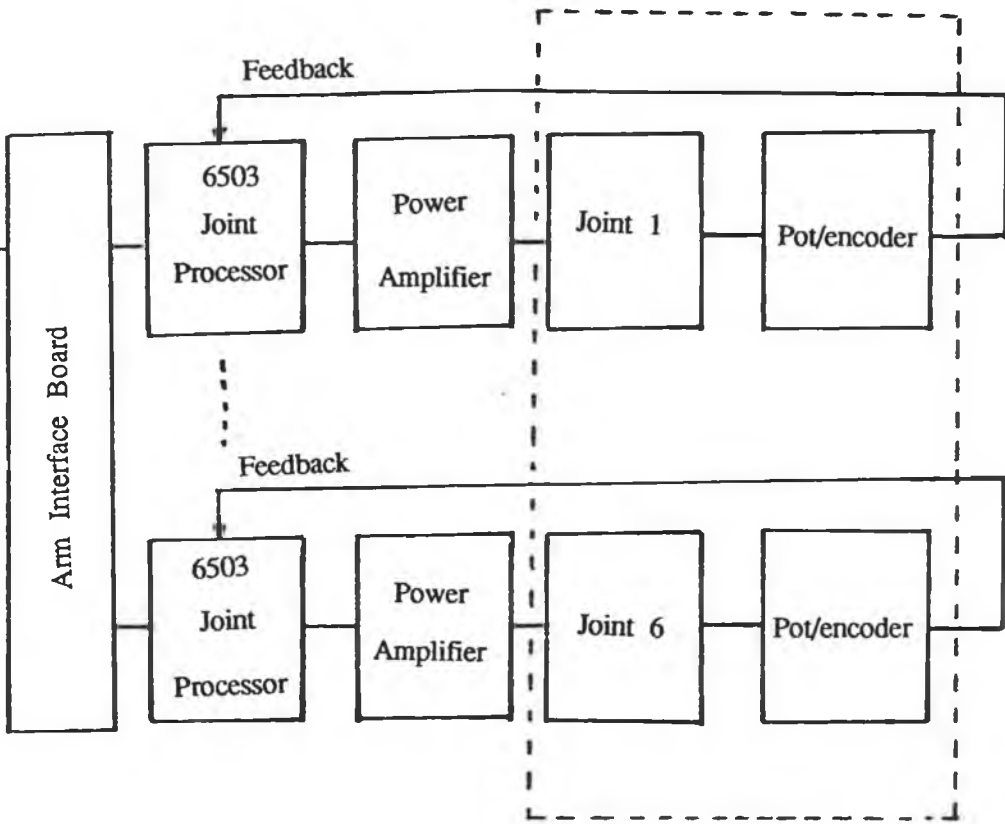


Figure 7.1



Puma 560 Hardware Configuration



## 7.1 Puma 560 Hardware Configuration

Figure 7.1 shows a simplified diagram of the Puma hardware. The Puma range of manipulators are controlled by VAL. VAL provides an operating system which controls the manipulator and allows the user to program various tasks [19]. Tasks are entered in the form of end effector position and orientation coordinates  $(x,y,z,o,a,u)$ . These coordinates are transformed to supply the controller boards with six joint space coordinates. Desired paths are followed by the continual updating of desired joint positions.

VAL has been designed as a selfcontained environment which makes interfacing with external equipment difficult. The user has no access to any hardware information via the keyboard, its software is stored in EPROMS which does not allow the user to examine or modify its routines. It is therefore impossible in the commercial system to access controller board outputs or even joint positions while the arm is tracking some desired path. In order to obtain suitable data for model testing purposes, it was necessary to record data using external equipment.

In the Puma series of manipulator heirarchical controller system the VAL handles path tracking, inverse kinematics, program execution etc. Joint control represents the lowest level. Each joint is controlled by its own digital controller board based around the Rockwell 6503 processor. Each controller board is provided with a set point every 28 mS by VAL, implemented on the DEC LSI-11/02 microcomputer. Potentiometer and incremental encoders are housed on the rotor of each servo motor. Approximately 30 times during each 28 mS window of the digital servo system, the signals from the encoders are compared to the setpoint and any necessary correction signals are generated. The control signals from the controller boards are amplified by the power amplifier units for each of the servomotors.

The Puma 560 mechanical manipulator is represented by the dotted box in Figure 7.1. This section of the manipulator has been modelled in the previous chapters. Although the model output positions are given in radians compared with the incremental encoder output provided in the actual manipulator, both outputs can easily be related to one another.

### 7.1.1 Acquiring Manipulator Input/Output Data

Many of the tests in this chapter require recorded input/output data for the manipulator. In the schematic diagram of the Puma 560 controller hardware configuration, Figure 7.1, the points at which the test data was obtained are shown. For open loop model tests, the input voltages to the three primary joint motors (waist, shoulder, and elbow) and output positions from the calibration potentiometers were recorded on different channels of a DASH-8 card mounted on an Ericsson personal computer.

The absolute joint positions were determinable from the calibration pots on the Puma 560 encoder-potentiometers units connected to each motor armature. The pot signals were tapped from the J-Bus connector on the back plane of the robot control module. Motor input voltages were tapped from the outputs of the three major power amplifier units in the power amp module [14]. Both signals were conditioned and filtered before being converted.

### 7.1.2 Acquiring Controller board Input/Output Data

Input/output data from the controller board is required in order to identify the controller parameters. Figure 7.2 shows a schematic diagram of the Puma controller board with its input and output signals.

All of the Puma digital controller boards have identical hardware and are interchangeable in the Puma system. During the initialisation procedure and execution of the path motions, controller parameters are downloaded from the main computer board to the separate controller boards. In order to obtain data suitable for identification from the controller board the following procedure was taken.

The Tektronix 9100 Series Digital Analysis System (DAS) was used to acquire data at the board inputs and outputs. The controller output is a 12 bit digital signal and is converted to an analog signal before the power amplifier. The incremental encoder input to the controller is in the form of a square wave signal. This signal is converted to an 8 bit code and is available on the controller board data bus. Setpoints are received by the controller board via the Q-bus.

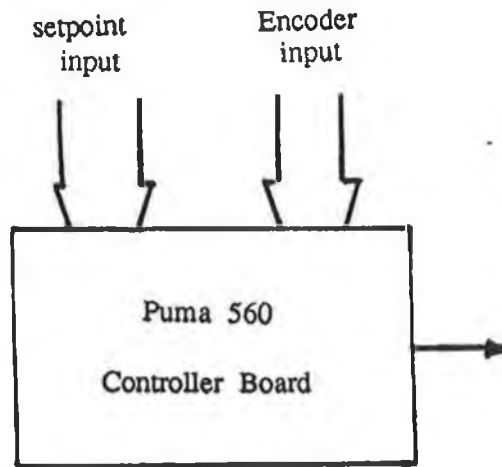


Figure 7.2 Schematic of Controller board

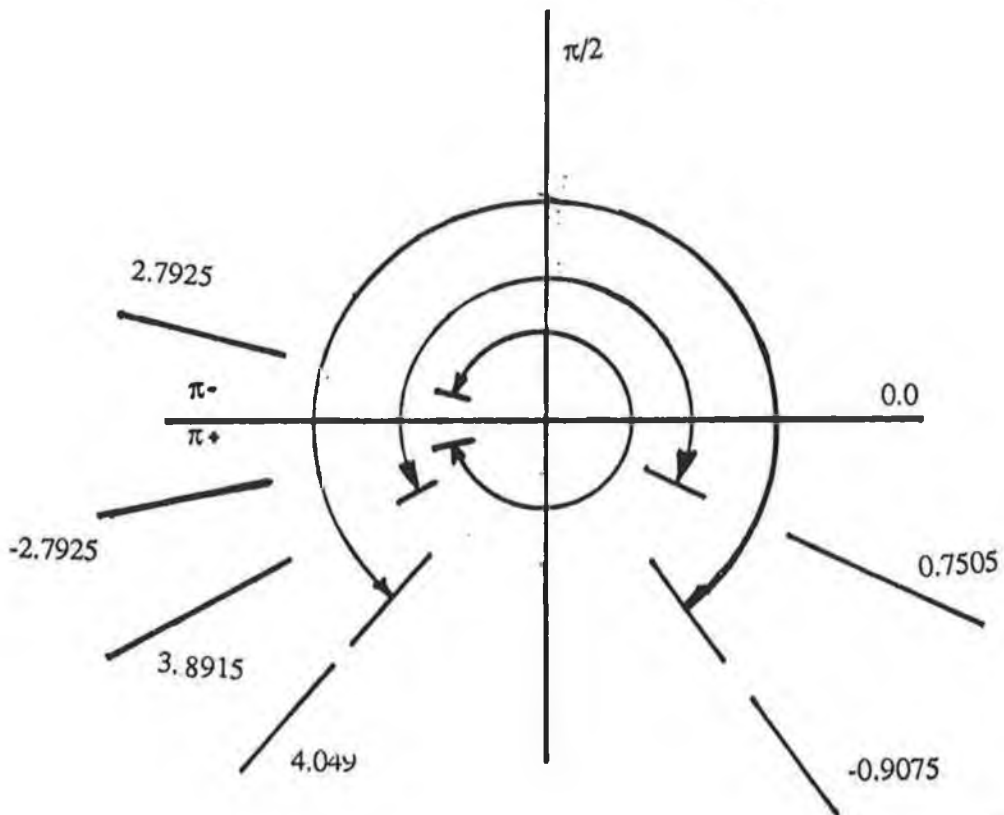


Figure 7.3 Redefinition of joint ranges

The data on the data bus was recorded. The source and destinations of the data was determined by monitoring the control lines on the Q-bus. This made it possible to separate encoder positions and setpoints from other data that appeared on the data bus while the controller software was executing. The controller output was recorded on a separate data acquisition input channel. The DAS has a memory capacity of about 4 kbytes. A sequence of data 4k long could hence be collected and later examined.

## 7.2 Motor Output Torque Tests

To avoid discontinuities from  $\pi$  to  $-\pi$  in the specifications of the joint 2 and 3 simulated angles, the joint angles of Fig 6.1 have been redefined as shown in Fig 7.3 This provides a clearer simulation output.

Several preliminary tests were made on the simulator.

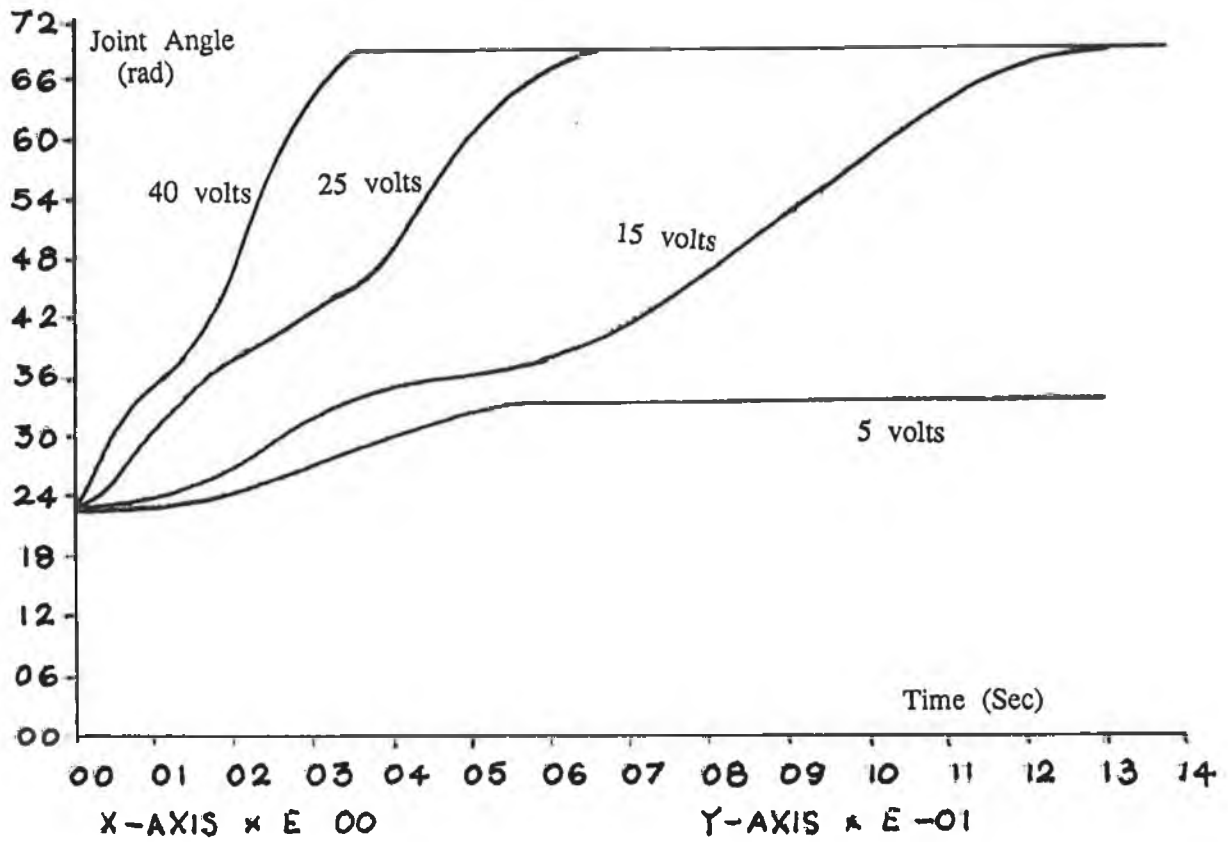
Each of the joints was driven with constant voltages varying from -40 volts to 40 volts in steps of 5 volts to check that the motors produced sufficient torque to drive the links at the full range of velocities specified for the Unimation Puma.

**Test 1** It was found that in order to drive joint 2 through its full range of positions, when the arm was fully extended, a minimum constant voltage of 7.5 volts was required. Viewing the actual Puma motor voltages for a similar motion showed that the minimum voltage required was 8.5 volts.

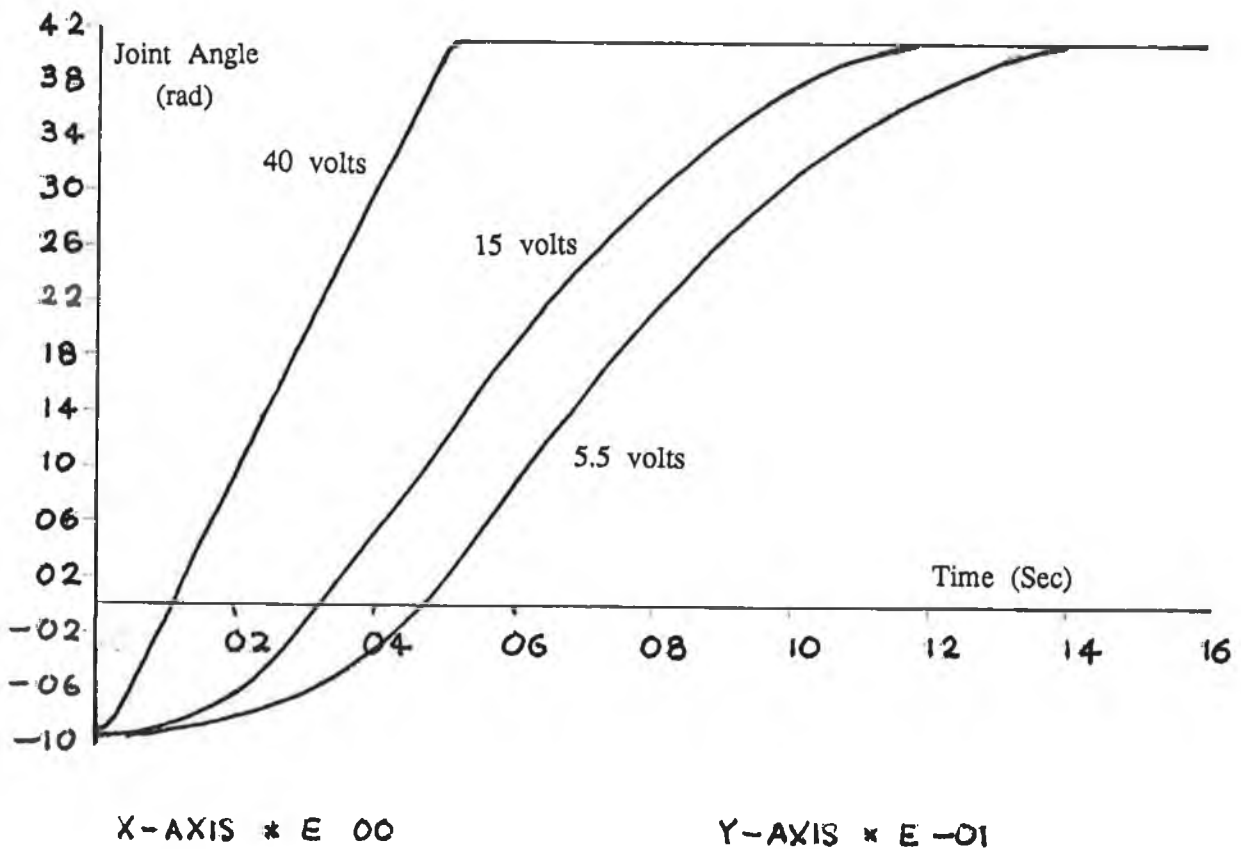
**Test 2** When link 3 was folded back completely, presenting its minimum gravity loading, simulation results showed that joint 2 could be driven by a minimum voltage of 4 volts. In this configuration, when the mass centre of link three was closest to joint 2, the same motion required 5 volts in the industrial robot.

**Test 3** The minimum voltage required to drive joint 3 through its complete range was measured on the industrial robot to be 5.5 volts. In the model, however, a smaller voltage of approximately 3.5 volts was needed.

The differences between simulated and actual voltage requirements can be accounted for by friction forces or incorrect link inertias



Plot 7.1 Simulated motion of joint 2 with constant i/p voltages



Plot 7.2 Simulated motion of joint 3 with constant i/p voltages

**Test 4** Individual joints on the Puma 560 were commanded to rotate at different constant velocities in both directions. Plot 7.1 shows the simulated motion of link 2 for a range of constant voltage inputs. The first and third joints were locked in an extended arm configuration for these tests. The graphs are all nonlinear due to the gravity loading which depends upon position.

**Test 5** Plot 7.2 shows the simulated motion for link 3 for constant input voltages ranging from 0 to 40 volts. Again the motion is nonlinear due to gravity loading.

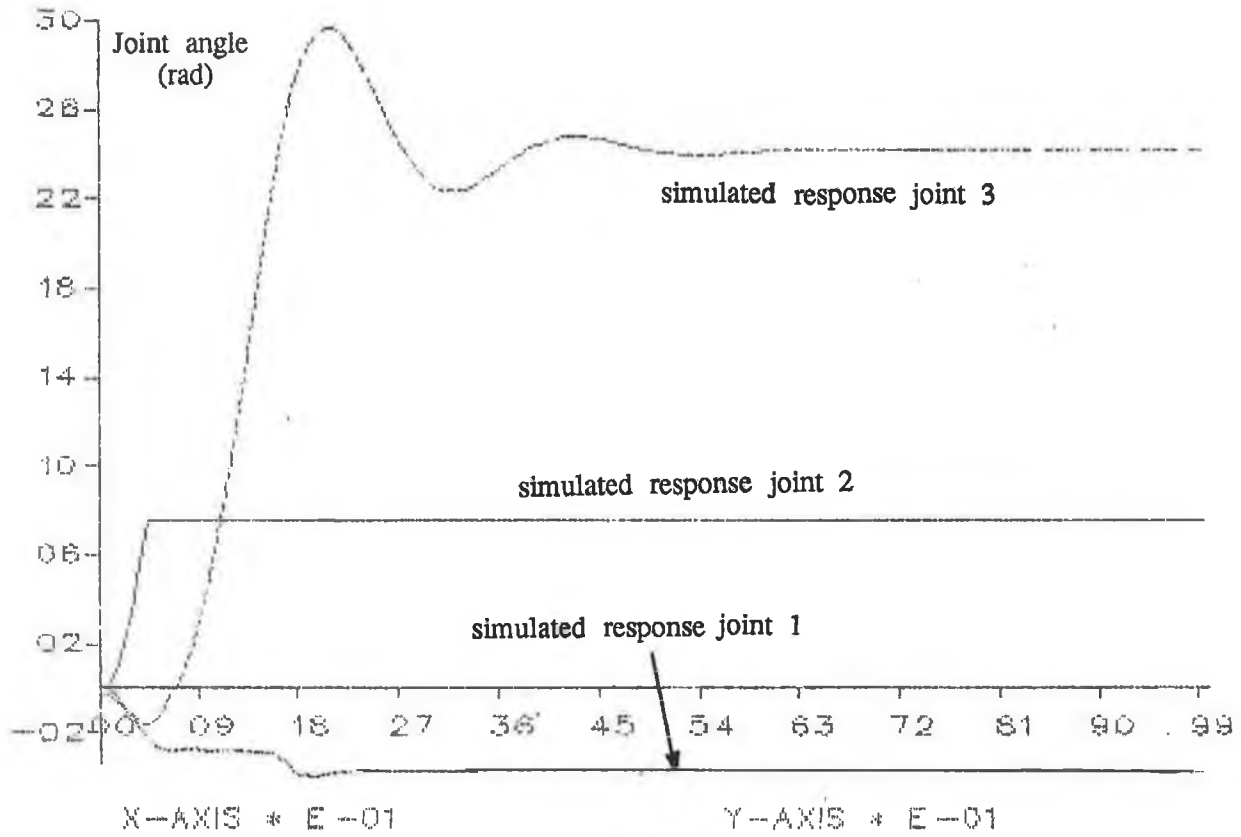
Tests 4 and 5 were repeated for negative voltages starting from the other link extremes. The link movements were found to be symmetrically identical.

### 7.3 Estimating Friction Coefficients using Gravity Tests

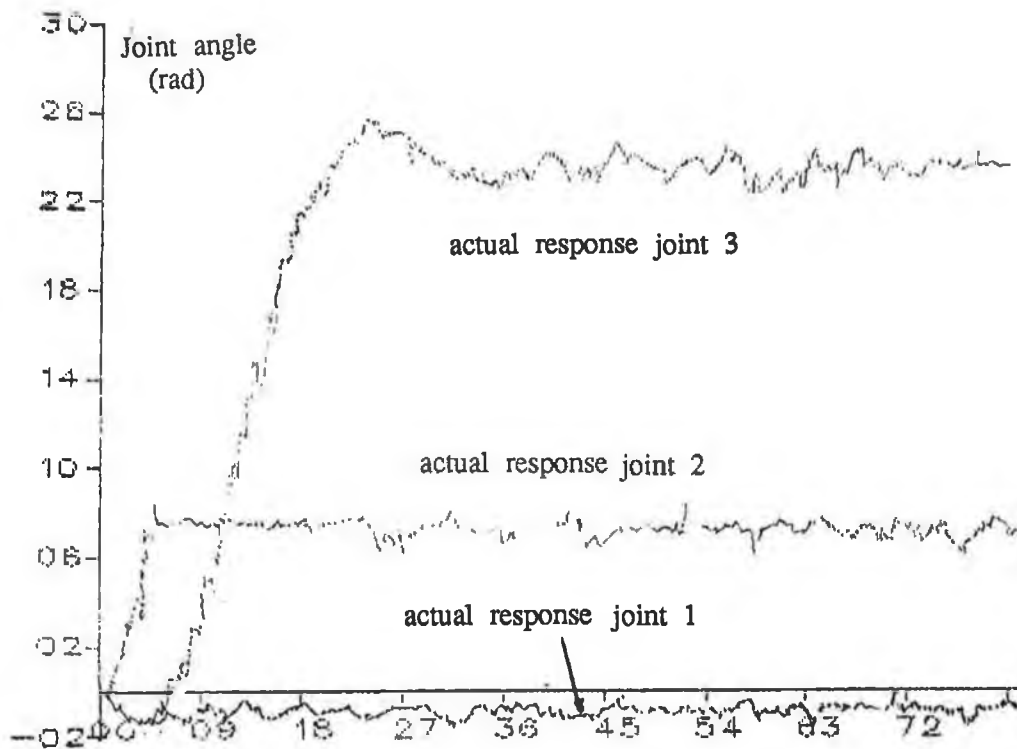
Gravity tests are used to provide initial estimates of friction forces for each of the three joints. Friction is modelled as coulomb friction.

The arm simulator was run in different starting positions with zero voltage inputs and links given freedom of motion. The joints were allowed to collapse under the force of gravity. The test when the simulator initial angles were all equal to zero, as shown in Figure 3.4 is described here as a typical example. For this test the friction forces were initially set to zero. Friction forces are later introduced to improve the model response. Plot 7.3 shows the simulated response. In order to compare the simulation response with the actual manipulator responses for a similar motion, the robot arm was supported and the brake release button was pressed. Plot 7.4 shows the measured response from each of the three joints of the Puma.

The simulated motion of link 2 compares well with the actual motion for this test. The simulated link 3 response has a larger overshoot than was recorded for the real system. Joint 1 is not affected by gravity however the movement of joints 2 and 3 have caused it to rotate fractionally. The extent of this rotation was more marked in the simulation than on the Puma. Introducing a carefully chosen friction coefficient for joint 3 produced the response shown in Plot 7.5. After a series of gravity tests it became apparent that friction forces



Plot 7.3 Frictionless simulated freefall motion of joints 1, 2 & 3



Plot 7.4 Measured Puma freefall motion

existed at all joints. Estimates of the friction forces present at joints 1 and 3 were made and introduced permanently into the model. It proved more difficult to estimate the friction forces at joint 2 due to lack of test data. It was not possible for this link to oscillate as in the case of link 3. Allowing it to collapse from greater heights ran the risk of damaging gearing and stop supports. The friction experienced at joint 2 was estimated to be twice that of joint 3 since the gearing ratio of joint 2 is twice that of joint 3.

The estimated values for friction coefficients are given in Table 7.1.

#### 7.4 Estimating Radii of Gyration using Self Inertial Tests

The self inertia experienced at any joint is due to the acceleration at that joint. The self inertia experienced at link  $i$  depends on the inertias of links  $j$  ( $j \geq i$ ) and on the manipulator configuration. The self inertia terms provide a means of measuring some of the link parameters. In the following tests the wrist joints are treated as a lumped mass on link 3. The radii of gyration of the manipulator links are related to the link moments of inertia as given in the set of equations 3.10. Section 4.3.1 gave the formulae for the manipulator dynamic equation coefficient terms. By measuring the input output responses of the manipulator joints in different configurations the values of link parameters can be estimated. The tests are designed so that the effect of different terms may be isolated.

In the following tests the various joints were manually controlled using the Puma manual control pendant. The manual control was set at its maximum speed position in order to achieve maximum acceleration and deceleration. Although the manual control does not allow the manipulator to attain its maximum velocity the acceleration to reach this speed is sufficient for inertial tests.

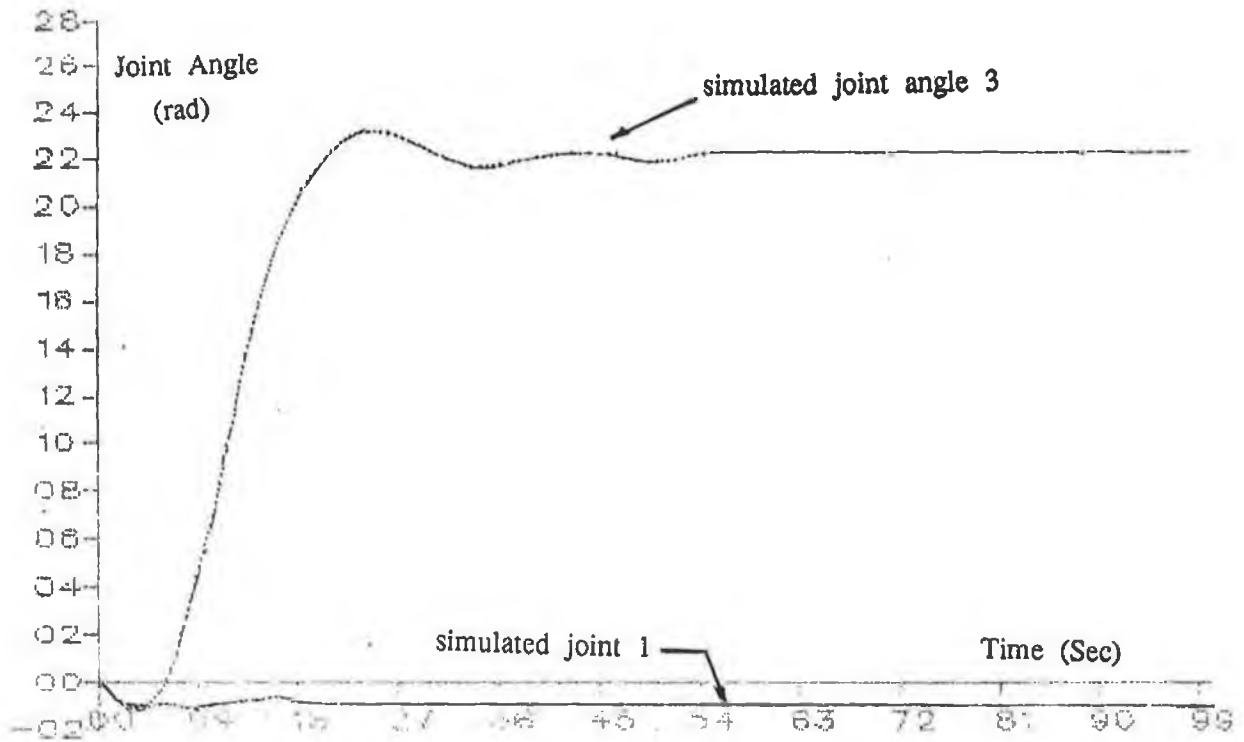
##### 7.4.1 Estimation of $k_{3zz}^2$

Because of the manner in which each of the inertial terms are coupled, parameters in link 3 were examined first. Symbolic formulæ for all inertial coefficients  $D_{ij}$  for the Puma 560 are given from eqn (4.18) as:

$$D_{33} = m_3(k_{3zz}^2 + a_3^2 + 2a_3\bar{x}_3)$$

specifies the self inertia for link 3. Only the  $z$  component of the radius of

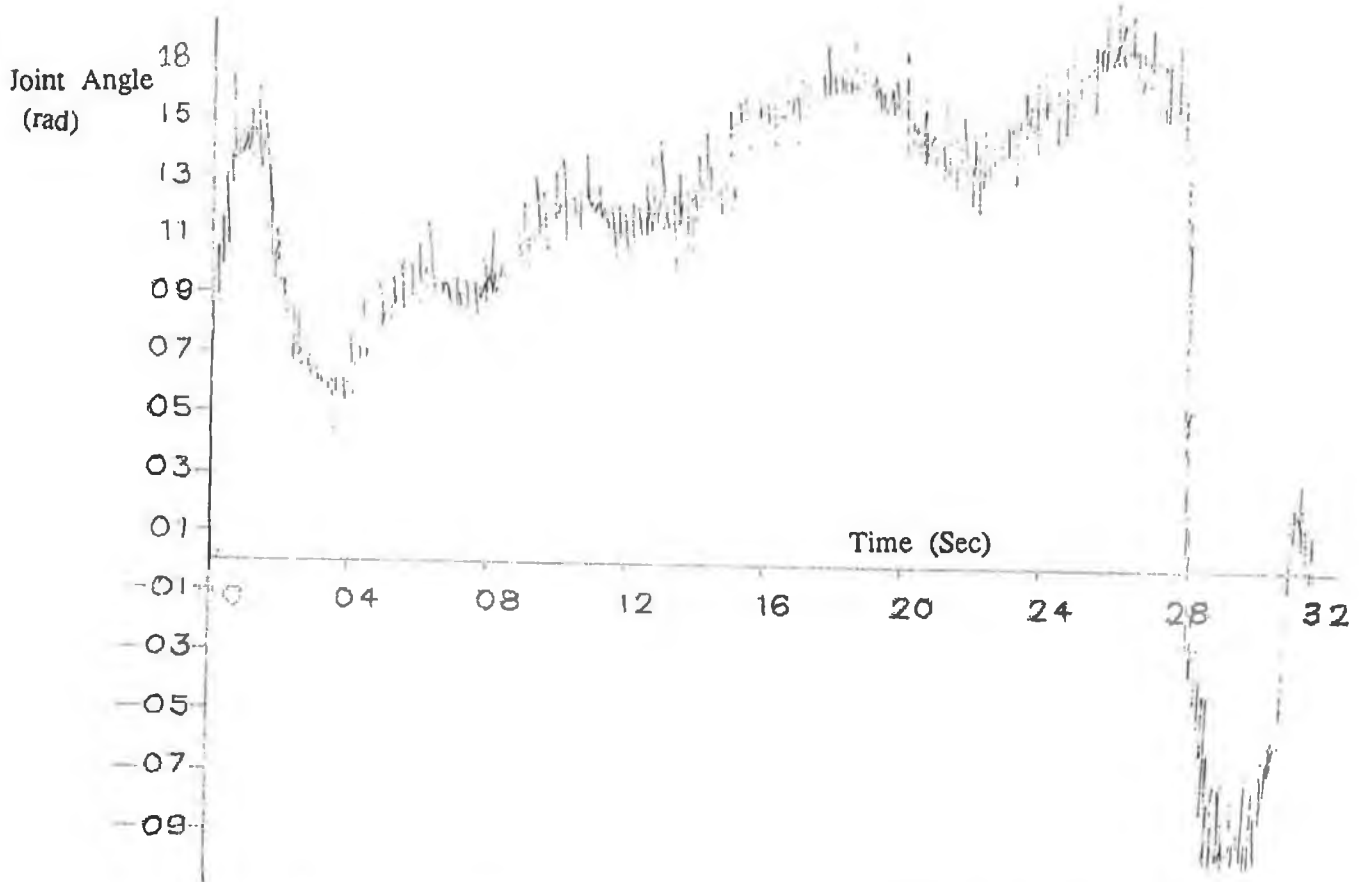




X-AXIS \* E -01

Y-AXIS \* E -01

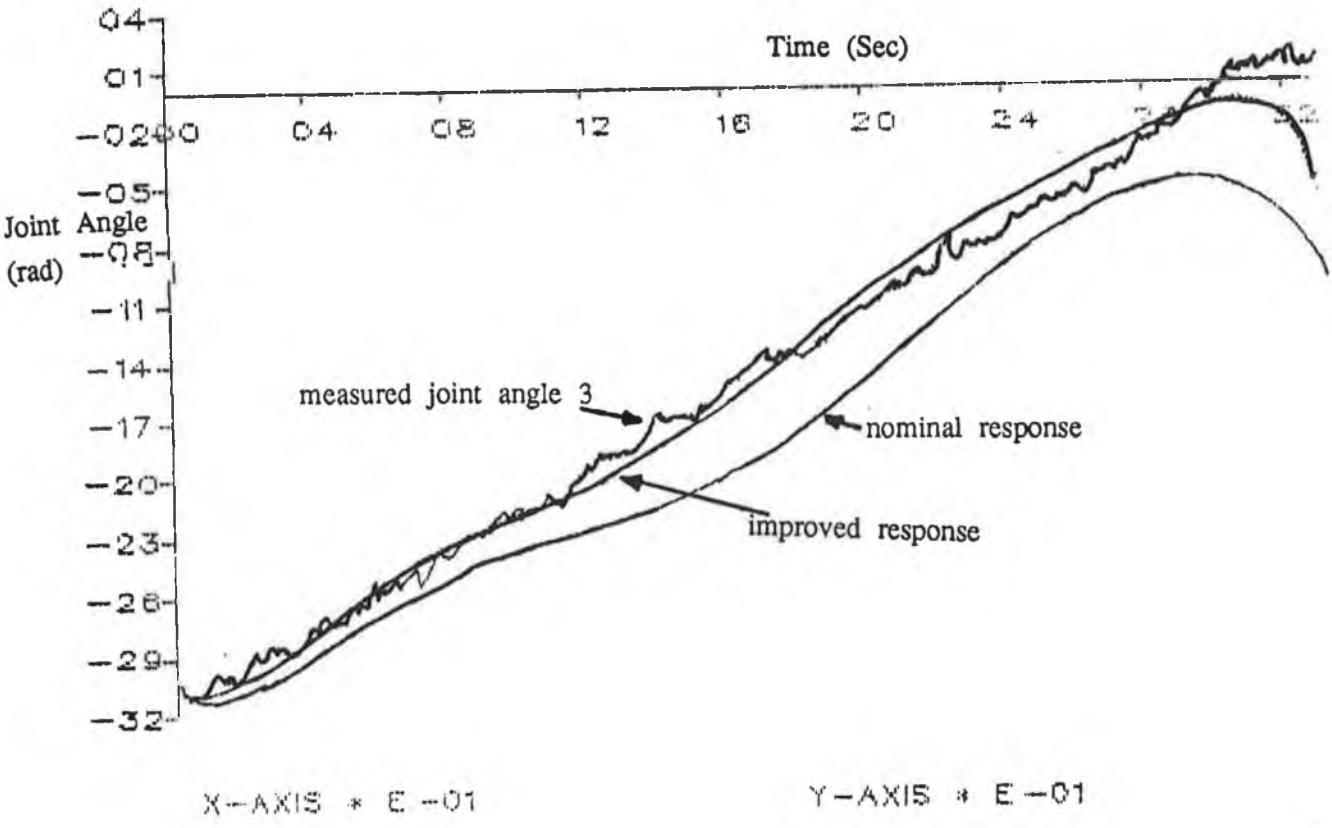
Plot 7.5 Simulated freefall motion of joints 1 & 3 with friction



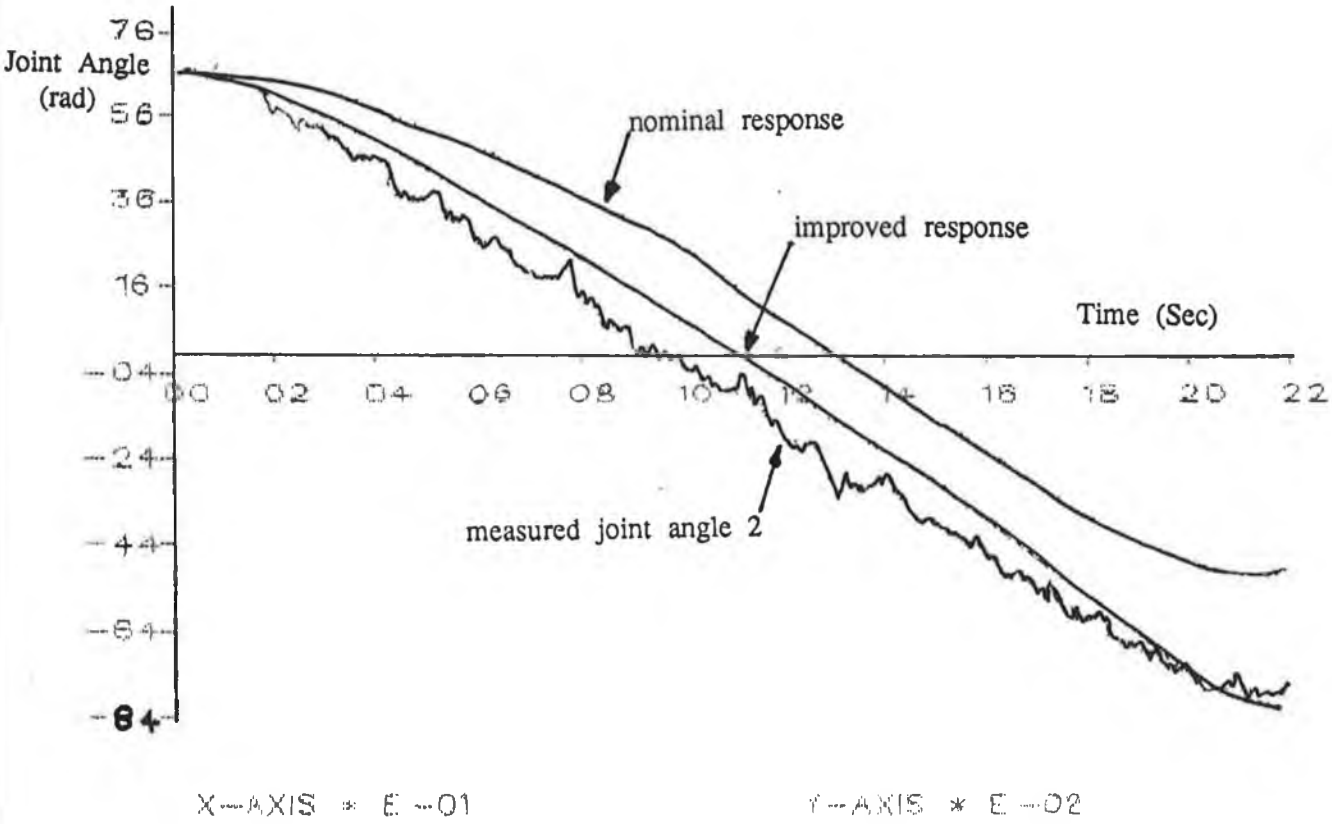
X-AXIS \* E -01

Y-AXIS \* E 00

Plot 7.6 Voltage i/p to joint 3 for constant velocity demand



Plot 7.7 Position profiles for constant joint 3 velocity demand, measured and simulated



Plot 7.8 Position profiles for constant joint 2 velocity demand, measured and simulated

gyration is present because link 3 rotates about the z axis of the reference frame. Locking joints one and two allows us to study the movement of the third link in isolation.

Link 3 was allowed to hang vertically. The link was then commanded to move at the maximum velocity possible using the puma manual control unit, to its horizontal position. This command involved an initial acceleration then a constant velocity followed by a deceleration period. The acceleration and deceleration periods are of particular interest when analysing the link inertial terms.

Link three was commanded to rotate about the joint 3 axis with the waist and shoulder links in locked positions. In this configuration link 3 is only affected by inertial and gravity terms. The input voltage to motor 3 was recorded and is plotted in Plot 7.6. Plot 7.7 shows three plots, a) measured joint 3 angle movement, and the simulated open loop movement of joint angle 3, b) nominal value of  $k_{3ZZ}^2$  and c) tuned value of  $k_{3ZZ}^2$ . Reducing  $k_{3ZZ}^2$  improved the likeness between measured and simulated responses. Repeating the above test at different points over the full range of possible positions allowed for joint 3 gave an accurate estimation of  $k_{3ZZ}^2$ . This value is given in Table 7.1

#### 7.4.2 Estimation of $k_{2ZZ}^2$

The same procedure may be carried out in order to estimate  $k_{2ZZ}^2$  using test data for link 2. From eqn (4.13) the self inertia of link 2 is given as :

$$D_{22} = m_2(k_{2ZZ}^2 + a_2 + 2a_2x_2) + m_3[(2a_2a_3 + 2a_2\bar{x}_3)C_3 + 2a_2\bar{z}_3S_3 + k_{3ZZ}^2 + a_2^2 + a_3 + 2a_3\bar{x}_3]$$

All elements except  $k_{2ZZ}^2$  on the R.H.S. of the equation are known. In order to obtain test data for joint 2, joint 3 was locked at  $-52^\circ$  to ensure it presented it's minimum inertia to joint 2. Figure 7.4 shows this configuration. Joint 2 was then commanded to accelerate to the full velocity possible with the manual control. Plot 7.8 shows measured joint movement and Plot 7.9 shows the corresponding input voltage profile. Again the simulator response was examined for the voltage shown in Plot 7.9 and the responses for nominal and tuned values of  $k_{2ZZ}^2$  are both plotted in Plot 7.9.  $k_{2ZZ}^2$  was increased to improve the

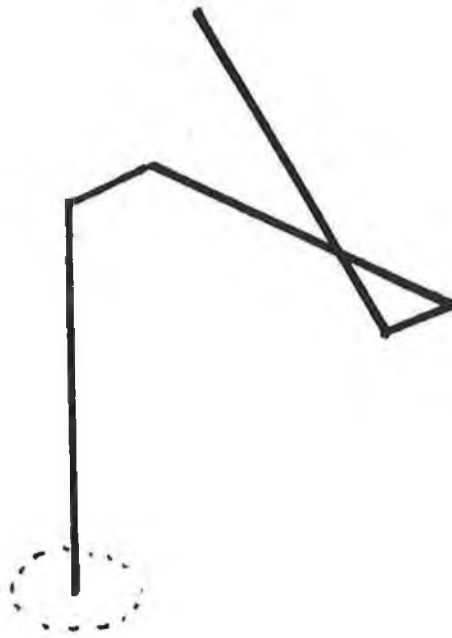


Figure 7.4 Position of link 3 to present minimum Inertia ot joint 2

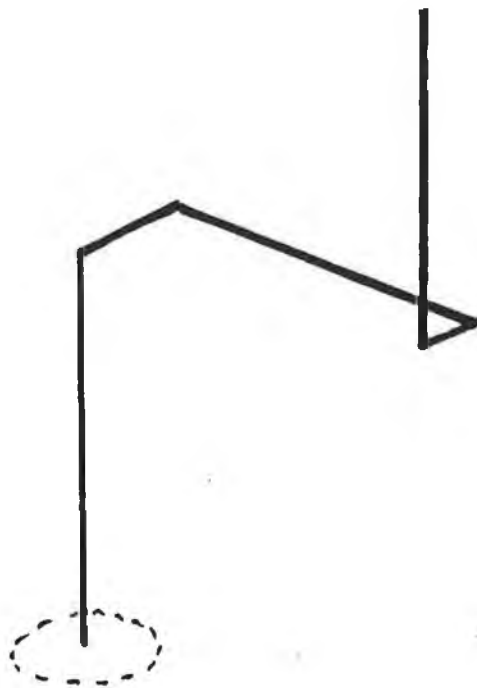


Figure 7.5 Manipulator position for final gyration Test

response. After several similar tests at various evenly distributed initial positions within the range of joint two the value of  $k_{2zz}^2$  was accurately determined and is entered in Table 7.1.

#### 7.4.3 Estimation of $k_{3xx}^2$ , $k_{3yy}^2$ & $k_{1zz}^2$

When determining  $k_{3xx}^2$  and  $k_{3yy}^2$  the self inertial term for joint 1 was used. An expression for  $D_{1,1}$  was given in equation (4.9) as.

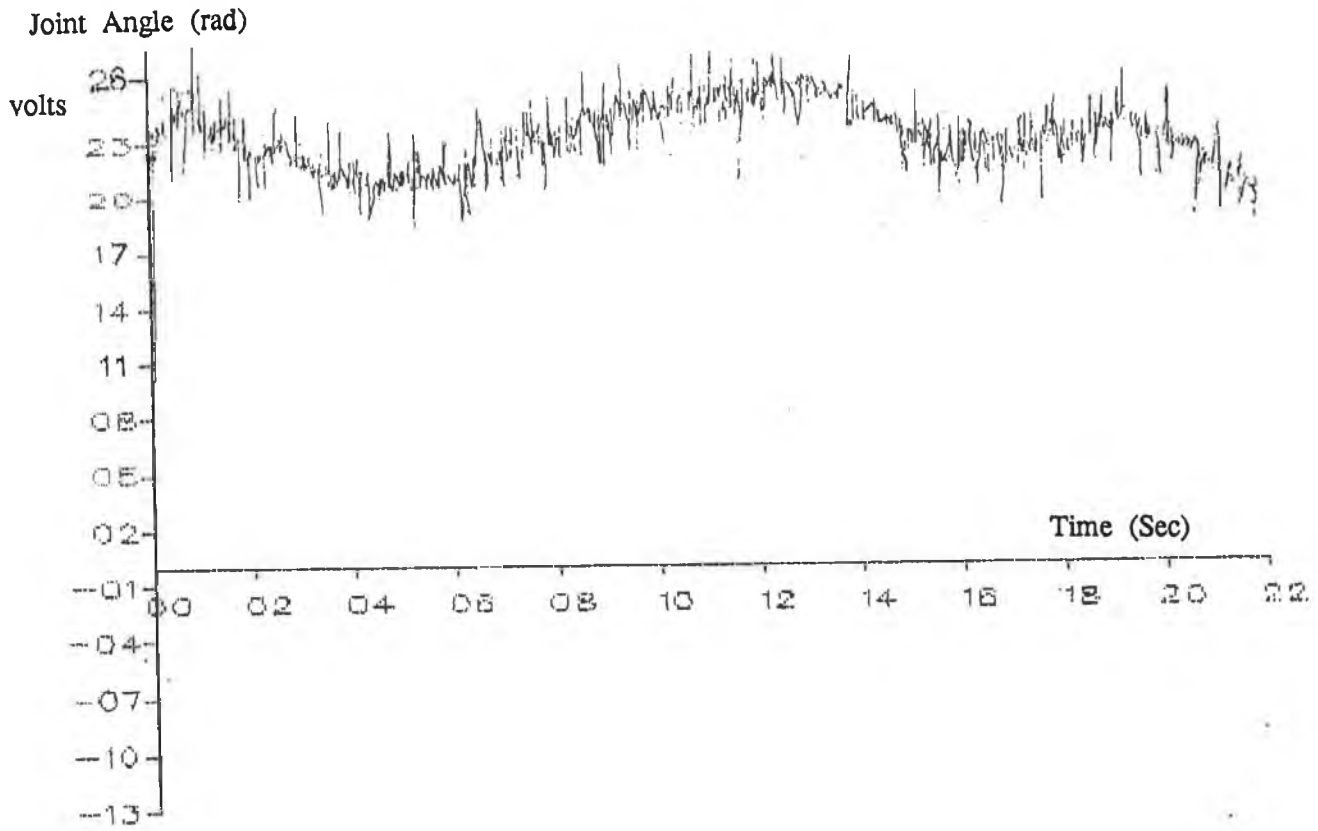
$$\begin{aligned}
 D_{1,1} &= m_1 k_{1zz}^2 \\
 &+ m_2 (k_{2xx}^2 S_2^2 + k_{2yy}^2 C_2^2 + a_2^2 C_2^2 + 2a_2 \bar{x}_2 C_2^2) \\
 &+ m_3 [ k_{3xx}^2 S_{23}^2 + k_{3zz}^2 C_{23}^2 + d_3 + a_2^2 C_2^2 + a_3^2 C_{23}^2 + 2a_2 a_3 C_2 C_{23} \\
 &+ 2\bar{x}_3 (a_2 C_2 C_{23} + a_3 C_{23}^2) + 2\bar{y}_3 d_3 + 2\bar{z}_3 (a_3 C_{23} S_{23} + a_2 C_2 S_{23}) ]
 \end{aligned}$$

During the following test we assume that the link parameters for links 1, 2 and 3 are in the correct proportion. Responses are recorded for different tests and the parameters then scaled to give the best fit to the measured data.

The first test had the robot arm in its extended upright position. This position corresponded to the three joint angles being set as follows :  $q_1$  was set to  $0^\circ$ ,  $q_2 = 90^\circ$  and  $q_3 = -90^\circ$  as shown in Figure 7.4. Substituting these joint angles into equation (4.9) shows that the term containing  $k_{2yy}^2$  equals zero and  $k_{2xx}^2$  has its maximum effect. Joint 1 was commanded to rotate at its maximum velocity. Plot 7.10 shows its movement and Plot 7.11 shows the driving voltage. The simulation response for the same input is plotted with the actual response in Plot 7.10. The simulator has a slower response because the  $k_{3yy}^2$ ,  $k_{2xx}^2$ ,  $k_{1zz}^2$  are fractionally too large. It was difficult to identify which contributed most, so all were scaled down proportionally. This resulted in the improved response also plotted in Plot 7.10. Table 7.1 contains the corrected values.

#### 7.4.4 Estimating the Remaining Radii of Gyration

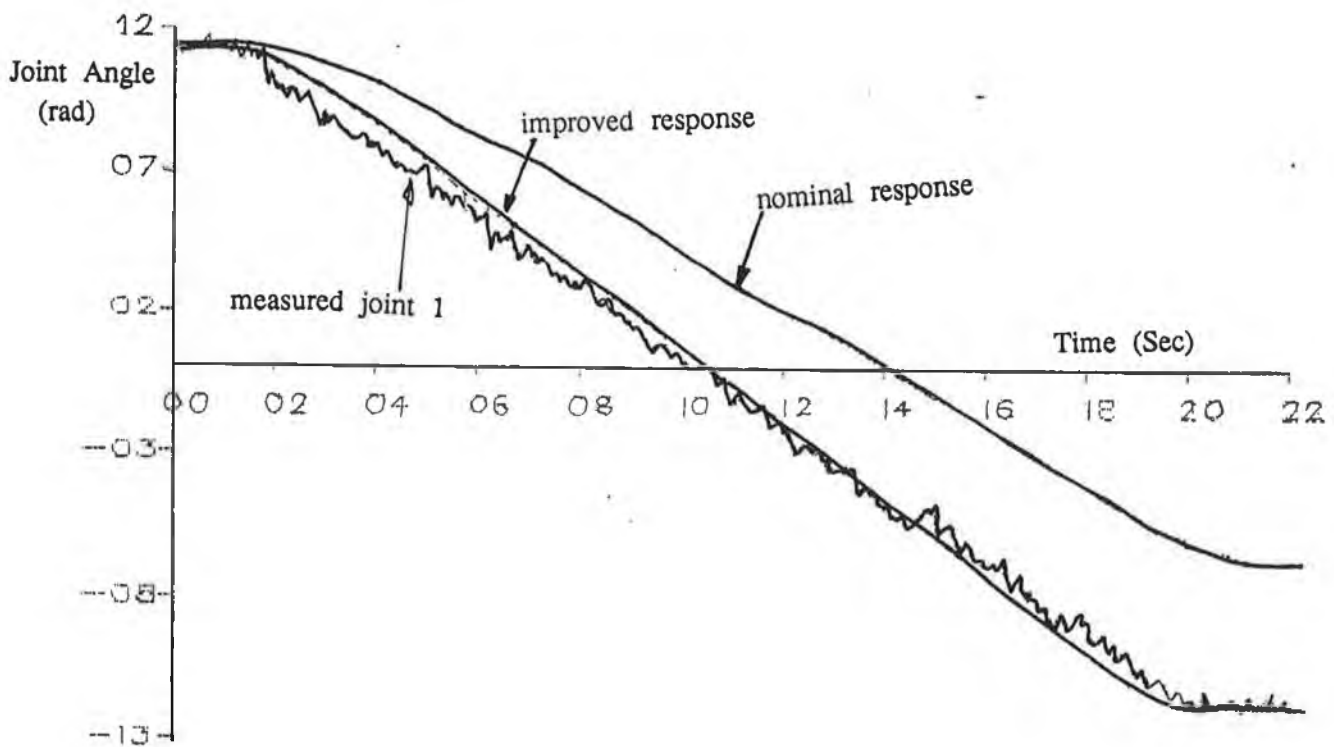
In order to estimate the value of  $k_{3yy}^2$  the arm was placed in the configuration shown in Figure 7.5. Substituting the three joint angles into equation 7.3 causes some of the terms to disappear. When in this configuration  $k_{2xx}^2$  has no effect while  $k_{2yy}^2$  has its maximum effect. Plot 7.12 shows the recorded movement of joint one when commanded to rotate at constant velocity in a negative direction.



X-AXIS \* E -01

Y-AXIS \* E 00

Plot 7.9 Voltage i/p to joint 3 for constant velocity demand



X-AXIS \* E -01

Y-AXIS \* E -01

Plot 7.10 Position profiles for constant joint 1 velocity demand, measured and simulated

Plot 7.13 shows the corresponding recorded input voltage. Plot 7.12 also shows the tuned and untuned simulated joint responses to the same voltage input.

The radius of gyration elements for each of the links was found (in a similar manner) by setting the arm in different configurations, causing different joints to accelerate recording the joint input and output response and tuning the simulator response by varying the causal parameter.

Table 7.1 gives the full list of tuned link parameters.

### 7.5 Open Loop Path Tracking Tests

This section examines how close the simulator can follow the Puma motion for a recorded set of motions when operating in open loop. In these tests the Puma 560 tool was commanded to travel in a straight vertical line between two points. Figure 7.6 shows a lined diagram of the manipulator initial and final positions. The command was given via the manual control module. The path tracking tests were run at a relatively slow speed.

The input voltages to the three joints were recorded while the manipulator moved through this path. These voltages were input to the simulator. Plots 7.14, 7.15 and 7.16 show the measured and simulated joint angle profiles for joint 1, 2 & 3. The simulated joint angle movements compare well with the measured angles. Both joint 2 and joint 3 experience a drop due to gravity at first. After the initial slip joint movements are satisfactorily similar. Simulated joint 3 movement is slower than the recorded response.

### 7.6 Considerations for Closed Loop Tests on the simulator

In order to emulate the Puma controller system, in software, on the dynamic model the following steps are taken:

- (1) Setpoints are supplied to the software controller program every 28 mS.
- (2) The controller program samples the model output variables every 28/30 mS.
- (3) Control signals will remain constant over each sampling interval.
- (4) The power amplifiers are modelled as constant gains.

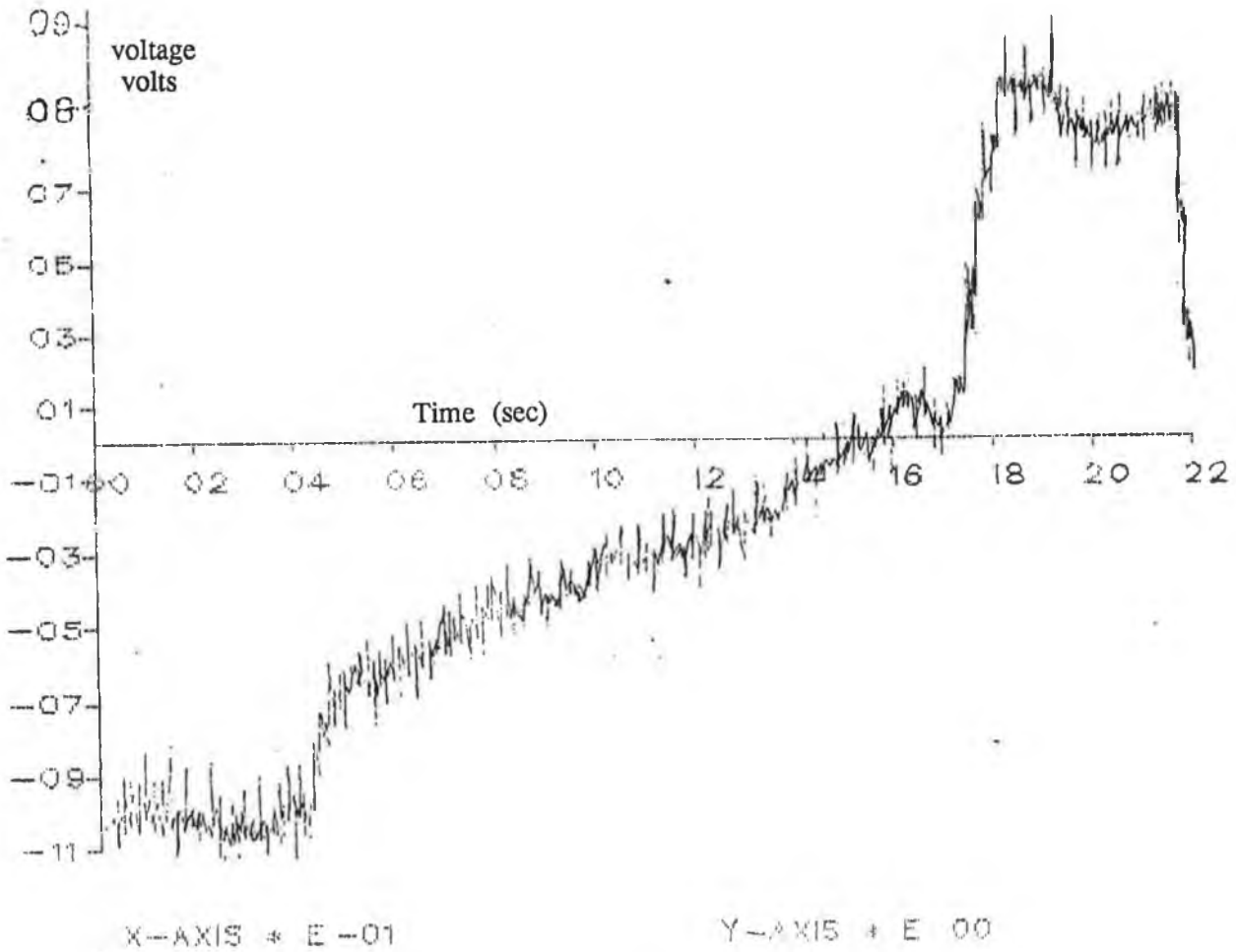
The controller discussed in the following section deal with the closed loop

TABLE 7.1  
Tuned Puma 560 link parameters

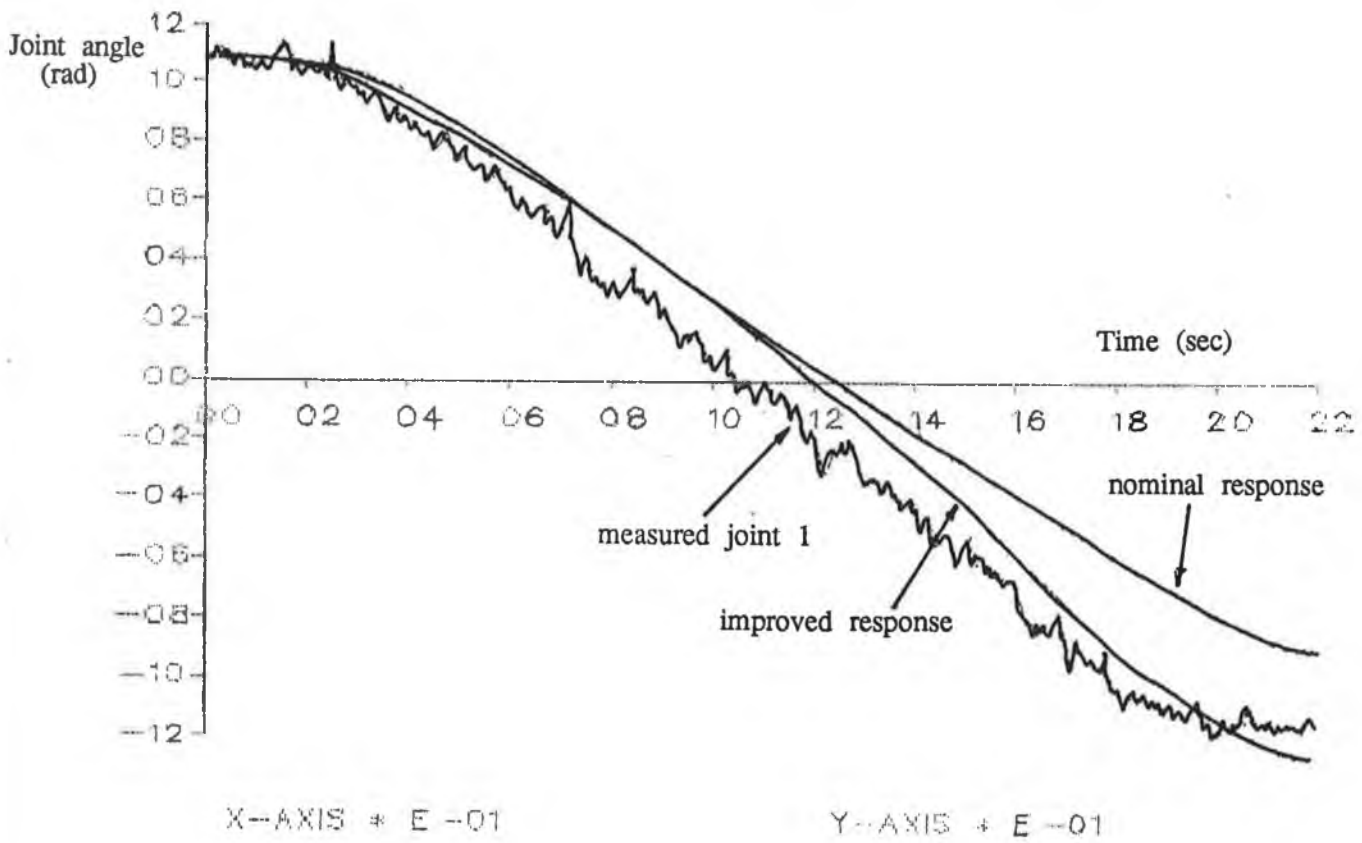
link	mass	radius of gyration			friction	reflected motor inertia
	$m_i$ (kg)	$k_{iXX}^2$ (m <sup>2</sup> )	$k_{iYY}^2$ (m <sup>2</sup> )	$k_{iZZ}^2$ (m <sup>2</sup> )	$H_i$ (Nm/m/s)	$Ia_i$ (kg-m <sup>2</sup> )
1	12.96	0.1802	0.1800	0.0141	0.0107	0.1816
2	22.37	-0.0516	0.1900	0.0031	0.0341	0.2361
3*	5.01	0.0510	0.0691	0.0027	0.0150	0.0590

\* Considering the last three links as a load added to link three.

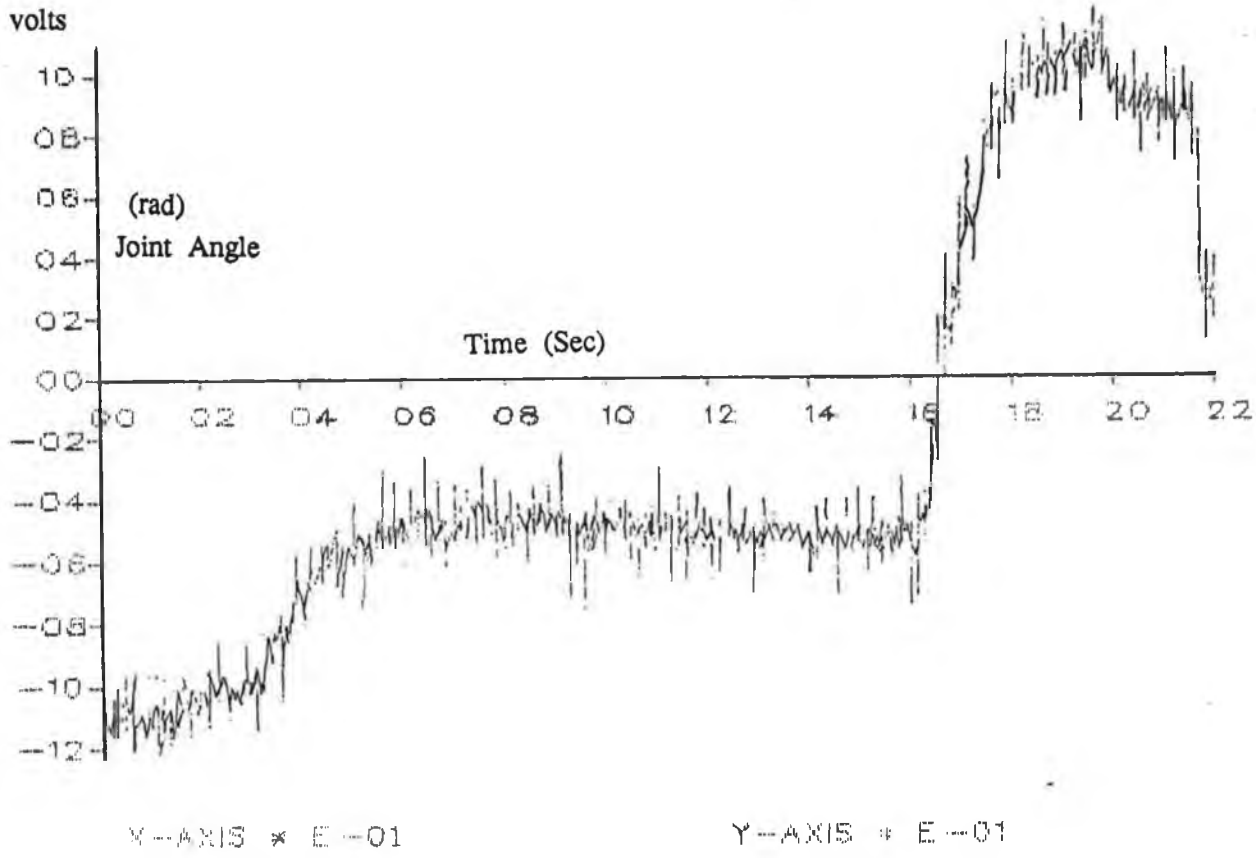




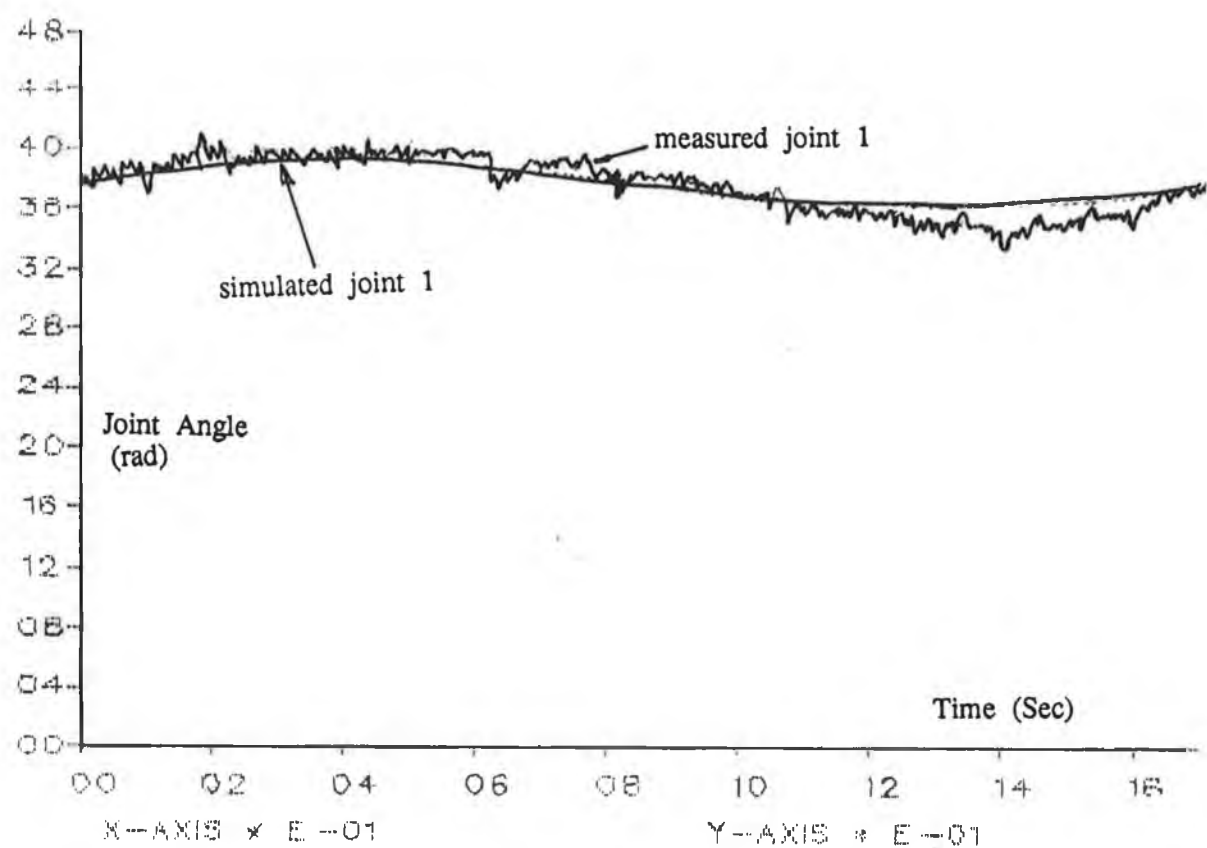
Plot 7.11 Voltage i/p to joint 1 for constant velocity demand



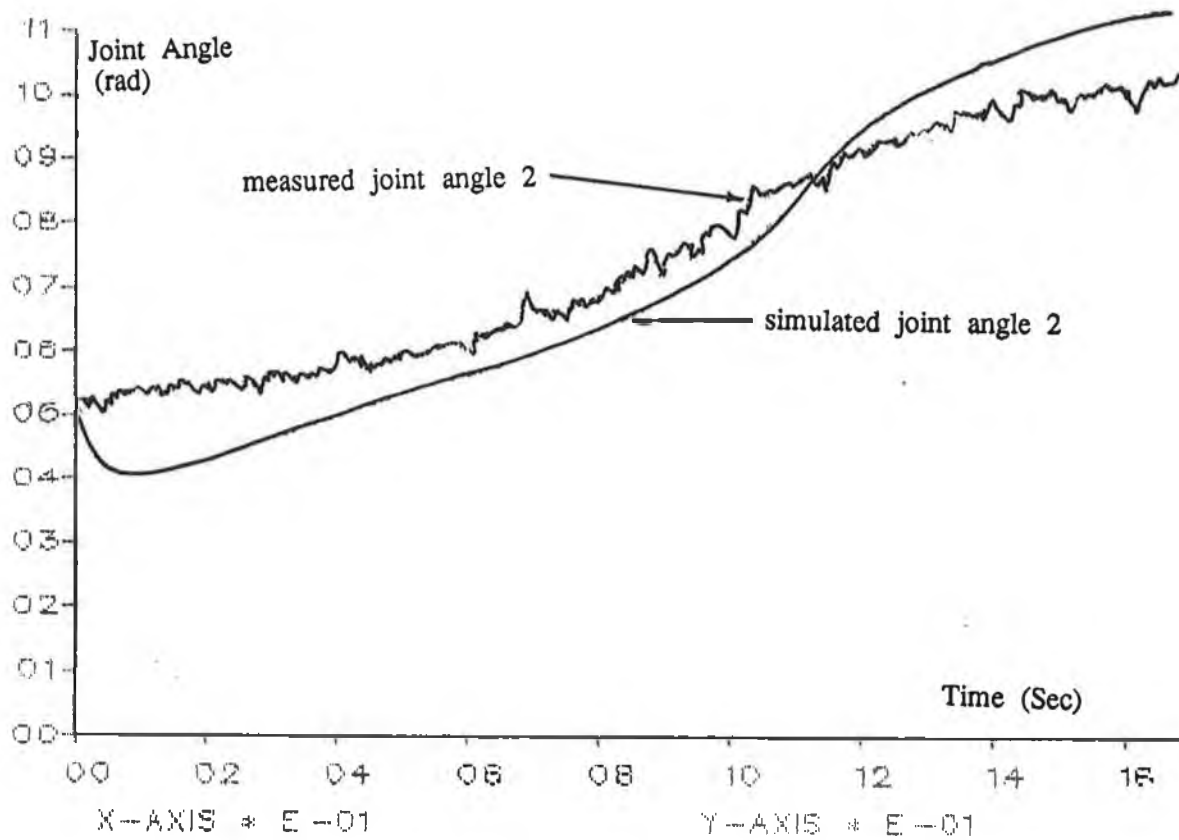
Plot 7.12 Simulated position profile for joint 1, negative velocity demand



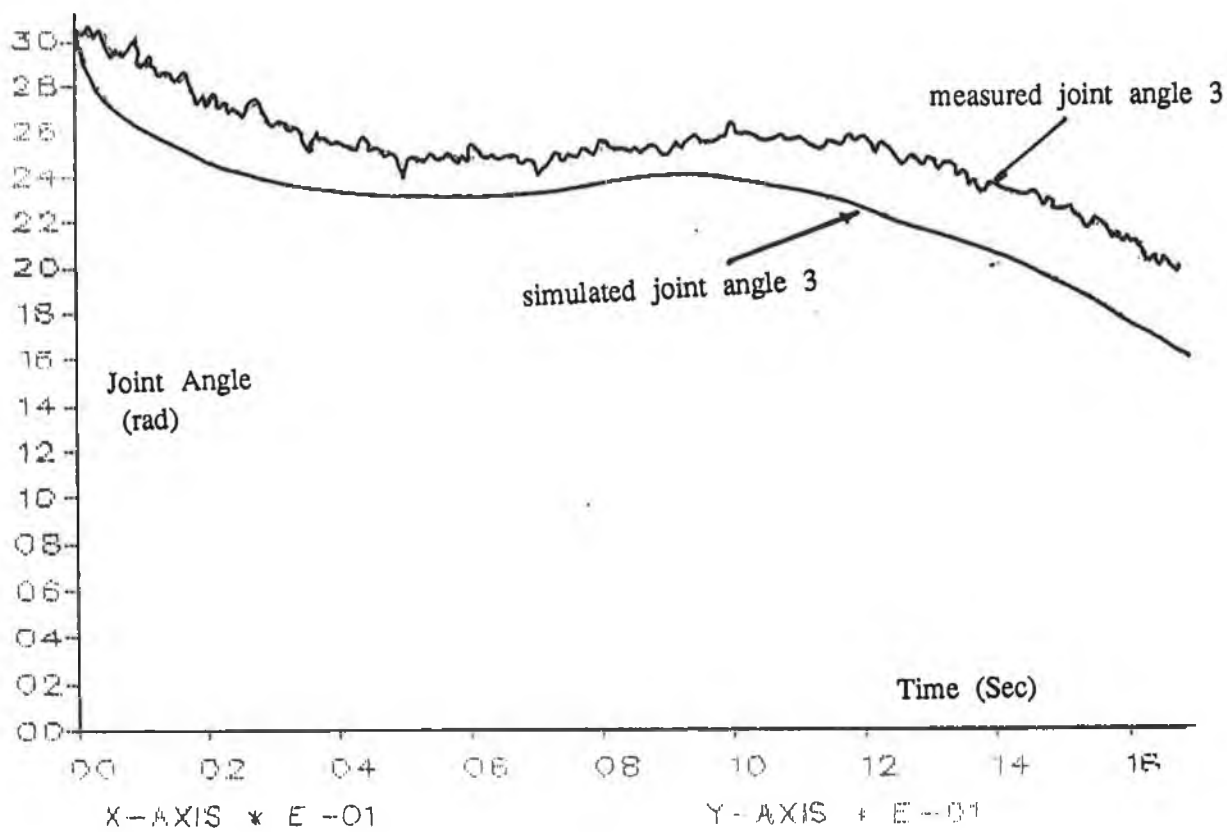
Plot 7.13 Recorded i/p voltage for constant velocity at joint 1



Plot 7.14 Motion of joint 1 during simultaneous open loop response of all three joints.



Plot 7.15 Motion of joint 2 during simultaneous open loop response of all three joints.



Plot 7.16 Motion of joint 3 during simultaneous open loop response of all three joints.

system sketched in Fig 7.1. The hardware to the left of the digital controller boards shown in Fig 7.1 is replaced by a data file which contains the setpoints.

Separate discrete Proportional Integral Differential (PID) controllers are used to control each of the three primary joints individually. The PID controllers are tuned using Zielgler-Nichols ultimate-sensitivity method [21]. Once PID values were chosen the closed loop system response was examined for different joint path commands. The setpoints are chosen to be ramp and step functions in time. The step function command provides an indication of overshoot and rise times.

The open loop tests have provided sufficient information to tune the model so that it more accurately represents the true manipulator dynamics. The tuned model is suitable for controller evaluation. The remainder of this chapter examines the closed loop model performance.

### 7.6.1 A Review of PID Control

Astrom [21] covers both discrete and analog PID control in detail. PID controllers have very robust characteristics. The performance of separate PID controllers on each of the model joints provides an indication of problems that arise in the multi-joint control problem. A brief summary of PID control is outlined below. Fig 7.7 shows a simple closed loop system. The controller is driven by the error signal between actual output and desired output. The structure and parameters of a controller greatly influences the closed loop system performance.

In the continuous time domain the equation for a PID controller is

$$u(t) = K_g \left( e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right) \quad (7.1)$$

where  $K_g$  = gain factor  
 $T_i$  = integral coefficient  
 $T_d$  = differential coefficient

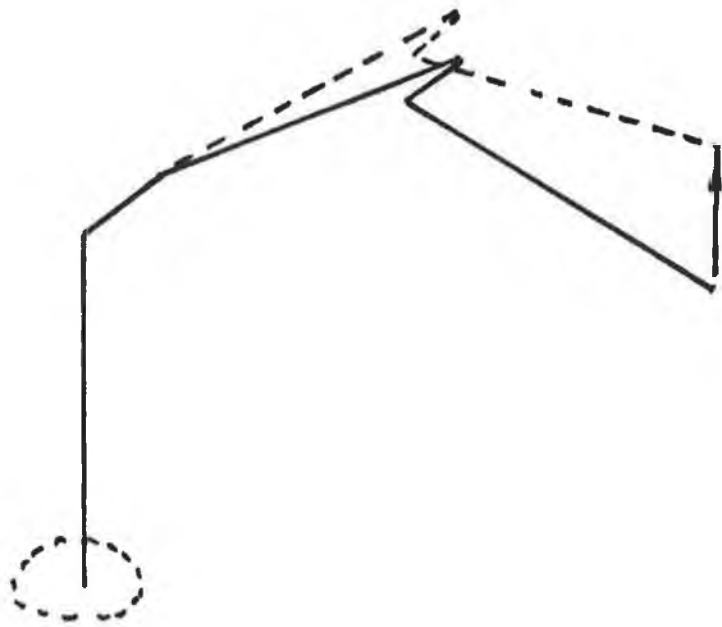


Figure 7.6 Vertical path recorded for openloop test

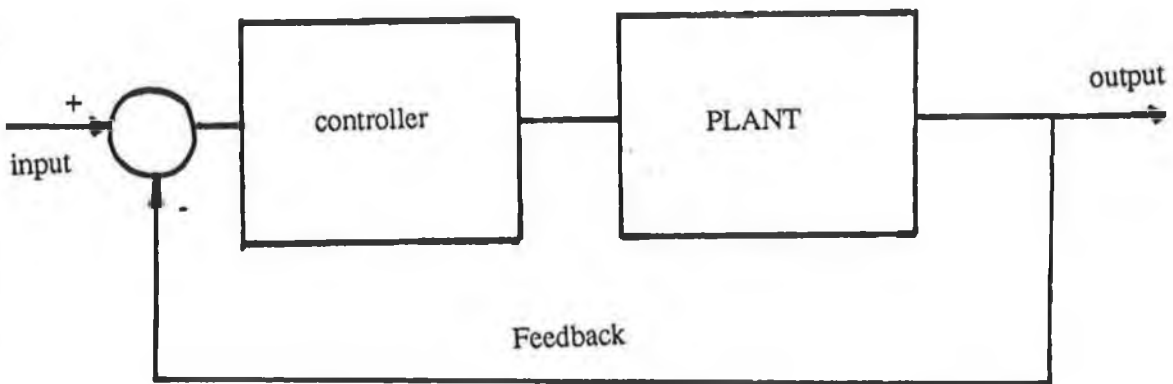


Figure 7.7 Closed loop System

In the complex frequency domain eqn (7.1) can be written

$$G(s) = K_g \left( 1 + T_d s + \frac{1}{T_i s} \right)$$

A discrete algorithm can be used by discretising eqn (7.1) making a simple backward difference approximation of the derivative part and an Euler approximation of the integral part. This provides accurate results when the sampling rate is short compared with the shortest time constant of the control system, The discretized equation is:

$$u(k) = K_g \left( e(k) + \sum_{i=0}^{k-1} e(i-1) + \frac{T_d}{h} (e(k) - e(k-1)) \right) \quad (7.2)$$

$h = \text{sample period}$

In order to save on-line calculation time the recursive version of eqn (7.2) is used as

$$U(k) = U(k-1) + q_0 e(k) + q_1 e(k-1) + q_2 e(k-2) \quad (7.3)$$

This discrete equation corresponds to the transfer function in the z domain

$$G(z) = \frac{U(z)}{e(z)} = \frac{q_0 + q_1 z^{-1} + q_2 z^{-2}}{1 - z^{-1}} \quad (7.4)$$

In this cases the relationship between the discrete and analog coefficients are

$$q_0 = K_g \left( 1 + \frac{T_d}{h} \right) \quad (7.5)$$

$$q_1 = K_g \left( 1 + 2 \frac{T_d}{h} - \frac{h}{T_i} \right) \quad (7.6)$$

$$q_2 = K_g \frac{T_d}{h} \quad (7.7)$$

## 7.6.2 Ziegler Nichols Tuning of Controller Parameters

The Ziegler Nichols ultimate-sensitivity method of determining the  $T_d$ ,  $T_i$  and  $K$  coefficients is now outlined. Firstly a proportional controller is used to control the system. The minimum proportional gain that causes the closed loop system to oscillate is recorded. This gain is referred to as  $K_{max}$ . The period of oscillation,  $T_p$ , is recorded. A suitable (averaged) set of P, I, and D parameters is then given from the Ziegler Nichols settings [21] as shown in Table 7.2.

Control method	Parameter choice		
	$K_g$	$T_i$	$T_d$
P	$0.5 K_{max}$		
PI	$0.45 K_{max}$	$T_p/1.2$	
PID	$0.6 K_{max}$	$T_p/2$	$T_p/8$

Table 7.2 Ziegler Nichols Controller Settings

## 7.7 PID Control of the Robot Dynamic Model

This section examines the closed loop performance of the Puma dynamic model. Single loop positional control is applied to the separate joints while the other joints are locked in position. It is found that small oscillations with decreasing magnitudes occur in the single joint control cases. One case of multi-joint control is presented in the final subsection.

### 7.7.1 PID control of Joint 1

The inertial load experienced by joint one depends upon the position of links two and three. Joint 1 is independent of gravity forces. Since the controller is required to control the manipulator through all reachable points, it must be designed to cater for the worst possible cases. Two load extremes exist for joint one: a) when link two and three are fully extended and the manipulator is

in its vertical poise (see Figure 3.2(a) ), and b) when link two and three are fully extended and the manipulator is in it's horizontal position, (see Figure 3.2(b) ).

Although the controller needs to cater for demands when joints two and/or three are accelerating or travelling at constant velocities, the controller in this section is designed when joints two and three are in locked positions. Values for the PID controller are found for the two extremes described above using Ziegler-Nichols values. This gives an indication of how dependent the PID parameters are on robot position if controller performance is to remain constant.

The Puma simulation was configured in its vertical position with joints two and three locked. The error signal to the controller was measured in radians and the output of the controller in volts. Values of  $K_{max} = 150$  and  $T_p = 0.45$  were found.

Using Table 7.2 values of  $T_d$ ,  $T_i$  and  $K$  were calculated. Plot 7.17 shows the closed loop system response for these values.

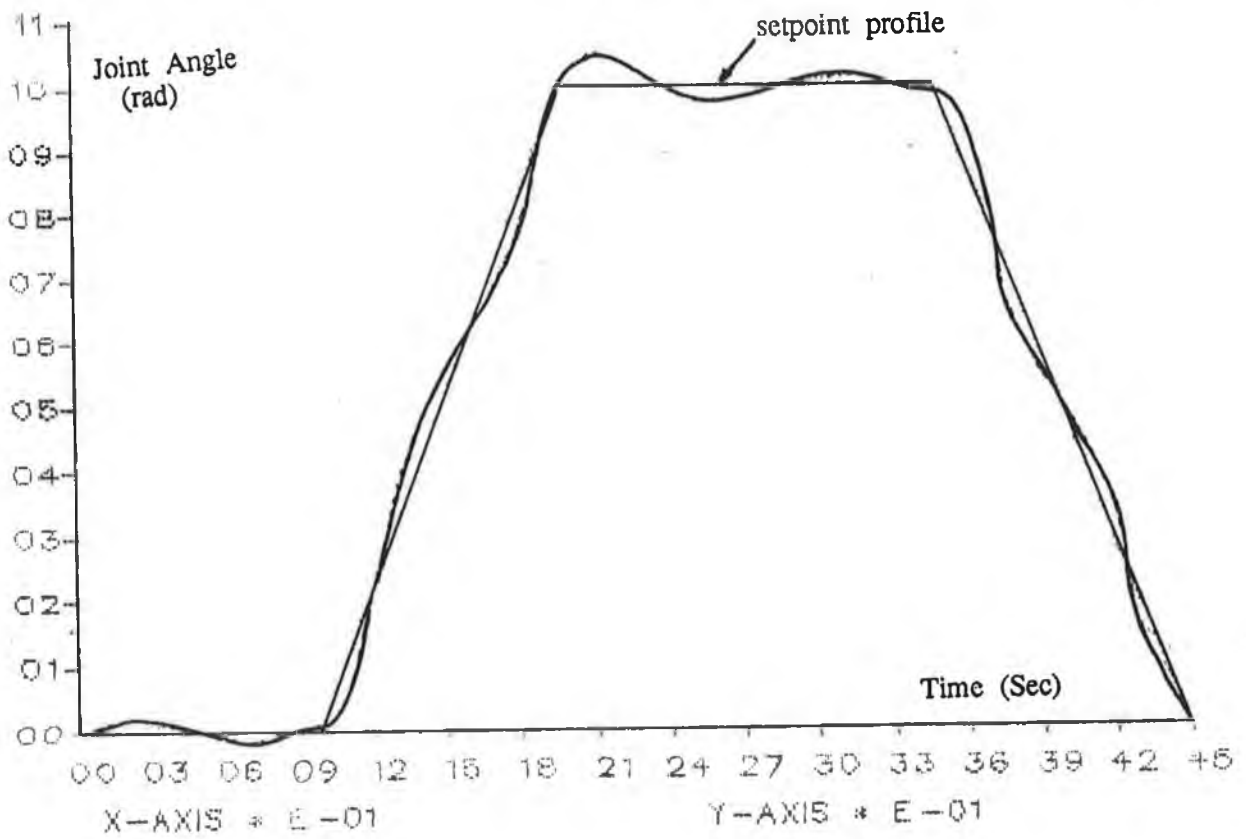
The Puma simulation was configured in it's horizontal extended arm position. In this position  $K_{max} = 220$  and  $T_p = 0.6$ . Plot 7.18 shows the simulated response with these PID parameters.

A certain amount of overshoot is present in both plots. This is removed by employing a smoother path of setpoints thus avoiding sudden changes in joint velocity.

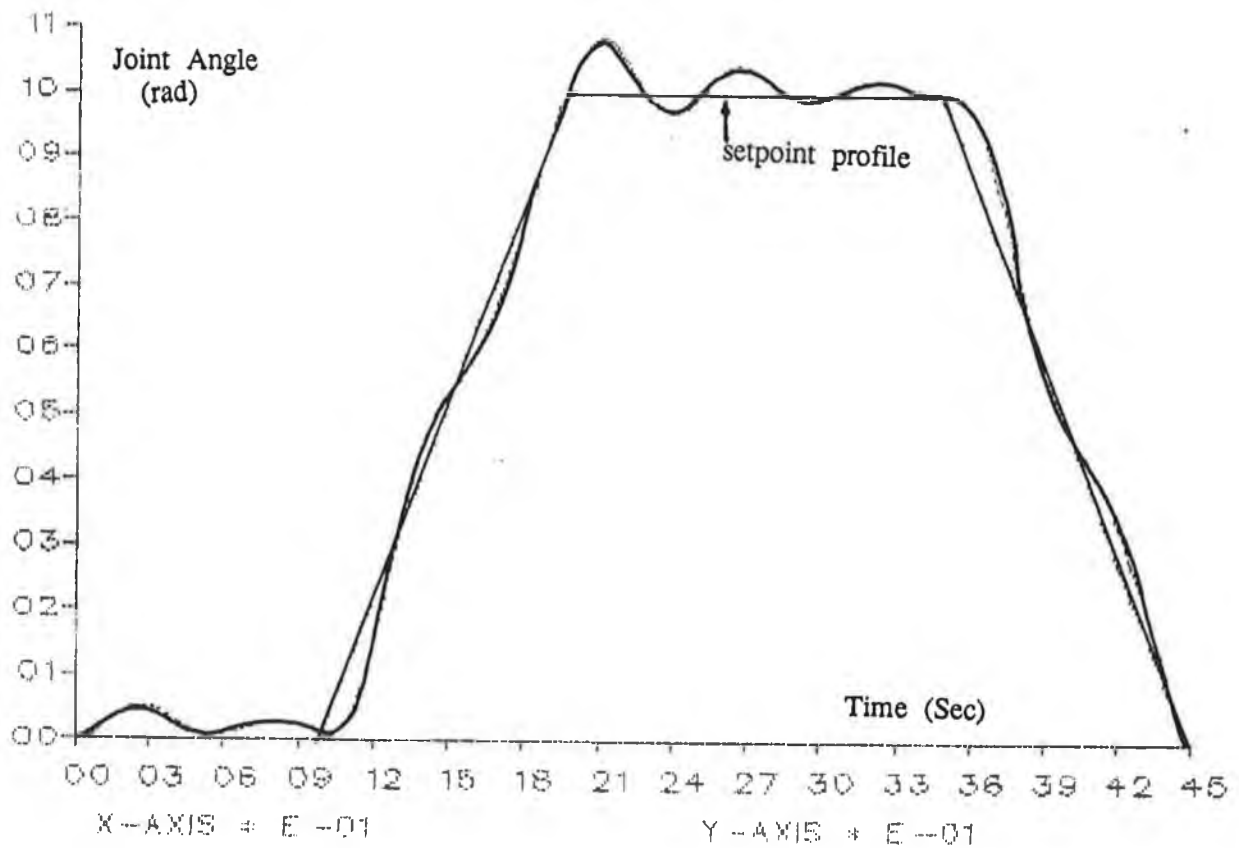
### 7.7.2 PID control of Joint 2

A similar procedure was employed to find controller parameters for joint 2. Links 2 and 3 are aligned thus presenting their maximum load to joint 2. Values of  $K_{max}$  and  $T_p$  were evaluated at setpoints over the complete range of joint 2 positions. The minimum value of  $K_{max}$  and the maximum value of  $T_p$  were chosen to ensure that the controller would remain stable over the full range of positions for joint 2. These values was found,  $K_{max} = 475$ ,  $T_p = 0.4$  which occurred close to the arm vertical position. Plot 7.19 shows the closed loop response to the PID controller chosen according to Table 7.2.





Plot 7.17 Simulated closed loop control of joint 1 using parameters chosen when  $K_{max} = 150$ ,  $T_p = 0.45$



Plot 7.18 Simulated closed loop control of joint 1 using parameters chosen when  $K_{max} = 220$ ,  $T_p = 0.6$

The procedure was repeated with joint three set at an angle of  $-0.9075$  rad (ie folded back against link two thus bringing its centre of mass as close as possible to joint 2).  $K_{\max}$  and  $T_p$  were found to be equal to  $K_{\max} = 385$  and  $T_p = 0.2$ . This occurred when joint two was  $0.17$  radians away from its vertical position. This position corresponded to when the centre of mass of the combined links was directly above the joint axis. PID parameters were chosen according to Table 7.2 and the path shown in Plot 7.20 was followed.

### 7.7.3 PID Control of Joint 3

Link 2 was locked in its vertical position and joint 3 was commanded to move to different positions over its full range.  $K_{\max} = 270$ ,  $T_p = 0.18$  were chosen for most stable control. Plot 7.21 shows the closed loop response when PID values were chosen according to these measurements and Table 7.2.

### 7.7.4 Multi Joint Control of The Dynamic Model

It has been shown that each joint can be adequately controlled by its PID controller if only one joint moves at a time. Essentially this control problem is that of positional control of a motor with a varying load. Simultaneous positional control of the three joints presents a larger problem and is now examined. The PID parameters were chosen as detailed in the previous section. The desired paths for each joint and closed loop joint responses are shown in Plot 7.22.

Joints two and three experience sustained oscillations when attempting to remain in a fixed position. Different setpoints were examined and similar oscillations were present in all cases. The oscillations are the result of coupling between the joints.

## 7.8 Puma 560 Controller Parameter Identification

Test data suitable for the identification of the joint 1 controller board was obtained while the manipulator was executing the repeated motion sketched in Fig 7.8. This motion corresponded to a repeated  $10$  mm end effector movement

in the x direction w.r.t. world coordinates while the robot arm was horizontal and extended fully. Three sets of data were obtained:

- (a) the controller setpoints, in radians
- (b) The joint position, in radians
- (c) Output control signal, in volts

It was necessary to convert the encoder and setpoint inputs to radians and the controller output to volts, so that both the simulation and actual manipulator controllers were operating in the same scales.

One estimate of joint one's controller transfer function was identified using a Recursive Least Squares (RLS) [22] algorithm on the first batch of test data as:

$$G(z) = \frac{6739 + 1430 z^{-1} + 5743 z^{-2}}{1 + 0.96 z^{-1} + 0.106 z^{-2}} \quad (7.9)$$

The PID controller z domain transfer function given in eqn (7.4) has no  $z^{-2}$  term in it's denominator. The RLS identification routine however has identified a  $z^{-2}$  denominator term. The coefficient of this term is approximately 9 times smaller than the  $z^{-1}$  coefficient and is ignored when calculating the PID parameters.

Several sets of data were recorded in the same operating region. Given the sampling period,  $h = 0.93$  mS, equations (7.2) to (7.4) are used to solve simultaneously for the controller PID values. Using the identified transfer function coefficient values from different sets of data about the operating region, PID values were found to be:

$$K_g = 137 \pm 13\%$$

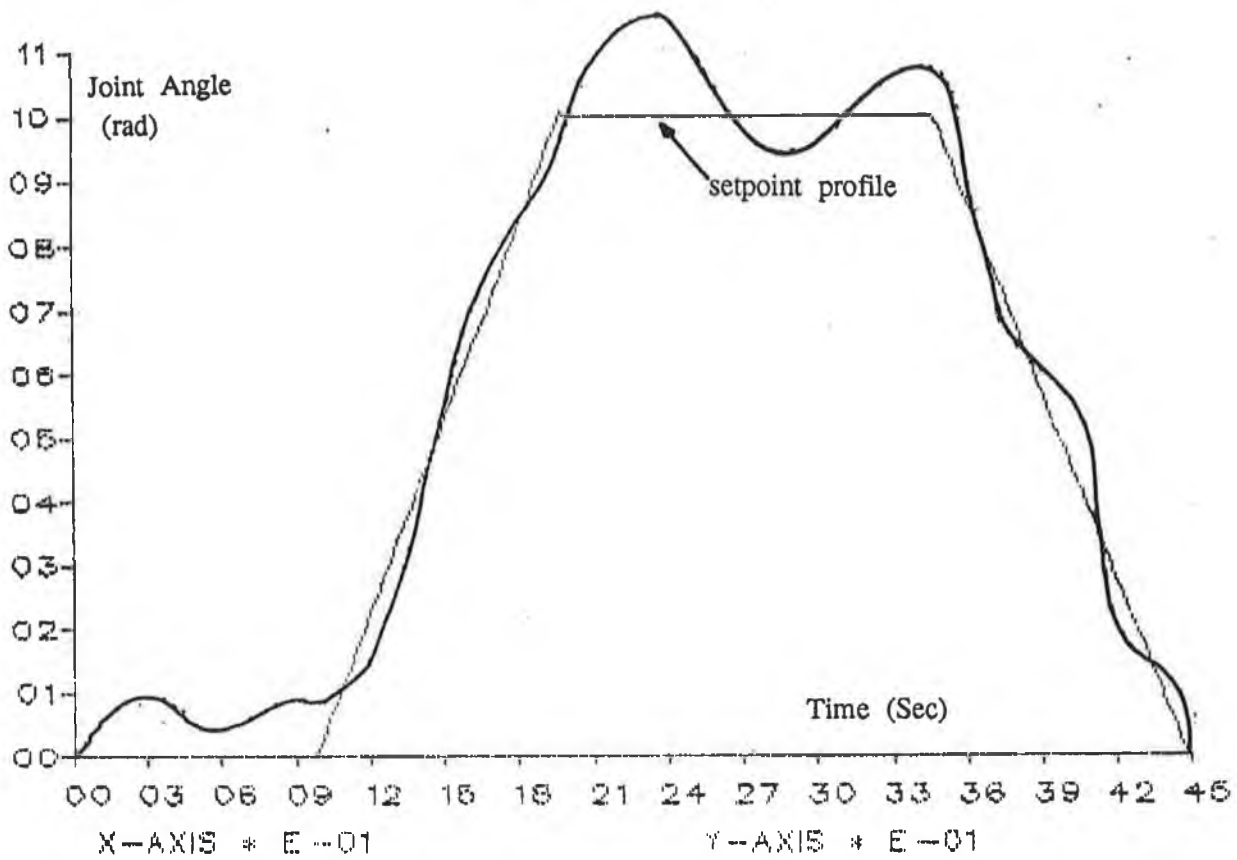
$$T_i = 0.315 \pm 15\%$$

$$T_d = 0.032 \pm 40\%$$

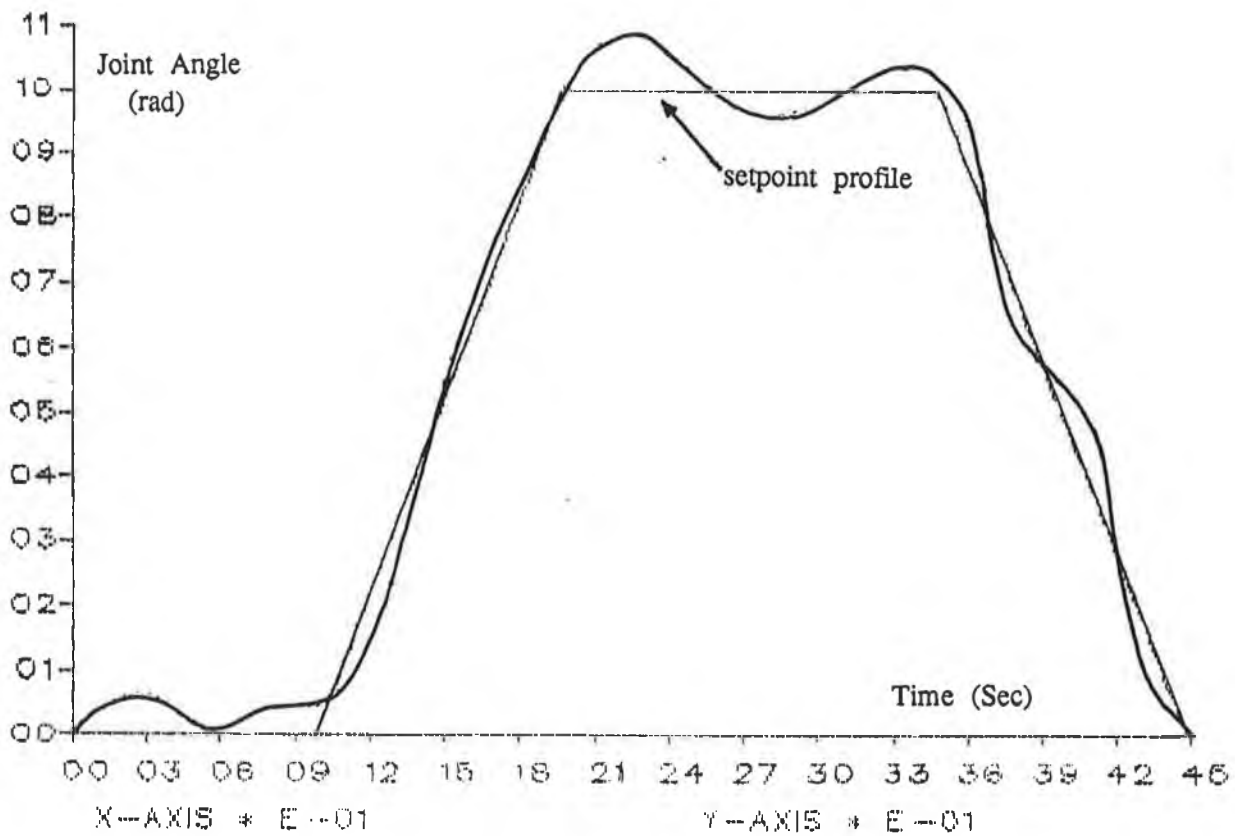
In order to provide a set of test data at a different position in the manipulator operating range, end effector paths were chosen about points close to the



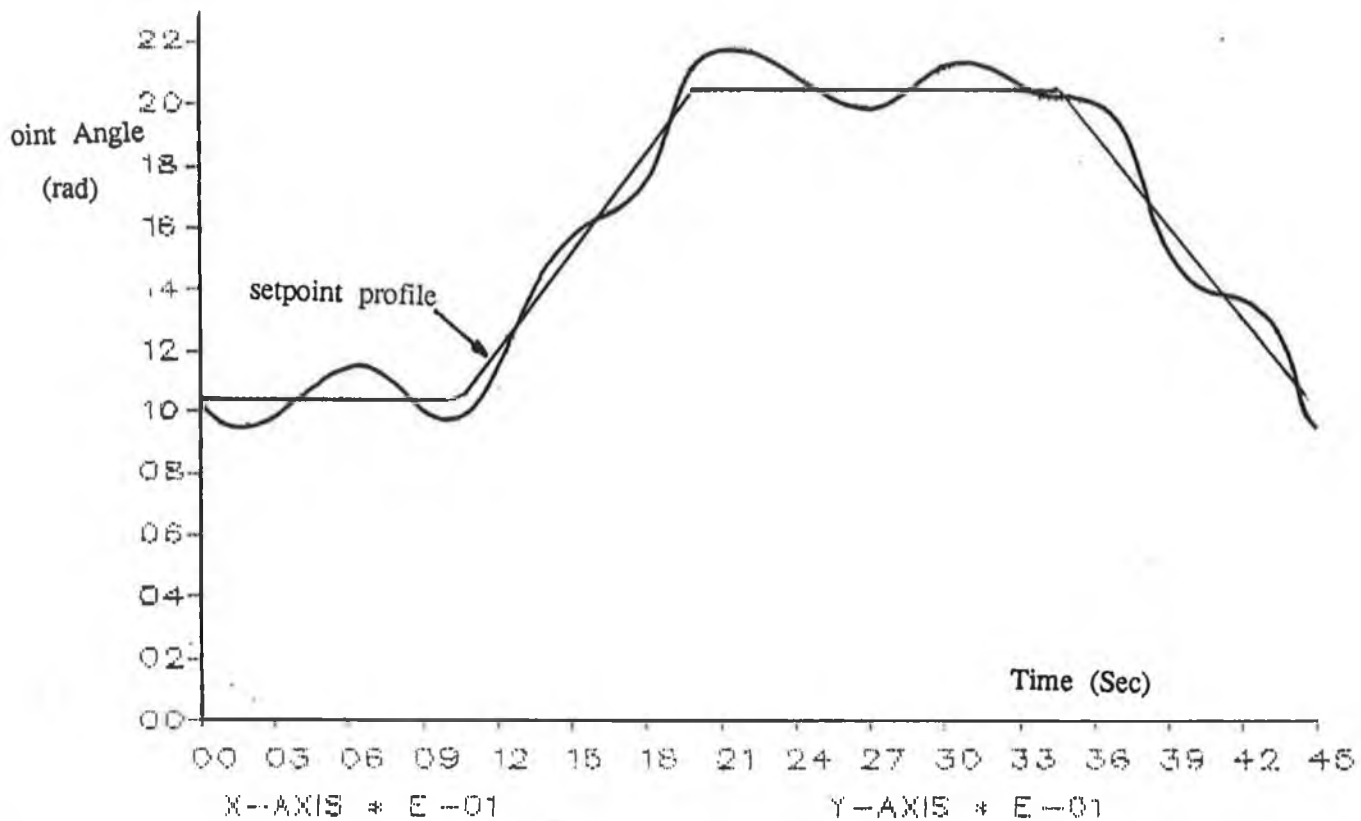
Figure 7.8 Motion for Controller board Parameter Identification



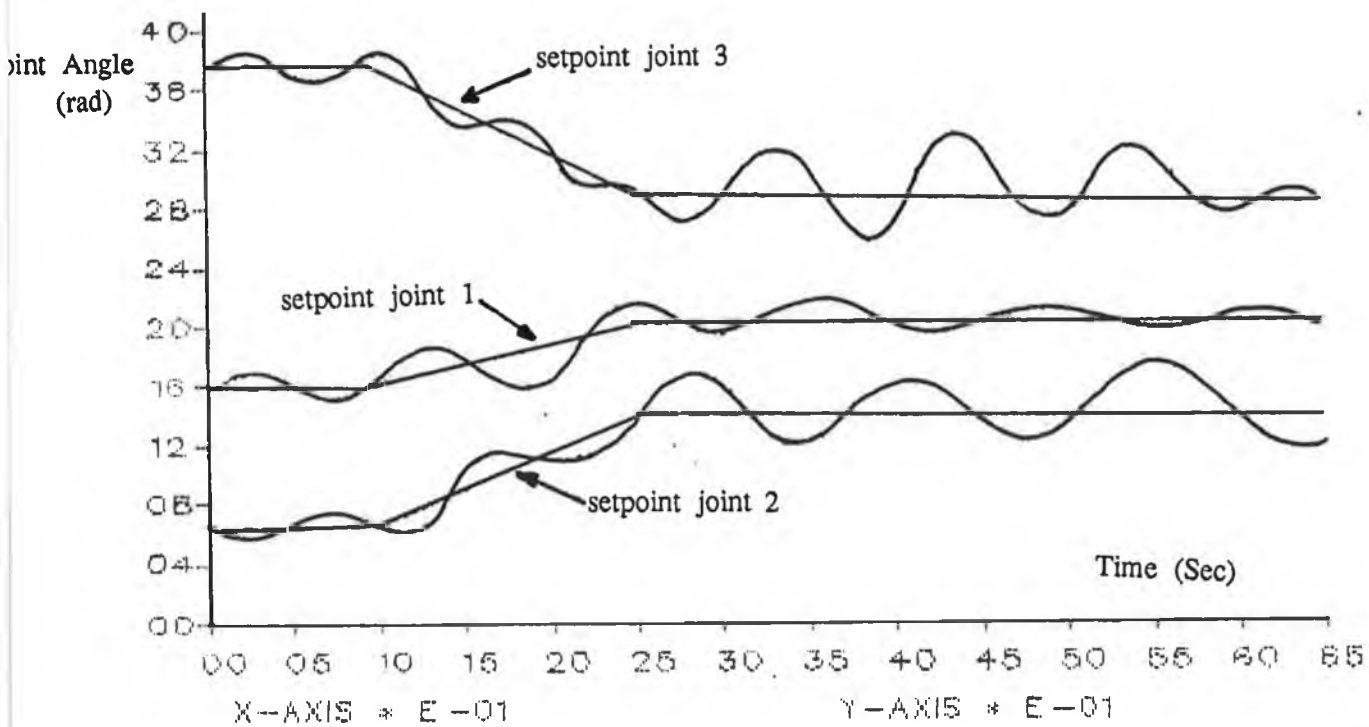
Plot 7.19 Simulated closed loop control of joint 2 using parameters chosen when  $K_{max} = 475$ ,  $T_p = 0.2$



Plot 7.20 Simulated closed loop control of joint 2 using parameters chosen when  $K_{max} = 385$ ,  $T_p = 0.18$



Plot 7.21 Simulated closed loop control of joint 3 using parameters chosen when  $K_{max} = 150$ ,  $T_p = 0.45$



Plot 7.22 Simultaneous control of joints 1, 2 & 3

manipulator's highest reachable point. Using VAL, the end effector was commanded to move in repeated paths in the horizontal plane. Plot 7.24 shows one of the joint movement profiles. Eqn (7.10) gives a typical controller transfer function as estimated using the RLS algorithm on one set of data.

$$G(z) = \frac{8618 + 15106 z^{-1} + 8913 z^{-2}}{1 + 1.13 z^{-1} + 0.0972 z^{-2}} \quad (7.10)$$

Using similar arguments as discussed above values of  $K$ ,  $T_i$  and  $T_d$  were found to be

$$K_g = 176 \pm 45\%$$

$$T_i = 0.42 \pm 27\%$$

$$T_d = 0.04 \pm 50\%$$

These values are different from those of the previous set of tests, indicating that a form of gain scheduling is employed in the Puma 560 manipulator controllers.

### 7.9 Simulation Results of Model Control Using Measured PID Parameters

The closed loop simulation tests on joint one in Section 7.4 used constant PID controller values for the full joint ranges. This section examines whether the performance of the first joint is much improved using the measured controller parameters.

With the arm fully extended joint one was commanded to follow a specified path as shown in Plot 7.25. This plot shows the closed loop responses using both the measured PID parameters and those estimated in Section 7.8. The simulator follows the desired path more accurately when controlled by the measured PID controller.

With the arm in its vertical position joint one was commanded to track the path shown in Plot 7.26. Again both measured and estimated values for the PID controllers were used. Both responses are plotted in Plot 7.26. It is difficult to determine which set of parameters produces the better results. Both controllers are slightly detuned.

## 7.10

### Summary

A complete set of tuned link mechanical parameters was derived by driving the third order dynamic model in open loop. The tuned model joint angles accurately followed measured angle profiles when driven by recorded input voltages.

Using Ziegler Nichols ultimate-sensitivity method PID controllers were designed for each of the simulator primary joints. Controller performances were tested while tracking paths specified by setpoints stored in a data files. The controllers adequately provided positional control for each joint while the other joints were locked. In this case dynamic coupling between the joints was ignored.

It was found that oscillations occurred during simultaneous joint control. Joints two and three both oscillated while attempting to remain at a fixed setpoint. Since oscillations in the single joint control situations were present but decaying while dynamic coupling was not present, sustained oscillations must be due to dynamic coupling between the joints. In order to remove the oscillations several steps can be taken. Firstly the controllers could be tuned better, gain scheduling could be used, as in the case of the Puma 560 and further, the control loops could be designed to provide velocity and torque control. This emphasises the need for more sophisticated controllers.

For the robot arm model PID values were estimated, using Ziegler Nichols values, for two load extremes, when the robot arm was fully extended in the vertical and horizontal positions. The controller parameters resident in the Puma joint one controller board were identified at these load extremes. It was found that the measured and estimated values for the parameters differed by 35% when in the horizontal position and 17% when vertical.

The experimentally measured values for the PID controller parameters were implemented on the software. The closed loop responses for the measured and estimated controllers were compared. It was found that the measured values provided better control. Since the measured PID parameters used on the Puma 560 were dependent upon manipulator configuration, it was deduced that gain scheduling was employed.



## CHAPTER 8

### CONCLUSIONS

The complete procedure for modelling the dynamics of the first three links of a Puma 560 industrial manipulator has been presented. Firstly a second order model was derived using a Euler-Lagrange formulation outlined by Paul. The general form of the second order model has been derived previously, however, it fails to present the full manipulator dynamics, and is therefore of limited value when designing a controller. Even the second order model is never completely specified. The coefficient terms of the second order equations are trigonometrical functions of joint angles, link sizes and mechanical parameters. Mechanical specification of a manipulator links is often limited to the sketches provided in manufacturers service manuals. Measurement of the link mechanical parameters have been made using experimental and analytical techniques. However proof of their accuracy has not been presented. If the accuracy of these parameters could be guaranteed, the second order model still ignores many of the manipulator main dynamics. Actuator dynamics and friction forces must be incorporated to approach some level of realism. The contents of this thesis have satisfied some of the difficulties with robotic modelling. The inclusion of actuator dynamics and friction forces in the second order model, resulted in a third order set of dynamic equations. Nominal values of the link mechanical parameters were estimated based on published values. Simulation of these equations made it possible to reestimate the link parameters using a series of carefully chosen tests.

The open loop response of the third order model was fully tested for voltage inputs recorded from the industrial manipulator when performing different tasks. When tuning was completed the model open loop responses represented the Puma response very accurately. Several conclusions can be drawn from this result.

- (1) That the main dynamics of the Puma 560 industrial manipulator can be described by a third order dynamic model.
- (2) That the first order model used for the d.c. servomotors was adequate.
- (3) That the simulation technique was accurate, and
- (4) That link mechanical parameters were estimated correctly

Once the model had been fully tested in the open loop state, the remainder of the thesis concentrated on closed loop model performance. The obvious first

choice for controlling the manipulator was to use a PID loop for each joint. Ziegler Nichols ultimate-sensitivity method was employed to obtain initial estimates for the PID parameters for each link. The minimum proportional gain which drove each joint unstable ( $K_{max}$ ) depended highly upon the position of the manipulator links. For each link the safest PID values were chosen to ensure stability over its full range of movements. This meant that the controller was highly untuned for much of the joint motion.

In the closed loop tests, the setpoint profiles were stored in data files. Each joint was commanded to follow a ramp motion, where the slope of the ramp represented the maximum velocity demanded by the manual controller. Overshoot was present at the abrupt velocity changes in the setpoint profile, but could be avoided if slow transitions in velocity were specified.

The controllers for each of the three joints were separately capable of path tracking at the maximum speed required by the actual Puma manipulator. Controlling each joint separately, however does not guarantee good overall manipulator control. Even though the single loop controllers would control individual joints in isolation (when the other links were locked in any position) it was found that simultaneous control of the three joints produced sustained oscillations in both path tracking and fixed point control. This was accounted for by the dynamic coupling between joints since these sustained oscillations were not present in the single joint control using the same PID parameters. To remove the oscillations, better tuned controllers for each operating area in the robot work space are required and the use of joint decoupling would also be advantageous.

In order to establish what controller parameters are used in the actual industrial manipulator, the Puma was put through a series of path tracking tests. While the Puma executed these tasks the input/output controller board data was recorded. The data was passed through a recursive least squares identification algorithm and the controller board PID parameters were obtained. It was discovered that the joint controllers were tuned differently for different manipulator joint positions and link configurations, indicating that gain scheduling is employed on the industrial manipulator. The measured PID parameters were used in the simulator controllers and the previous set of path tracking tests were repeated. Improved responses were obtained using the measured PID parameters. Several conclusions may be made from these results.

- (1) Ziegler Nichols can provide an adequate first approximation for manipulator PID control.
- (2) Overshoot may be avoided by using smooth velocity changes.
- (3) Single joint control is relatively simple when dynamic coupling is not present eg. when the other joints are locked.
- (4) Dynamic coupling between links produces stability problems for servo and positional control.
- (5) Decoupling may be required to improve performance.
- (6) The Puma 560 industrial manipulator employs gain scheduling.
- (7) The closed loop simulated model performs well when a tuned PID controller is used.

The project has been successful in its objective to obtain an accurate dynamic model of the Puma 560 industrial manipulator. The model has proved its accuracy in both the open and closed loop situations. Simple PID controllers succeeded to control each joint separately but failed in simultaneous path tracking due to detuned controllers and lack of decoupling between links. The success of this modelling exercise paves the way for the design of more efficient controllers.

## BIBLIOGRAPHY

## BIBLIOGRAPHY

- [1] Nof Shimon ( Edited by ) : Handbook of industrial robots. John Wiley & Sons Inc. 1985
  
- [2] Harris & Billings : Self-tuning and adaptive control: theory and applications 2<sup>nd</sup> edition. Peter Peregrinus Ltd. for IEE, 1985.
  
- [3] Koivo et al : Adaptive linear controller for robotic manipulators. IEEE Transactions on Automatic Control, Vol AC-28 no. 2 Feb 1983
  
- [4] Luh : Anatomy of industrial robots and their controls. IEEE Transactions on Automatic Control, Vol AC-28 no. 2 Feb 1983
  
- [5] Carr S & Anderson G et al : An LQG approach to selftuning with applications to robotics. Robot Control: Theory and Applications. Peter Pergrinus Ltd. for IEE, 1988.
  
- [6] Paul R : Robot manipulators mathematics, programming and control. The MIT Press, 1981
  
- [7] Craig J : Introduction to robotics, mechanics and control. Addison-Wesley Publishing Company, 1986.
  
- [8] Stone H et al : Dynamic modeling of a three DOF robotic Manipulator. IEEE Transactions on Systems Man and Cybernetics, Vol. SMC-14, NO.4 July/Aug 1984.
  
- [9] Leahy et al : Efficient Dynamics for the Puma 600. Robotics and Automation, San F. CA. 1986.
  
- [10] Denavit J. & Hartenberg R : A kinematic notation for lower-pair mechanisms based on matrices. ASME Journal of Applied Mechanics (June 1955),

- [11] Hollerbach J M : A recursive Lagrangian formulation of manipulator dynamics and a comparative study of dynamics formulation complexity. IEEE Transactions on Systems Man and Cybernetics. Vol. SMC-10, No. 11, Nov 1980.
- [12] Silver W : On the equivalence of Lagrangian and Newton-Euler dynamics for manipulators. Int. Journal of Robotic Research, Vol 1 No.2, pp. 60-70
- [13] Goldstein H : Classical mechanics 2<sup>nd</sup> edition. Addison-Wesley Publishing Company, Student Series, 1980.
- [14] Synom K : Mechanics, third edition. Addison-Wesley Publishing Company, 1979
- [15] Paul R. et al : Estimation of inertial parameters of the Puma 560 manipulator. ASME Mech. Vol 7, No.64, pp 17-23
- [16] Bejczy et al : Nonlinear feedback control of a Puma 560 robot arm by computer. Proceedings of 24<sup>th</sup> Conference on Decision and Control, Dec 1985.
- [17] Armstrong et al : The explicit dynamic model and inertial parameters of the Puma 560 arm. Robotics and Automation, San F. CA. 1986.
- [18] Tourassis et al : The inertial characteristics of dynamic robot models. Mechanism and Machine Theory. Vol 20, NO. 1, pp, 45-52, 1985.
- [19] Unimation : 500 Series, Users guide to VAL. A Westinghouse Company, 1983.
- [20] Unimation : Puma 500 Mk 2 electrical and mechanical drawings. A Westinghouse Company, 1983.
- [21] Astrom & Wittenmark : Computer controlled systems theory and design. Prentice-Hall Information and System Sciences Series, 1984.
- [22] Ljung & Söderström : Theory and practice of recursive identification. The MIT Press, 1983

- [23] Sedgewick R : Algorithms. Addison–Wesley Publishing Company, Student Series, 1983
- [24] Bently J. : The principles of measurement systems. Longman Group Ltd. 1983.
- [25] Snyder W : Industrial robots, computer interfacing and control. Prentics–Hall Industrial Robots Series, 1985.
- [26] Sheingold (Editor) : Transducer interfacing handbook. Analog Devices, Inc. 1980
- [27] Bazerghi et al : An kinematic exact model of the Puma 600 manipulator. IEEE Transactions on Systems Man and Cybernetics, Vol. SMC-14, NO. 3 May/June 1984.
- [28] Paul R : Kinematic control equations for simple manipulators. IEEE Transactions on Systems Man and Cybernetics, Vol. SMC-11, NO.6 June 1981.
- [29] Birk J : An overview of the basic research needed to advance the state of knowledge in robotics. IEEE Transactions on Systems Man and Cybernetics, Vol. SMC-11, NO.8 Aug 1981.
- [30] Gruver W et al : Industrial robot programming languages: A comparative evaluation. IEEE Transactions on Systems Man and Cybernetics, Vol. SMC-14, NO.4 July/Aug 1984.
- [31] Burdick J : An algorithm for the generation of efficient manipulator dynamic equations. Robotics and Automation, San F. CA. 1986.
- [32] Chung et al : On the dynamic characteristics of a balanced Puma 760 robot. IEEE Transactions on Systems Man and Cybernetics, Vol.35, NO.2 May 1988.
- [33] Kabuka et al : Microprocessor-based architecture for control of a six joints robot arm. IEEE Transactions on Systems Man and Cybernetics, Vol.35, NO.2 May 1988.

[34] Goldenberg A : An approach to real - time control of robots in task space. application to the control of Puma 560 without VAL. IEEE Transactions on Systems Man and Cybernetics, Vol. 35, NO.2, May 1988.

[35] Lee C & Zeigler : Geometric approach in solving inverse kinematics. IEEE Transactions on Aerospace and Electronic Systems Vol. EAS-20, No. 6, Nov 1984.



## APPENDIX A

## APPENDIX A

### ANALYSIS OF THE PSEUDO INERTIAL MATRIX $J_i$

The matrix  $J_i$  is known as the pseudo inertial matrix for link  $i$ . It has a similar format to the inertial tensor matrix which is well documented in literature Goldstein [6]. In brief however  $J_i$  provides information about the distribution of mass of the link relative to the links coordinate frame. Figure A.1 presents a link and its associated coordinate frame. A point volume  $dv$  is sketched, and the vector,  $r$ , from the frame origin to this volume is shown. When rotating this point volume about the  $x$  axes it will have a differential moment of inertia equal to

$$d I_{xx} = (y^2 + z^2) \rho dv$$

$$\rho = \text{material density and } dm = \rho dv.$$

Similarly the moments of inertia about the  $y$  and  $z$  axes are given as

$$d I_{yy} = (x^2 + z^2) dm$$

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

The total mass moments of inertia about the various axes for the complete link volume may be calculated as

$$I_{xx} = \int_{\text{link } i} (y^2 + z^2) dm$$

$$I_{yy} = \int_{\text{link } i} (x^2 + z^2) dm$$

$$I_{zz} = \int_{\text{link } i} (x^2 + y^2) dm$$

The cross products of inertia for the link are similarly formulated to be

$$I_{xy} = \int_{\text{link } i} xy dm$$

$$I_{xz} = \int_{\text{link } i} xz \, dm$$

$$I_{yz} = \int_{\text{link } i} yz \, dm$$

All link rotations will be described relative to the frame shown, cross products of inertia are thus assumed to be zero, Goldstein [6]. In order to form  $J_i$  for each of the links it is necessary to measure the moments of inertia about each of the principal axes. A further step in relating the pseudo inertial matrix to the inertial values is to notice that

$$\begin{aligned} \int_{\text{link } i} x^2 \, dm &= \frac{1}{2} \int_{\text{link } i} (y^2 - z^2) \, dm + \frac{1}{2} \int_{\text{link } i} (x^2 + z^2) \, dm \\ &\quad + \frac{1}{2} \int_{\text{link } i} (y^2 + x^2) \, dm \\ &= \frac{-I_{ixx} + I_{iyy} + I_{izz}}{2} \end{aligned}$$

Likewise

$$\int_{\text{link } i} y^2 \, dm = \frac{I_{ixx} - I_{iyy} + I_{izz}}{2}$$

and

$$\int_{\text{link } i} z^2 \, dm = \frac{I_{ixx} + I_{iyy} - I_{izz}}{2}$$

If  $\bar{r}_i$  is the centre of mass for link  $i$  and  $\bar{r}_i = (\bar{x}_i \bar{y}_i \bar{z}_i \ 1)$  then

$$\int_{\text{link } i} i_x \, dm = \bar{x}_i m_i$$

$$\int_{\text{link } i} i_y \, dm = \bar{y}_i m_i$$

$$\int_{\text{link } i} i_z \, dm = \bar{z}_i m_i$$

These last three components are the first moments of the link. Using the above information the pseudo inertial matrix  $J_i$  is rewritten in terms of the links inertias as

$$J_i = \begin{bmatrix} \frac{-I_{ixx} + I_{iyy} + I_{izz}}{2} & I_{ixy} & I_{ixz} & m_i \bar{x}_i \\ I_{ixy} & \frac{I_{ixx} - I_{iyy} + I_{izz}}{2} & I_{iyz} & m_i \bar{y}_i \\ I_{ixz} & I_{iyz} & \frac{I_{ixx} + I_{iyy} - I_{izz}}{2} & m_i \bar{z}_i \\ m_i \bar{x}_i & m_i \bar{y}_i & m_i \bar{z}_i & m_i \end{bmatrix}$$

where

- $I_{ixx}$  represents the moment of inertia about the x axis of the link i coordinate frame,
- $I_{iyy}$  represents the moment of inertia about the y axis of the link i coordinate frame,
- $I_{izz}$  represents the moment of inertia about the z axis of the link i coordinate frame i,
- $I_{ixy}$ ,  $I_{ixz}$ ,  $I_{iyy}$ , represent the cross products of inertia for link i, about the various coordinate frame axes,
- $m_i \bar{x}_i$ ,  $m_i \bar{y}_i$ ,  $m_i \bar{z}_i$ , are the first moments of link i.

and  $\bar{x}_i$ ,  $\bar{y}_i$ ,  $\bar{z}_i$  = the x, y, & z components of the link centre of mass w.r.t. link i coordinate frame and where  $m_i$  equals the mass of link i.

The introduction of a new quantity referred to as the radius of gyration gives a clear indication of the link mass distribution. The radius of gyration is the radius such that if all the mass of the link were situated a distance  $k_i^2$  from the axis, its moment of inertia would be  $I_i$ . The radius of gyration is related to the link inertia and the mass by the following equalities:

$$m_p k_{p_{xx}}^2 = I_{p_{xx}}$$

$$m_p k_{p_{yy}}^2 = I_{p_{yy}}$$

$$m_p k_{p_{zz}}^2 = I_{p_{zz}}$$

$$m_p k_{p_{xy}}^2 = I_{p_{xy}}$$

$$m_p k_{p_{yz}}^2 = I_{p_{yz}}$$

$$m_p k_{p_{xz}}^2 = I_{p_{xz}}$$

It is normal to assume that the cross products of inertia, ie. the final three terms, are approximately zero, Goldstein [4].  $J_i$  may thus be rewritten as

$$J_i = \begin{bmatrix} \frac{-m_p k_{p_{xx}}^2 + m_p k_{p_{yy}}^2 + m_p k_{p_{zz}}^2}{2} & 0 & 0 & m_i \bar{x}_i \\ 0 & \frac{m_p k_{p_{xx}}^2 - m_p k_{p_{yy}}^2 + m_p k_{p_{zz}}^2}{2} & 0 & m_i \bar{y}_i \\ 0 & 0 & \frac{m_p k_{p_{xx}}^2 + m_p k_{p_{yy}}^2 - m_p k_{p_{zz}}^2}{2} & m_i \bar{z}_i \\ m_i \bar{x}_i & m_i \bar{y}_i & m_i \bar{z}_i & m_i \end{bmatrix}$$

## APPENDIX B

## APPENDIX B

### INVESTIGATION OF 4<sup>TH</sup> ORDER RUNGE-KUTTA NUMERICAL INTEGRATION ON A SYSTEM WITH A DRIVING INPUT

Fig B.1 shows the simplest numerical integration method. The Euler method approximates the curve  $x = f(t)$  by a polygon whose slope, at each time  $t_r$ , is given by the tangent to the curve  $x = f(t)$  at  $t_r$ . It is a first order method with a truncation error per step of order  $h^2$ . Errors occur because the slope of  $f(t)$  changes over the interval  $h$ . A better approximation of the slope, over the interval, will result in a closer estimate of the function.

The fourth order Runge Kutta integration method provides a closer approximation of the functions slope over each interval by taking a weighted sum of the slopes about each point  $t$ .

For an  $n^{\text{th}}$  order equation written as

$$\dot{x} = f(x,t) \quad \text{B.1}$$

$$x_i = f_i(x_1, x_2 \dots x_n, t) \quad 1 \leq i \leq n \quad \text{B.2}$$

the formula for advancing the solution one step is

$$x_{i,r+1} = x_{i,r} + 1/6 (k_{i,1} + 2k_{i,2} + 2k_{i,3} + k_{i,4}) \quad 1 \leq i \leq n \quad \text{B.3}$$

where  $x_{i,r+1} = x_i(t_{r+1}) = x_i(t_0 + (r+1)h) \quad 1 \leq i \leq n$

$$k_{i,1} = hf_i(x_{1,r}, x_{2,r} \dots x_{n,r}, t_r)$$

$$k_{i,2} = hf_i(x_{1,r} + 0.5k_{1,1}, \dots x_{n,r} + 0.5k_{n,1}, t_{r+0.5})$$

$$k_{i,3} = hf_i(x_{1,r}+0.5k_{1,2}, \dots, x_{n,r}+0.5k_{n,2}, t_{r+0.5})$$

$$k_{i,4} = hf_i(x_{1,r}+0.5k_{1,3}, \dots, x_{n,r}+0.5k_{n,3}, t_{r+1})$$

for an  $n^{\text{th}}$  order system with an external input

$$\dot{x} = f(x,u,t) \quad \text{B.8}$$

$$\dot{x}_i = f(x_1, \dots, x_n, u_i, t) \quad 1 \leq i \leq n \quad \text{B.9}$$

This appendix examines whether the Runge Kutta algorithm needs to be modified when applied to a system with a piecewise continuous input.

In the derivation of the Runge Kutta algorithm the partial derivatives of  $f$  w.r.t. the state variables and time, are evaluated at points in the integration interval. If the system has an external input then the function must also be differentiated with respect to the input. When the input is held constant over the interval the partial derivatives wrt the input will be zero giving no cause for adapting the standard formula. The input will be treated like any other system state and equations B.3 to B.7 may be rewritten as

$$\dot{x}_i = f_i(x_1, x_2, \dots, x_n, u_i, t) \quad 1 \leq i \leq n$$

the formula for advancing the solution one step is

$$x_{i,r+1} = x_{i,r} + 1/6 (k_{i,1} + 2k_{i,2} + 2k_{i,3} + k_{i,4}) \quad 1 \leq i \leq n$$

where  $x_{i,r+1} = x_i(t_{r+1}) = x_i(t_0 + (r+1)h)$

$$k_{i,1} = hf_i(x_{1,r}, x_{2,r}, \dots, x_{n,r}, u_{i,r}, t_r)$$

$$k_{i,2} = hf_i(x_{1,r}+0.5k_{1,1}, \dots, x_{n,r}+0.5k_{n,1}, u_{i,r+0.5}, t_{r+0.5})$$

$$k_{i,3} = hf_i(x_{1,r}+0.5k_{1,2}, \dots, x_{n,r}+0.5k_{n,2}, u_{i,r+0.5}, t_{r+0.5})$$

$$k_{i,4} = hf_i(x_{1,r}+0.5k_{1,3}, \dots, x_{n,r}+0.5k_{n,3}, u_{i,r+1}, t_{r+1})$$



since

$$\frac{\delta f}{\delta t} = \frac{\delta f}{\delta y} \frac{\delta y}{\delta t} + \dots + \frac{\delta f}{\delta u} \frac{\delta u}{\delta t}$$

but

$$\frac{\delta u}{\delta t} = 0 \quad \text{over the interval thus}$$

thus

$$\frac{\delta f}{\delta u} = 0 \quad \text{over the interval}$$