THE UNIVERSITY OF ALBERTA

The Effect of Misalignments on Manipulator Performance

by

Ian W. A. Kermack

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

OF Master of Science

Mechanical Engineering

EDMONTON, ALBERTA

Fall 1986

ISBN 0-315-32323-X

THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR Ian W. A. Kermack

TITLE OF THESIS The Effect of Misalignments on Manipulator Performance

DEGREE FOR WHICH THESIS WAS PRESENTED Master of Science

YEAR THIS DEGREE GRANTED Fall 1986

(SIGNED) ....

PERMANENT ADDRESS:

14316 101 Ave

Edmonton Alberta

T5N-0K6

DATED October 15 1986

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled The Effect of Misalignments on Manipulator Performance submitted by Ian W. A. Kermack in partial fulfilment of the requirements for the degree of Master of Science.

Roger W Toogood
Supervisor

J R Colbourne

A W Lipsett

V G Gourishankar

Date 14 Oct '86

Abstract

This thesis investigates the effect of misalignments on the analysis and response of robot manipulators. The identification and incorporation of misalignments into the kinematic models of manipulators is of fundamental importance to the success of future offline programing techniques.

The kinematics of misaligned manipulators is reviewed and a convention for describing the geometry of misaligned manipulators is developed. This convention is used to investigate the forward and inverse kinematic solutions for misaligned manipulators. The presence of misalignments was found to have little impact on the forward solution, however the inverse solution becomes much more complicated.

Manipulator dynamics is reviewed and an efficient algorithm for the generation of symbolic equations of motion is presented. It was found that the symbolic equations could be evaluated more efficiently than the Hollerbach or Newton-Euler equations. The symbolically generated equations are used to investigate the dynamic response of misaligned manipulators. It was found that misalignments had minimal impact on the manipulator's dynamic response. Therefore the manipulator's dynamic model need not include misalignments.

Acknowledgements

Sincere gratitude is expressed to Dr. Roger Toogood for his energy and enthusiasm as well as his guidance, patience, and unyielding support throughout the study.

Appreciation is extended to the staff in the Department of Mechanical Engineering for two very rewarding years of study and support.

Special thanks are given to my parents for their constant encouragement and respect for higher learning.

My deepest appreciation is extended to my wife Jennifer for her love and support throughout my graduate program.

Finally, the author gratefully acknowledges the financial support provided by the Department of Mechanical Engineering.

Table of Contents

List of Tables

List of Figures

I. Introduction

Automation technology has been applied almost exclusively in very large scale manufacturing systems. These systems produce very large quantities of identical items and therefore the economies of scale have been able to offset the major investment in hard automation equipment. The major drawback of hard automation has been its inflexibility. In order to modify the end product a major investment of time and capital are required to reconfigure the system. Flexible manufacturing systems promise to change this and as a result will bring the economies of automation to smaller scale production.

Flexible manufacturing systems will reduce batch sizes while maintaining automation economy because of the versatility of the machines and the ease and speed with which they can be reprogrammed. Off-line programming will play a major role in simplifying the reprogramming task and is therefore of vital importance to the success of flexible manufacturing.

This thesis deals with a particular problem faced by off-line programming. Off-line programming demands very accurate models of manipulators. Therefore any discrepancies between the actual and theoretical parameters which describe the manipulator will affect the accuracy of any tasks being programmed off-line. The discrepancies between actual and ideal parameters are termed misalignments. This thesis will investigate the effect of misalignments on kinematic and dynamic analysis, and on manipulator performance.

The programming method most commonly used in industry today is lead-through programming. With this method the manipulator is manually led through the desired task, either physically or with a teach pendant, while the joint positions are recorded. Unfortunately this method requires that work processes are suspended while reprogramming is performed. The production downtime involved with such stoppages can become costly, especially in assembly line conditions which may require several processes to be suspended while modifications may be required to only a few manipulators. As a result there has been a recent trend toward off-line programming. It is desired to be able to simulate the tasks to be performed, generate and then download a task program to the manipulator controller. The new task could then be immediately executed with little interruption of the manufacturing process. Clearly off-line programming could result in greatly reducing the downtime required to accomodate process changes.

Off-line programming and higher level control strategies demand very accurate geometric models of both the manipulator and the workspace. Using the Denavit-Hartenberg convention (described further in Chapter II), manipulator geometry can be specified by four parameters (length, twist, offset, and rotation) for each link in the kinematic chain. Commercially available robots have an ideal set of parameters, derived from the original design, which define the ideal geometry. However each individual manipulator may have slightly different geometry as the result of manufacturing errors and tolerances, wear, accidents, or otherwise. These deviations between the ideal parameters and the actual parameters are the source of the misalignments. Because of the increasing need for very accurate mathematical models (especially for off-line programming) there is a growing interest in misalignments, their effects, and identification.

Dubowsky and Maatuk (1985) investigated kinematic errors resulting from misalignments (in planar mechanisms) and described a method of parameter identification. Mooring (1983) showed that the Denavit-Hartenberg transform may become ill-conditioned if very slight joint axis misalignments are present and proposed an alternative transform developed by Suh and Radcliff (1978). Mooring used this new transform to study the effect of misalignments on positioning accuracy and found that very slight misalignments could produce significant positioning errors. Simulating a PUMA 600 manipulator, with both joints two and three misaligned by one degree, Mooring found the position error to be a nonlinear function (of the joint angles) reaching a maximum of 32.56 mm. Clearly, this is unsatisfactory for assembly operations. Mooring proposed an algorithm for parameter identification suggesting that it would be more cost effective to accomodate misalignments through identification and control algorithms than to eliminate them by building robots to extremely high tolerances. Foulley and Kelly (1984) have proposed the use of a single compensation matrix to correct the positioning discrepancy. While this method can correct for positioning errors it does not accurately describe each link and as a result is unsuitable for dynamic analysis.

Manipulator dynamics is playing an important and ever expanding role in the simulation, analysis, and control of modern manipulators. The feedback control systems used today regard dynamic interactions as disturbances which the control system must reject. During low speed motions simple feedback systems can accomodate the relatively small disturbances resulting from dynamic effects. However in higher speed motions the dynamic interactions may dominate the equations of motion and the resulting perturbations may be too large to be rejected by simple feedback. Higher level control systems which incorporate manipulator dynamics are now being pursued and may improve system performance.

> " Preliminary results show that feed forward computation of the inverse dynamics could improve the performance of the feedback control system because perturbations due to dynamic interactions do not have to be rejected." Brady et al. (1982)

Torque-based control systems are also being investigated. Brady et al. (1982) These systems use the equations of motion to generate the nominal joint torques required to drive a manipulator along some specified trajectory. A torque based controller then drives the system along the desired trajectory and position feedback ensures accurate tracking.

Fast efficient algorithms for the generation and evaluation of the equations of motion are extremely important to the implementation of control systems which utilize dynamics and have been pursued actively for the past decade.

Manipulator dynamic theory has evolved from two basic approaches, the Lagrangian formulation based upon Lagranges equation, and the Newton-Euler formulation which is based upon D'Alembert's principle.

The Lagrangian formulation was first developed by Uicker (1965) who derived the Lagrangian and subsequent equations of motion for general spatial linkages. Kahn (see Brady et al. (1982)) then particularized these equations to open loop manipulator linkages. Due to the complexity and computational intensiveness of the Uicker-Kahn equations, simplifications have been made. Bejczy (1974) simplified the equations by neglecting the Coriolis and centripetal terms. This led to a vast reduction in complexity and computation. However the approximation is only valid for very low speed motions in which the gravity terms dominate the equations of motion Paul (1981).

The inefficiency of the Uicker-Kahn equations results largely because each generalized force is evaluated as a separate entity, distinct from all other forces acting on the system. Therefore a lot of needless recomputation is performed each time a force is evaluated. Waters (see Brady et al. (1982)) recognized that the generalized coordinates could be expressed in a form which allowed the Uicker-Kahn equations to be rewritten in a much more efficient, recursive form. Hollerbach (1980) further refined Waters' equations resulting in an efficient, recursive dynamic formulation.

The Newton-Euler equations, based upon Newton's equations and D'Alembert's principle, were first applied to open loop kinematic chains by Hooker and Margulies (1965). Vukobratovic (1982) and Stephanko (1982) then applied these equations to biomechanical structures such as the human body. It has only been very recently that these equations were applied directly to robotic systems by Luh et al. (1981). The Newton-Euler equations, although lacking the elegance and simplicity of the Lagrangian formulations are more computationally efficient. This arises due to the naturally recursive nature of the equations and the use of 3X3 rotation matrices and vector translations rather than the 4X4 homogeneous transformations employed in the Lagrangian formulations. Hollerbach (1980) has shown however, that the recursive Lagrangian formulation employing

3X3 matrices results in significant increases in efficiency as compared to the 4X4 equations. However, the efficiency still does not equal that of the Newton-Euler formulation.

The equations of motion as derived from either the Lagrangian or Newton-Euler formulation are represented by a set of coupled, nonlinear differential equations in matrix form. A numerical approach is most commonly used to evaluate these equations in which the matrix operations are performed at each manipulator state in question. Another approach which has been pursued recently is to generate the equations of motion symbolically. This is done by performing the matrix operations a single time using symbols, as opposed to numeric values for the matrix elements. It is possible that the symbolic expressions generated can be evaluated more quickly than the numerical approach, resulting in a more efficient algorithm. This increase in execution speed results because the looping, testing and logical transfer operations required by the numerical method are eliminated and because multiplications by zero and one are not performed in the symbolic equations. It should be noted however that the efficiency of evaluation depends greatly upon the extent of simplification performed upon the symbolic equations. If little or no simplification is performed the equations of motion may become so large and complex that their evaluation may take much longer than the numerical methods.

Symbolic generation of manipulator equations of motion is not a new idea. Luh and Lin (1981) developed an algorithm for symbolically deriving the equations of motion based upon the Newton-Euler formulation. In order to simplify the equations, similar terms are compared and insignificant terms are neglected. No comparisons between efficiency of symbolic and numerical techniques were performed. More recently Vecchio et al (1982) developed a program for deriving the equations of motion, based upon the Lagrangian formulation, using the symbolic language REDUCE. Unfortunately, no comparisons between symbolic and numerical efficiencies were undertaken by the authors.

In chapter 2 the Denavit-Hartenberg and Mooring conventions are reviewed and their shortcomings discussed. An alternative transform is then presented which overcomes these shortcomings. This new transformation will then be used to investigate the effects of misalignments upon positioning accuracy and dynamic behavior.

Chapter 3 will investigate the kinematic and dynamic analysis of misaligned manipulators. The forward and inverse kinematic solutions will be investigated and numerical techniques for solving the inverse problem will be presented. Manipulator dynamic analysis will then be reviewed.

Chapter 3 will go on to investigate the relative efficiencies of the numerical and symbolic methods. An algorithm for symbolic generation based upon the Lagrangian formulation has been developed and it will be shown that symbolic generation can result in a significantly more efficient method for the evaluation of the equations of motion.

The symbolic equations of motion will then be used to investigate the forward dynamic solution. The forward solution solves for the manipulator joint forces given the kinematic state at any given instant in time. The forward solution can be used during simulation to determine if the desired trajectory can be executed within the alloted time, given that the joint actuators can produce limited forces. The forward solution may also be used in high level control systems which will predict the joint forces required to correct the trajectories in real time.

Chapter 4 will investigate the effects of misalignments on both positioning and orientation accuracy. General expressions for these errors will be derived and the relative magnitudes of errors resulting from misalignments in each parameter will be reviewed. It will be shown that the angular misalignments have the greatest impact on position and orientation accuracy. Furthermore, a general error matrix will be developed which will define the position and orientation errors as functions of the manipulator state.

The effect of misalignments on the joint forces required to traverse a given trajectory will also be investigated in chapter 4. The effects of misalignments on joint forces will be shown to be relatively small. However, these perturbations may affect the response of a manipulator if a control system cannot account for these perturbations. This thesis will not investigate how manipulators actually respond if they are misaligned as this requires investigating how misalignments affect various control systems. This would be a topic of research in itself and as such is beyond the scope of this work.

Chapter 5 will then summarize the results of this investigation.

II. Description of Robot Kinematics

This chapter will deal with the kinematic description of misalignments using homogeneous transformations. The Denavit-Hartenberg convention will be reviewed and the problems arising with misalignments will be discussed. Readers who are unfamiliar with homogeneous transforms are referred to Paul (1981).

A. Denavit-Hartenberg Convention

Normally six parameters, five independent, describe the relative position and orientation between the coordinate frames defining consecutive links. However, the Denavit-Hartenberg convention places restrictions upon the location and orientation of each coordinate frame and as a result only four parameters are required.

The Denavit-Hartenberg convention specifies that the origin of the coordinate system of link n is located at the intersection of the common normal between the axes of joints n and n+1, and the axis of joint n+1 ,as shown in figure II.1. If joint axes n and n+1 intersect, the origin of link n is specified to be at the point of intersection. If the joint axes are parallel the origin is chosen to make the offset (d_n) zero for the next link whoose coordinate frame

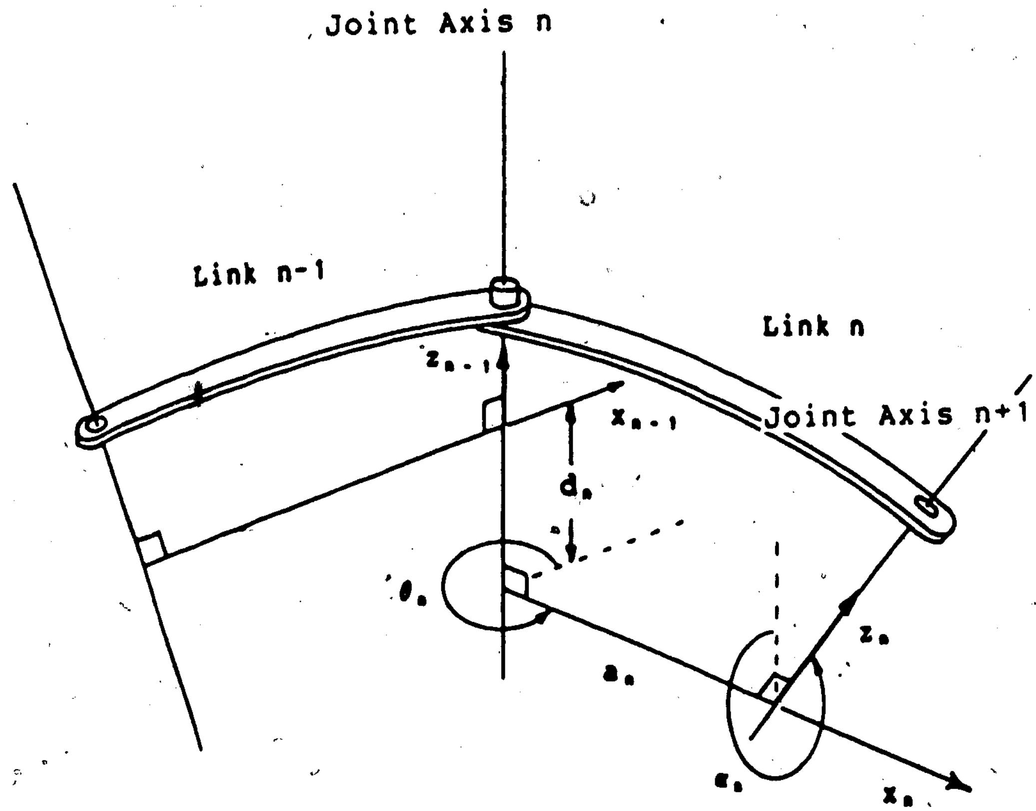

Figure II.1 The Denavit-Hartenberg Convention is defined. The z_n axis is colinear with joint axis n+1. The x_n axis is colinear with the common normal existing between joint axes n and n+1. If the axes intersect, the direction of the x_n axis is parallel or antiparallel to the cross product z_{n-1} X z_n. The rotation is zero when x_{n-1} and x_n are parallel and have the same direction. (Figure II.1)

The Denavit-Hartenberg transformation matrix A_n is the result of the following rotations and translations:

1. rotate about z_{n-1}, an angle θ_n, bringing x_{n-1} to x_n*
2. translate along z_{n-1}, an offset d_n, bringing x_n* to x_n
3. translate along x_n, the length a_n
4. rotate about x_n, the twist angle α_n

i.e. A_n=Rot(z,θ_n)Trans(z,d_n)Trans(x,a_n) Rot(x,α_n)

or, carrying out the matrix multiplications,

$$A = \begin{bmatrix} C\theta & -S\theta\, C\alpha & S\theta\, S\alpha & a\, C\theta \\ S\theta & C\theta\, C\alpha & -C\theta\, S\alpha & a\, S\theta \\ 0 & S\alpha & C\alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where

$$C\theta = Cos(\theta_n)$$

$$S\theta = Sin(\theta_n)$$

$$C\alpha = Cos(\alpha_n)$$

$$S\alpha = Sin(\alpha_n)$$

Note that the subscript n has been omited from the matrix elements for the sake of clarity. This will be consistent throughout the duration of this investigation.

For revolute joints the joint variable is θ with a, α, and d constant. For prismatic joints the variable is d with θ, α, and a constant.

The link coordinate frames are easily defined and conveniently located if the joint axes remain in parallel planes or intersect orthogonally. In these cases the coordinate frames are located at the joint on the link as illustrated in Figure II.2. However if the axes are not parallel in the same plane or intersect at some angle other than 90 degrees, the link coordinate system will not be located on the physical link. Mooring (1983) has shown that if the joint axes intersect and there is some small angle between them the resulting coordinate system may be very far removed from the actual link as illustrated in Figure II.3. The offset can then become a very large number, tending to infinity as the degree of misalignment decreases. As a result, this may cause the A_n matrix to become ill-conditioned, where one element of the matrix is many orders of magnitude larger than the others. Numerical difficulties then arise. This is particularly evident when the inverse of A_n is required. This is a disadvantage of the Denavit-Hartenberg convention when dealing with misalignments.

During simulations of manipulators and their trajectories, the location of each link, as well as the endpoint, is often of concern. This is especially true if the manipulator is to operate in a confined workspace where obstacles must be avoided. When using the Denavit-Hartenberg convention to describe manipulators,coordinate frames may be far removed from the physical link. As a result another transform, which locates the physical endpoint of the link within the link's coordinate system, is required to locate the endpoint in space. It would be much more convenient if the link coordinate system always represented the endpoint of the link itself.

It is apparent that the Denavit-Hartenberg convention is not a satisfactory method for describing manipulators which have misalignments present. New conventions are required which alleviate the problems associated with ill conditioning and coordinate frame dislocation.

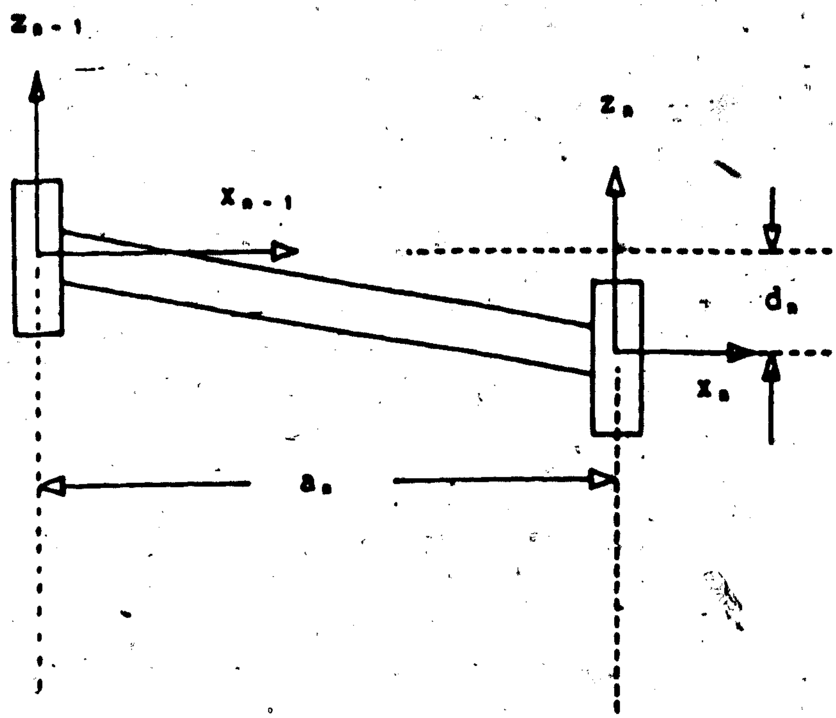

Figure II.2 Typical Ideal Link Between Parallel Revolute Axes

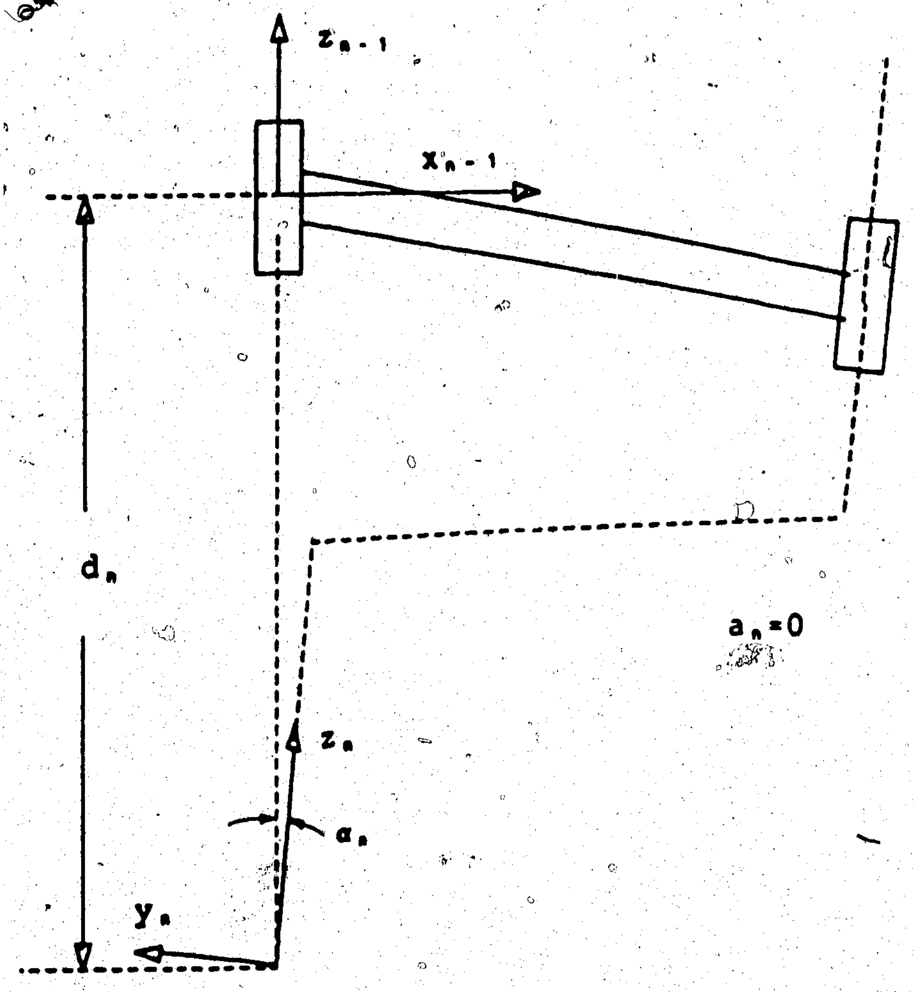

Figure II.3 The Effect of a Slight Misalignment on Coordinate System Location

B. Moorings Convention

Mooring (1983) overcame the problems, resulting from misalignments, associated with the Denavit-Hartenberg convention by adopting a transformation developed by Suh and Radcliff (1978). This transformation utilizes a unit vector which describes the direction of the rotation axis (z_n), a rotation angle (ϕ_n) descibing the relative motion between links, and a point (p_n) through which the rotation axis passes as shown in Figure II.4. Since there is no constraint on coordinate system location, each coordinate frame may be located coincident with each joint. As a result the transform is well conditioned and the problems associated with dislocated coordinate frames are alleviated.

Moorings convention is:

$$D = \begin{bmatrix} u_x{}^2V+C & u_xu_yV-u_zS & u_xu_zV+u_yS & D_{14} \\ u_xu_yV+u_zS & u_y{}^2V+C & u_yu_zV-u_xS & D_{24} \\ u_xu_zV-u_yS & u_yu_zV+u_xS & u_z{}^2V+C & D_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where u_x, u_y, u_z are the components of unit vector $\underline{u}$ which defines the orientation of the joint axis. The joint rotation angle ϕ_n descibes the relative rotation between the successive links, and p_x, p_y, p_z define the position of the joint axis. The z axis of the associated coordinate frame is coincident with the joint axis and the x axis is coincident with the vector p.

D_{14},D_{24},D_{34} are given by:

1. $D_{14}=(1-D_{11})*p_x-D_{12}*p_y-D_{13}*p_z$
2. $D_{24}=-D_{21}*p_x+(1-D_{22})*p_y-D_{23}*p_z$
3. $D_{34}=-D_{31}*p_x-D_{32}*p_y+(1-D_{33})*p_z$

and

1. $V=vers(\phi_n)= 1-cos(\phi_n)$
2. $C=cos(\phi_n)$
3. $S=sin(\phi_n)$

There are two major problems with Moorings convention which limit its applicability in manipulator kinematics and dynamics. The first problem is one of universality: Moorings convention only applies to revolute joints, unlike the Denavit-Hartenberg convention which may be used for both revolute and prismatic joints. The ability to analyze misalignments in both revolute and prismatic joints is clearly necessary.

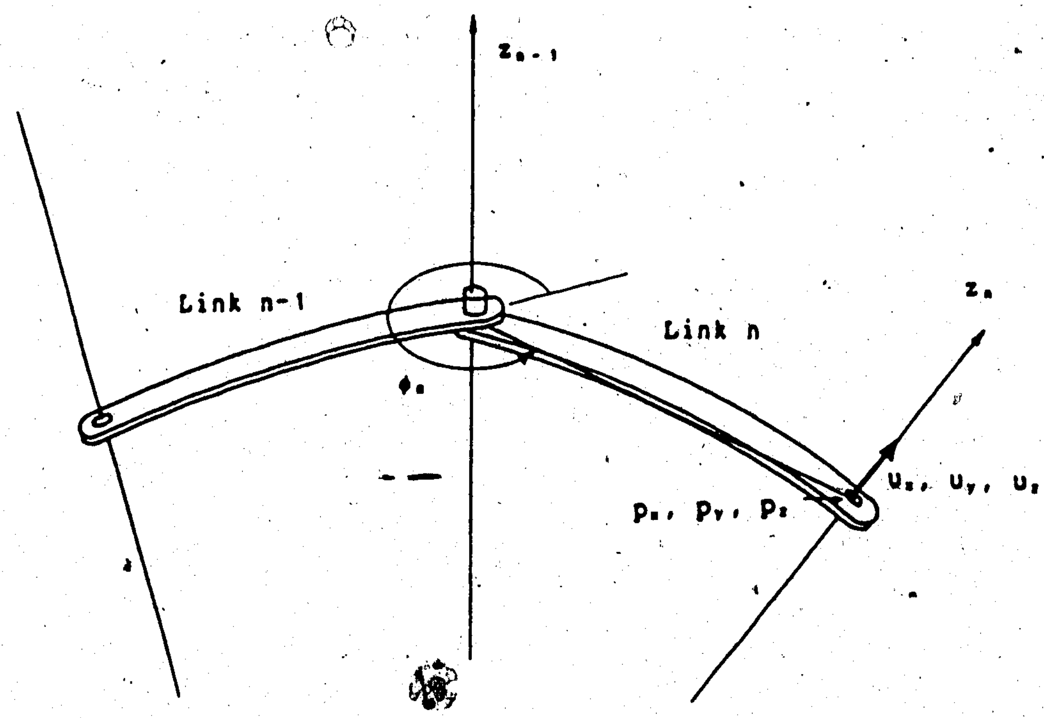

Figure II.4 Moorings Convention

The other problem arises from the use of partial derivatives of the transform matrices in the development of the equations of motion. The partial derivatives of the Denavit-Hartenberg A matrix with respect to the joint variable theta is

$$\frac{\partial A}{\partial \theta} = \begin{bmatrix} -S\theta & -C\theta\, C\alpha & C\theta\, C\alpha & -a\, S\theta \\ C\theta & -S\theta\, C\alpha & S\theta\, S\alpha & a\, C\theta \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

This results in a very sparse matrix and this in turn simplifies the resulting equations of motion. The Mooring convention however, results in a more complicated partial derivative matrix:

$$\frac{\partial D}{\partial \phi} = \begin{bmatrix} u_x^2 S - S & u_x u_y S - u_z C & u_x u_z S + u_y C & D'_{14} \\ u_x u_y S - u_z C & u_y^2 S - S & u_y u_z S - u_x C & D'_{24} \\ u_x u_z S - u_y C & u_y u_z S + u_x C & u_z^2 S - S & D'_{34} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

where:

$$D_{14}=(1-D_{11})p_x-D_{12}p_y-D_{13}p_z$$

$$D_{24}=-D_{21}p_x+(1-D_{22})p_y-D_{23}p_z$$

$$D_{34}=-D_{34}p_x-D_{32}p_y+(1-D_{33})p_z$$

This results in equivalent but more complex equations. If the equations of motion are evaluated numerically this added complexity does not present a large problem. However, if it is desired to generate the equations of motion symbolically, simplicity of equations is of utmost importance. It will be shown (Chapter 3) that evaluating the symbolic equations is much faster than numerical methods and therefore symbolic generation is desirable. In view of these problems a new convention is required.

C. Modified Denavit-Hartenberg Convention

A modified Denavit-Hartenberg transform is proposed which alleviates the problems, resulting from misalignments, associated with the Denavit-Hartenberg and Mooring conventions. The modified transform allows the link coordinate frame to be located coincident with the joint through the use of an additional link parameter termed the skew angle (γ_n). Subsequently five parameters are now required to define each coordinate system. It will be seen that in the ideal case γ_n is zero and the modified transform becomes identical to the Denavit-Hartenberg transform.

Figure II.5 illustrates the modified convention. The link parameters length (a_n), offset (d_n), and rotation (θ_n) specify the location of the coordinate frame while the parameters twist (α_n) and skew (γ_n) define the orientation. The modified transform, D, is obtained by specifying the origin of the coordinate frame at a desired location on joint n+1 and proceeding with the following transformations such that the z_n axis is colinear with the joint n+1 axis:

1. rotate about the z_{n-1} axis the rotation (θ_n) clockwise such that the transformed x-z plane passes through the origin
2. translate along the transformed z axis the offset (d_n) such that the transformed x axis passes through the origin

3. translate along the transformed x axis the length (a_n) such that the coordinate system is coincident with the defined origin
4. rotate about the transformed x axis the twist angle (α_n)
5. rotate about the transformed y axis the skew angle (γ_n)

The resulting modified transform, D, given by

$$D = Rot(z,\theta_n)Tran(z,d_n)Tran(x,a_n)Rot(x,\alpha_n)Rot(y,\gamma_n)$$

$$= A\ Rot(y,\gamma_n)$$

is:

$$D = \begin{bmatrix} C\theta C\gamma - S\theta S\alpha S\gamma & -S\theta C\alpha & C\theta S\gamma + S\theta S\alpha C\gamma & a\ C\theta \\ S\theta C\gamma + C\theta S\alpha S\gamma & C\theta C\alpha & S\theta S\gamma - C\theta S\alpha C\gamma & a\ S\theta \\ -C\alpha S\gamma & S\alpha & C\alpha C\gamma & d \\ 0 & 0 & 0 & \end{bmatrix}$$

where:

$S\gamma = Sin(\gamma_n)$

$C\gamma = Cos(\gamma_n)$

Note that if the skew (γ_n) is zero the modified transform, D, reduces to the Denavit-Hartenberg form.

Figures II.3 and II.6 illustrate two links which have intersecting joint axes. The link in Figure II.3 is defined by the Denavit-Hartenberg convention and the associated coordinate frame is dislocated from the link end point. The link in Figure II.6 is defined by the modified convention and the resulting link coordinate frame is conveniently located at the n+1 joint centre. The Denavit-Hartenberg convention results in a very large offset and a link length of zero. The modified convention alternatively specifies the length and offset which represent the actual link.

As with the original Denavit-Hartenberg convention, and unlike Moorings, prismatic joints can be accomodated with the modified convention.

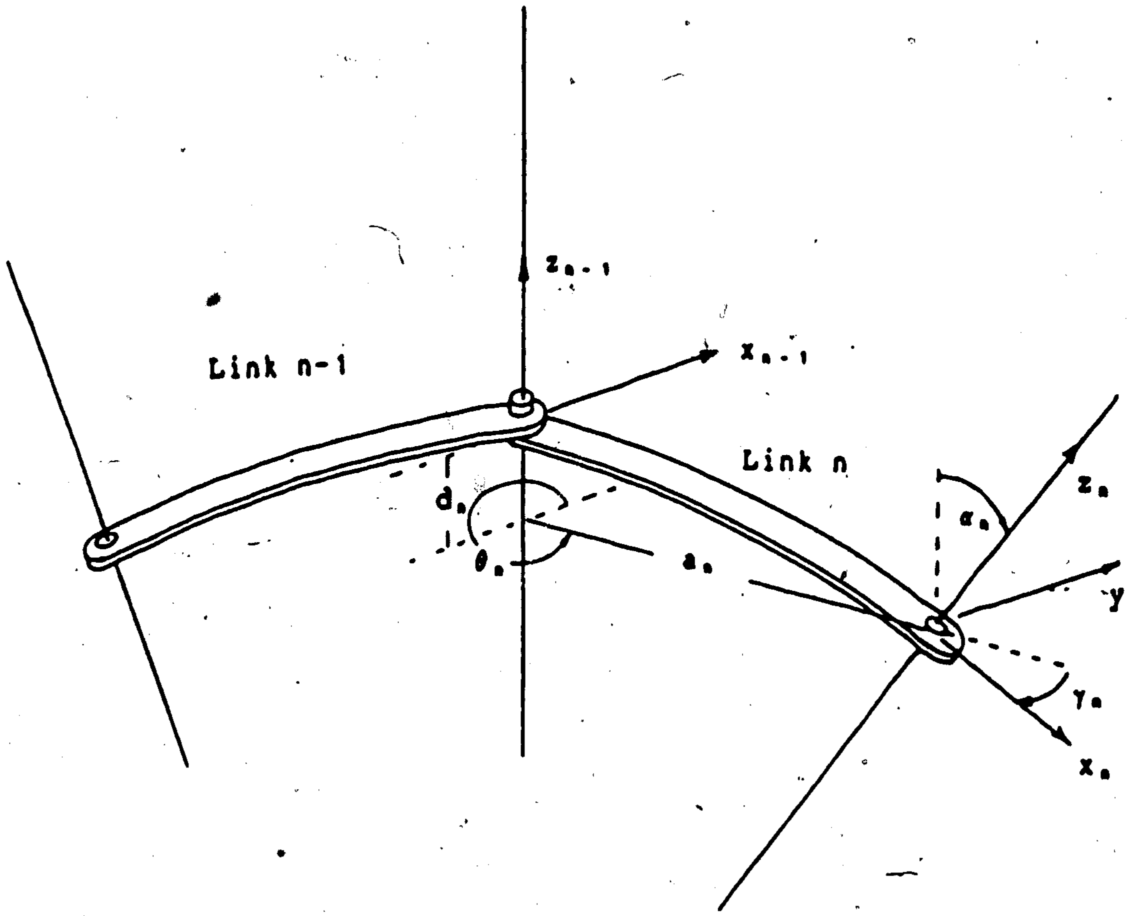

Figure II.5 Modified Denavit-Hartenberg Convention

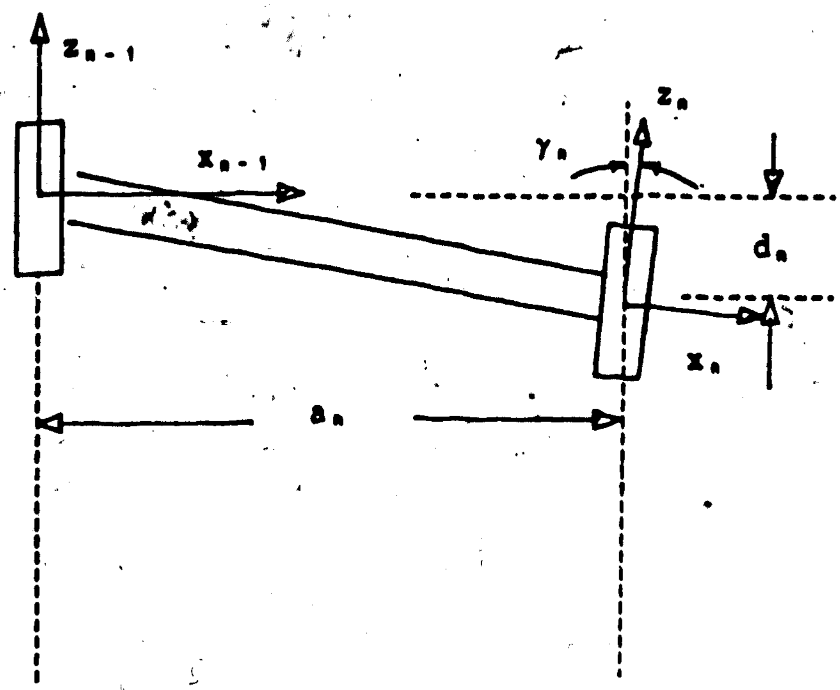

Figure II.6 A Typical Link With Intersecting Joint Axes Defined Using The Modified Denavit-Hartenberg Convention

The partial derivative matrix of the modified transform, while more complicated than the unmodified version, is less complicated than Moorings convention.

$$\frac{\partial D}{\partial \theta} = \begin{bmatrix} -S\theta C\gamma - C\theta S\alpha S\gamma & C\theta C\alpha & -S\theta S\alpha + C\theta S\alpha C\gamma & -a\ S\theta \\ C\theta C\gamma - S\theta S\alpha S\gamma & -S\theta C\alpha & C\theta S\alpha + S\theta S\alpha C\gamma & a\ C\theta \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

This modified Denavit-Hartenberg convention will be used in the following chapters for the kinematic and dynamic analysis of manipulators with misaligned joints.

III. Kinematic and Dynamic Theory and Analysis

A. Kinematics

Kinematic analysis is concerned with the relation between the joint positions and velocities and the location, orientation, and velocity of each link. The forward kinematic solution is concerned with solving for the location and orientation of any link given that the joint coordinates are specified. This problem is solved very neatly by multiplying successive transforms together such that the T_n matrix is obtained. The location and orientation of the nth coordinate frame are then specified by the elements of T_n.

The inverse kinematic problem is that of solving for the joint coordinates given that the desired endpoint and oriention of the manipulator are known. Paul (1981) presents a very effective method for solving the inverse problem which usually results in explicit equations defining the joint coordinates. The elements of the T_n matrix are known from the desired position and orientation of the endpoint. T_n is also known to be equal to the product of the A matrices for each of the n links:

$$T_n = A_1 A_2 --- A_n$$

Subsequently n matrix equations can be written by successively pre-multiplying the above equation by the A matrix inverses. The matrix elements on the left hand side of the ith equation are functions of the elements of T_n and the first i joint variables. The matrix elements on the right hand side of the ith equation are either zero, constants, or functions of the ith+1 through nth joint variables. Twelve algebraic equations result from each of the n matrix equations. From these 12*n algebraic equations, explicit expressions defining the joint coordinates may be obtained. For more detailed discussions and worked examples see Paul (1981).

The Inverse Kinematic Solution For Misaligned Manipulators

The presence of misalignments and the implementation of the modified Denavit-Hartenberg transform, D, presents no added difficulties to the forward kinematic solution, assuming the misalignments are known.

The inverse solution however, becomes much more complicated, and explicit analytical expressions may be impossible to obtain. This is due to the fact that highly coupled, non-linear equations result from using the method outlined by Paul (1981). In lieu of explicit and independent expressions, numerical techniques may be used to solve the coupled equations for the joint coordinates.

An attempt to solve the problem for a three degree of freedom manipulator (as shown in Figure III.1) was made using the Gauss-Seidel method for non-linear algebraic equations. However, more than one solution for the joint coordinates exists for every endpoint. As a result the solution often jumped intervals and would converge on the unwanted solution. This problem was most apparent in regions close to any of the global coordinate axes. Under-relaxation was tried in an attempt to reduce the problem however there was little improvement. When using the Gauss-Seidel method it is sometimes helpful to rearrange the order in which the equations are iterated. However none of the rearrangements tried alleviated the problems. It is possible that with the proper set of equations and order of iteration the

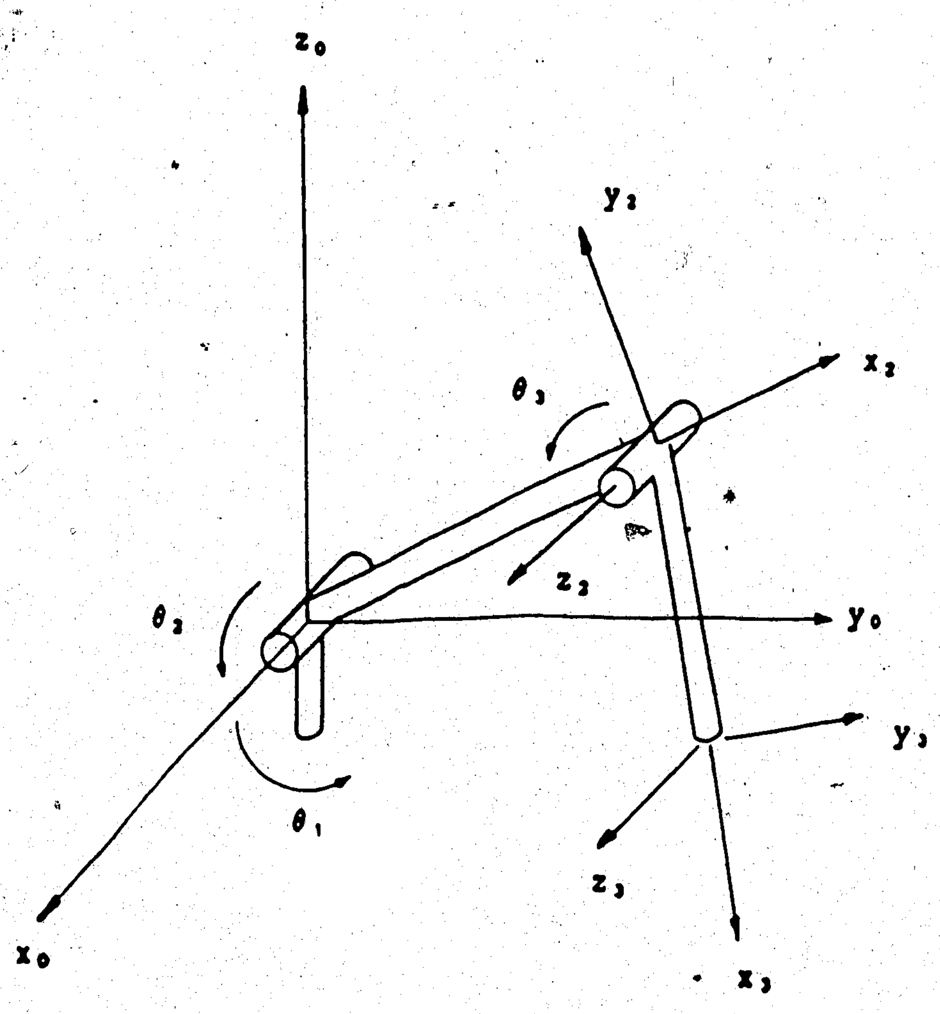

Figure III.1 Three Degree of Freedom Manipulator Used in the Analysis of Misalignments

Gauss-Seidel method could be a satisfactory method of solving the inverse problem. However finding the combination could prove very difficult especially as the number of degrees of freedom increases.

The elements of the T_n matrix which specify the position and orientation of the manipulator endpoint are non-linear functions of the n joint coordinates. Therefore twelve nonlinear equations exist which define the position and orientation as functions of the n joint variables. Newton-Raphson iteration was used in an attempt to solve the inverse problem by solving these equations for a three degree of freedom manipulator. Severe oscillation problems occured when trying to solve the equations in regions near any of the coordinate axes. Under-relaxation was provided in an attempt to reduce the oscillation, however the amount of relaxation required to provide stability resulted in excessively slow convergence. As a result this method was inefficient and unsuitable for solving the inverse problem.

The method which did prove successful in solving the inverse problem for the three degree of freedom manipulator over all regions except in the immediate vicinity of the z axis was a simple numerical technique. In this method an initial guess at the solution was obtained from the inverse equations for an ideal manipulator. Each joint coordinate was incremented by plus or minus a small deviation and that combination of new coordinates which produced a new endpoint closest to the desired solution was used as the new guess for the next iteration. This method proved to be surprisingly efficient, generally requiring fewer than sixteen iterations to obtain convergence within 0.001 % of the desired endpoint.

Solutions to the inverse problem in regions close to the global z axis could not be obtained by any of the numerical techniques attempted. This is due to the degeneracy of the global z axis. It can be seen in Figure III.2 that an infinite set of solutions exist for any endpoint along the z axis. Therefore difficulties in obtaining a solution by any numerical method should be expected in this region.

The inverse kinematic problem for misaligned manipulators is a complicated problem which warrants further study. A numerical technique has been devised which solves the problem over all regions other than the z axis, however other techniques such as gradient search techniques may also be effective. As the number of degrees of freedom increases the exhaustive search method will become less efficient due to the rapidly increasing number of combinations arising from perturbing each joint slightly.

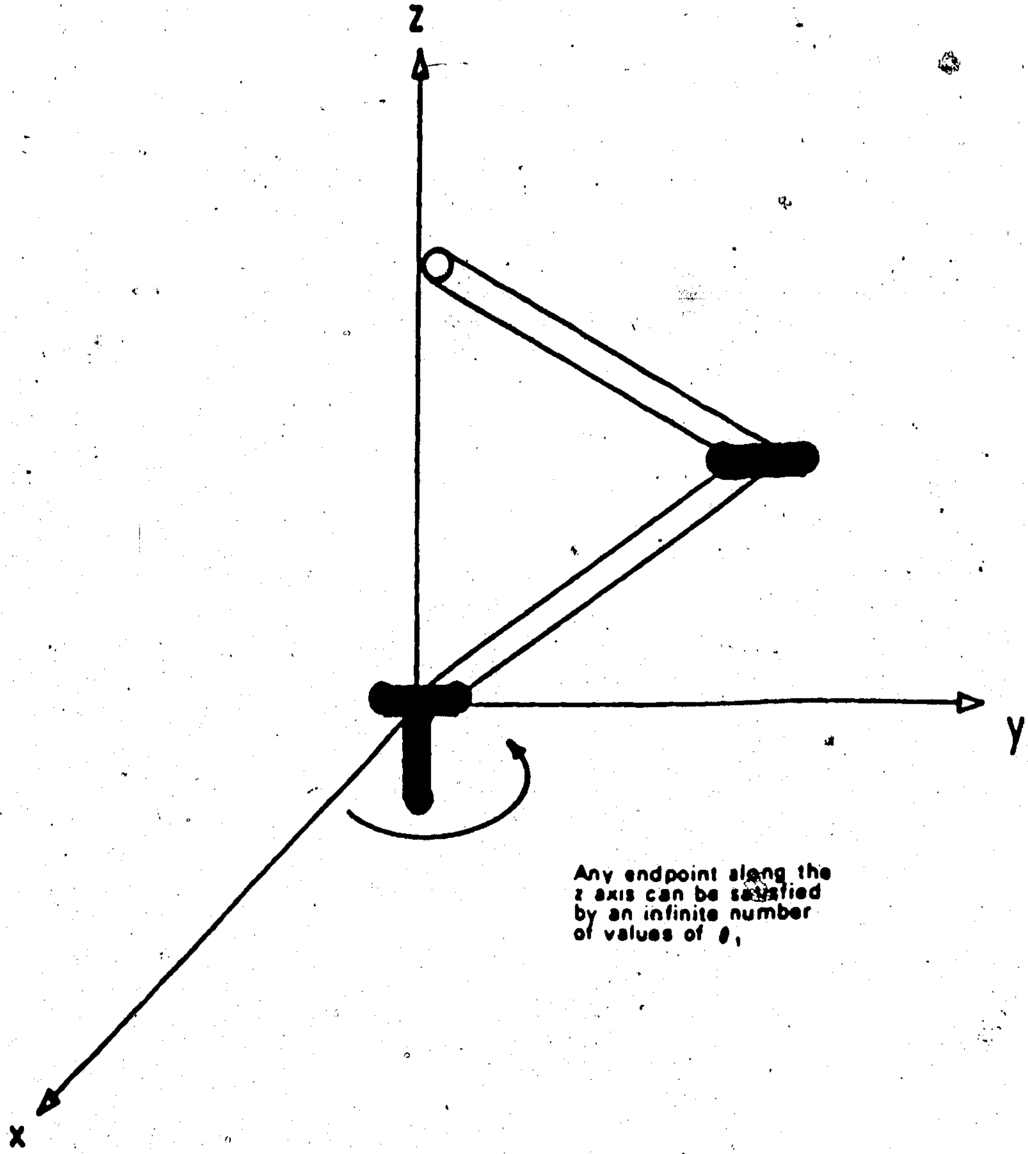

Figure III.2 The z-axis Degeneracy

B. Dynamics

The past two decades have witnessed a great deal of research aimed at improving the efficiency of dynamics algorithms. Table III.1 compares the relative efficiencies of the four most common formulations. The Newton-Euler formulation is the most efficient, requiring approximately one fifth as many addition and multiplication steps as the 4X4 Hollerbach equations. When compared to the Uicker-Kahn equations the Newton-Euler formulation requires about one seventieth as many computational steps. This represents a drastic improvement in efficiency.

The speed and efficiency of the evaluation of the equations of motion is of paramount importance in manipulator control and simulation algorithms. In order to incorporate manipulator dynamics into control algorithms, it is essential that the equations be evaluated very rapidly. Paul (1981) indicates that these equations must be evaluated at least sixty times per second. This is an ambitious task but one which the Newton-Euler formulation has been able to accomplish. Despite the efficiency of the Newton-Euler equations, the reduction of computational requirements remains a research objective.

The computational efficiency of the 4X4 Hollerbach equations can be improved by approximately fifty percent by reformulating the equations employing 3X3 rotational transforms and vector translations. This increase in

Table III.1 Computational Complexity of Various Dynamic Formulations [Brady et al. (1982)]

		Computational Complexity	
		For n Degrees of Freedom	6 degrees of freedom
Uicker-Kahn	*	$25n^4+66n^3+129n^2+42n-96$	67984
	+	$32n^4+86n^3+171n^2+53n-128$	51456
Hollerbach 4x4	*	830n-592	4388
	+	675n-464	3586
Hollerbach 3x3	*	412n-277	2195
	+	320n-201	1719
Newton-Euler	*	150n-48	852
	+	131n-48	738

efficiency results because 3X3 matrix multiplication requires less than one half the number of operations required by 4X4 multiplication. The 3X3 equations lack the elegance and simplicity of the 4X4 formulation and as a result are more difficult to program. The 4X4 equations are much easier to program than either the 3X3 Hollerbach equations or the Newton-Euler equations.

The ease of programming the 4X4 Hollerbach equations stems from the use of homogeneous transforms. However, the fourth row of a homogeneous transform, consisting of zeros and a one, contains no essential information. Therefore computational steps involving the fourth row are needless and inefficient. Hollerbach (1980) notes that reductions in computation could be realized through software which optimizes the multiplication of homogeneous transforms. This optimization would neglect the operations imposed by the fourth row and would improve the efficiency by approximately fifty percent. At the same time it would preserve the algorithmic simplicity of the 4X4 equations. Unfortunately the efficiency of this algorithm still does not compare favourably with the Newton-Euler formulation even though it is easier to program.

Symbolic Generation

The efficiency of the equations may be improved by generating the equations of motion symbolically. In doing so operations involving zeros and ones need never be performed and increases in efficiency should result. Computing time should also be reduced due to the elimination of looping, testing, and logical transfer operations. This results from the significant computational overhead imposed by the incrementing and reseting of counters and the relocation of data.

In order to investigate the viability of symbolic generation the program DYNAM was developed. DYNAM is an interactive program which generates symbolic equations of motion for arbitrary open chain manipulators with up to 6 degrees of freedom. As the equations are being generated DYNAM writes a FORTRAN callable subroutine which will evaluate the equations of motion given the manipulators kinematic state. As a result, the equations of motion for any arbitrary manipulator can be generated and programmed automatically with minimal input by the user. The equation generation is based upon the 4X4 Hollerbach formulation. Appendix I contains a discussion of the DYNAM algorithm and the recursive Lagrangian equations. Figure III.3 illustrates the relationship between the user simulation program, the equations of motion, and DYNAM.

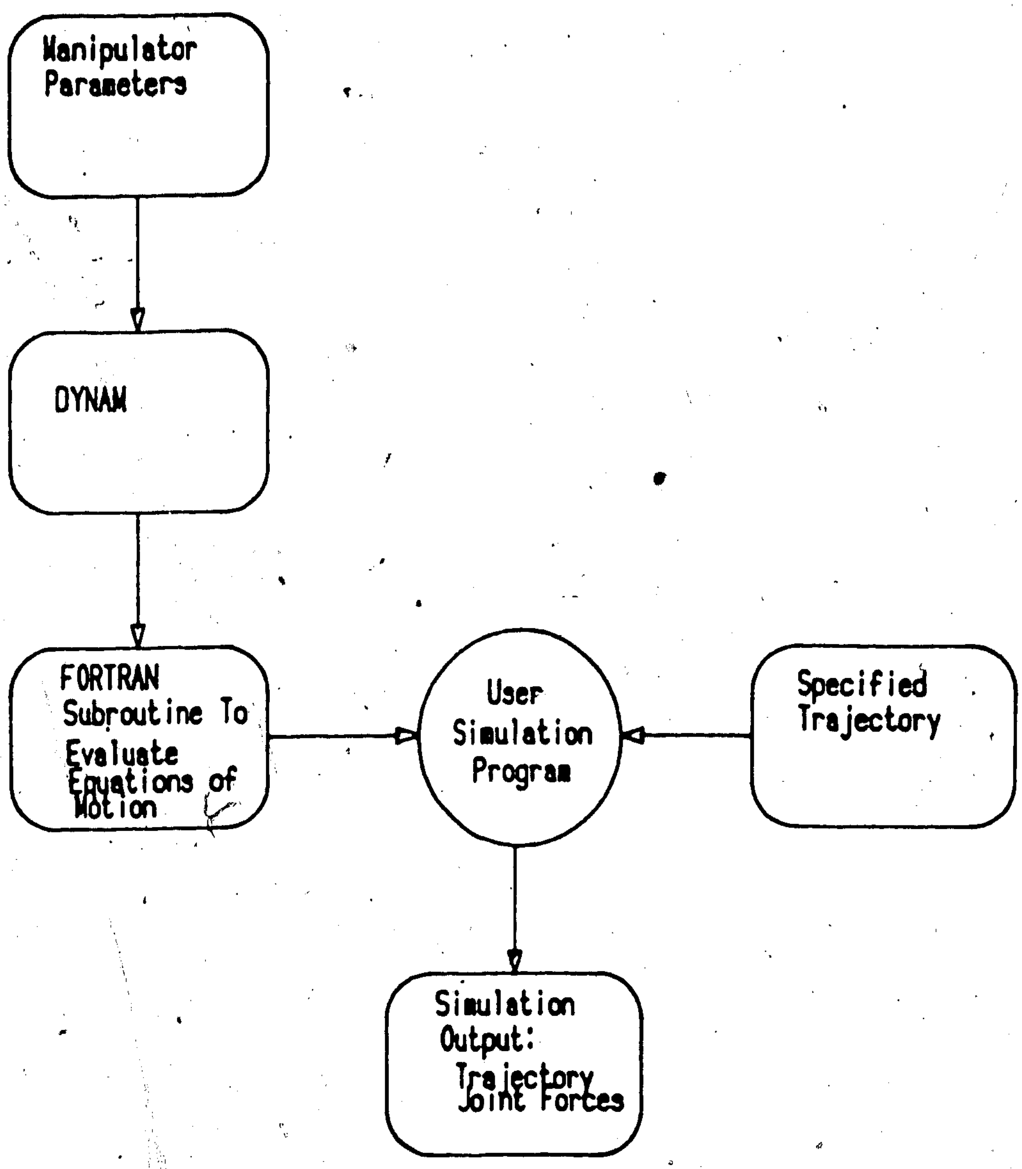

Figure III.3 The Relationship Between DYNAM and the User Simulation Program

Preliminary investigation quickly indicated that the symbolic equations for manipulators with three or more degrees of freedom grow at an astounding rate. Therefore, some form of equation simplification becomes essential. The equations for manipulators with revolute joints contain many trigonometric functions. As a result vast reductions in equation complexity can be obtained through simplification utilizing trigonometric identities. The equations of motion for a two degree of freedom manipulator with non-zero inertia terms, as generated by DYNAM with no simplification, require more than 798 multiplication and 399 addition steps in order to be evaluated. Evaluating the equations numerically utilizing the Hollerbach 4X4 equations would require 1068 multiplication and 886 addition steps while the Newton-Euler would involve 252 and 214 respectively. However if the symbolic equations are reduced through the use of trigonometric identities and grouping, only 41 multiplication and 40 addition steps are required. This represents a drastic reduction in computation and increase in efficiency.

The simplification performed upon the two degree of freedom equations was carried out by hand and was a long and tedious process. The simplification of the significantly longer and more complicated equations of a higher degree of freedom manipulator would be a long and arduous task (if it could be performed at all). Therefore it becomes essential that simplifications be performed automatically by the computer during the generation process. This could be an extremely difficult task given that trigonometric simplifications are required. Languages such as REDUCE2 can expand and simplify algebraic expressions, however no facility exists for trigonometric simplifications. As a result, some other method of equation simplification is required.

A simple and effective algorithm for automatic simplification has been implemented in DYNAM. As the equations of motion are being generated, each symbolic string is tested to see if more than two multiplications or one addition is required. If this is the case the string is replaced with a new variable and the corresponding algebraic equation is written into the FORTRAN subroutine. This method of simplification eliminates a lot of needless recomputation and reduces storage requirements because a given string may be used several times in the equations of motion and it need only be evaluated and stored a single time. Much less complicated equations result from this method of simplification. Hence for the two degree of freedom example discussed previously, only 203 multiplication and 104 addition steps are required for equation evaluation. It is apparent that still greater efficiency could be realized if full trigonometric simplification were possible. More research in this area could lead to extremely efficient algorithms for equation of motion evaluation.

Table III.2 compares the computational efficiency of the symbolic equations to the Hollerbach and Newton-Euler formulations for two and three degree of freedom revolute manipulators. The simplified symbolic expressions require less than one third the total number of operations needed by the 3X3 Hollerbach equations and only one fifth of those needed by the 4X4 equations. The total number of computational steps, between the Newton-Euler and simplified equations, is within 10% for the three degree of freedom case. However, the symbolic equations should be significantly more efficient with respect to the computational time required for equation evaluation. This is due to the greater efficiency of machine code produced by the symbolic equations which have none of the looping, testing, or logical transfer instructions required by the Newton-Euler equations.

Comparing the actual number of arithmetic operations required by the symbolic equations to other algorithms becomes impractical for manipulators with more than three degrees of freedom. However, comparisons of the computer time required to evaluate the equations by different methods is feasible. Table III.3 shows that, for a three degree of freedom revolute manipulator, the symbolic expressions are evaluated in approximately one fourtieth of the time required by the 4X4 Hollerbach equations. Assuming that the Newton-Euler equations can be evaluated in approximately one fifth the time of the Hollerbach equations the symbolic

Table III.2 Computational Complexity of Symbolic Equations

		Computational Complexity	
		Two Degrees of Freedom	Three Degrees of Freedom
Symbolic Unsimplified	x	More than 798	Unavailable
	+	More than 399	Unavailable
Symbolic Simplified	x	203	450
	+	104	243
Hollerbach 4x4	x	1068	1898
	+	886	1561
Hollerbach 3x3	x	547	959
	+	439	759
Newton-Euler	x	252	402
	+	214	342

equations should be approximately eight times faster than the Newton-Euler. This represents a dramatic improvement in efficiency.

For a six degree of freedom manipulator the symbolic equations can be evaluated in one sixth the time required by the Hollerbach equations. Therefore, the symbolic equations generated and simplified by DYNAM should be approximately as efficient as the Newton-Euler formulation.

It can be seen that the Hollerbach equations for a six degree of freedom manipulator require twice as long to evaluate than do the three degree of freedom equations. This is to be expected because the Hollerbach equations are linear with respect to degrees of freedom.(refer to Table III.1) It is also noted that the simplified symbolic equations are not linear. The three degree of freedom equations are evaluated in one thirteenth the time required by the six degree of freedom equations. A non-linear relationship to be expected as the equations themselves are not linear with respect to degrees of freedom. However, if the symbolic equations were completely reduced they would always be evaluated in less time than the Hollerbach or Newton-Euler methods.

Table III.2 indicates that the symbolic expressions for a three degree of freedom manipulator require approximately one fourth as many operations as the 4X4 Hollerbach expressions. However, Table III.3 indicates that the symbolic expressions can be evaluated in approximately one

Table III.3 Relative Execution Time Comparisons Between The Symbolic Equations And The Hollerbach 4x4 Algorithm

	Degrees of Freedom	
	Three	Six
Simplified Symbolic	1	13
Hollerbach 4x4	42	87

fourtieth of the time. This difference represents the overhead imposed by looping, testing, and logical transfer operations. Clearly the elimination of these steps results in a substantial improvement in computational efficiency.

The examples illustrated in Tables III.2 and III.3 have been based upon revolute manipulators. The equations of motion for revolute manipulators can be much more involved than those of manipulators with prismatic joints. Mutually perpendicular prismatic joints minimize joint interactions and therefore simplify the equations of motion. Comparing the execution time of revolute and prismatic manipulators indicates that a six degree of freedom manipulator with prismatic second and third joints has equations of motion which can be evaluated 25% faster than a purely revolute manipulator. It must be noted that the numerical methods of equation generation do not realize any reduction in computional complexity because of prismatic joints. This increase in efficiency is only available to symbolic methods.

Further reductions in equation complexity may also be obtained by eliminating some of the inertia terms of certain links. Generally the fourth, fifth, and sixth links of a six degree of freedom manipulator are very short and have small inertias when compared to the first three links and typical loads. If the inertia terms (I_{xx}, I_{yy}, I_{zz}) for the final three links are neglected, the resulting equations of motion can be evaluated in less than one half the time of a standard six degree of freedom manipulator. The ability to take advantage of physical properties which minimize equation complexity is another advantage of symbolic generation over numerical methods.

The results shown here indicate that symbolic generation can be an efficient, viable, and versatile method for evaluating the equations of motion. Still greater improvements in efficiency could be realized if full trigonometric simplification and regrouping were to employed. This will be a very difficult task to automate however and was not attempted in the present work.

The Forward Dynamic Solution

The forward solution is directed at evaluating the joint forces given the manipulator's kinematic state at some instant in time. Any of the algorithms previously discussed for equation of motion evaluation may be used to solve the forward problem. Obviously, efficient algorithms are most desirable.

The first step in the forward solution is to define some desired trajectory and then specify some time history of path execution. The joint positions may then be evaluated as functions of time and joint velocities and accelerations may be obtained through successive numerical differentiation. Finally the equations of motion are evaluated and the joint forces required to traverse the path may be obtained.

Figure III.4 illustrates the joint torque functions required by a three degree of freedom revolute manipulator, to traverse the straight line trajectory shown in Figure III.5. Straight line trajectories are representative of those used in assembly operations as well as simple welding and painting motions. The ideal link parameters (kinematic and dynamic) are listed in Appendix II and are representative of a light duty commercial manipulator. This manipulator will be used throughout the body of this thesis.

The path time history chosen for this example was a shifted cosine function as shown in Figure III.6. It can be seen in Figure III.4 that this type of time history results in instantaneous changes in joint torques at the start and finish of the trajectory. These abrupt changes would not be advisable in actual practice as they could induce vibration and overshoot problems. Therefore some other type of path history, likely based upon polynomials, which produces smooth acceleration transitions at all times would be required for actual implementation. The cosine path history will be used however, for the duration of this investigation.

It can be seen in Figure III.6 that the time is the ratio of time to total duration of trajectory, and as such is nondimensional. Likewise, the trajectory parameter s is nondimensional parameter ranging from 0 to 1. This parameter describes the position of the endpoint relative to the total length of the trajectory.

The forward dynamic solution will be used in chapter 4 to investigate the effect of misalignments on the joint torques required to move a manipulator along a desired trajectory.

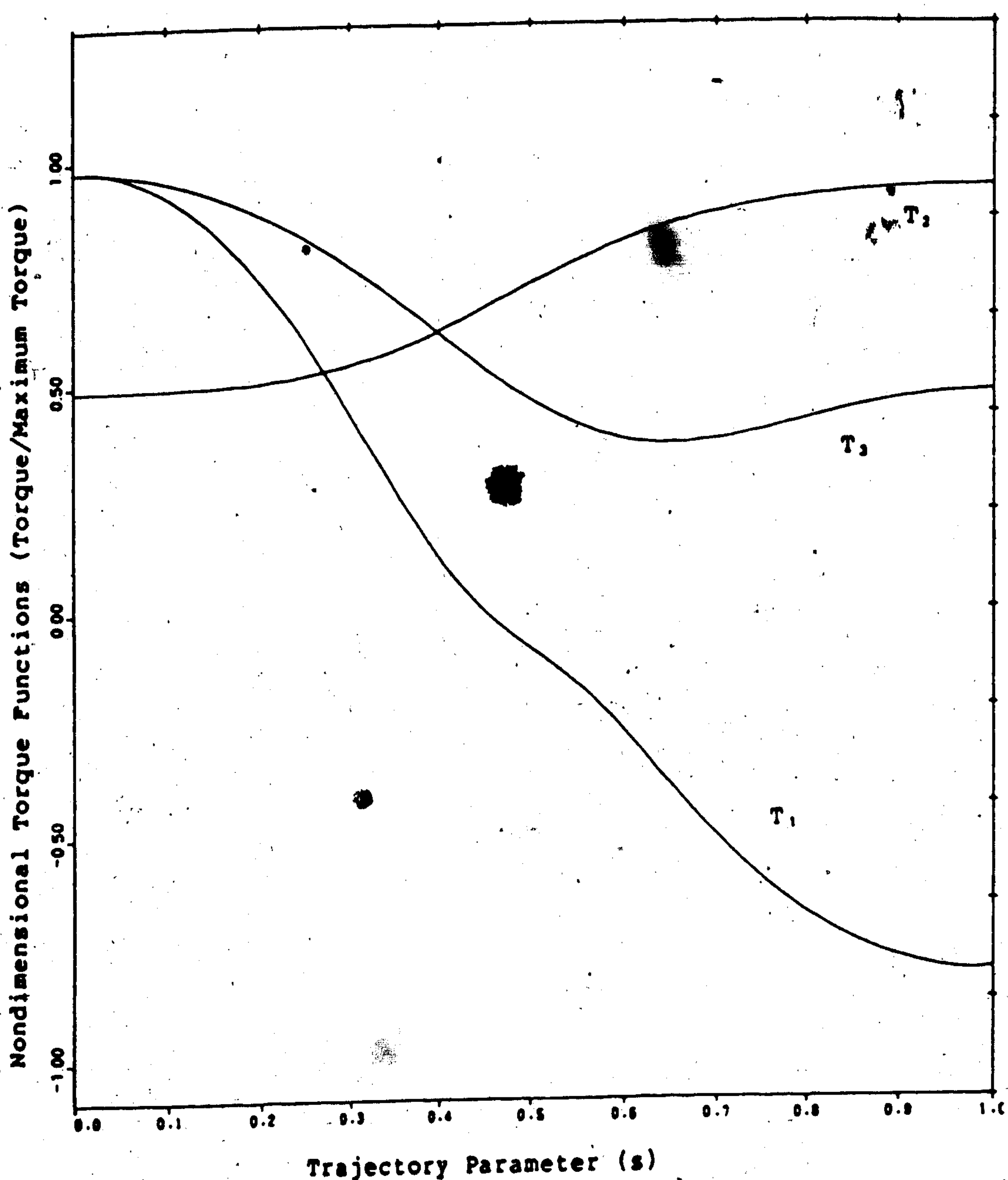

Figure III.4 Joint Torque Functions Required by a Three Degree of Freedom Manipulator to Traverse a Typical Path

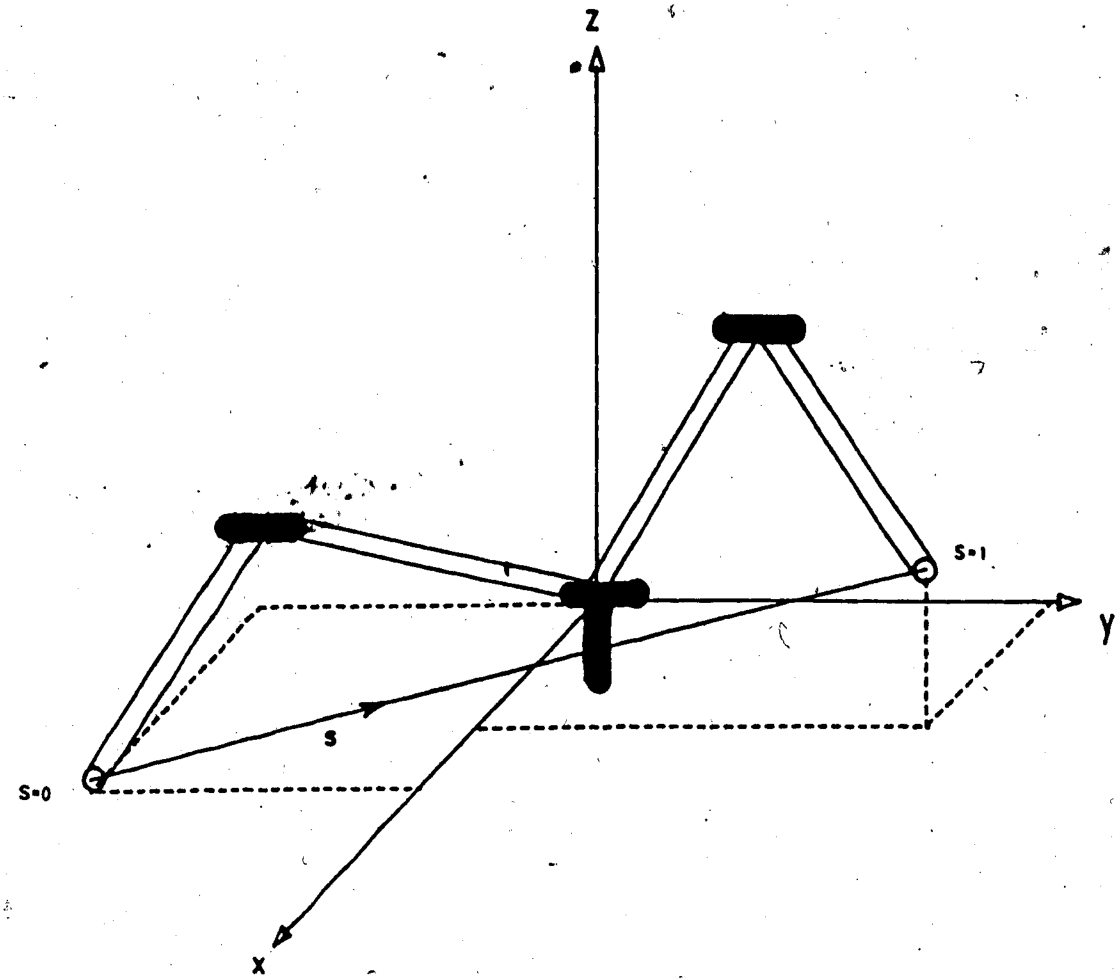

Figure III.5 Typical Three Degree of Freedom Manipulator and Typical Trajectory

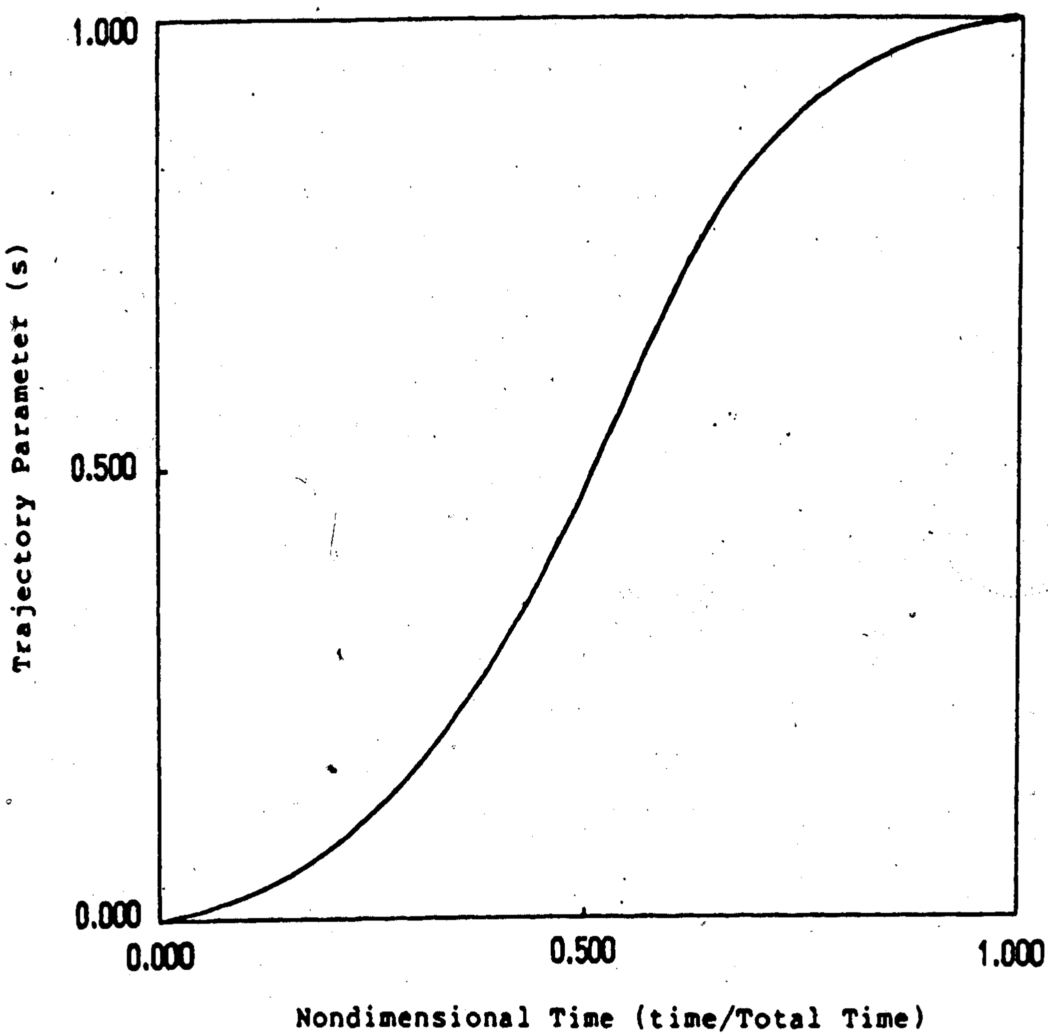

Figure III.6 Path Time History

IV. The Effect of Misalignments on Performance

Misalignments affect both the static positioning accuracy and dynamic response of robots. The magnitude of these effects depends upon which parameters, or combinations thereof, are misaligned, the magnitude of the misalignments, and the configuration of the manipulator.

A. Positioning Errors

Robot manufacturers often specify very high tolerances for manipulator repeatability. However, accuracy is rarely mentioned. Repeatability is a measure of the tolerance within which a manipulator will return to a predefined point and is therefore related to the control system. Accuracy however, is a measure of the tolerance within which a manipulator will reach a desired point given the ideal joint coordinates which have been calculated based upon the ideal kinematic parameters. As such, accuracy is affected by the kinematic parameters themselves. Offline programming requires very accurate mathematical models of manipulators and therefore the identification of misalignments is very important.

The modified Denavit-Hartenberg convention defined five link parameters which may be used to describe each link in a robot system. Errors in any of these parameters will have varying effects upon both the positioning and orientation accuracy.

The position and orientation of the nth link is defined by the elements of the T_n matrix. The position alone is defined by the elements of the fourth column and may be represented as functions of 1st through nth link parameters:

$$x_n = f(a_1, d_1, \alpha_1, \ldots, \gamma_n, \theta_n)$$

$$y_n = g(a_1, d_1, \alpha_1, \ldots, \gamma_n, \theta_n)$$

$$z_n = h(a_1, d_1, \alpha_1, \ldots, \gamma_n, \theta_n)$$

The position error may result from misalignments in any one or more of the links. This error may be defined as;

$$|E| = \{\epsilon_x^2 + \epsilon_y^2 + \epsilon_z^2\}^{1/2}$$

Where ϵ_x is the difference between the x coordinate of the ideal and the misaligned manipulator. Similarly ϵ_y and ϵ_z represent the differences between the ideal and misaligned y and z coordinates.

$$\epsilon_x = f_i(\ldots) - f_m(\ldots)$$

$$\epsilon_y = g_i(\ldots) - g_m(\ldots)$$

$$\epsilon_z = h_i(\ldots) - h_m(\ldots)$$

Where $f_i(\ldots)$, $g_i(\ldots)$, and $h_i(\ldots)$ represent the ideal x,y,z coordinates and $f_m(\ldots)$, $g_m(\ldots)$, and $h_m(\ldots)$ represent the endpoint coordinates of the actual (presumed misaligned) manipulator

The terms ϵ_x,ϵ_y,and ϵ_z can be expressed as differentials;

$$\varepsilon_x = \frac{\partial f_i}{\partial a_1}\Delta a_1 + \ldots + \frac{\partial f_i}{\partial \theta_3}\Delta\theta_3$$

$$\varepsilon_y = \frac{\partial g_i}{\partial a_1}\Delta a_1 + \ldots + \frac{\partial g_i}{\partial \theta_3}\Delta\theta_3$$

$$\varepsilon_z = \frac{\partial h_i}{\partial a_1}\Delta a_1 + \ldots + \frac{\partial h_i}{\partial \theta_3}\Delta\theta_3$$

Where $\Delta a_1 \ldots \Delta\theta_n$ are the misalignments for the parameters of the n links. The error in each coordinate, resulting from parameter misalignments, can be represented by the sum of the partial derivatives of the respective equation with respect to each paramater (evaluated at the ideal state) multiplied by the parameter misalignment. In this fashion general equations for position error can be generated. Orientation errors can be investigated in a similar manner, however this analysis will be postponed until later in the chapter.

The effects of misalignments in each of the parameters of a three degree of freedom manipulator will be analyzed. In this analysis, misalignments in only link 2 will be pursued so as to minimize equation complexity. Extension of this analysis to multiple misalignments is straight forward.

The kinematic equations defining the endpoint position for a three degree of freedom revolute manipulator with ideal 1st and 3rd links and a misaligned 2nd link are:

$$f = a_3 C_3 (C_1 D_{21} + S_1 D_{27}) + a_3 S_3 (C_1 D_{22} + S_1 D_{28}) + a_2 C_1 C_2 + d_2 S_1$$

$$g = a_3 C_3 (S_1 D_{21} - C_1 D_{27}) + a_3 S_3 (S_1 D_{22} - C_1 D_{28}) + a_2 S_1 C_2 - d_2 C_1$$

$$h = a_3 D_{24} C_3 + a_3 D_{25} S_3 + a_2 S_2$$

Where:

$D_{21} = C\theta_2 C\gamma_2 - S\theta_2 S\alpha_2 S\gamma_2$

$D_{22} = -S\theta_2 C\alpha_2$

$D_{24} = S\theta_2 C\gamma_2 + C\theta_2 S\alpha_2 S\gamma_2$

$D_{25} = C\theta_2 C\alpha_2$

$D_{27} = -C\alpha_2 S\gamma_2$

$D_{28} = S\alpha_2$

The position error arising due to misalignments in the second link is:

$$|\epsilon| = (\epsilon_x^2 + \epsilon_y^2 + \epsilon_z^2)^{1/2}$$

where, with all other misalignments equal to zero:

$$\epsilon_x = C_1C_2\Delta a_2 + S_1\Delta d_2 + S_1a_3S_3\Delta\alpha_2 - S_1a_3C_3\Delta\gamma_2 - C_1(a_3S_{23} + a_2S_2)\Delta\theta_2$$

$$\epsilon_y = S_1C_2\Delta a_2 - C_1\Delta d_2 - a_3C_1S_3\Delta\alpha_2 + a_3C_1C_3\Delta\gamma_2 - S_1(a_3S_{23} + a_2S_2)\Delta\theta_2$$

$$\epsilon_z = S_2\Delta a_2 - a_3C_1S_3\Delta\alpha_2 + a_3C_1C_3\Delta\gamma_2 + (a_3C_{23} + a_2C_2)\Delta\theta_2$$

The values Δa_2, Δd_2, $\Delta\alpha_2$, $\Delta\gamma_2$, and $\Delta\theta_2$ represent the misalignments in each of link 2's parameters. The effect of individual misalignments is demonstrated as follows.

Position Errors Due to Length and Offset Misalignments

The position error resulting from a misalignment in the length (a_2) of link 2 is given by:

$$|E|_{a_2} = \left\{\left(\frac{\partial f_i}{\partial a_2}\right)^2 + \left(\frac{\partial g_i}{\partial a_2}\right)^2 + \left(\frac{\partial h_i}{\partial a_2}\right)^2\right\}^{1/2} \Delta a_2$$

Carrying out the partial differentiation and simplification a constant position error results:

$$|E|_{a_2} = |\Delta a_2|$$

Similarly, a constant position error:

$$|E|_{d_2} = |\Delta d_2|$$

results from a misalignment in the offset (d_2).

These results should be expected as the kinematic equations are linear with respect to both the length (a_n) and the offset (d_n). Position errors resulting from length and offset misalignments will be constant for manipulators regardless of which link or combination of links is misaligned. Note that it is possible that two or more links could be misaligned such that the total error would be zero.

Position Errors Due to Twist and Skew Misaligments

The position error resulting from a misalignment in the twist (α_2) of link 2 is given by:

$$|E|_{\alpha_2} = \{(\frac{\partial f_i}{\partial \alpha_2})^2 + (\frac{\partial g_i}{\partial \alpha_2})^2 + (\frac{\partial h_i}{\partial \alpha_2})^2\}^{1/2} \Delta\alpha_2$$

Performing the partial differentiation and simplification a variable position error results:

$$|E|_{\alpha_2} = |\, a_3 S_3 \Delta \alpha_2|$$

Performing an analogous operation for skew (γ_2) misalignments a similar expression is obtained:

$$|E|_{\gamma_2} = |\, a_3 C_3 \Delta \gamma_2|$$

These expressions represent position errors which vary nonlinearly from zero to a maximum value of $a_3 * \Delta\alpha_2$ or $a_3 * \Delta\gamma_2$ respectively. Unlike length and offset errors, the errors resulting from twist and skew misalignments will depend upon which joint or joints are misaligned. In the example shown the errors are functions of the length and rotation of link 3. In general, the errors arising due to a twist or skew misalignment in joint i will be a function of the parameters of links i+1 through n. The effect of slight misalignments in twist or skew can result in substantial position errors due to the multiplying effect of the link lengths. This problem becomes more acute for the larger manipulators due to the increased length of each link. This effect was noted by Mooring (1983). He found that, for a PUMA 600, a one degree error in the skew of link 2 and 3 produced a maximum position error of 35mm.

Position Errors Due to Rotation Misalignments

The position error resulting from a misalignment in the rotation angle (θ_2) is given by:

$$|E|_{\theta_2} = \{(\frac{\partial f_i}{\partial \theta_2})^2 + (\frac{\partial g_i}{\partial \theta_2})^2 + (\frac{\partial h_i}{\partial \theta_2})^2\}^{1/2} \Delta\theta_2$$

Once again, performing the partial differentiation and simplification, a variable position error results:

$$|E|_{\theta_2} = \{a_3^2 + a_2^2 + 2\, a_2 a_3 C_3\}^{1/2} \Delta\theta_2$$

The maximum position error occurs when θ_3 equals 0 degrees, i.e. arm extended, and is given by:

$$|E|_{\theta_2} = (a_2 + a_3)\, \Delta\theta_2$$

The minimum occurs at θ_3 equals 180 degrees, i.e. arm doubled back on itself, and is given by:

$$|E|_{\theta_2} = (a_2 - a_3)\, \Delta\theta_2$$

In general the position error, in an n degree of freedom manipulator, arising due to a misalignment in the ith rotation variable (θ_i) will be a function of the ith through n link parameters. As with the twist and skew errors, substantial position errors can arise due to slight errors in the rotation variable. Errors in rotation are most likely to arise as part of the control hardware or software. Problems with angular position encoders could easily result in rotation misalignments.

Multiple Misalignments

The effects of individual misalignments have been examined. However, actual manipulators will generally have some combination of misalignments. General expressions for the position errors resulting from combinations of misalignments can be obtained by the method outlined. However, the equations become quite involved and manual derivations become lengthy. Therefore a more general method of deriving the equations of position and orientation errors is needed.

The position and orientation of the nth manipulator link is defined by the T_n matrix. The incremental change in each element of T_n arising due to misalignments in each of the 5*n link parameters (p_m) can be written as:

$$\Delta T_n = \sum_{m=1}^{5n} \left(\frac{\partial T_n}{\partial p_m} \Delta p_m \right)$$

This matrix can be termed the error matrix E_n.

The change in orientation resulting from the misalignments will be defined by the [illegible] three columns of of the error matrix while the change in position will be defined by the fourth column. Given the partial derivatives of the T_n matrix with respect to each link parameter general expressions for position and orientation error may be obtained.

The T_n matrix is the product of the 1st through nth modified Denavit-Hartenberg matrices, D.

$$T_n = D_1 D_2 \ldots D_n$$

The partial derivative of T_n with respect to a parameter, p_m, of link m can be obtained by the product of the D matrices and the partial derivative of D_i with respect to p_m. Recall that the partial of D_i with respect to p_j is zero.

$$\frac{\partial T_n}{\partial p_m} = D_1 \ldots D_{i-1} \frac{\partial D_i}{\partial p_m} D_{i+1} \ldots D_n$$

The elements of E_n which describe the position of a typical 3 degree of freedom revolute manipulator (Figure III.1) have been generated and the effect of multiple misalignments on the position error along an arbitrary straight line trajectory (Figure III.5) has been investigated as follows.

The position error as a function of the normalized trajectory parameter (s) has been plotted in Figure [illegible] In this analysis the length and offset of each link are assumed to be misaligned by 0.254 mm. The twist, skew, and [illegible] are assumed to be misaligned by 1.0 degrees. The position error varies nonlinearly over the trajectory from a maximum value of 8.89 mm to a minimum value of 6.40 mm. The maximum position error occuring over the entire workspace can be determined by maximizing the position elements of E_n and is found to be 33.02 mm, which is similar to that found by Mooring.

Clearly these are unacceptable discrepancies in systems which may require accuracy to fractions of millimeters. If offline programming is to be successful, misalignments must be identified and incorporated into the manipulator model.

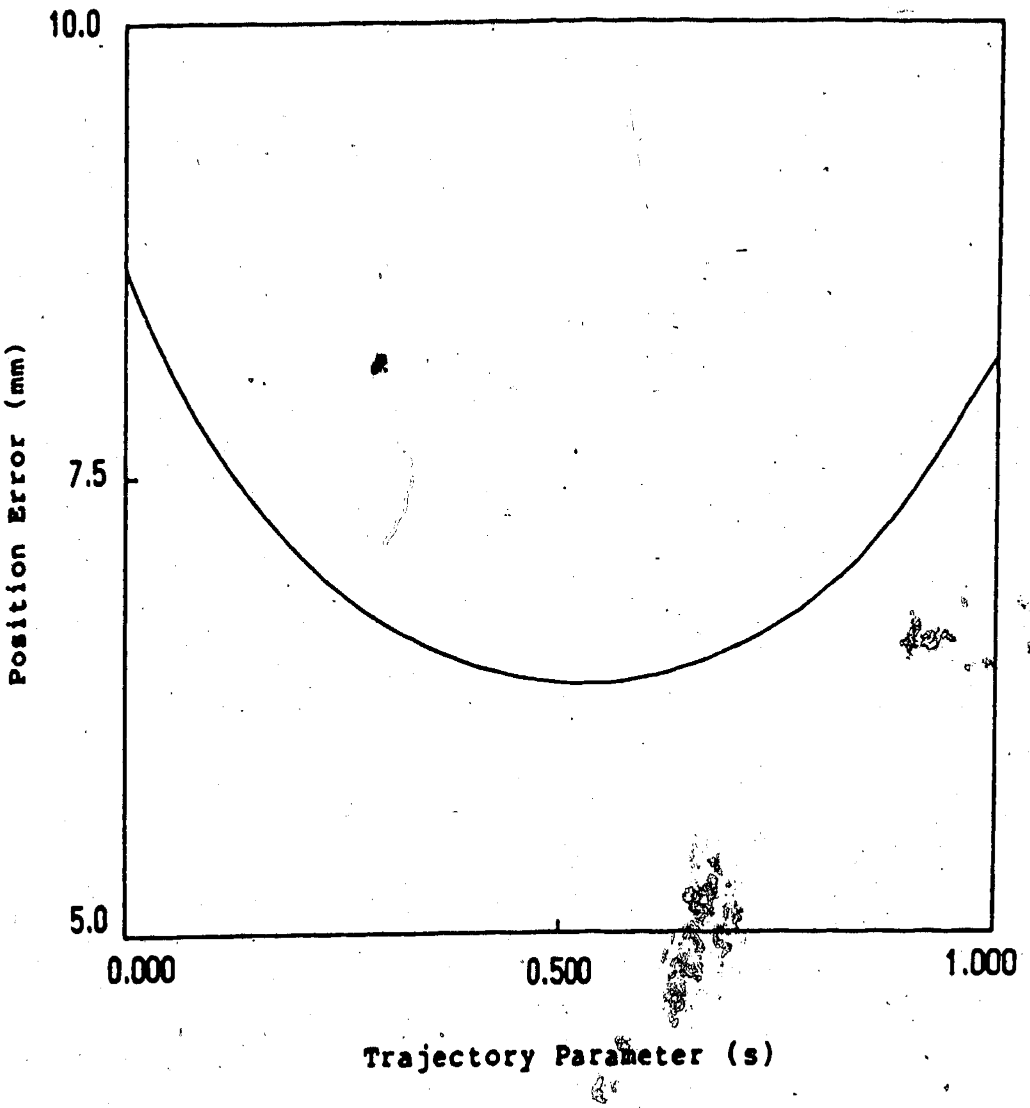

Figure IV.1 Position Error Along a Typical Trajectory

Orientation Errors

Orientation errors arising due to misalignments can also be investigated using the error matrix. The first three rows and columns of E_n define the incremental change in the direction cosines, which define the link orientation, arising from misalignments. The complete error matrix for a three degree of freedom manipulator with misalignments in the parameters of link two is:

$$E_3 = \begin{bmatrix} E_{1,1} & E_{1,2} & E_{1,3} & E_{1,4} \\ E_{2,1} & E_{2,2} & E_{2,3} & E_{2,4} \\ E_{3,1} & E_{3,2} & E_{3,3} & E_{3,4} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Where:

$$E_{1,1} = - C_1 S_{23}\, \Delta\theta_2 + S_1(\Delta\alpha_2\, S_3 - \Delta\gamma_2 C_3)$$

$$E_{1,2} = - C_1 C_{23}\, \Delta\theta_2 + S_1(\Delta\alpha_2\, C_3 - \Delta\gamma_2 S_3)$$

$$E_{1,3} = C_1(\Delta\gamma_2\, C_2 + \Delta\alpha_2\, S_3)$$

$$E_{1,4} = C_1 C_2\, \Delta a_2 + S_1\, \Delta d_2 + S_1 a_3(S_3\, \Delta\alpha_2 - C_3\, \Delta\gamma_2)$$
$$- C_1(a_3\, S_{23} + a_2 S_2)\Delta\theta_2$$

$$E_{2,1} = - S_1 S_{23}\, \Delta\theta_3 + C_1(\Delta\gamma_2\, C_3 - \Delta\alpha_2\, S_3)$$

$$E_{2,2} = - S_1 C_{23}\, \Delta\theta_2 - C_1(\Delta\alpha_2\, C_3 + \Delta\gamma_2\, S_3)$$

$$E_{2,3} = S_1(\Delta\gamma_2 C_2 + \Delta\alpha_2\, S_2)$$

$$E_{2,4} = S_1 C_2\, \Delta a_2 - C_1\, \Delta d_2 - C_1 a_3(S_3\, \Delta\alpha_2 + C_3\, \Delta\gamma_2)$$
$$+ S_1\, (a_3 S_{23} + a_2 S_2)\, \Delta\theta_2$$

$$E_{3,1} = C_{23}\, \Delta\theta_2$$

$$E_{3,2} = - S_{23}\, \Delta\theta_2$$

$$E_{3,3} = - C_2\, \Delta\alpha_2 + S_2\, \Delta\gamma_2$$

$$E_{3,4} = S_2\, \Delta a_2 - C_1 a_3(S_3\, \Delta\alpha_2 + C_3\, \Delta\gamma_2)$$
$$+ (a_3 C_{23} + a_2 C_2)\Delta\theta_2$$

Note that the orientation error is independent of the length and offset parameters as expected.

Figure IV.2 shows the path of a typical three degree of freedom manipulator moving along the global x axis. If the manipulator were ideal link 3 would be coplanar with the x-z plane at all times. However misalignments will cause the link to be displaced, relative to the x-z plane, by some angle ζ which will vary during the course of the trajectory. Figure IV.3 illustrates the error angle (ζ) as the angle between the ideal x axis (x_{3i}) and the actual, misaligned axis (x_{3m}). The terms Δm_1 and Δn_1 represent the elements of the error matrix $E_{3(1,2)}$ and $E_{3(1,3)}$

Figure IV.4 plots the magnitude of the orientation error ζ as a function of the normalized trajectory paramater s. In this analysis the twist, skew, and rotation have each been assumed to be misaligned by 1.0 degree. Note that the orientation error varies from a maximum of 1.65 degrees to a minimum of zero and then through the x-z plane to a final error of 0.06 degrees.

It has been shown that general expressions for the position and orientation errors can be obtained through the generation of the error matrix. Length and offset misalignments result in constant position errors over the workspace and have no effect upon the orientation. Errors arising due to misalignments in the twist, and skew are nonlinear functions of the parameters of links i+1 through n. The error resulting from a misalignment in the rotation is a nonlinear function of the parameters of links i through n. Position and orientation are critical for certain assembly, welding, and material handling operations. The position and orientation errors which can result from relatively minor misalignments have been shown to be substantial.

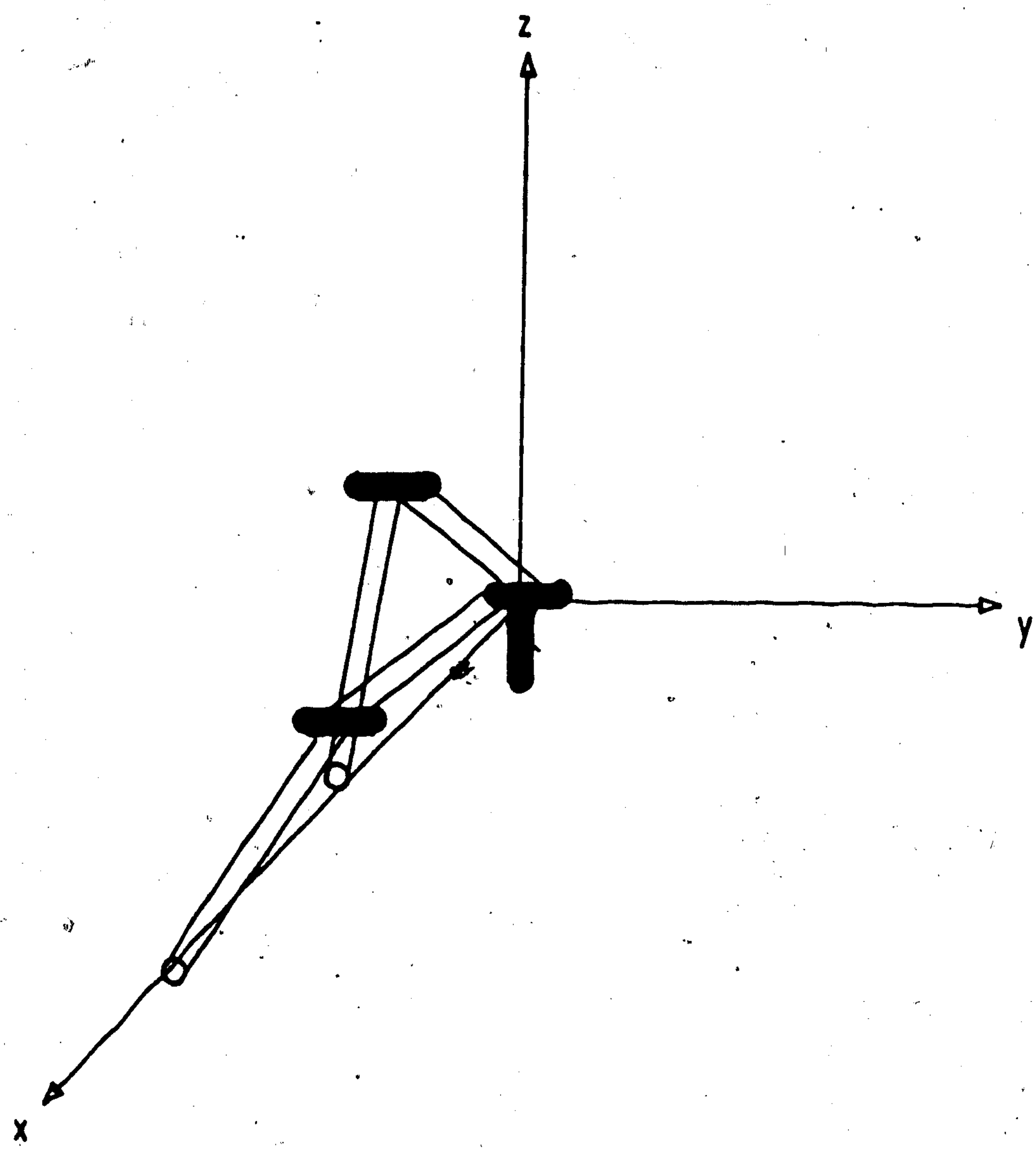

Figure IV.2 A Path Which Traverses the x-axis

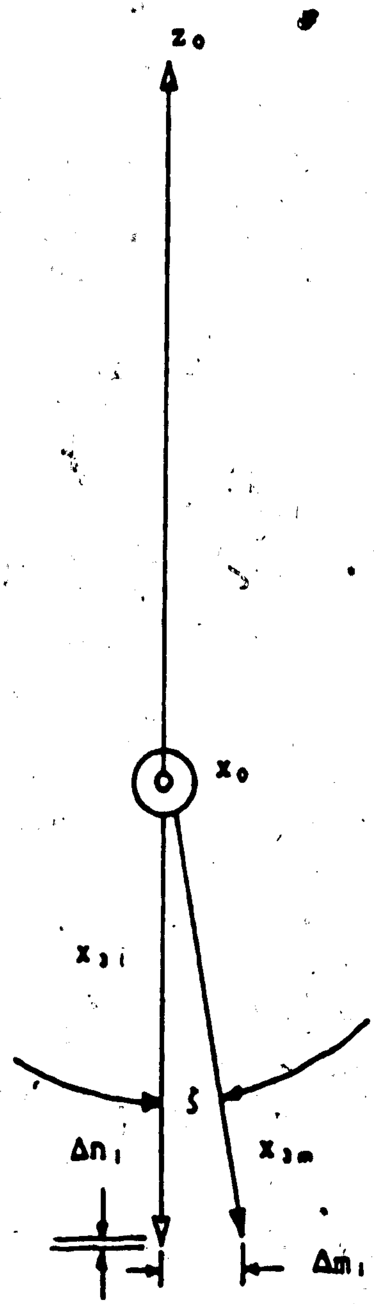

Figure IV.3 Definition of the Orientation Error

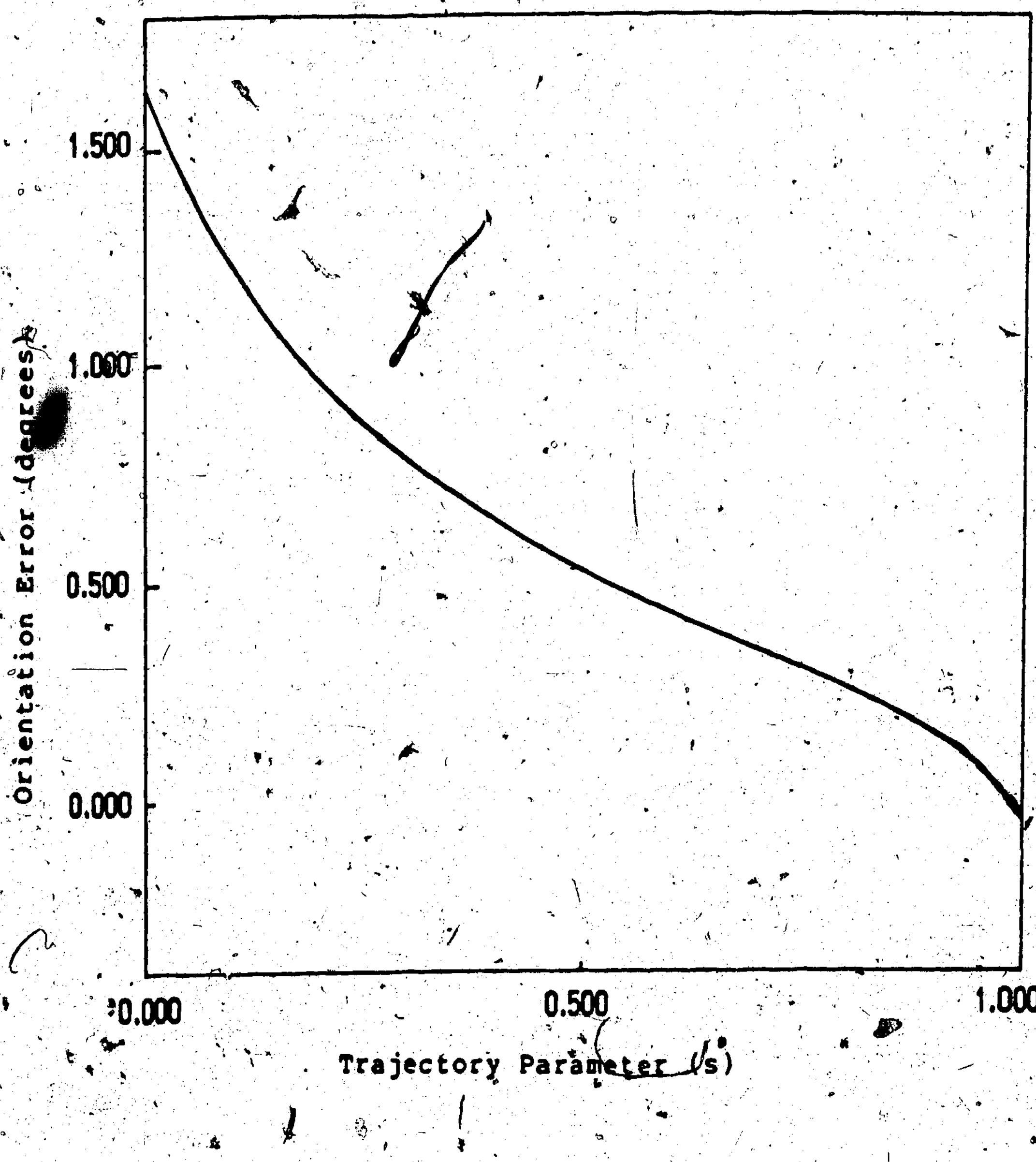

Figure IV.4 Orientation Error as a Function of the Trajectory Parameter

As a result the determination of misalignments is very important if an accurate model of a manipulator is to be obtained. Furthermore, misalignments must be accounted for if offline programming of manipulators is to be successful.

B. Dynamic Analysis of Misaligned Manipulators

Each manipulator link has five kinematic parameters which define the link coordinate system and orientation. Position and orientation errors were completely defined by misalignments in these five parameters. However, each link also has associated with it ten dynamic parameters which define the mass and inertia properties. The three product of inertia terms (I_{xy}, I_{xz}, I_{yz}) are generally zero leaving seven parameters which locate the centre of mass and describe the moments of inertia of each link. As a result a total of twelve parameters are required to define each link kinematically and dynamically. In order to thoroughly investigate the effect of misalignments on dynamic performance the concept of misalignments, or parameter errors, would have to be expanded to include errors in the seven dynamic parameters of each link.

This section will deal with obtaining the joint torques required by a misaligned manipulator to traverse a prescribed trajectory. It will be shown that joint torque functions are almost identical for the ideal and misaligned manipulators. Therefore the parametric analysis of the dynamic response of misaligned manipulators seems to be of little value.

Joint Force Deviations Along a Prescibed Path

The joint forces required for an ideal manipulator to traverse a given path, in a specified time, were obtained in chapter 3 by evaluating the equations of motion for an ideal manipulator at each point along the trajectory. The equations of motion for misaligned manipulators can be generated by incorporating the modified Denavit-Hartenberg transform into either the Lagrangian or Newton-Euler formulations. The program DYNAM has incorporated the modified transform into Hollerbach's recursive Lagrangian formulation and thus the equations of motion for misaligned manipulators can be generated symbolically. Using these equations of motion the dynamic response of misaligned manipulators has been investigated. The relative magnitude of dynamic errors resulting from misalignments in the kinematic parameters will be demonstrated by considering the effect of kinematic misalignments on the forward solution.

The Effect of Kinematic Parameter Misalignments on Joint Forces

The joint forces required to traverse a typical trajectory (Figure III.5) over a specified time period (2 seconds) were evaluated as outlined in chapter 3. The same trajectory was then simulated with the length, offset, twist, and skew, misaligned by 0.25 mm and 1.0 degrees respectively. Errors of these magnitudes, 0.25%, are reasonable given the machining processes involved in the manufacture of commercial manipulators.

Figure IV.5 plots the joint torque functions for both the ideal and misaligned manipulators. It can be seen that the differences are extremely small. The deviations in joint torques are plotted in Figure IV.6. The maximum torque deviation is associated with joint # 1 and has the value of 2.9 %. The maximum error in joints #2 and #3 are 0.10 % and 0.42% respectively. It is noted that the deviations vary non-linearly throughout the trajectory.

Table IV.1 illustrates the relative sensitivity of the equations of motion to individual misalignments in length, offset, twist, and skew. It can be seen that in this case the equations are most sensitive to errors in the twist and skew parameters. The relative sensitivity of the equations may, however, vary depending on the particular manipulator and trajectory involved.

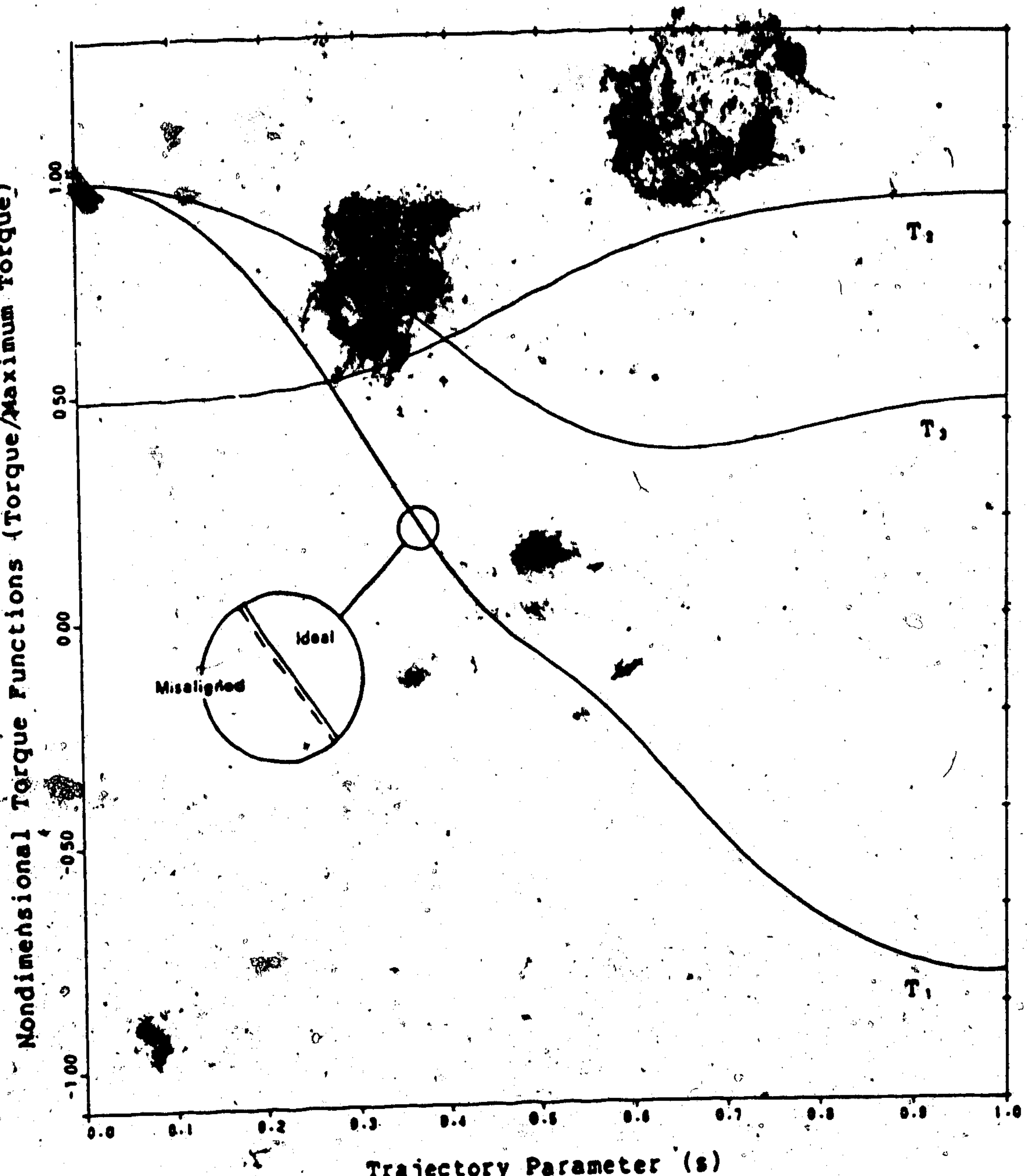

Figure IV.5 Joint Force Functions for an Ideal and a Kinematically Misaligned Manipulator

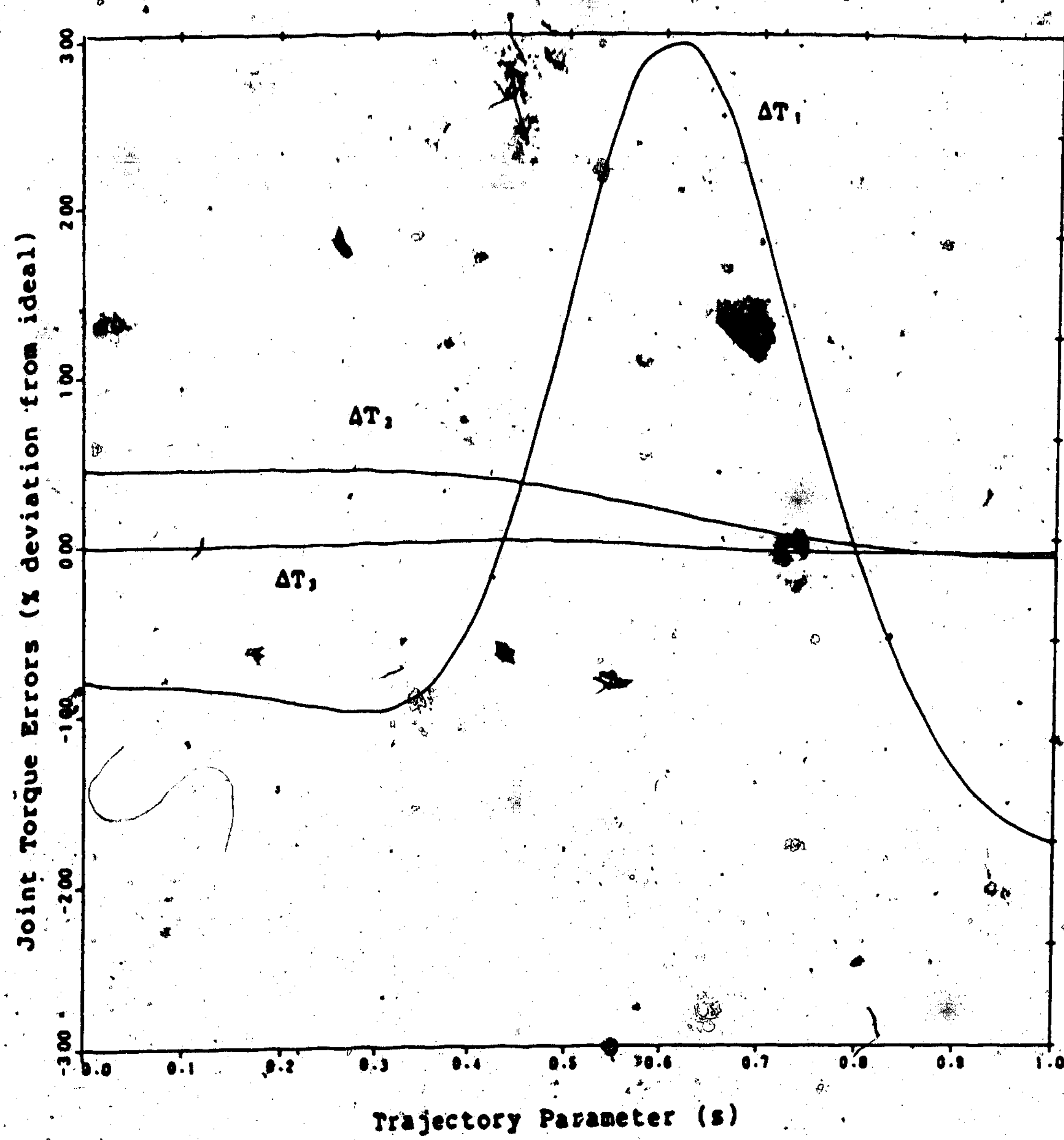

Figure IV.6 Joint Force Deviations Arising Due To Misalignments

This investigation indicates that misalignments do not have a significant impact on the dynamic response of manipulators. Therefore it is likely that the dynamic model incorporated into a control system does not need to include the effect of misalignments. This is an important result because the inclusion of a misaligned dynamic model would lead to a less efficient control algorithm.

Table IV.1 The Effect of Individual Misalignments on the Equations of Motion

Parameter Error	Joint Torque Errors		
	Joint 1	Joint 2	Joint 3
$\Delta a_2=0.25$	0.72%	0.04%	0.08%
$\Delta d_2=0.25$	0.11%	0.10%	0.05%
$\Delta \alpha_2=1.0$	2.40%	0.37%	0.10%
$\Delta \gamma_2=1.0$	2.00%	0.33%	0.02%

V. Conclusions and Recommendations

This thesis has investigated the effects of joint misalignments on the analysis and performance of robot manipulators. The identification and incorporation of misalignments into the kinematic model of manipulators is of fundamental importance to the success of future offline programming techniques. However, it was found that misalignments have very little effect on manipulator dynamics. The main points and conclusions which have arisen from this investigation are presented below:

Manipulator Geometry

The Denavit-Hartenberg convention does not adequately describe the geometry of misaligned manipulators for the following reasons:

1. If consecutive joint axes are not parallel and co-planar or they do not intersect orthogonally, the joint coordinate system can become far removed from the physical end point of the link itself. This leads to the ill-conditioning of the A matrix and can result in severe numerical difficulties, particularly when the inverse of the A matrix is required.
2. When misalignments are present, another transform which

defines the actual endpoint of the link with respect to the link coordinate system is required to locate the endpoint in the workspace.

Mooring proposed the adoption of a transform developed by Suh and Radcliff for the geometric description of misaligned links. The Mooring convention however has two major problems:

3. The Mooring convention does not [illegible] to prismatic joints.
4. The partial derivative matrix of the Mooring transform is relatively complicated and as such is not appropriate for the symbolic generation of the equations of motion.

A modified Denavit-Hartenberg convention was proposed which overcomes the problems illustrated above. With this convention a fifth link parameter termed skew (γ) was introduced.

1. The fifth link parameter accomodates non-parallel joint axes while maintaining the coincidence of the link coordinate system with the physical end point of the link. As a result, the A matrix is always well conditioned and the need for supplementary transformations to locate link endpoints is eliminated. Furthermore, link inertia properties remain essentially unchanged from the ideal configurations.
2. The partial derivative of the modified transform, while more involved than the Denavit-Hartenberg convention, is

much less complicated than the Mooring convention. As a result, the modified convention is more appropriate for use in the symbolic generation of the equations of motion.

The modified convention was then used to investigate the effects of misalignments on the kinematic and dynamic analysis, and performance of misaligned manipulators.

Kinematic Analysis

1. The forward kinematic solution is not adversely affected by the presence of misalignments.
2. Explicit equations for the inverse solution are extremely difficult, if not impossible, to obtain for misaligned manipulators. As a result, numerical methods must be employed to obtain solutions to the inverse problem.
3. The Gauss-Seidel and Newton-Raphson methods for nonlinear algebraic equations were used in order to solve the inverse problem. However, severe oscillation problems were encountered in the regions near any of the global coordinate axes. Under-relaxation helped, however the amount of relaxation required to produce stability resulted in excessively slow convergence.
4. A simple yet effective numerical technique was proposed which was successful in obtaining solutions to the

inverse problem in all regions other than those very close to the z axis. The Z axis is likely to present problems with any numerical technique due to the degenerate nature of this axis.

Dynamic Analysis

1. The equations of motion for misaligned manipulators can be obtained with any of the Lagrangian formulations by utilizing the modified Denavit-Hartenberg convention.
2. A program called DYNAM was developed and used to investigate the viability of the symbolic generation of the equations of motion.
3. It was found that without some form of simplification and reduction the symbolic equations very quickly become extremely large and unmanageable.
4. A simple yet effective algorithm for equation simplification was presented which resulted in a very efficient method of equation evaluation. The symbolic equations for a three degree of freedom revolute manipulator can be evaluated in approximately one fourtieth of the time required by the Hollerbach 4 x 4 formulation. The equations for a six degree of freedom revolute manipulator can be evaluated in approximately one sixth of the time.
5. If the symbolic equations could be completely simplified through the use of trigonometric reduction and

regrouping, the efficiency of the symbolic equations would see a dramatic improvement. It must be noted that completely reduced symbolic equations represent the most efficient equations of motion possible.

6. The complexity of the symbolic equations is non-linear with respect to the number of degrees of freedom, whereas both the Newton-Euler and Hollerbach formulations are linear. However, if the the symbolic equations could be completely reduced, they would always be evaluated in less time.

7. Symbolic generation can take advantage of certain manipulator geometries and dynamic properties which can lead to a reduction in equation complexity. The equations of motion for manipulators with one or more prismatic joints can be evaluated in less time than those of a purely revolute manipulator. Similarly, inertia properties of the end effector of a six degree of freedom manipulator can be eliminated if they are found to be insignificant, and the resulting equations of motion can become much less complicated.

8. The symbolic equations for each joint are independent of each other and would therefore be appropriate for parallel processing architectures. This could lead to further improvement in the efficiency of equation evaluation.

Position and Orientation Errors

1. The Error matrix E_n has been derived and can be used to determine the errors in position and orientation which arise due to any combination of misalignments.
2. The position error resulting from length and offset misalignments will be constant regardless of which link or combination of links is misaligned. As a result, it is possible that two or more links could be misaligned such that the total position error would be zero. Misalignments in the length and offset of a link have no effect on errors in the orientation of a manipulator.
3. The position error resulting from misalignments in the twist (α) or skew (γ) of link i will be nonlinear functions of the parameters of links i+1 through n. The effect of slight misalignments in these parameters can result in substantial position errors due to the multiplying effect of the link lengths. Very large manipulators will therefore be particularly sensitive to accuracy or identification of these parameters.
4. Position errors resulting from misalignments in the joint rotation variable (θ_i) will be functions of the ith through n link parameters. As with twist and skew, substantial errors can arise due to misalignments in the rotation variable. These misalignments will most likely arise due to problems with the control system such as position encoders, or the control software.

5. Orientation errors are independent of the length and offset parameters, however they are nonlinear functions of the twist, skew, and rotation parameters.
6. Both position and orientation are critical for certain assembly and welding operations. It was shown that modest misalignments in the second link of a three degree of freedom manipulator could result in substantial position and orientation errors. Therefore the identification of misalignments and their inclusion in the mathematical model of manipulators is of paramount importance to the success of offline programming. Furthermore, the manipulator control system must incorporate the misalignments in the kinematic model if the desired trajectory is to be traversed.

Dynamic Response

The dynamic response of a misaligned manipulator was investigated and the following was noted:

1. The effect of misalignments on the joint torques required to traverse a given path was found to be relatively small. A three degree of freedom manipulator with the length, offset, twist, and skew misaligned by 0.25 mm and 1.0 degrees respectively was analyzed. It was found that the maximum joint torque deviation occurred on joint number 1 and had a magnitude of 2.9%. The maximum errors in joint torques number two and three were 0.10 and 0.42 respectively. These errors are relatively small and it is likely that they would be insignificant and easily controlled by the manipulator's control system (Assuming of course that the control system included the misalignments in its kinematic model of the manipulator).
2. It was found that the equations were most sensitive to misalignments in the twist and skew parameters. The relative sensitivity may, however, vary depending on the particular manipulator and geometry involved.

Recommendations

The following recommendations can be made:

1. Different numerical methods for solving the inverse kinematic problem should be pursued. Gradient search techniques may be efficient methods of solution.
2. Great improvements in the efficiency of the dynamic models of manipulators could be realized by further simplifying and reducing the symbolic equations of motion. Symbolic dynamic models will likely have a significant impact on manipulator control and simulation. High level control systems are going to demand extreme efficiency of the dynamic model and completely reduced symbolic equations will result in the most efficient models possible.
3. The effect of errors in the inertia parameters on the dynamic response should be investigated as these may have a significant effect on manipulator response. Furthermore, the effect of changes in joint friction should be investigated as this will certainly have an impact on the joint torques required to traverse a desired path. The implication on torque based control systems may be significant.
4. The effect of misalignments on the manipulator control system should be investigated. It will be very important to determine how a manipulator with misalignments will respond when the controller is also modelled. Certain

control systems may have a greater tolerance for misalignments than others. It would certainly seem that torque based control systems would be very sensitive to misalignments and an accurate model of both kinematic and dynamic misalignments would be important.

Bibliography

Ayres, M, Robotics: Applications and Social Implications, Ballinger Pub., 1980.

Bejczy, A.K., "Robot Arm Dynamics and Control", Jet Propulsion Laboratory, California Institute of Technology, 1979.

Brady, Hollerbach, Johnson, Lozano-Perez, Mason, Robot Motion, Planning & Control. MIT Press, 1982

Byung, and Kang , "Suboptimal Control of Industrial Manipulators with Weighted Minimum Time-Fuel Criterion." IEEE Transactions on Automatic Control. Vol. AC-30 No. 1 Jan. 1985

Dubowsky, S, and Maatuk, J, "A Parameter Identification Study of Kinematic Errors in Planar Mechanisms," Journal of Engineering For Industry, Vol. 97 May 1985.

Engelberger, J, Robotics in Practice, American Management Associations, 1980

Foulley, L.P., and Kelly, R.B., "Improving The Precision of a Robot", IEEE 1st International Conference on Robotics, 1984.

Goldstein , H, Classical Mechanics, Addison-Wesley, 1980.

Hollerbach, J.M., "A Recursive Formulation of Lagrangian Manipulator Dynamics" IEEE Trans. Systems, Man, Cybernetics SMC-10, 11, 1980.

Hooker, W.W., and Margulies, G., "The Dynamical Attitide for an n-Body Satellite", J. Astronautical Sciences 12,4, 1965.

Luh, J.Y.S., and Lin C.S., "Automatic Generation of Dynamics Equations for Mechanical Manipulators", Proc. Joint Automatic Control Conference, 1981.

Mooring, B.W., "The Effect of Joint Axis Misalignments on Robot Positioning Accuracy", ASME Computers in Engineering, Vol. 2, 1983.

Paul, R.P., Robot Manipulators. MIT Press, 1981

Suh C., and Radcliffe C.W., Kinematics & Mechanism Design, John Wiley & Sons, 1978.

Uicker, J.J., On the Dynamic Analysis of Spatial Linkages Usisng 4 by 4 Matrices, Ph.D Thesis, Department of

Mechanical Engineering and Astronautical Sciences, Northwestern University, 1965.

Vechio L., Nicosia S., Nicola F., and Lentini D., "Automatic Generation of Dynamical Models of Manipulators", Journal of Dynamic Systems, Measur[illegible] Control, 1982.

Vukobratovic, M, Scientific Fundamentals of Robots 1, Dynamics of Manipulation Robots. New York, Springer-Verlay, 1982

Appendix A: Symbolic Generation of the Equations of Motion

DYNAM is an interactive computer program which generates symbolic equations of motion for arbitrary manipulators of up to six degrees of freedom. The algorithm is based upon the recursive Hollerbach 4x4 dynamics formulation.

The Hollerbach formulation evaluates manipulator joint forces as follows:

$$F_i = \text{Trace}\left[\frac{\partial T_i}{\partial q_i} B_i\right] - \bar{g}^T \frac{\partial T_i}{\partial q_i} c_i$$

Where c_i and B_i are evaluated from $i=n$ down to $i=1$, and are defined by:

$$c_i = m_i \, {}^i a_i + C_{i+1} \, c_{i+1}$$

$$B_i = J_i \, \ddot{T}_i^{\,T} + D_{i+1} \, B_{i+1}$$

T_i and T_i are evaluated from $i=1$ to n and are defined by:

$$\ddot{T}_i = \ddot{T}_{i-1} D_i + 2\dot{T}_{i-1} \frac{\partial D_i}{\partial q_i} q_i + T_{i-1} \frac{\partial^2 D_i}{\partial q_i^2} \dot{q}^2$$

$$+ T_{i-1} \frac{\partial D_i}{\partial q_i} \ddot{q}_i$$

$$\dot{T}_i = \dot{T}_{i-1} D_i + T_{i-1} \frac{\partial D_i}{\partial q_i} q_i$$

where:

$$T_0 = [I]$$

$$\dot{T}_0 = \ddot{T}_0 = [0]$$

$$B_{n+1} = [0]$$

$$c_{i+1} = [0]$$

J_i = inertia matrix

A flow chart for the evaluation of the equations is shown in Figure A.1. DYNAM generates the symbolic equations following this algorithm.

The program was written in Pascal for two reasons:

1. Pascal handles character and string manipulation very conveniently.
2. The versatility of the data structures available with Pascal tend to reduce algorithm complexity and improve self documentation.

Software has been developed for the symbolic manipulation of the following tasks:

1. Matrix addition
2. Matrix multiplication
3. Transpose of a matrix
4. Multiplication of a matrix by a constant
5. Trace of a matrix
6. Multiplication of a matrix by a vector
7. Generation of the first and second derivatives of the A and D matrices.

Of the above tasks the addition and multiplication algorithms proved to be the most complex. These algorithms must perform the operations while minimizing equation complexity. Many of the operations required by these algorithms require special consideration. Some of these tasks are:

1. The algorithms must not add zeros
2. Multiplications by zero must not be included in the

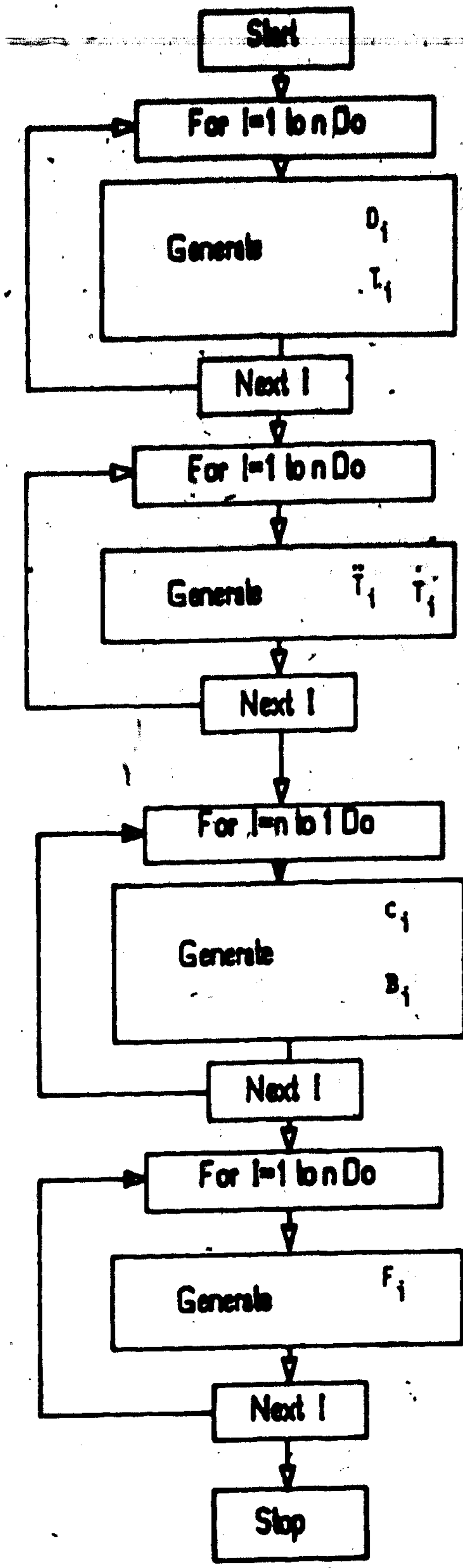

Figure A.1 Algorithm For Equation Generation Using the 4x4 Recursive Hollerbach Formulation symbolic equations.

3. Multiplications by one must not be performed
4. Algebraic hierarchy must be maintained while minimizing the proliferation of unnecessary groupings and brackets
5. Multiplication of positive and negative quantities must be handled appropriately

DYNAM has incorporated a simplification algorithm which minimizes the complexity of the symbolic equations. This simplification routine replaces all strings which contain more than one addition and/or two multiplications with a single variable. This algorithm was relatively simple to incorporate yet resulted in a dramatic improvement in the efficiency of equation evaluation.

DYNAM creates a FORTRAN callable subroutine which contains the symbolic equations for a given manipulator. This subroutine will evaluate the joint forces given the manipulators kinematic state.

Appendix B: Manipulator Kinematic and Dynamic Parameters

All of the analysis in this thesis has been based upon an idealized manipulator whose parameters are similar to those of the commercially available PUMA 600. The kinematic and dynamic properties are shown in Table B.1.

Table B.1 The Kinematic and Dynamic Properties of a Three Degree of Freedom Manipulator

	Link 1	Link 2	Link 3
Length (m)	0	.4572	.4318
Offset (m)	0	0	0
Twist (degrees)	90	0	0
Skew (degrees)	0	0	0
Mass (kg)	4.51	15.91	11.36
$\bar{x}$ (m)	0	-.128	-.085
$\bar{y}$ (m)	0	0	0
$\bar{z}$ (m)	0	0	0
Ixx (kg-m**2)	0.0038	0.1237	0.0074
Iyy (kg-m**2)	0.0038	0.1237	0.0074
Izz (kg-m**2)	0.017	0.9897	0.707