# AN ADAPTIVE CONTROL STRATEGY
# FOR INDUSTRIAL ROBOTIC SYSTEMS

A Thesis

Presented to

the Faculty of the Graduate School

University of Missouri - Columbia

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Michael S. King

Dr. J. Kenneth Blundell Thesis Supervisor

May, 1984

The undersigned, appointed by the Dean of the Graduate Faculty, have examined a thesis entitled,

AN ADAPTIVE CONTROL STRATEGY FOR

INDUSTRIAL ROBOT SYSTEMS

Presented by Michael S. King

A candidate for the degree of Master of Science, and hereby certify that in their opinion it is worthy of acceptance.

[redacted]

# ABSTRACT

Industrial robots in use today lack the ability to perceive and interact with their environment. This limitation is a major obstacle confronting robotic systems developers. This work outlines an adaptive control strategy which allows a robot to work within a time-varying environment. After developing the kinematic model of the robot and its relationship to its environment, the adaptive control strategy is simulated. The performance of the system is measured by using an index of performance and comparing the simulation results against a series of non-adaptive models. The results indicate that an adaptive control strategy significantly increases the performance of a robotic system operating in a time-varying environment.

CONTENTS

# LIST OF ILLUSTRATIONS

# 1.0 INTRODUCTION

The field of robotics is a challenging, multi-discipline area of emerging technology. A review of current research and industrial literature suggests that industrial robots are becoming a key element of this nation's efforts to reindustrialize.

## 1.1 Background

Although robotics research and the use of robots in the industrial workplace is increasing, early efforts in remote manipulators date back to the second world war (1), where teleoperators were used to handle radioactive materials. The word "robot" came into use as the result of a 1921 play, R.U.R. (Rossum's Universal Robot), written by Karel Capek (2). Robot is derived from the Czech word for "forced labor".

The modern definition of a robot comes from the Robot Institute of America:

> "A robot is a reprogrammable, multifunctional manipulator designed to move material, parts, tools or specialized devices, through variable

programmed motions for the performance of a variety of tasks."

With this definition in mind, the basic components of an industrial robot may be illustrated (Figure 1).

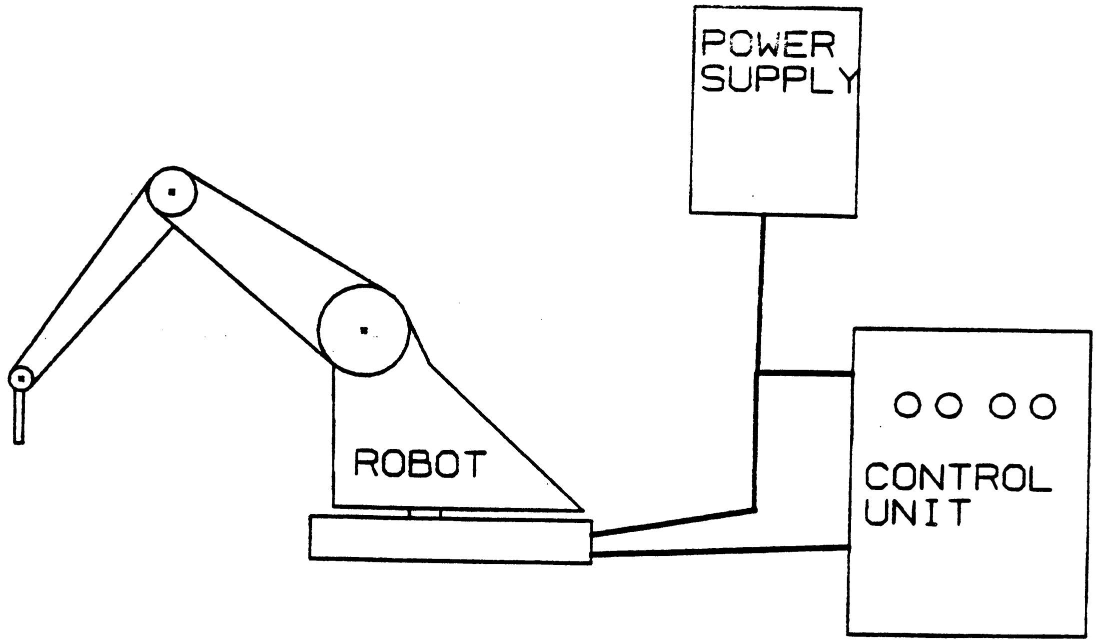


Figure 1.  3 Basic Elements of a Robot

The manipulator supplies the ability to "...move material, parts, tools or specialized devices...", the power supply provides the energy for the system to operate, and the controller makes the robot "...reprogrammable, multifunctional...". While Figure 1 is very simplistic, it does typify the configuration of most of the industrial robots in use today.

## 1.2 Current Control Strategies

Technological advances are being made in virtually every application area concerning industrial robots. Conferences, such as the AUTOFACT conference or the International Symposium on Industrial Robots, are yearly showcases of the lastest hardware and software for industrial use as well as forums for research findings. Robotics, however, has a malady similar in nature to that of the computer industry: namely, highly advanced hardware inovations abound but the software needed to take full advantage of these technological marvels is severely limited. In a sense, the two fields are intertwined. The advent of low cost, high performance microprocessors has broadened the potential areas of application of computers into previously unforseen areas (i.e. kitchen appliances, "personal computers"). The simultaneous impact on the robotics industry is the ability to create specialized, high speed sensor systems to aid the robot in adapting to its environment. These systems, working in conjunction with a sufficiently robust control architecture, would allow such tasks as assembly operations with force compliance to become routine work. This potential is currently being lost due to the lack of adequate control software for commercial robots.

Two methods of control are employed on most of the robots in use today: Point-to-Point (PTP) and Continuous Path (CP) (3). PTP controlled robots have the following characteristics:

a) Path definition is via discrete points in Cartesian space.
b) Path modification during program execution is generally difficult.
c) Since only the position of the end of the robot (i.e. the end tooling or gripper) is of concern, individual member motions are not generally controlled or even known.
d) Typical application areas are materials handling, tool changing, and machine loading and unloading.

CP controlled robots exhibit the following features:

a) Path definition is via manual movement of the arm through a desired trajectory, with path information captured on a time-based sampling interval (typically at 60 - 80 Hz)
b) Path modification during program execution is generally difficult.
c) The robot follows a smooth, continuous, con-

trolled path without noticeable speed changes from point to point.

d) Typical application areas are spray painting, polishing, and welding.

Recent literature (4)-(11) suggests that the standard control features available for industrial robots today cannot adequately provide the flexibility needed for advanced applications. Indeed, current research trends are in the areas of cognitive reasoning, image processing, advanced sensory perception and interfaces, and adaptive control. All of these research topics can broadly be placed into the field of Artificial Intelligence (AI). AI is an area within Computer Science which deals with problem solving, machine perception, and expert systems. Advances in the field of AI have been slow, due mainly to the limited capabilities of current software/hardware architectures. However, the potential benefit of AI research to the robotics industry is clear. The robotic control strategies for the late 80's and beyond will be based on advanced software systems utilizing AI techniques.

### 1.3 Present Work

The object of this study is to introduce the concepts of adaptive control into the field of robotic systems development. Several authors (2),(4), and (6), have presented candidate implementations of adaptive control strategies either at the robot (primitive) level or the supervisory (task) level. The current work examines the interaction of these two levels of adaptive control and the effects this interaction has on a particular robotics application.

## 2.0 FUNDAMENTALS

In order to positionally control a robot, some form of a kinematic model must be developed. The model must be concise and as computationally efficient as possible to allow it to be implemented in a real-time control strategy. For simple robot configurations (i.e. two or three degrees of freedom - DOF), direct trigonometric solutions are viable. But, as the number of DOF increases, the trigonometric solutions become unwieldy. Therefore, it would be beneficial to have a universal modelling scheme that would apply to any manipulator. Such a scheme has been developed, using matrix methods developed by Denavit and Hartenberg (12) for closed kinematic chains. Their basic formulation was extended to robotic manipulators (open kinematic chains) by Pieper and later by Paul, et al (1),(13).

### 2.1 Notation

The notational scheme to be used for the body of this material is as follows: lower-case characters will be used to denote point vectors in 3-D Euclidean space (e.g. v = a$\underline{i}$ + b$\underline{j}$ + c$\underline{k}$) and upper-case characters will be used to denote matrices and/or coordinate frames (e.g. A = |mxm| matrix). Other notational considerations will be presented when appropriate to clarify a particular section of development.

## 2.2 Homogeneous Coordinates and Transformations

A position (point) vector in space may be represented in a number of different ways. In the Cartesian coordinate system, a position vector may be written as

$$v = a\underline{i} + b\underline{j} + c\underline{k} \qquad (2.1)$$

where $\underline{i}$, $\underline{j}$, and $\underline{k}$ represent unit vectors along the x,y, and z axes, respectively. In order to represent v in homogeneous coordinates, an extra component, called the scale factor, is added to v. The representation of v then becomes

$$v = |x,y,z,w| \qquad (2.2)$$

where

$$\begin{aligned} a &= x/w \\ b &= y/w \\ c &= z/w \end{aligned} \qquad (2.3)$$

The representation of position vectors in homogeneous coordinates will facilitate the development of transformation matrices that include rotation, translation, perspective, and scaling transformations.

The concept of homogeneous coordinates is that any n-dimensional vector may be transformed, in n+1 dimensional space, without losing the original position vector. After transformation, it is always possible to recover the vector by reversing the transformation and/or dividing by the scale factor w. When using homogeneous coordinates to represent the position and orientation of objects and robots in their environment, the scale factor is unity.

Homogeneous transformations are 4x4 matrices that map a position vector expressed in homogeneous coordinates from one reference frame into another. The general form of a homogeneous transformation matrix (14) is

T =

| 3x3 rotation matrix | 3x1 translation matrix |
|---|---|
| 1x3 perspective transformation | scaling factor |

(2.4)

The physical interpretation of Eq.(2.4) is as follows:

a) the upper left 3x3 submatrix represents rotations about the coordinate axes;

b) the upper right 3x1 submatrix represents translations along the coordinate axes;

c) the lower left 1x3 submatrix represents perspective transformations; and

d) the lower right diagonal element is the global scale factor.

For the proposed use of homogeneous transformations in robotics applications, the perspective transformation is set to zeros and the scale factor is always unity.

The individual submatrices described above may themselves be represented as homogeneous transformations. Using these various representations, pure rotations, translations, or any combination thereof may be developed. Then, objects in 3-D space may be defined with respect to a global reference frame, to each other, or to a manipulator. The ability to form complex relationships between different reference frames using simple transformations and matrix algebra is the fundamental reason for adopting the homogeneous coordinate representation.

The homogeneous transformations representing pure rotations about the x, y, or z axes in Cartesian coordinates are notationally represented as Rot($x,\theta$), Rot($y,\theta$), Rot($z,\theta$), respectively. The notation is read, "Rotation about the x-axis by an angle $\theta$...". The corresponding transformations are

$$\mathrm{Rot}(x,\theta) = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & \mathrm{Cos}\theta & -\mathrm{Sin}\theta & 0 \\ 0 & \mathrm{Sin}\theta & \mathrm{Cos}\theta & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \tag{2.5}$$

$$\mathrm{Rot}(y,\theta) = \begin{vmatrix} \mathrm{Cos}\theta & 0 & \mathrm{Sin}\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\mathrm{Sin}\theta & 0 & \mathrm{Cos}\theta & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \tag{2.6}$$

$$\mathrm{Rot}(z,\theta) = \begin{vmatrix} \mathrm{Cos}\theta & -\mathrm{Sin}\theta & 0 & 0 \\ \mathrm{Sin}\theta & \mathrm{Cos}\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \tag{2.7}$$

To show how these rotations may be applied, consider a position vector, $p = 5\underline{i} + 2\underline{j} + 6\underline{k}$. First, rotate p about the z-axis by 90 degrees to position vector q. Using Eq.(2.7), where $\mathrm{Sin}\theta = 1$ and $\mathrm{Cos}\theta = 0$, we have

$$q = \mathrm{Rot}(z,90)p \tag{2.8}$$

$$\begin{vmatrix} -2 \\ 5 \\ 6 \\ 1 \end{vmatrix} = \begin{vmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} 5 \\ 2 \\ 6 \\ 1 \end{vmatrix} \tag{2.9}$$

Figure 2 illustrates the result of this transformation.

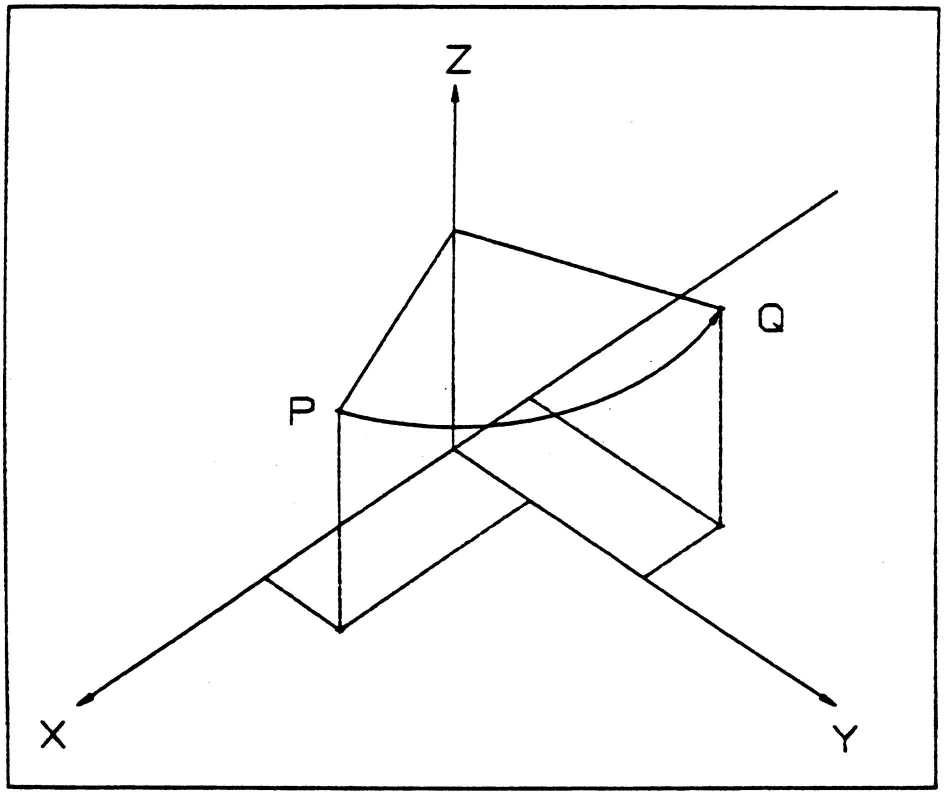


Figure 2. Rot(z,90)

If q were next rotated 90 degrees about the x-axis to r, Rot(x,90), we obtain (using Eq.(2.5)),

$$r = Rot(x,90)q \tag{2.10}$$

$$\begin{vmatrix} -2 \\ -6 \\ 5 \\ 1 \end{vmatrix} = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} -2 \\ 5 \\ 6 \\ 1 \end{vmatrix} \tag{2.11}$$

This result is shown in Figure 3.

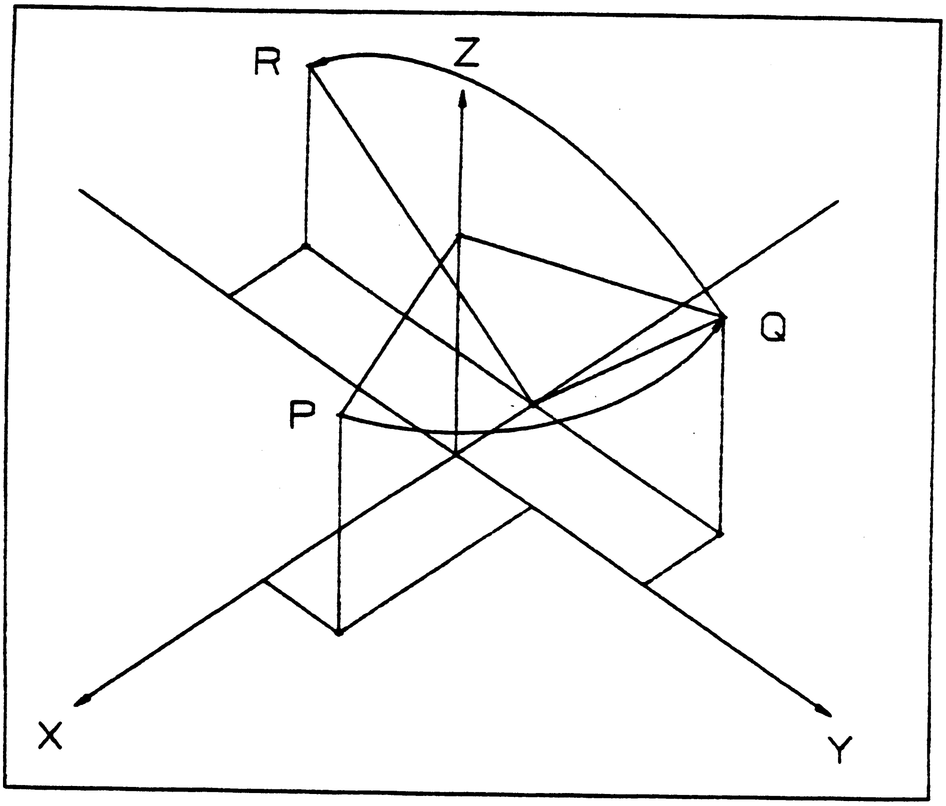


Figure 3. Rot(x,90)

By combining Eq.(2.8) and (2.10), a single rotational transformation may be developed to accomplish the same movements from p to r. First, substitute for q in Eq.(2.10) using Eq.(2.8) to obtain

$$r = Rot(x,90)Rot(z,90)p \qquad (2.12)$$

Next, perform the indicated multiplication of the rotation transformations

$$Rot(x,90)Rot(z,90) = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \qquad (2.13)$$

$$\mathrm{Rot}(x,90)\mathrm{Rot}(z,90) = \begin{vmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \tag{2.14}$$

Now, applying the new, compound rotational transformation to p, we obtain

$$\begin{vmatrix} -2 \\ -6 \\ 5 \\ 1 \end{vmatrix} = \begin{vmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} 5 \\ 2 \\ 6 \\ 1 \end{vmatrix} \tag{2.15}$$

which is the same result obtained in Eq.(2.11). The order in which the rotations are performed is important. Since matrix multiplication is non-commutative, a different final position would be arrived at if the rotation about the x-axis was performed first. The preceding example demonstrates the ease of creating compound transformations from the simple homogeneous "building blocks".

The homogeneous transformation representing pure translation along the x, y, or z axes in Cartesian coordinates is notationally represented as Trans(a,b,c). The notation is read "Translate the distance a along x, b along y, c along z". The corresponding transformation is

$$\mathrm{Trans}(a,b,c) = \begin{vmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{vmatrix} \tag{2.16}$$

Translation transformations may be included in vector transformations involving rotations in a manner analogous to that illustrated in Eq.(2.13). The result is a single homogeneous transformation which includes rotations and translations.

Since the perspective transformation is null and the scale factor always unity for robotics applications, the general form of the homogeneous transformation may be represented as

$$T = \begin{vmatrix} nx & ox & ax & px \\ ny & oy & ay & py \\ nz & oz & az & pz \\ 0 & 0 & 0 & 1 \end{vmatrix} \tag{2.17}$$

Equation (2.17) is the standard form for a 4x4 homogeneous transformation matrix (14). The columns may be thought of as column vectors n,o,a, and p. The upper left 3x3 rotational submatrix describes the orientation of some coordinate frame, D, with respect to some global coordinate frame, G. The vectors n,o, and a form an orthogonal set of vectors such that the cross product, o x a, yields n. The upper right 3x1 submatrix describes the position of the frame D with respect to G (see Figure 4).

The position and orientation of frame D {n',o',a',p'} may be arrived at as follows. First rotate about the z-axis of G 90 degrees (Rot(z,90)), then rotate about the x-axis of G -90 degrees (Rot(x,-90)), and finally translate the rotated

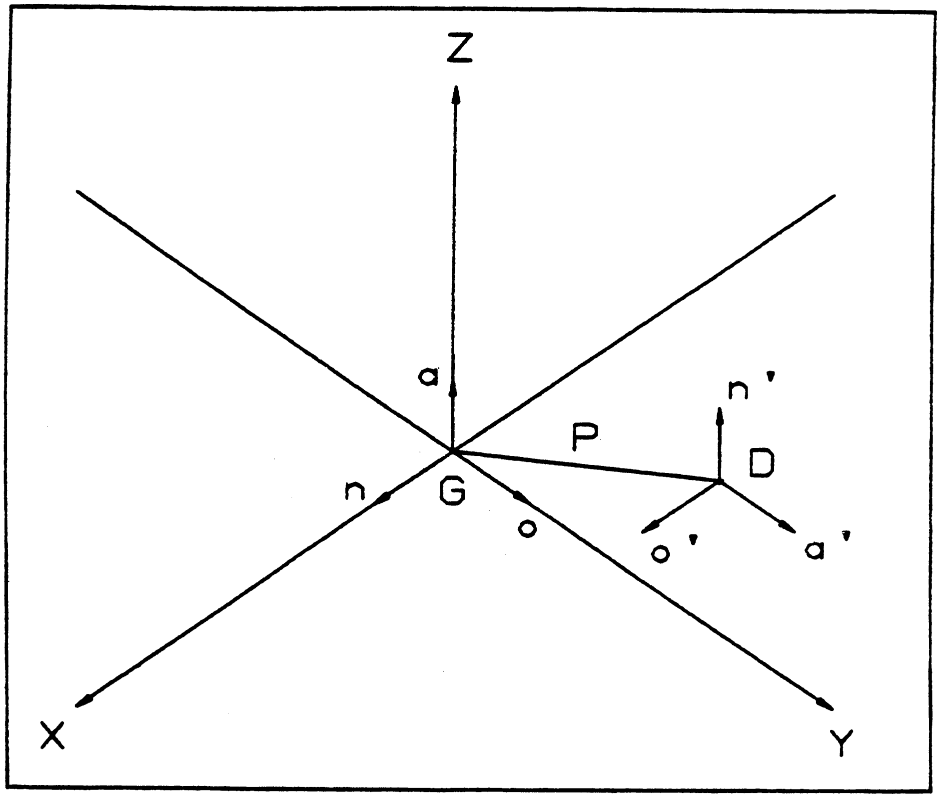


Figure 4. General Transformation Representation

frame D an amount $3\underline{i} + 5\underline{j} + 10\underline{k}$ (Trans(3,5,10)). The symbolic correspondence between G and D may then be written as

$$D = \text{Trans}(3,5,10)\text{Rot}(x,-90)\text{Rot}(z,90)G \qquad (2.18)$$

Upon evaluating the translation and rotation transformations, a general relationship between G and D, called T, is found to be

$$T = \text{Trans}(3,5,10)\text{Rot}(x,-90)\text{Rot}(z,90) \qquad (2.19)$$

or

$$T = \begin{vmatrix} 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 5 \\ 1 & 0 & 0 & 10 \\ 0 & 0 & 0 & 1 \end{vmatrix} \qquad (2.20)$$

Utilizing Eq.(2.20), it is now possible to describe vectors, objects, or even additional coordinate frames with respect to either D or G. To describe a vector, s, in frame G in terms of D, s is premultiplied by the transformation T, obtaining

$$^{D}s = Ts_{G} \qquad (2.21)$$

where the superscript "D" denotes "s with respect to D" and the subscript "G" denotes "s with respect to G".

If s were defined in terms of frame D, then it could be transformed into frame G by post multplying sD by T to get Gs. The ability to describe and move vectors, objects or coordinate frames using simple matrix algebra is a powerful aid in robotic systems development. With this capability, the task of programming path information into the robot controller, offline, may be accomplished by describing the path in the reference (global) coordinate frame instead of in terms of joint coordinates. The only requirement is that a homogeneous transformation exist which describes the position and orientation of the robot with respect to the global frame. Then, using this transformation, the global path data is transferred into the controller's memory with the data now represented in terms of the robot's coordinate frame.

Before leaving the subject of homogeneous transformations, one additional relationship is needed, the inverse of Eq.(2.17). The inverse of the general homogeneous transformation will be used repeatedly throughout the development of the kinematic model of the robot. It will also be used in the description of the robot's working environment. The physical interpretation of the inverse of T, denoted $T^{-1}$, is simply a description of the position and orientation of some reference coordinate frame with respect to some transformed frame. Using Eq.(2.20), the inverse of T may be written as

$$T^{-1} = \begin{vmatrix} 0 & 0 & 1 & -10 \\ 1 & 0 & 0 & -3 \\ 0 & 0 & 0 & -5 \\ 0 & 0 & 0 & 0 \end{vmatrix} \tag{2.22}$$

Since, in matrix algebra, a matrix multiplied by its inverse should yield the identity matrix, I

$$\begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} = \begin{vmatrix} 0 & 0 & 1 & -10 \\ 1 & 0 & 0 & -3 \\ 0 & 1 & 0 & -5 \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 5 \\ 1 & 0 & 0 & 10 \\ 0 & 0 & 0 & 1 \end{vmatrix} \tag{2.23}$$

which is shown in Eq.(2.23). The general form of $T^{-1}$ is, given T equal to

$$T = \begin{vmatrix} nx & ox & ax & px \\ ny & oy & ay & py \\ nz & oz & az & pz \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (2.24)$$

then the inverse is

$$T^{-1} = \begin{vmatrix} nx & ny & nz & -p.n \\ ox & oy & oz & -p.o \\ ax & ay & az & -p.a \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (2.25)$$

where the "." represents a vector dot product. This may be proven by post-multiplying Eq.(2.24) by Eq.(2.25).

## 2.3 Application to Robotics

In the previous two sections, the notational considerations and use of homogeneous transformations has been outlined. A more complete treatise on homogeneous transformations may be found in (1) or (14). With this background, the application of homogeneous transformations to robotics may be considered.

Any robot or manipulator may be thought of as an open kinematic chain. Each link in the chain is connected by some form of joint, usually revolute (purely rotational) or prismatic (purely translational). By placing a body-fixed coordinate system on each link, and then relating these various coordinate systems using homogeneous transformations, the relative position and orientation of the links may be described (1). The traditional notation for a homogeneous transformation which describes the relation of one link to another is called an A matrix (12). An A matrix is used to define the interlink relationships between adjacent pairs of links in a kinematic chain. In order to describe the position and orientation of the second link in a two-link chain in the global reference frame, the matrix product

$$T_2 = A_1 A_2 \tag{2.26}$$

is evaluated. This product of A matrices is continued until all the interrelationships between links are defined. For robotics, this product, Ti, represents the position and orientation of the manipulator in 3-D space.

If a five link robot is considered, then Ti becomes

$$T_5 = A_1 A_2 A_3 A_4 A_5 \tag{2.27}$$

Since a five link robot may have 5 DOF, one for each link-joint pair, $T_5$ represents 3 DOF for positioning and 2 DOF for orientation.

The primary concern when developing a control strategy for a robot is the position and orientation of the tool center point (TCP) in relationship to its work. If the center point between the fingers of the end effector or "hand" is taken as the TCP, and the coordinate system in Figure 5 is assigned as shown, then physical significance can be related to $T_5$.

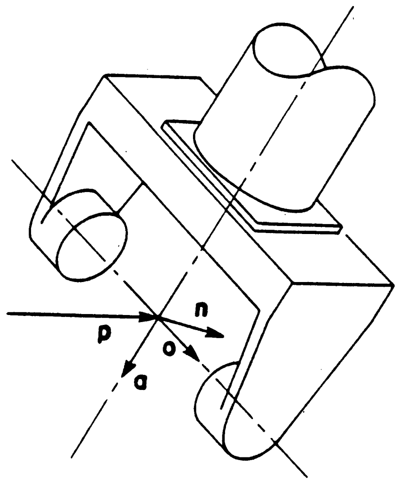


Figure 5. Hand-based Coordinate System

The vector p gives the position of the hand. The vector a, called the approach vector, points in the direction that the hand would approach an object. The vector o, called the orientation vector, gives the orientation of the hand from fingertip to fingertip. The fourth vector n, the normal vector, forms an orthogonal set of vectors with o and a such that

$$n = o \times a \tag{2.28}$$

Using vectors n,o,a, and p, the manipulator's $T_5$ transformation matrix becomes

$$T_5 = \begin{vmatrix} nx & ox & ax & px \\ ny & oy & ay & py \\ nz & oz & az & pz \\ 0 & 0 & 0 & 1 \end{vmatrix} \qquad (2.29)$$

What may be concluded from Eq.(2.29) is, if $T_5$ is known for a particular robot, then the position and orientation of the TCP (or hand) may be calculated for any possible orientation the robot might assume. The knowledge of T is essential in order to control the path of the robot during task execution. The development of a particular robot's T matrix is the topic of the next section.

## 3.0 DEVELOPMENT OF THE KINEMATIC MODEL

In Section 2, the concept of homogeneous transformations and their application to robotics was presented. In order to utilize homogeneous transformations for the control of a robot, a series of coordinate system assignments must be made. Once this task is accomplished then the A matrices, which describe the interlink relationships, may be developed. From these A matrices, the T matrix for the robot may be obtained using Eq.(2.27). With T defined, the motion variables for the robot may be solved for, giving a closed-form solution to the kinematics problem.

### 3.1 Coordinate System Assignments for the Robot

In this work, a particular robot has been selected for the kinematic model. The robot, called MICROBOT, is a product of MICROBOT, INC., Mountain View, CA.. The justification for selecting this particular robot out of the large number available to industry/acedemia is the School of Engineering at UMKC has several of these robots available for instructional and research purposes.

The first step in developing the kinematic model is the assignment of coordinate systems to the robot. To accomplish this, an understanding of how rigid links in a kinematic chain are related to one another is necessary (see Figure 6)(14).

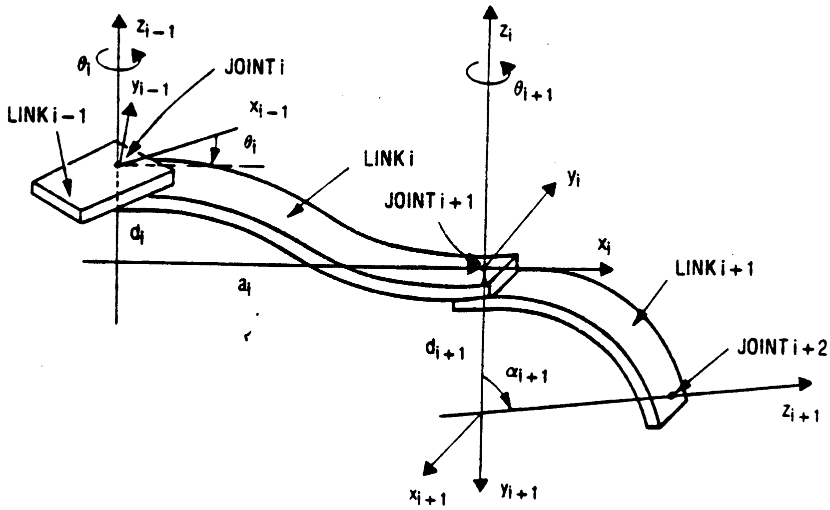


Figure 6. Link Coordinate System and Parameters

A joint axis, for joint i, is established at the point of connection between link i-1 and link i. This joint axis, zi-1, has two normals to it, one for each link. The relative position of the two links, di, is the distance along zi-1 between the two normals. The angle between the links, $\theta$i, is measured in a plane perpendicular to a plane formed by zi-1 and the link i normal. di and $\theta$i are called the "distance" and "angle" between adjacent links (14).

A link i is connected to at most two additional links, i-1 and i+1. Therefore, a given link will have two joint axes associated with it. The important point, in a kinematic sense, is the links maintain a fixed relationship to one another. This relationship is specified by two additional parameters, ai and $\alpha$i. ai is the shortest distance along a common normal between the two joint axes, zi-1, zi. $\alpha$i is the angle made between the two joint axes, measured in a plane perpendicular to ai. ai and $\alpha$i are called the "length" and "twist angle" of link i, respectively (14).

The four parameters described above, di, $\theta$i, ai, and $\alpha$i, completely define the relationship and structure of any two adjacent links in a kinematic chain. Because the MICROBOT has only revolute (purely rotational) joints, the joint variable to be solved for in order to arrive at a kinematic model is $\theta$i.

The coordinate system assignments for the MICROBOT are made in the following manner. Working outward from the base (see Figure 7), assign joint axis $z_0$ to the base joint, with the origin of this frame positioned to coincide with the connection point of link 1 and link 2 (the shoulder joint). The origin of the end of the arm coordinate frame, link 5, is positioned to coincide with link 4. An additional frame will be specified later to relate the hand to the arm.

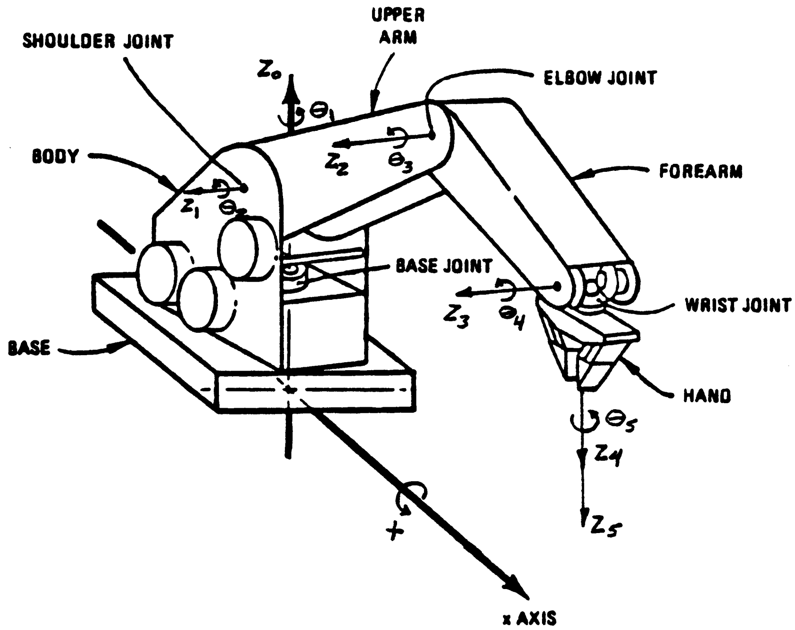


Figure 7. MICROBOT Coordinate System Assignments

## 3.2 Specification of $T_5$ for the MICROBOT

The A matrices which describe the interlink rotational and translational relationships may be derived using the following algorithm (1).

a) Rotate about zi-1 an angle θi;

b) translate along zi-1 a distance di;

c) translate along rotated xi-1 = xi a length ai;

and

d) rotate about xi, the twist angle $\alpha$i.

Steps (a) through (d) may be expressed in terms of four homogeneous transformations which will relate the coordinate frame of link i to link i-1. Notationally, (a) through (d) become

$$Ai = Rot(z,\theta)Trans(0,0,d)Trans(a,0,0)Rot(x,\alpha) \qquad (3.1)$$

Recalling the homogeneous transformations represented above, (i.e. Eq.(2.7), (2.16), and (2.5)), the general form of the A matrix becomes

$$A = \begin{vmatrix} Cos\theta & -Sin\theta Cos\alpha & Sin\theta Sin\alpha & aCos\theta \\ Sin\theta & Cos\theta Cos\alpha & -Cos\theta Sin\alpha & aSin\theta \\ 0 & Sin\alpha & Cos\alpha & d \\ 0 & 0 & 0 & 1 \end{vmatrix} \qquad (3.2)$$

With this definition of the A matrix, a 4x4 homogeneous transformation, the $T_5$ transformation describing the position and orientation of the end of the manipulator may be obtained. Table 3.1 gives the values of the link parameters for the MICROBOT.

TABLE 3.1 Link Parameters for the MICROBOT

| Link | Variable | $\alpha$ | a | d | Cos$\alpha$ | Sin$\alpha$ | Range |
|---|---|---|---|---|---|---|---|
| 1 | $\theta_1$ | +90 | 0 | 0 | 0 | 1 | ±90 |
| 2 | $\theta_2$ | 0 | $L_2$ | 0 | 1 | 0 | +144,-35 |
| 3 | $\theta_3$ | 0 | $L_3$ | 0 | 1 | 0 | +0,-149 |
| 4 | $\theta_4$ | -90 | 0 | 0 | 0 | -1 | ±90 |
| 5 | $\theta_5$ | 0 | 0 | 0 | 1 | 0 | ±180 |

The following shorthand notation will be used during the development of the $T_5$ matrix for the MICROBOT.

$$\sin\theta i = Si$$
$$\cos\theta i = Ci$$
$$\sin(\theta i + \theta j) = Sij$$
$$\cos(\theta i + \theta j) = Cij$$

Using Eq.(3.2) and the parameters from Table 3.1, the A matrices of the MICROBOT are:

$$A_1 = \begin{vmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \tag{3.3}$$

$$A_2 = \begin{vmatrix} C_2 & -S_2 & 0 & L_2C_2 \\ S_2 & C_2 & 0 & L_2S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \tag{3.4}$$

$$A_3 = \begin{vmatrix} C_3 & -S_3 & 0 & L_3C_3 \\ S_3 & C_3 & 0 & L_3S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \tag{3.5}$$

$$A_4 = \begin{vmatrix} C_4 & 0 & -S_4 & 0 \\ S_4 & 0 & C_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \tag{3.6}$$

$$A_5 = \begin{vmatrix} C_5 & -S_5 & 0 & 0 \\ S_5 & C_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \qquad (3.7)$$

In order to calculate $T_5$, the product of the A matrices must be evaluated, starting at the end of the manipulator and working back toward the base.

$$^4T_5 = A_5 = \begin{vmatrix} C_5 & -S_5 & 0 & 0 \\ S_5 & C_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \qquad (3.8)$$

$$^3T_5 = A_4\,{}^4T_5 = \begin{vmatrix} C_4C_5 & -C_4S_5 & -S_4 & 0 \\ S_4C_5 & -S_4S_5 & C_4 & 0 \\ -S_5 & -C_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \qquad (3.9)$$

Any time the revolute joint axes are parallel, the following trigonometric identities may be employed to simplified the expressions.

$$\begin{aligned} SinACosB + CosASinB &= Sin(A+B) = SAB \\ CosACosB - SinASinB &= Cos(A+B) = CAB \end{aligned} \qquad (3.10)$$

$$^2T_5 = A_3\,{}^3T_5 = \begin{vmatrix} C_{34}C_5 & -C_{34}S_5 & -S_{34} & L_3C_3 \\ S_{34}C_5 & -S_{34}S_5 & C_{34} & L_3S_3 \\ -S_5 & -C_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \qquad (3.11)$$

$$^1T_5 = A_2\,{}^2T_5 = \begin{vmatrix} C_{234}C_5 & -C_{234}S_5 & -S_{234} & C_{23}L_3+C_2L_2 \\ S_{234}C_5 & -S_{234}S_5 & C_{234} & S_{23}L_3+S_2L_2 \\ -S_5 & -C_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \qquad (3.12)$$

Performing the last multiplication, $T_5$ is found to be

$$T_5 = A_1{}^1T_5 = \begin{vmatrix} nx & ox & ax & px \\ ny & oy & ay & py \\ nz & oz & az & pz \\ 0 & 0 & 0 & 1 \end{vmatrix} \tag{3.13}$$

where

$$\begin{aligned}
nx &= C_1C_{234}C_5 - S_1S_5 \\
ny &= S_1C_{234}C_5 + C_1S_5 \\
nz &= S_{234}C_5 \\
ox &= -C_1C_{234}S_5 - S_1C_5 \\
oy &= -S_1C_{234}S_5 + C_1C_5 \\
oz &= -S_{234}S_5 \\
ax &= -C_1S_{234} \\
ay &= -S_1S_{234} \\
az &= C_{234} \\
px &= C_1(C_{23}L_3 + C_2L_2) \\
py &= S_1(C_{23}L_3 + C_2L_2) \\
pz &= S_{23}L_3 + S_2L_2
\end{aligned} \tag{3.14}$$

Equations (3.13) and (3.14) represent the $T_5$ transformation matrix for the MICROBOT. As mentioned previously, the n, o, a, and p column vectors that constitute $T_5$ are the position and orientation of the end of the manipulator with respect to some, as yet unspecified, reference frame. Also, a check of the validity of this development may be performed by evaluating the cross product of o and a (o x a). The result of this operation should be the first column of the $T_5$ matrix, n, since n, o, and a form an orthogonal set of vectors. By using the values indicated in Eq.(3.14) for o = $ox\underline{i} + oy\underline{j} + oz\underline{k}$ and a = $ax\underline{i} + ay\underline{j} + az\underline{k}$, n is found to be equal to the cross product of o and a.

The preceding development has resulted in the formulation of the $T_5$ homogeneous transformation matrix for the MICROBOT.

With the expression for $T_5$ (Eq.(3.13) and (3.14)) and knowing the joint variable values, the position and orientation of the end of the manipulator may be found for any possible configuration. However, it would be more useful from a control standpoint if the joint variables, $\theta_1$ - $\theta_5$, could be found given a position in 3-D space. Since the path the robot is to follow to perform a given task is generally known in the reference coordinate frame, what is needed is a series of closed-form expressions relating the $\theta i$'s to the reference frame. These expressions will be obtained in the next section.

## 3.3 Solution for the Joint Variables, $\theta i$

$T_5$ is the product of the homogeneous transformations which describe the relative position of one manipulator link with respect to another (the A matrices),

$$T_5 = A_1 A_2 A_3 A_4 A_5 \tag{3.15}$$

In Eq.(3.15) above, $A_1$ is the position and orientation of link 1, $A_2$ is the position and orientation of link 2 with respect to link 1 and so on. By successive pre-multiplications of the inverse of the A matrices, Eq.(3.15) gives five matrix equations that will be used to solve for the joint variables $\theta_1$ - $\theta_5$.

$$A_1^{-1}T_5 = {}^1T_5 \tag{3.16}$$

$$A_2^{-1}A_1^{-1}T_5 = {}^2T_5 \tag{3.17}$$

$$A_3^{-1}A_2^{-1}A_1^{-1}T_5 = {}^3T_5 \tag{3.18}$$

$$A_4^{-1}A_3^{-1}A_2^{-1}A_1^{-1}T_5 = {}^4T_5 \tag{3.19}$$

First, it will be necessary to find the inverse of $A_1$ - $A_4$. Using Eq.(2.25), the inverse A matrices are

$$A_1^{-1} = \begin{vmatrix} C_1 & S_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ S_1 & -C_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \tag{3.20}$$

$$A_2^{-1} = \begin{vmatrix} C_2 & S_2 & 0 & -L_2 \\ -S_2 & C_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \tag{3.21}$$

$$A_3^{-1} = \begin{vmatrix} C_3 & S_3 & 0 & -L_3 \\ -S_3 & C_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \tag{3.22}$$

$$A_4^{-1} = \begin{vmatrix} C_4 & S_4 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -S_4 & C_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \tag{3.23}$$

The method to be used in solving for the joint variables, $\theta_1$ - $\theta_5$, will be to search each of the matrix equations, (3.15) - (3.19), to find expressions in which one of the $\theta i$'s we are looking for is present alone. Since, in Eq.(3.15) - (3.19), the equal sign implies element by element equality, there are 12 non-trivial expressions per equation from which the solution for a particular $\theta i$ may be found.

Now, starting with Eq.(3.16) and (3.20) plus Eq.(3.12) for $^1T_5$,

$$\begin{vmatrix} C_1 & S_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ S_1 & -C_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} nx & ox & ax & px \\ ny & oy & ay & py \\ nz & oz & az & pz \\ 0 & 0 & 0 & 1 \end{vmatrix} =$$

$$\begin{vmatrix} C_{234}C_5 & -C_{234}S_5 & -S_{234} & C_{23}L_3+C_2L_2 \\ S_{234}C_5 & -S_{234}S_5 & C_{234} & S_{23}L_3+S_2L_2 \\ -S_5 & -C_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \qquad (3.24)$$

Carrying out the indicated multiplication on the left hand side (LHS) of Eq.(3.24), it becomes

$$LHS = \begin{vmatrix} C_1nx+S_1ny & C_1ox+S_1oy & C_1ax+S_1ay & C_1px+S_1py \\ nz & oz & az & pz \\ S_1nx-C_1ny & S_1ox-C_1oy & S_1ax-C_1ay & S_1px-C_1py \\ 0 & 0 & 0 & 1 \end{vmatrix} \qquad (3.25)$$

By taking the (3,4) elements from Eq.(3.24) and (3.25), an expression is found which isolates $\theta_1$.

$$S_1px - C_1py = 0$$

$$S_1/C_1 = py/px$$

The tangent of an angle is $\sin\theta/\cos\theta$, so $\theta_1$ is found to be

$$\theta_1 = \tan^{-1}(py/px) \qquad (3.26)$$

The use of the arctangent will be the standard procedure throughout the rest of the solutions for $\theta i$. The rational is that it eliminates angular ambiquities that arise whenever the sine, cosine, arcsine and arccosine functions are used. This translates to degeneracies due to positions the manipulator might assume (1).

Having found $\theta_1$, the LHS of Eq.(3.24) is completely defined and we now search for additional expressions which are a function of a single joint variable. Elements (1,3) and (2,3) give the following:

$$(1,3):\quad C_1ax + S_1ay = -S_{234}$$

$$(2,3):\quad az = C_{234}$$

and, by multiplying through (1,3) by -1 and dividing, we obtain

$$S_{234}/C_{234} = (-C_1ax - S_1ay)/az$$

and $\theta_{234}$ is

$$\theta_{234} = \mathrm{Tan}^{-1}((-C_1ax - S_1ay)/az) \qquad (3.27)$$

With $\theta_{234}$ known, three additional $\theta i$'s may be solved for ($\theta_3$, $\theta_5$, $\theta_2$). The development of the solutions follows as above, so only the results are shown.

$$\theta_5 = \mathrm{Tan}^{-1}((oz/S_{234}(S_1ox - C_1oy)) \qquad (3.28)$$

$$\theta_3 = \mathrm{Tan}^{-1}\left|\frac{-(1 - C_3{}^2)^{1\,2}}{\dfrac{(C_1px + S_1py)^2 + pz^2 - L_2{}^2 - L_3{}^2}{2L_2L_3}}\right| \qquad (3.30)$$

$$\theta_2 = \mathrm{Tan}^{-1}\left|\frac{(C_3L_3 + L_2)pz - S_3L_3(C_1px + S_1py)}{(C_3L_3 + L_2)(C_1px + S_1py) + S_3L_3pz}\right| \qquad (3.31)$$

Since the MICROBOT has three parallel revolute joints out to the end of link 3, no new information will be obtained until link 3 is constrained, so we look to solve Eq.(3.18).

$$A_3^{-1}A_2^{-1}A_1^{-1}T_5 = {}^3T_5$$

The LHS of the above equation is

$$LHS = \begin{vmatrix} C_3 & S_3 & 0 & -L_3 \\ -S_3 & C_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} C_2 & S_2 & 0 & -L_2 \\ -S_2 & C_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} f_{11} & f_{12} & f_{13} & f_{14} \\ f_{21} & f_{22} & f_{23} & f_{24} \\ f_{31} & f_{32} & f_{33} & f_{34} \\ f_{41} & f_{42} & f_{43} & f_{44} \end{vmatrix} \qquad (3.32)$$

where

$$f_{11} = C_1nx + S_1ny$$

$$f_{21} = nz$$

$$f_{31} = S_1nx - C_1ny$$

$$f_{41} = 0$$

.
.
.
etc

Carrying out the indicated multiplication and equating candidate terms with the right hand side of the equation above, which is

$$^3T_5 = \begin{vmatrix} C_4C_5 & -C_4S_5 & -S_4 & 0 \\ S_4C_5 & -S_4S_5 & C_4 & 0 \\ -S_5 & -C_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$\theta_4$ is found to be

$$\theta_4 = \mathrm{Tan}^{-1}\left|\frac{(C_2C_3 - S_2S_3)(C_1ax + S_1ay) + (S_2C_3 + C_2S_3)az}{(C_2S_3 + S_2C_3)(C_1ax + S_1ay) + (S_2S_3 - C_2C_3)az}\right|$$

(3.33)

This completes the development of the solutions for the joint variables, $\theta_1$ - $\theta_5$, for the MICROBOT. With these solutions, we are now able to specify the position and orientation of the robot, given $T_5$ as input. The approach would be to read the current location of the robot, which would give values to the o, a, and p vector components that compose $T_5$, then use the solutions obtained in this section to calculate the individual joint angles. The implication of the existence of these closed-form expressions for the joint variables is that we now have a tool for predicting the joint angles (and, indirectly joint rates) needed in order to follow paths in space.

## 4.0 TASK DESCRIPTION

Based on the preceding development, in which the tools necessary to effectively model and control the kinematics of a robot were defined, this section will look at the specific application of these tools to a materials handling problem. The environment in which the MICROBOT is to work will be described, along with the methods used for path development and the constraints that the robot must operate within.

### 4.1 Working Environment

Many of the industrial robots in use today are primarily involved with materials handling applications. Simple pick and place robots have been utilized for many years in industry, mainly because of their reliability and the ease with which they could be reprogrammed (i.e. adjustment of limit switches).

However, because the pick and place or "bang-bang" type of robot does not have an advanced controller associated with it, its flexibility is limited to very "structured" tasks. As will be seen, the task currently under investigation does not fit the above category.

The work place into which the MICROBOT will be placed is shown in Figure 8. The physical layout of the area is as follows. Parts are supplied to the robotic transfer station by the input conveyor. The robot is positioned such that its maximum amount of base rotation (180 degrees) is utilized. The center of rotation for the base is at (-8,8,0). Both of the conveyors are eight inches wide and continuous belt types. Point B is the beginning of tracking position (BOT) for the transfer. Point K is the end of tracking position (EOT) for the transfer. There are no physical obstructions in the working volume of the robot except the two conveyor belts. The tolerance distance for the input tracking phase of the transfer (distance from B to F) is eight inches, as is the output tolerance distance (distance from G to K). All positions defined for the current task are within the working volume of the MICROBOT.

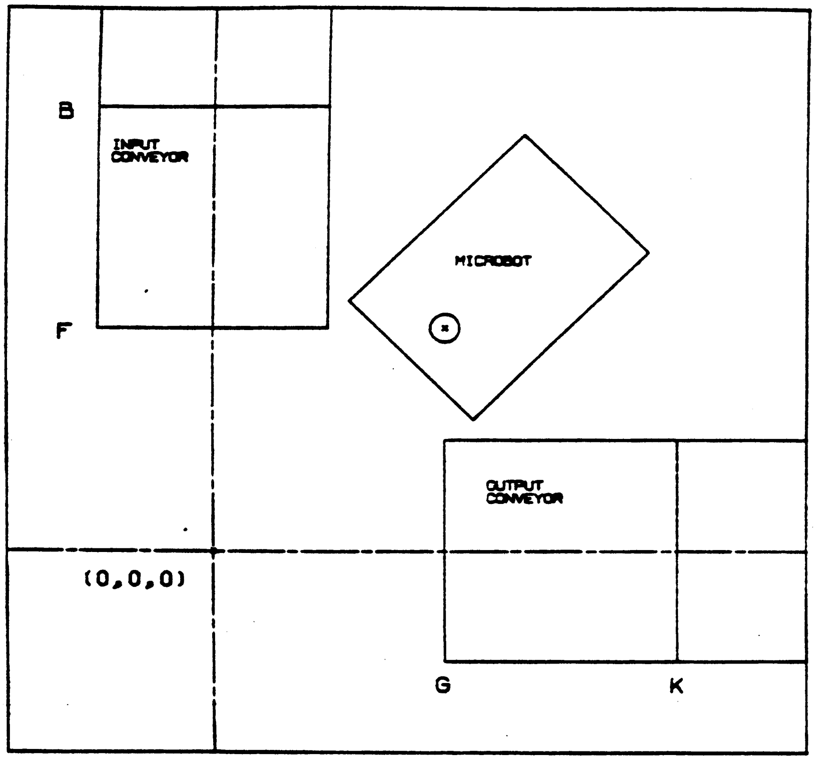

Figure 8. Working Environment

## 4.2 Path Development

The MICROBOT is a point to point (PTP) type robot as described in Section 1.2. Because of this, the path to be defined must consist of discrete points in 3-D space. This is not a problem, however, since there is a homogeneous transformation which will allow any path the robot can physically follow to be programmed into the MICROBOT's controller (the $T_5$ matrix plus the expressions for the joint variables). The full path that the MICROBOT must follow in order to perform its transfer operation successfully is shown in Figure 9. This figure is a three-dimensional representation of the working environment. The annotations "R I, R II, R III, R IV" are the four regimes which the intended path is divided. Each regime has a specific set of operational requirements associated with it. The operational requirements will be enumerated in the constraints section.

The R I portion of the path starts at the BOT:(-16,0,3), procedes along the centerline of the input conveyor (i.e. linear tracking type motion), up to and including Point F:(-8,0,3). The minimum tracking height during this portion of the path is 1.0 inches. LO denotes the point at which the robot picks the part up off of the input conveyor. All five joint variables are changing during this motion.

The R II portion of the path starts at F and procedes to Point G:(0,8,3) at a constant height of 3.0 inches. This represents 90 degrees of base rotation, the only joint involved in the move.

The R III portion of the path begins at G, follows the centerline of the output conveyor, up to and including the EOT:(0,16,3). The minimum tracking height is 1.0 inches. TD denotes the point at which the part has been released by the MICROBOT. All of the joint variables change during this

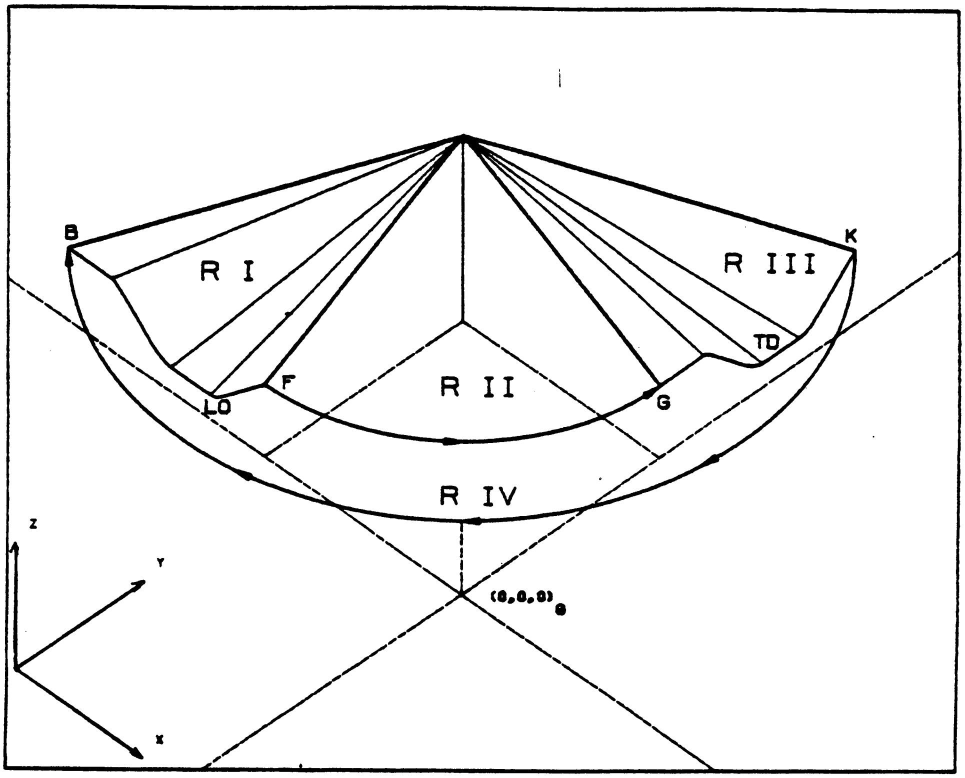


Figure 9. Path Definition for the Robot

motion.

The R IV portion of the path is the return portion of the transfer cycle. It starts at EOT and procedes back to B, a base rotation of 180 degrees. The base joint and, over two segments the hand roll joint, are involved in the motion.

The discrete points which define the path segments in R I and R III were generated using two techniques. First, the linear portions of the paths were created by simply dividing the segments into equal distance pieces based on the conveyor being tracked (i.e. x divisions along the input conveyor, y divisions along the output conveyor). The portions of the path that involved increasing or decreasing the tracking height were obtained using a Fowler-Wilson cubic spline routine. This routine assures that the curve tangents at the endpoints of the spline are continuous, which is desirable to avoid joint angle rate of change requests from becoming infinite. This is analogous to a function having a continuous first derivative.

The path segments in R II and R IV were generated by simple subdivision of the arcs decribed by the desired motions. In all, the total path contains 460 points in space, with the average interpoint distance being approximately 0.11 inches. This corresponds to 459 individual moves for the robot to be controlled through.

## 4.3 Operational Constraints

As mentioned above, each of the four regimes along the robot's transfer path have operational constraints. The operating policy for the robotic transfer station is as follows.

a) The input parts are being supplied via a manual assembly process with a probabilistic distribution (mean = 22.5872 seconds)

b) During R I tracking, the robot must be synchronized with the speed of the input conveyor belt and track a straight line.

c) At liftoff (LO), the maximum transfer rate of the robot is confined to the maximum rate of change of the shoulder joint.

d) The part must be rotated 90 degrees during transfer in R II to be properly aligned and ready for delivery to the output conveyor.

e) During output tracking, the robot must be synchronized with the speed of the output conveyor and track a straight line.

f) The time taken in order to transfer the part to the TD point must be such that the parts leave the transfer station equally spaced apart (45.00 inches ±.25).

g) During the return portion of the transfer cycle, R IV, the hand must be re-oriented such that it is in the proper position by the time the robot reaches the BOT.

h) The cycle time for the transfer should be less than or equal to the mean of the input parts distribution.

The MICROBOT, as supplied from the manufacturer, can not meet the requirements of the application described above. The task requires not only the ability to vary the joint rates of the robot, but also demands an active participation between the robot and its environment. It is the combination of a probabilistic input parts distribution, coupled with the requirement for equally spaced parts on the output conveyor, that makes an adaptive control strategy the only viable solution to this application if a flexible piece of automation is to be used.

# 5.0 THE ADAPTIVE CONTROLLER

The time-varying nature of most manufacturing operations makes the application of flexible automation to such tasks very difficult. While the robotics hardware being developed is capable of being driven to meet the requirements of an uncertain environment, sensory technology and the control algorithms to utilize that technology are lagging well behind. It is this lag between what is now and what is needed in industry that drives the applied research efforts in the robotics field.

## 5.1 Elements of Control

Adaptive control applied to the manufacturing world is not a new concept. Research efforts in the area of adaptively controlled machine tools dates back to the early 1960's (15). The problems encountered during the early stages of development of adaptive control were a combination of economic and technological constraints. These deficiencies are equally relevant today. The technological constraints were primarily due to the inability to measure the necessary process variables, in real time, needed to calculate the machine's index of performance (IP). The accurate accounting of the machine's IP is of fundamental importance if the tool is to be optimally controlled. The economic constraints arose when the early versions of these machine tools were to be put into production. The researchers quickly discovered that in order to make adaptively controlled systems viable for production, a subset of the capabilities of the prototype controllers was all that could be offered and still remain competitive in the marketplace. Fortunately, many of the early technical and economic problems are being resolved today.

Adaptive control shares traits with two other common forms of control strategy (i.e. feedback and optimal). Like feedback control, adaptive control must have sensory input from the system variables to determine the current state of the system. Also, like optimal control, adaptive control uses a measurement of the overall system performance. As mentioned above, this is the index of performance or IP. Unlike either of these two control strategies, adaptive control systems are designed to operate in a time-varying environment.

Adaptive control consists of three functions (see Figure 10).

1) Identification. Based on sensory data feedback, the controller must calculate the cur-

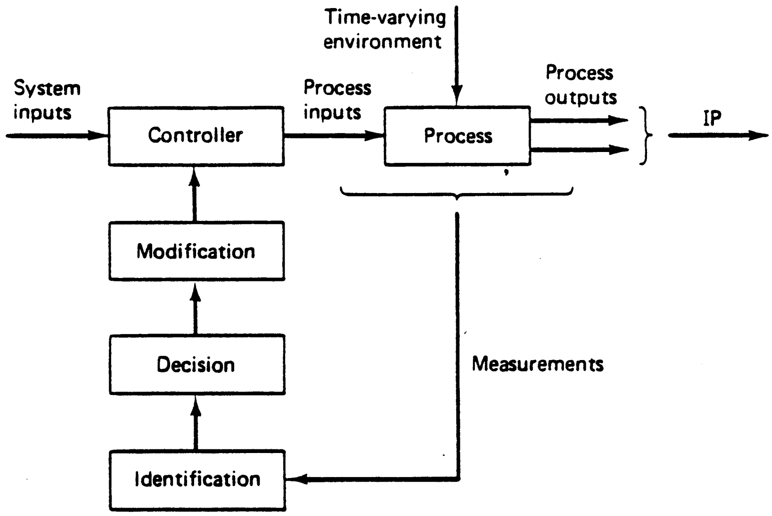


Figure 10. Three Functions of Adaptive Control

rent state of the system. Once this has been accomplished, the current state is compared to some desired state or goal for the system's performance. This might included estimating a model of the current system state in order to achieve the current system IP.

2) Decision. With the current state of the system known, the controller must determine which of the system variables to modify in order to attain the desired operating level. This might include updating the controller's "world" model to reflect a transition to a new

operating envelope or some modification to the internal parameters of the controller itself.

3) Modification. Once the software has determined the necessary changes to be made to meet the IP of the system, the modifications may then be made. Modifications are the physical changing of some system variable such as to drive the system towards the optimum IP. This is a hardware function, unlike the decision function which is logic based.

Having defined what an adaptive control system consists of, the control strategy for the materials handling problem may now be addressed.

## 5.2 Model Development

The method for developing flexible manufacturing stations or "cells" that is most commonly used today is via simulation. Because of the high cost of most applications in which a combination of numerically controlled machining centers, robots, and automated materials handling systems are to be used, simulation of the system prior to commissioning is very beneficial. Simulation studies allow the critical system performance parameters to be identified early in the development cycle. They also permit many different possible configurations to be proposed and tested without disruption of the current manufacturing facility. In addition, once the system is installed, the model used to develop the cell or manufacturing line is useful to aid in the assessment of system impact due to changing operating conditions or a different product mix. Because of the benefits listed above, the adaptive control strategy for the materials handling problem being investigated in this work will be simulated.

The operational contraints, as outlined in Section 4.3, give the requirements for the adaptive controller. The two most critical constraints imposed on the transfer station are the probabilistic nature of the upstream process (the parts supply) and the requirement for equally spaced parts on the output conveyor. The variablility of the parts supply necessitates the interaction of the controller and the input conveyor. This also requires the cycle time for the transfer task to vary from part to part, which means the robot's velocity must be adapted on a part by part basis to ensure that each part is transferred. The second constraint requires the controller to interact with the output conveyor to keep the parts being transferred equally spaced.

Because of the importance of the two contraints mentioned above, they are selected to represent the index of performance of the transfer station. The relationship to be established between these two system constraints will be developed in the time domain. The input parts arrive at irregular intervals to the transfer station and their inter-arrival times are stored from part to part. The time required to transfer a part depends on the combination of the arrival time of the next part to the station and the time required to transfer the current part to the output conveyor. Therefore, the IP for the MICROBOT transfer station is defined to be

$$IP = \frac{\text{arrival time of next part}}{\text{time between part deliveries}} \tag{5.1}$$

The IP, as defined above, should be maintained as close to one as possible. A value of one would indicate that the inter-arrival times of the parts and the transfer station cycle time are exactly the same. Since it was previously stated that the input parts supply is probabilistic in nature, the IP will seldom equal one. Instead, it will reflect the degree to which the controller is capable of modifying the input conveyor speed, the velocity of the robot in the various regimes, and the output conveyor speed in order to maintain equally spaced parts on the output conveyor.

There are several operating parameters that the controller works with, based on the mean of the inter-arrival times of the parts. The speed of the input conveyor should be 3.0 in/sec. The speed of the output conveyor should be maintained at 2.0 in/sec. If the parts were to arrive at time intervals equal to the mean of their distribution (22.5872 sec), then with these conveyor speeds all of the parts would be processed and be equally spaced on the output conveyor. This combination of parameters represents the ideal conditions for the station to work in. Therefore, the model for the controller uses these parameters in the decision portion of the control strategy to determine what needs to be adjusted in order to operate near this ideal state. The current system status is identified by interrogating the system velocity state vector, which contains the velocities needed in each of the regimes to process the current part. The velocity state vector is calculated at the beginning of transfer for each part and depends on the arrival time of the next part to the station. Because the controller is always "looking ahead" one part, it is possible to modify one or both of the conveyor's speeds during a part transfer such that the current part is properly spaced at output and the next part will not fall off the end of the input conveyor.

## 5.3 Control Algorithm

The adaptive control strategy that is to be used for the materials handling problem is outlined below. This algorithm is based on an ideal transfer time of 22.5872 seconds, which is the mean of the parts inter-arrival times. The nominal input conveyor speed is 3.0 in/sec and the nominal output conveyor speed is 2.0 in/sec. These speeds were selected to give the lowest transfer time while not having to operate the robot at its upper limit of performance. This provides the desired flexibility to allow the controller to request higher tip velocities from the robot when parts are coming in very close together (i.e. short inter-arrival times). For the following discussion, refer to Figure 11.

2.2) If the current input speed is less than the nominal, then there will be more time available to track the part in R I. This time difference is subtracted from the time to traverse R II and a new robot tip velocity is calculated for R II. The R IV tip velocity is unchanged and a new cycle time is calculated.

2.3) If the current input speed is equal to the nominal speed, then no tip velocity adjustments are necessary in R II and the new cycle time is calculated.

3.0) Check the arrival time of the next part against the ideal cycle time.

3.1) If the difference is greater than zero, then the part is coming in slower than is required and the input conveyor speed should be increased. Since the robot must track the current part at the current input speed, there is a lag time before the input conveyor's speed can be adjusted. This is calculated by dividing the distance between the BOT and LO by the current input velocity. A pseudo-distance is used to determine how much to increase the speed of the input conveyor. This pseudo-distance is calculated by multiplying the time left for the next part to reach the BOT (the new cycle time calculated in (2.1) minus the time lag) by the current input speed. The new input conveyor speed is then determined by dividing the pseudo-distance by the time left before the BOT.

3.2) If the difference is less than zero, then the part is coming in faster than is required. The time lag and the pseudo-distance the part is away from the BOT is calculated again, as in (3.1). For this case, there is time that must be made up since the part is coming in faster than nominal. The time to be made up is subtracted from the nominal time to traverse R IV (9.9855 sec). Then, the number of base joint steps required to move the robot through R IV is divided by the new time to traverse R IV, thus giving the average rate that will be expected of the base joint in order to make up the time difference in R IV alone. If the requested rate is less than or equal to the maximum joint rate for the base joint, then the time can be made up in R IV and a new cycle time is calculated without the need to slow down the input conveyor. If the time cannot be made up by decreasing the time to traverse R IV, then the joint rate for the base joint is set to its maximum value and the time that this adjustment made up is subtracted from the total time to be made up. Then, a new cycle time is calculated and the remaining time to be made up is added to it. The adjustment of the input conveyor speed to a lower value is then found by dividing the pseudo-distance the next part is away from the

BOT by the new cycle time plus the excess time to be made up.

4.0) Build the new velocity state vector for the current part.

5.0) Using the new velocity state vector, calculate the time required to traverse R I, R II, and the output tracking portion of R III (G to TD). The time to the start of output tracking (TTSOT) is found by adding the times in R I, R II, and the residual time left over from the previous cycle. The residual time is the time required for the robot to finish the R III (TD to K) and R IV moves of the previous cycle. The time to touchdown (TTTD) for the current part is the TTSOT plus the time to traverse the distance from G to TD at the nominal output conveyor speed. Since the robot's new cycle time is different than the ideal cycle time, the output conveyor's speed must be modified in order to keep the part spacing equal. However, the robot is required to track the output conveyor in a manner analogous to the input tracking task, so modification of the output conveyor's speed must be done prior to the robot reaching G. The lower limit to which the speed of the output conveyor might be adjusted is found by dividing the lower limit of the part spacing (44.75 inches) minus the pseudo-distance traveled during output tracking at the nominal output conveyor speed by the TTSOT. The upper limit of the output conveyor speed before tracking begins is calculated in the same manner only the upper part spacing limit (45.25 inches) is used. The actual speed selected at which to run the output conveyor to ensure equal spacing between parts is the average of the upper and lower speed limits calculated above. This speed is maintained until the robot gets to G, then it is reduced to the nominal output conveyor speed.

6.0) Start the current transfer cycle.

6.1) Dependent upon the location of the robot along the predefined transfer path, send the velocity state vector component for the current regime, along with the load information to the robot.

6.2) When the robot has reached the part touchdown position in R III, record the time it took to get there and add the previous cycle's residual time to obtain the time between part deliveries.

6.3) After the current cycle has been completed, check the input conveyor speed against the nominal. If it is greater than the nominal speed, then add the nominal speed to the current speed and divide by two. This averaging technique is to keep the pseudo-distance the controller thinks the next part is away from the transfer station from becoming unbounded. If the current input conveyor speed is less than the nominal speed, then leave it alone.

7.0) Go back to (1.0) and start processing the next part.

The control algorithm, as outlined above, allows the adaptive controller to make the necessary adjustments to the various system parameters to ensure that every part is transferred as closely as possible to the ideal cycle time and maintain the required equal spacing of the parts on the output conveyor. A review of this procedure should make obvious the premise that, using only feedback control or optimal control, the operational requirements for this application could not be met.

## 6.0 SYSTEM SIMULATION

The previous five sections have built upon one another to develop the necessary components to simulate the operation of the materials handling problem. In this section, the individual components will be brought together in the form of a simulation model of the transfer station.

### 6.1 Description of the System Model

The first component of the system model is the kinematic description and location of the robot in the working environment. Referring to Figure 8, the robot's base joint center of rotation was at a point (-8,8,0) in the global reference frame. Because of the way in which the coordinate frames were assigned to the robot in Section 3.1, the robot must be calibrated to the conveyor belts based on the location of the shoulder joint. The shoulder joint for the MICROBOT is 7.68 inches above the datum on which the robot sits. The datum for the system model is at zero height (z = 0) in the global reference frame. To calibrate the robot, a homogeneous transformation is used which describes the position and orientation of the robot with respect to the global frame. This transformation is denote Z, and is equal to

$$Z = \begin{vmatrix} .707 & .707 & 0 & -8.0 \\ -.707 & .707 & 0 & 8.0 \\ 0 & 0 & 1 & 7.68 \\ 0 & 0 & 0 & 1 \end{vmatrix} \tag{6.1}$$

The Z transformation shows that the robot has been rotated -45 degrees about the z axis of the global reference frame, then translated by the vector $-8.0\underline{i} + 8.0\underline{j} + 7.68\underline{k}$ in the global frame.

With Z defined, the position and orientation of the end of the robot may be determined by premultiplying the $T_5$ transformation matrix by Z. However, as was previously stated, the end effector or hand for the MICROBOT was not included in the formulation of the $T_5$ matrix. The relationship of the hand to the end of the arm is defined by another homogeneous transformation, denoted E. The reason for not including the E transformation into the $T_5$ matrix initially was that in the event of alternate hands or other tooling becoming available for the MICROBOT, the kinematic model of the arm itself would not have to be redefined, just the E transformation. The E matrix for the currently available hand is

$$E = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & wl1 \\ 0 & 0 & 0 & 1 \end{vmatrix} \tag{6.2}$$

where

$$wl1 = L_1 + \sqrt{L_2{}^2 - (G - G_0{}^2)/2}$$

$L_1$ = 1.884 in
$L_2$ = 1.700 in
$G_0$ = 1.520 in

The variable wll represents the variation of the length of the hand based on the distance, G, between the fingertips (see Figure 12).

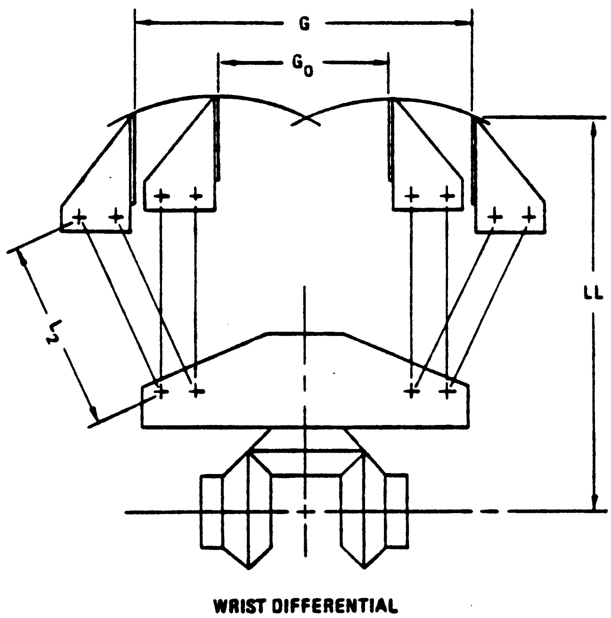


Figure 12. Hand Length Variation versus Hand Opening

This variation is taken into account to more accurately position the robot in space.

With Z and E defined, the position and orientation of the end of the hand may be defined for any obtainable point in the robot's work volume. There are two additional transformations necessary to completely define the relationship of the robot to its environment. The position of the part on the input conveyor is needed so the robot knows what to track, and the relationship of the hand to the part is needed to be able to pick the part up. The transformations, denoted P and PG, respectively, are shown below.

Regime I

$$P = \begin{vmatrix} 1 & 0 & 0 & XG \\ 0 & 1 & 0 & YG \\ 0 & 0 & 1 & ZG \\ 0 & 0 & 0 & 1 \end{vmatrix} \tag{6.3}$$

$$PG = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \tag{6.4}$$

Regime III

$$P = \begin{vmatrix} 0 & -1 & 0 & XG \\ 1 & 0 & 0 & YG \\ 0 & 0 & 1 & ZG \\ 0 & 0 & 0 & 1 \end{vmatrix} \tag{6.5}$$

In R I, the coordinate frame of the part is aligned with the global frame. In R II, the part's frame has been rotated +90 degrees about the global frame. The relationship between the hand and the part is constant. The hand's coordinate frame is always positioned such that the approach vector is in the minus z direction of the part frame and the orientation vector is in the minus y direction of the part frame. The XG, YG, ZG components in the P transformation are merely the global points the part travels through during the transfer cycle. For this model, these points correspond to the path data previously described in Section 4.2.

The transformations described above may now be put into the form of a transformation equation. This equation will accept the global point to be reached and the hand opening as input and give the value of $T_5$ as output. Once $T_5$ is known, the joint variables, $\theta_1$ - $\theta_5$, may be obtained using the solutions arrived at in Section 3.3. The general transformation equation for this configuration is

$$ZT_5E = PPG \tag{6.6}$$

and, solving for $T_5$

$$T_5 = Z^{-1}PPGE^{-1} \tag{6.7}$$

Equation (6.7) will give the value of $T_5$ for each position along the path that was developed for the robot to follow. Instead of explicitly solving this equation for each move for each part processed, Eq.(6.7) was solved for the defined path prior to the execution of the simulation since the velocities along the path and not the path itself is being controlled. What is needed as input to the simulation model to account for the robot's movements is the changes in the joint variable, $\theta_1$ - $\theta_5$, for each defined point along the path, together with the length of the move. The input to the system model which represented the robot was then reduced to the 3-D distance of the move and the corresponding $d\theta$s for each move.

The velocities attainable by the MICROBOT were modeled using a linear approximation based on performance data supplied by the manufacturer (16). Each joint of the robot is driven by a four coil, DC stepper motor working through a gear reduction unit located in the base of the robot. The motors, as used on the MICROBOT, produce 96 steps per revolution before going through the gear set. The linear equations give an estimate of the maximum rate (in steps/sec) that each motor is capable of producing without slippage (i.e. lost steps) based on the load the robot is transferring. This maximum rate is tested against the tip velocity requested by the controller to ensure that no motor is driven beyond its limits. If the requested tip velocity is within the limits of all motors, then the motor with the maximum number of steps is selected to base the time for a particular move on and the time is return to the controller. If a tip velocity is requested which falls outside any motor's limit, then that motor's rate is set to its maximum value and all of the other motor's rates and the requested tip velocity is adjusted according. The time for the move is then calculated and returned to the controller.

## 6.2 Assumptions

The simulation model is an idealization of a physical process. As such, certain assumptions must be made in order to produce a model that is computationally realizable and yet still retain the basic physical system characteristics. The assumptions pertaining to this simulation are as follows.

a) The conveyor belts are assumed to have infinite speed variability with negligible lag time between speed modification and response.
b) The positional uncertainty of the end of the robot was not taken into account(i.e. the dropping of steps due to voltage fluctuations in the power supply).
c) The parts are assumed to be previously oriented and presented in a consistent manner with respect to the input conveyor.
d) The part inter-arrival times were fit to a Weibull probability density function so that a closed form of the inverse pdf (i.e. the cdf) could be used to model the parts supply. Figure 13 shows the pdf of the input part distribution as modelled for the simulation study. Figure 14 shows the cdf used to simulate the inter-arrival times of the parts. The equations corresponding to these functions are:

pdf:

$$f(x)=0.0012(T-17.6019)^{3.9056}EXP(-(0.1840T-17.6019)^{4.9056}) \qquad (6.8)$$

cdf:

$$F^{-1}(u) = 17.6019 + 5.4342(-\ln(1.0-u))^{0.2038} \qquad (6.9)$$

The cdf is used to calculate the time of arrival of the next part. A uniformly distributed random number, u, is generated and substituted into Eq.(6.9). The time corresponding to this random probability is then obtained. Equations (6.8) and (6.9) represent the three parameter form of the Weibull. The number 17.6019 represents the location parameter of the distribution, which allows it to be shifted along the time line. The original data supplied was centered with a mean of 4.9853 seconds. The location parameter was employed such that the times

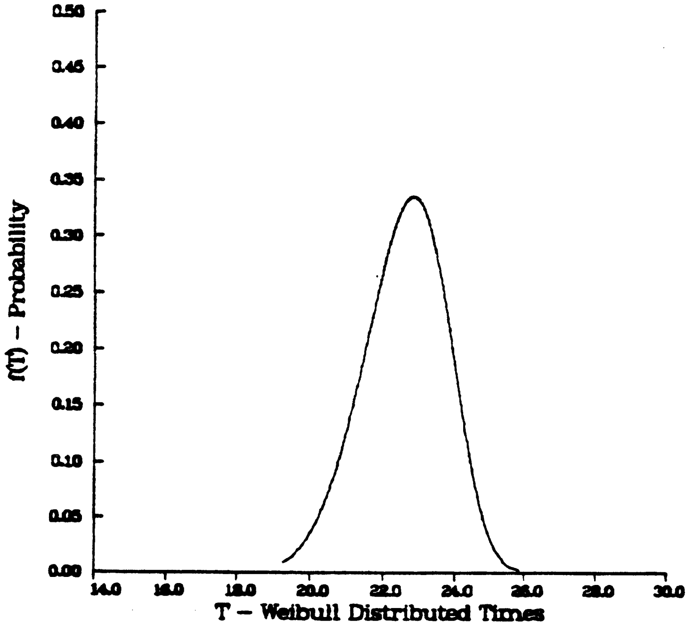


Figure 13. Weibull pdf of the Inter-arrival Times

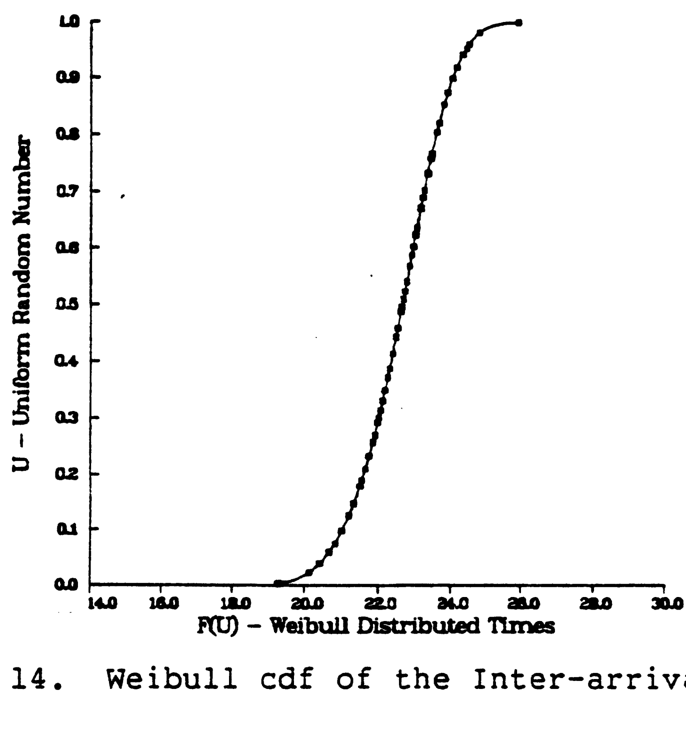


Figure 14. Weibull cdf of the Inter-arrival Times

were in the range appropriate for the robot to work within. The shape and scale parameters of the distribution were not effected.

e) The software associated with the robot has the capability to generate the appropriate velocity and acceleration curves to prevent discontinuities from occurring.

f) The parts being transferred were assumed to be 2.0 in cubes weighing 4 ounces each

## 6.3 Description of the Simulation Trials

In order to verify that the transfer station operates more efficiently with the adaptive controller in place than without, a second system model was developed. The second model used the same path data, the same kinematic model and the results derived from it and the same basic assumptions as the adaptive model used. The only difference was the absence of the adaptive controller. The operating policy was as follows. If a part was coming to the transfer station with an arrival time greater than the mean, then the robot would wait for it. If the part was coming in faster than the robot could process it, the part fell off of the input conveyor and the robot waited for the next part to come. The number of parts lost and the IP for the non-adaptive system were calculated.

For the purpose of comparing the adaptive and non-adaptive models, 500 parts were processed through the transfer station. All system dimensions are in inches, all velocities in inches/second, and all time is measured in seconds.

# 7.0 ANALYSIS OF RESULTS

The results of the simulation study were much easier to analyse than expected. The system model which utilized the adaptive controller proved to be clearly superior when operating in the uncertain environment in which it was placed. The general model response characteristics are discussed first, then a comparision is made between the two models, and finally the conclusions are drawn based on the data collected.

## 7.1 Model Performance

The characteristics of the kinematic model of the robot are shown in the next sequence of figures. Figure 15 shows a typical coordinate change with respect to time profile for a part transfer. The sharp peak in the Y coordinate is due to the change of direction at the beginning of the R IV portion of the cycle. Figure 16 illustrates the angular changes in the joint variables with respect to time for a typical part transfer. The sharp peaks in the $\theta_1$ and $\theta_5$ joint angles near 12.5 seconds denotes the time the return cycle began. This figure also shows how the hand is reoriented back to its initial tracking position during the R IV transfer.

The next sequence of illustrations (Figure 17 - 21) show the joint rates necessary to track and transfer a part. The rates are plotted in steps per second because the robot works primarily in this frame of reference. Also, the maximum rates that any joint can achieve is tested against the linearized motor models, which work in steps per second.

Figure 22 is a graphical depiction of how the adaptive controller handles a part. In the minus portion of the time line, the controller identifies the time of arrival of the next part. As can be seen, for this particular part coming in, a time lag of 2.1913 seconds was necessary prior to increasing the input conveyor belt speed. While the next part is coming in, the controller has already started the current cycle, with the appropriate velocity state vector specified to ensure that the present part is transferred according to the operating policy and will still be back in time to get the next part.

## 7.2 Model Comparison

The purpose of the simulation was to test the premise that an adaptive control strategy was more efficient when applied to a time-varying environment than a non-adaptive control strategy. Figure 23 is a comparison of the average time between part deliveries to the output conveyor. The lower curve is the adaptive controller results, while the upper curve represents the non-adaptive model. The straight line indicates the ideal cycle time of 22.5872 seconds. The data points are lumped averages over ten part lots. It is clear by Figure 23 that the adaptively controlled system performs above the requirements as indicated by the position of the model's part delivery curve. The performance numbers indicate that the adaptive system actually ran within 2.1 percent of the requested cycle time for the line. The non-adaptive model's performance was 41 percent out of spec, based on the operating policy established for it. The one key point not directly shown on Figure 23 is the fact that the non-adaptive system missed 206 parts due to its inability to adapt itself to its environment. This translates into a 41.2 percent reduction in through put at the station, not accounting for the lost revenue due to the parts hitting the floor.

Figure 24 is the same data shown in Figure 23, only a running average calculation was performed instead of a lumped average. It is intended to show the steady-state trends of the two systems. The bottom curve is the adaptive model. It is clear that the adaptive model settled much faster than did the non-adaptive model, and the adaptive system appears to have established a new set point from which to operate. This is indicative of the robustness of the control strategy.

The sensitivity of the non-adaptive model to the predefined cycle time of the robot was also examined. Since the non-adaptive model missed 206 parts while operating at the same robot cycle time as was used for the adaptive model, the robot's cycle time for additional comparison was reduced in one half second increments until the minimum feasible cycle time was reached. The time was taken off the R IV portion of the transfer path because this move represented only base joint rotation and could be readily adjusted. The results of this sequence of runs is shown in Table 7.1.

Table 7.1 Effect of Different Robot Cycle Times

| Ideal Cycle Time | Actual Cycle Time | Parts Missed | Parts ----- hr | Vavg R IV | IP |
|---|---|---|---|---|---|
| 22.5872 | 54.1660 | 206 | 66.4624 | 3.5594 | 0.3901 |
| 22.0000 | 35.6057 | 113 | 101.1074 | 3.7818 | 0.4388 |
| 21.5000 | 29.0548 | 64 | 123.9039 | 3.9943 | 0.4655 |
| 21.0000 | 25.8622 | 35 | 139.1995 | 4.2321 | 0.4817 |
| 20.5000 | 23.8200 | 14 | 151.1336 | 4.5000 | 0.4933 |
| 20.0907 | 23.0209 | 5 | 156.3796 | 4.7459 | 0.4984 |

Table 7.1 indicates an increase in the performance of the non-adaptive system by decreasing the the time spent in R IV. However, even at the minimum cycle time, the non-adaptive system still misses five parts and shows an IP

of only .4984. This may be compared to the average IP for the adaptive system which was 1.0214. An additional consideration of the performance increase shown in Table 7.1 is the decrease in the reliability of the robot. While the system will transfer more of the parts that enter its work volume, the robot is continuously being operated at its maximum performance limits with every part transferred. This will lead to a substantially reduced mean time to failure of the robot, thus increasing the overall downtime of the system. With the adaptive strategy, only the required rates for any given transfer cycle are requested by the controller. The ideal operating conditions for the robot were established such that the joint motors would be operating at 75 percent of rated output, with peak output rates available on request.

The final indication of the two systems' relative performance is the examination of their respective IP's over the 500 part cycle (Figure 25-30). The top set of data are direct calculations of the instantaneous IP of the adaptive system. The profile of the data appears well bounded and indicates the ability of the adaptive system to handle severe perturbations in the input part supply and still hold the tight tolerance around unity. The non-adaptive system appears to be oscillating with a regular frequency. This is to be expected since the system does not have the capability to interact with its environment. By examining each figure, the effect of decreasing the robot's cycle time for the non-adaptive model is shown. The triangles that appear to be lower than the rest of the instantaneous IPs indicate that a part has been missed and the non-adaptive system must wait until the next part arrives in order to continue transferring parts.

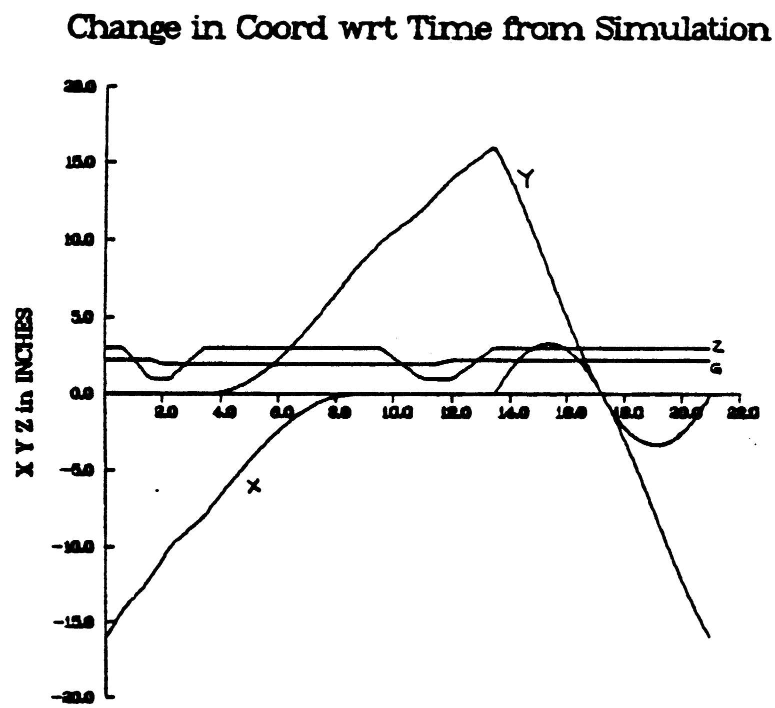


Figure 15. Change in Coordinates wrt Time

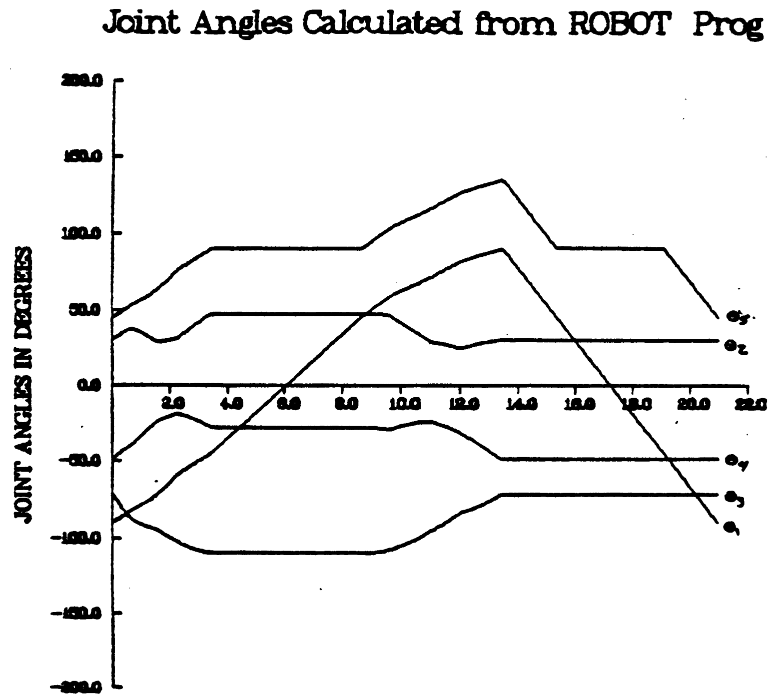


Figure 16. Change in Joint Angles wrt Time

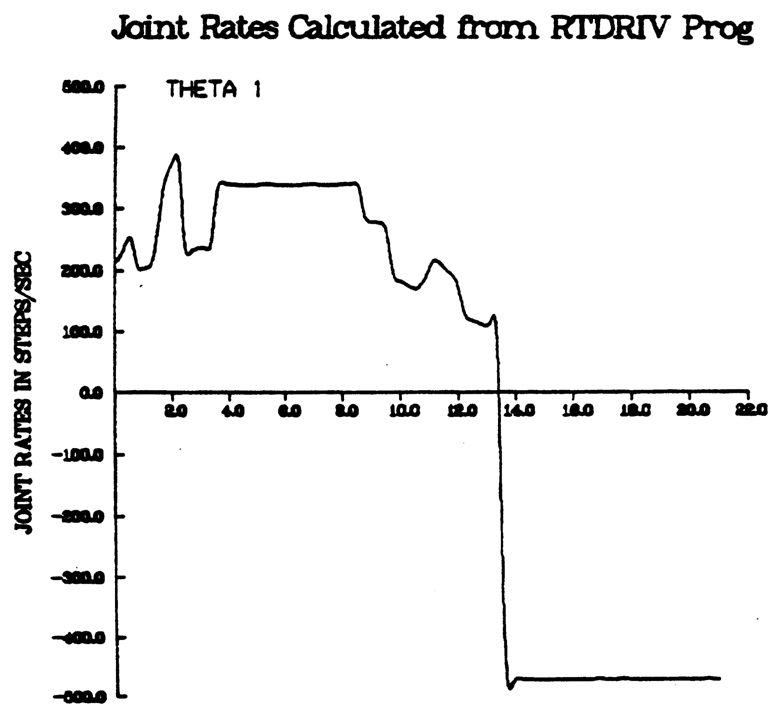


Figure 17. Theta 1 Joint Rates

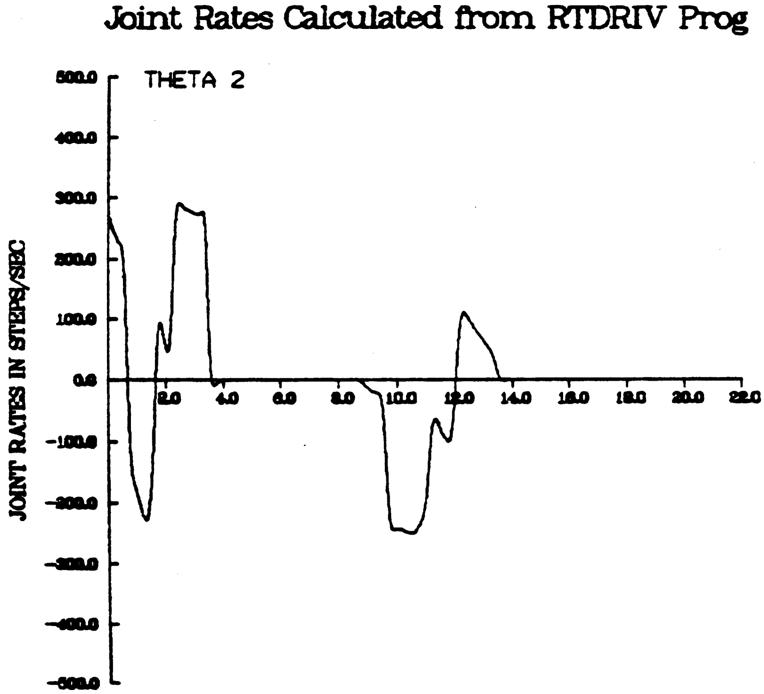


Figure 18. Theta 2 Joint Rates

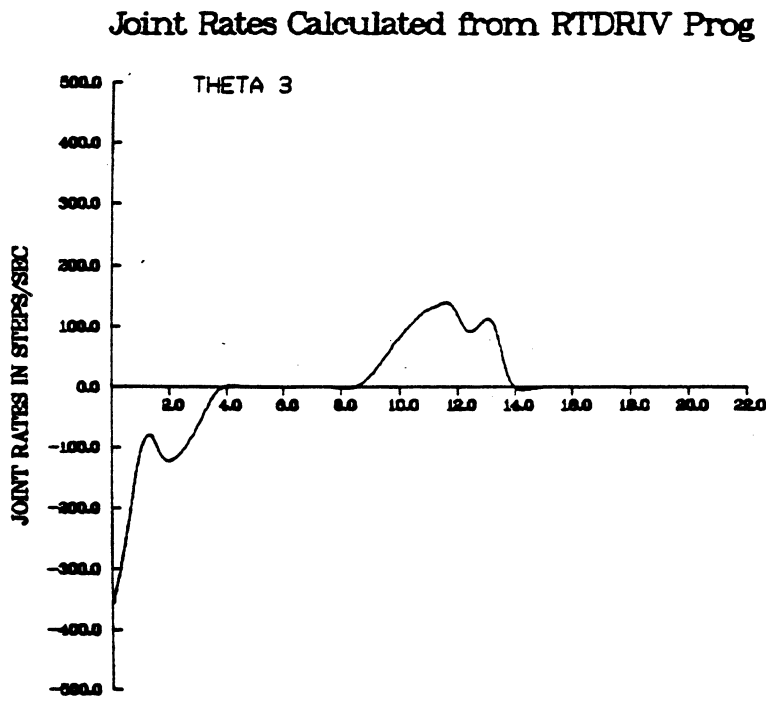


Figure 19. Theta 3 Joint Rates

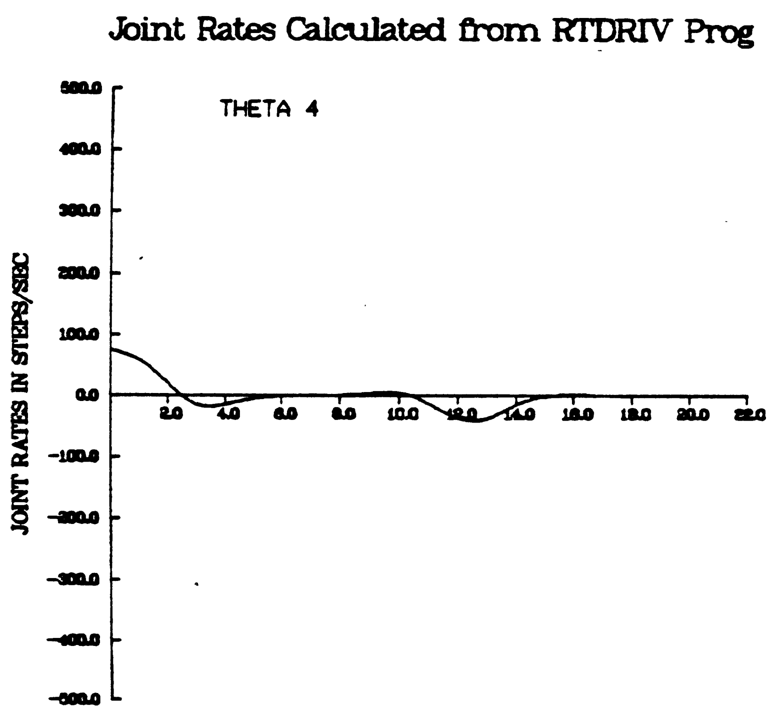


Figure 20. Theta 4 Joint Rates

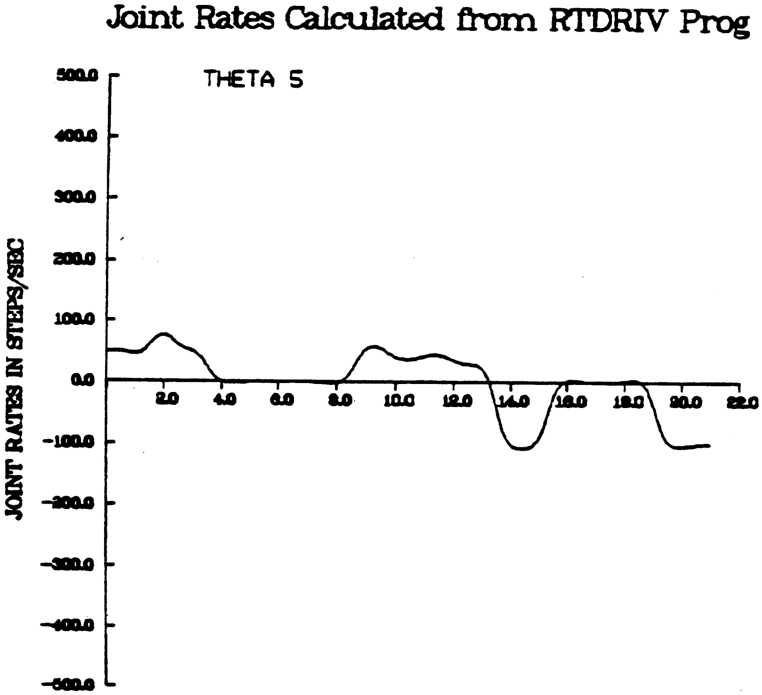


Figure 21. Theta 5 Joint Rates

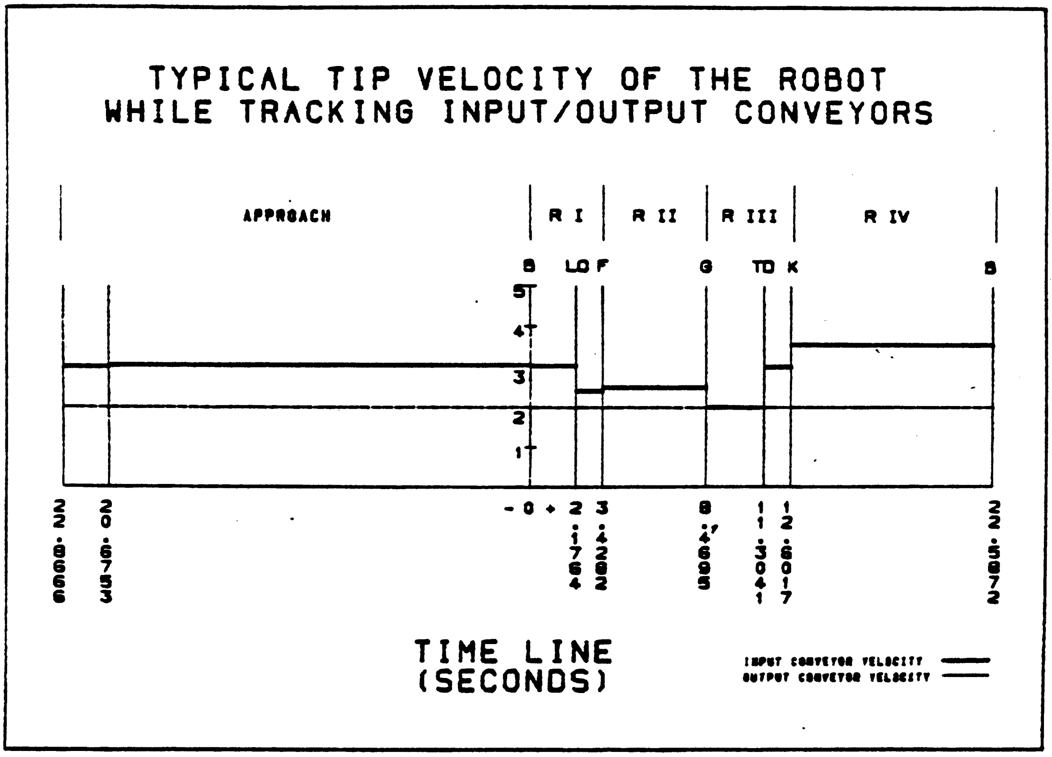


Figure 22. Robot Tip Velocity of a Transfer

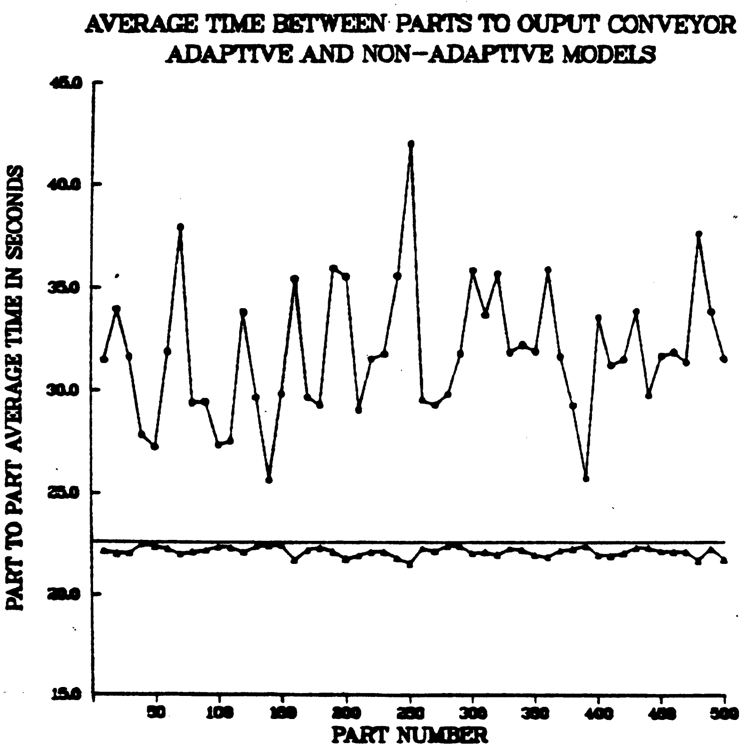


Figure 23. Time Between Parts Delivered (lumped)

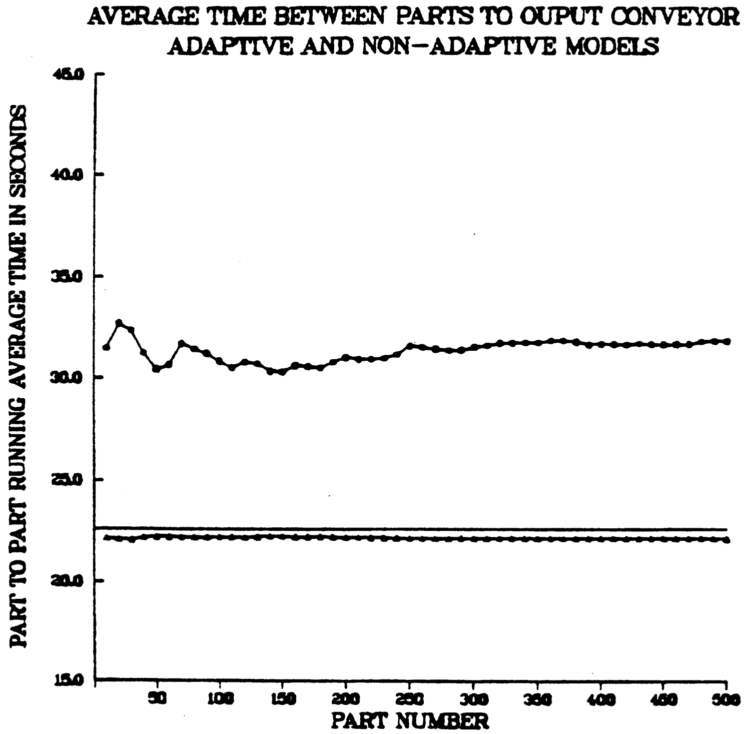


Figure 24. Time Between Parts Delivered (running)

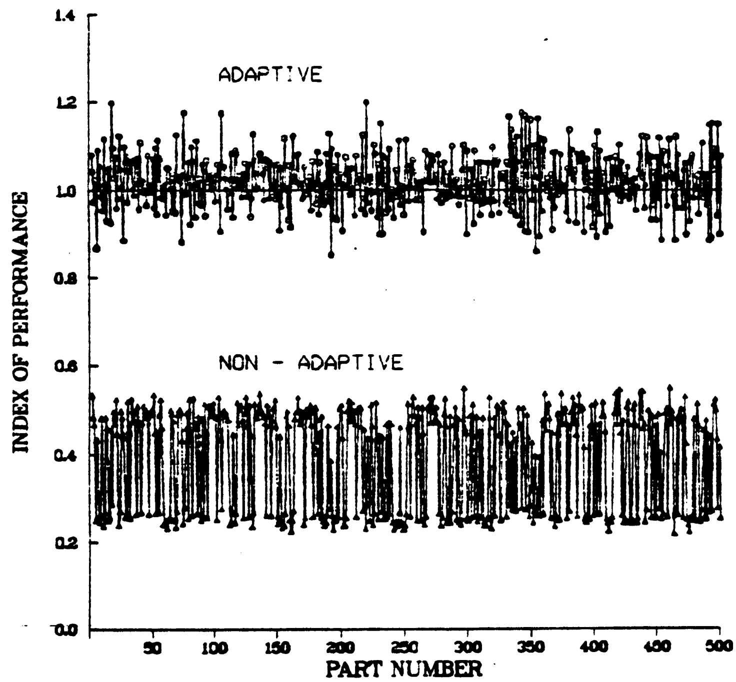


Figure 25. IP-Ideal Cycle Time is 22.5872 sec.

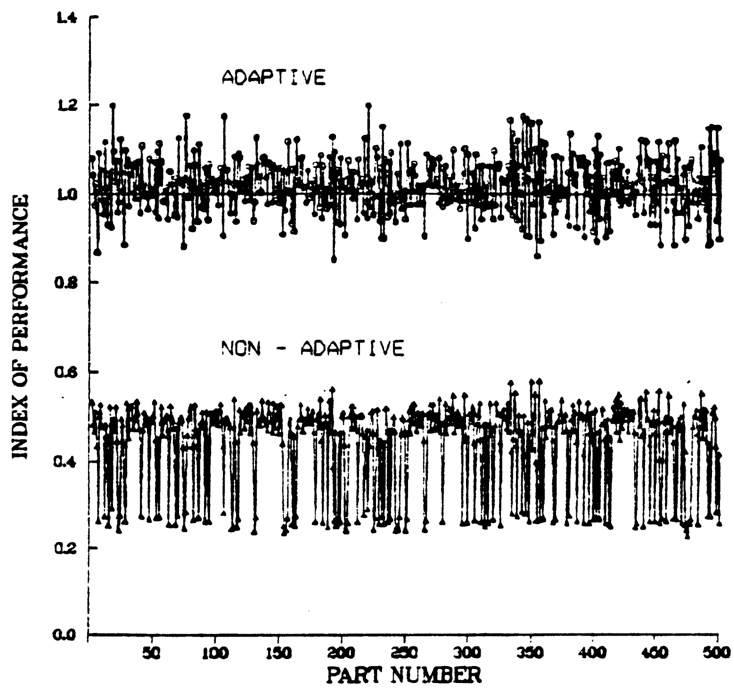


Figure 26. IP-Ideal Cycle Time is 22.0000 sec.

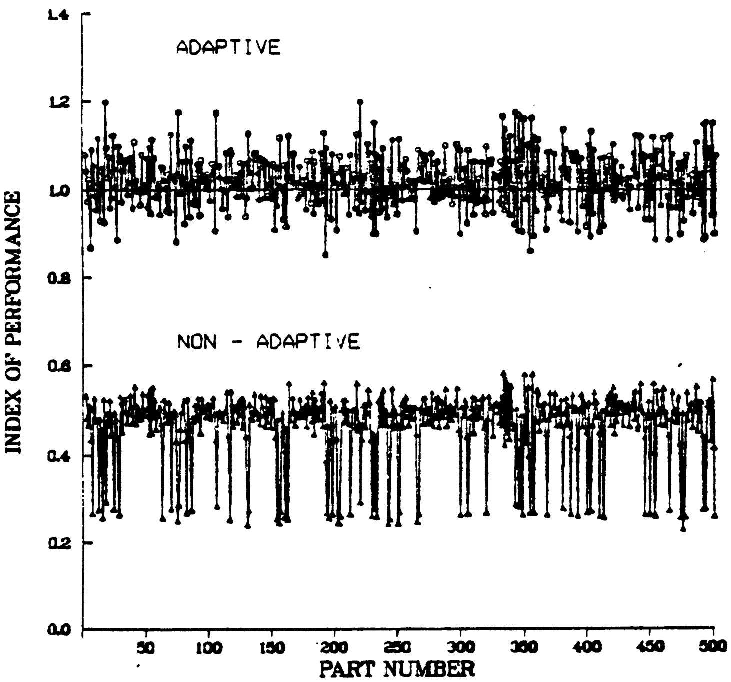


Figure 27. IP-Ideal Cycle Time is 21.5000 sec.

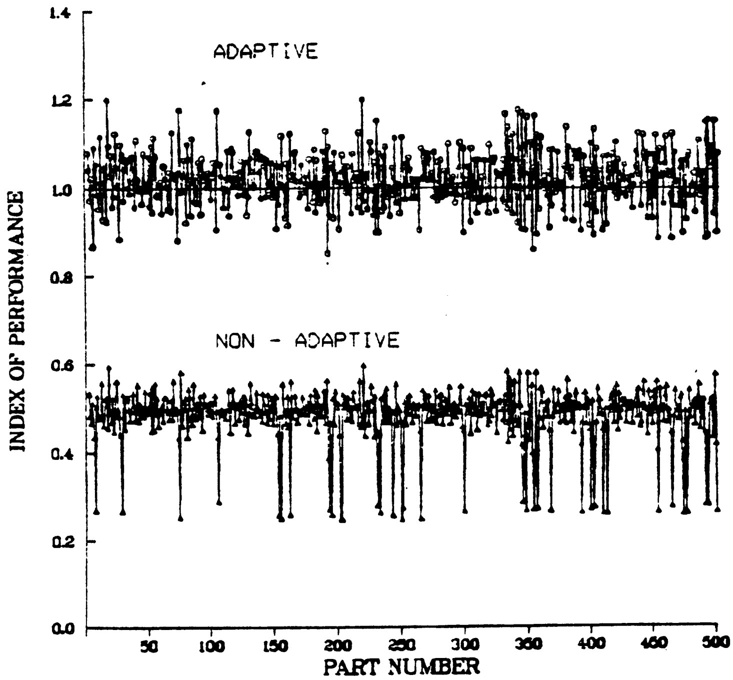


Figure 28. IP-Ideal Cycle Time is 21.0000 sec.

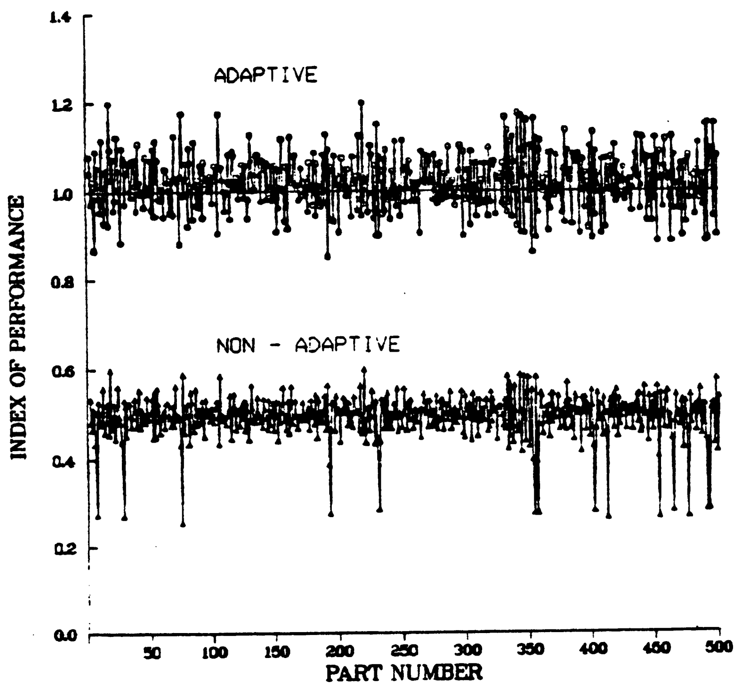


Figure 29. IP-Ideal Cycle Time is 20.5000 sec.

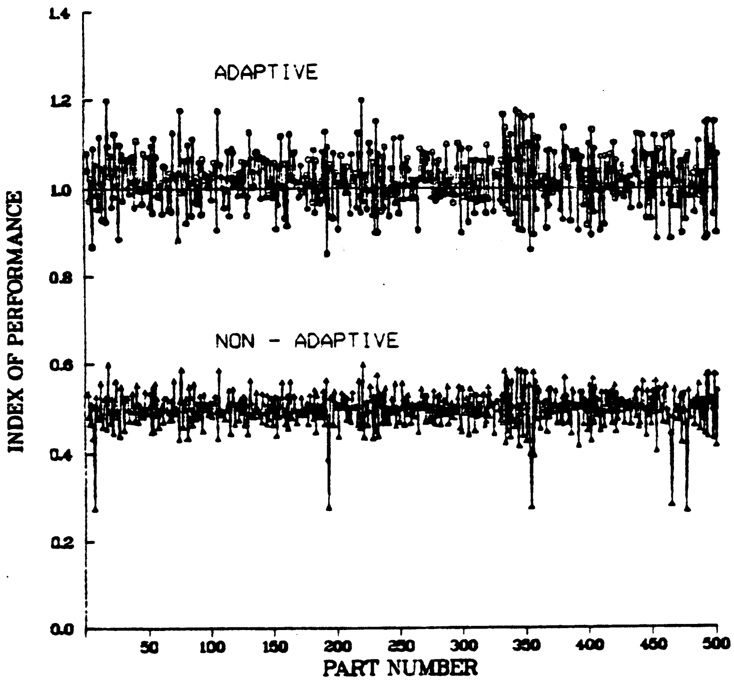


Figure 30. IP-Ideal Cycle Time is 20.0907 sec.

## 7.3 Conclusions

The object of this work was to apply an adaptive control strategy to a manufacturing problem with a high degree of uncertainty. The problem that has been outlined and subsequently solved using adaptive control techniques between a robot and two manufacturing lines indicates that more efforts must be applied in the area of adaptive control for robotic systems.

Areas discussed in this work that would, if realized, be the most beneficial to robotic systems development are Artificial Intelligence and advanced sensory technology. With an AI system at the heart of the adaptive controller (i.e. the decision function), advanced assembly tasks would be mundane activities. By giving the robot the ability to perceive its environment and respond accordingly, based on knowledge it is given and generates during operation, the concepts of the totally automated factory could be very attainable. Presently, industrial robots are being utilized in areas involving high risk to human life and to continuous or repetitive tasks. As advances are made in the control strategies and sensory technology, the robot will truly become the vital link in the factory of the future.

REFERENCES

(1) Paul, R. P., Robot Manipulators: Mathematics, Programming, and Control ,MIT Press,Cambridge, Massachusetts, 1982.

(2) Albus, J. S., Brains, Behavior, ' Robotics ,BYTE Books, Peterborough, N.H., 1981.

(3) Tanner, W. R.,ed. Basics of Robotics: Volume 1, Fundamentals ,2nd Ed., SME, Dearborn, MI., 1981; pp 3-12.

(4) Colleen, H.,"Adaptive Control: Giving Robots the Power to Cope." Robotics Today ,Spring 1980.

(5) Hohn, R. E.,"Application Flexibility of a Computer-Controlled Industrial Robot" Basics of Robotics: Volume 1, Fundamentals ,2nd Ed., SME, Dearborn, MI., 1981; pp 224-42.

(6) Demers, K. P. and P. M. Walsh,"Sensor-Based Real-Time Robot Control Systems." Robotics Today ,Vol. 5, No.3 1983; pp 69-72.

(7) Freund, E.,"Fast Nonlinear Control with Arbitrary Pole-Placement for Industrial Robots and Manipulators." International Journal of Robotics Research ,Vol. 1, No.1 Spring 1982; pp 65-78.

(8) Munson, J. H.,"Robot Planning, Execution, and Monitoring in an Uncertain Environment." 2nd International Conf. on AI ,London, September 1971; pp 338-49.

(9) Ejiri, M., et al,"An Intelligent Robot with Cognition and Decision-Making Ability." 2nd International Conf. on AI ,London, September 1971; pp 350-58.

(10) Baumann, E. W.,"CAD Model Input for Robotic Sensory Systems." AUTOFACT 4 Conf. Proc. ,Nov. 30 - Dec. 2, 1982 Philadelphia, PA.; pp 11.25-.34.

(11) Robertson, G. I.,"Heirarchical Control of Intelligent Robot and Vision Allows Plug-In System Integration." AUTOFACT 4 Conf. Proc. ,Nov. 30 - Dec. 2, 1982 Philadelphia, PA.; pp 11.35-.50.

(12) Denavit, J. and Hartenberg, R. S.,"A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices." Jour. of Applied Mechanics ,Vol. 22, June 1955; pp 215-21.

(13) Paul, R. P., B. Shimano, and G. Mayer,"Kinematic Control Equations for Simple Manipulators." Trans. on Systems, Man, and Cybernetics ,Vol. SMC-11,No.6 June 1981; pp 449-55.

(14) Lee, C. S. G.,"Robot Arm Kinematics, Dynamics, and Control." IEEE Computer Journal ,Dec. 1982; pp 62-80.

(15) Groover, M. P., Automation, Production Systems, and Computer-Aided Manufacturing ,Prentice-Hall, New Jersey 1980.

(16) Robotics Reference and Applications Manual ,MICROBOT, Inc. Moutain View, CA. 1981.