SYNTHESIS OF ROBOTS/MANIPULATORS FOR A PRESCRIBED WORKING SPACE

By

YING-CHIEN TSAI

Bachelor of Science and Engineering
National Cheng Kung University
Taiwan, Republic of China
1969

Master of Science and Engineering
National Cheng Kung University
Taiwan, Republic of China
1973

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
May, 1982

SYNTHESIS OF ROBOTS/MANIPULATORS FOR A

PRESCRIBED WORKING SPACE

Thesis Approved:

Thesis Adviser

Dean of the Graduate College

ACKNOWLEDGMENTS

My sincere gratitude is expressed to all who have helped in any way to make this study possible. In particular, I am grateful to Dr. A. H. Soni, chairman of my advisory committee, for providing me many opportunities to grow in the mechanism science area, for suggesting and encouragement to carry this problem through, for helping my financial problem and for taking care of my family. I am also grateful to my other committee members, Dr. R. Lowery, for his encouragement and guidance, Dr. L. Zirkle and Dr. J. Chandler, for suggestions and encouragement.

I am fortunate to have met a group of "mechanism men," Dr. A. G. Patwardhan, Dr. A. Hossne, Dr. V. Srinivasan, Mr. K. L. Ting, Mr. M. R. Gudavalli and Mr. G. Naganathan. To them I offer my gratitude for friendship and discussions.

My special appreciation is due to my parents and relatives for their financial support and encouragement to make my Ph. D. program possible. I am also obliged to Mrs. J. R. Walsh for her continuous encouragement.

I would like to express my appreciation to my wife, Shu-Ling, for her encouragement, taking care of the family and for her patiently typing the manuscript.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure | Page

CHAPTER I

INTRODUCTION

1.1 History of the Robot System

The term "Robot" has been used as parts of the subject in novels and movies for a long time. Under writers' and/or film makers' description, these robots are humanoid not only in structure but also in capabilities. They can walk, they can talk, they can hear, they can see. They can help their master answer the door, clean the house, wash dishes, do baby sitting, and sell goods at department stores -- they even can love and hate. Like Artoo Deetoo in the movie "Star Wars," they even have some capabilities superior than what human-beings have. But all these robots mentioned above are so-called "Idealized Robots". They may be come existence in the future, but not now.

The engineering, scientific and industrial community working in the area of robotics generally use the term "robot" to describe classes of machines which are capable of taking certain actions which were only associated with humans. We are here more concerned about the function and not the appearance. And what distinguishes a robot from other machines is that it does not necessarily always repeat the same pattern of actions.

The development of manipulator systems started, about fifty years ago, with the need of handling radioactive materials. In 1947, the Argonne National Laboratory initiated a project to develop a device which could duplicate the hand motions of a person at a remote control station. They built the first manipulator with six degrees of freedom. And this work resulted in the development of a series of mechanically coupled "master-slave" manipulator systems. In the master-slave system the motion of the master was duplicated by the slave system, and the forces on the slave system were to provide feedback to the master system. The disadvantage of the mechanically coupled master-slave system is that the master and slave units must be located fairly close to each other. In order to overcome this disadvantage, later on, Argonne National Laboratory developed electronically controlled master-slave manipulator systems and Oak Ridge National Laboratory developed hydraulic master-slave manipulator systems. And several other companies produced some of these master-slave manipulator systems including General Electric's "Man-Mate" and La Calhene's "MA-23" in France. Nowadays these kinds of master-slave systems allow people to work remotely in distant, physically difficult or hazardous environments. People use these systems in nuclear power plants, underground, undersea, and outerspace; people even apply these systems in micro-surgery whereby the surgeon's precision can be increased by several orders of magnitude.

But all these master-slave manipulator systems mentioned above need a human operator to monitor, make decisions, and control it all the time. For these reasons this kind of systems is not good for tedious and repetitive tasks or remote control at a very long distance which may cause time delay. Sheridan described "supervisory controlled manipulators" which operated partially under computer control and partially under human control, and Ernst described automatic manipulator systems which carried out the tasks completely under computer control, involving sensory information about the environment in 1961. The first generation of automatic inustrial robot was built by Illinois Institute of Technology Research Institute in 1971. Now we have thousands of first generation industrial robots in this country. "Unimate" and "Puma" by Unimation Inc., "T-3" by Cincinnati Milacron, . . . etc., are the best known modern industrial robots. They have been used in automotive and other mass production industries for welding, painting, assembling, . . . etc. Some other famous robot manufacturers in other countries are KAWASAKI in Japan, VW in West Germany, ASEA in Sweden, COMAU (a FIAT owned company) in Italy, RENAULT in France, . . . etc.

First generation industrial robots are controlled by minicomputers or microcomputers. One can either input the coordinates of working positions and some other data to the computer and let computer calculate the working path, or one can teach the robot by leading it to do the work once, the computer will memorize the whole process, and then the robot will repeat what it did before. So this kind of industrial robots are good for doing tedious and repetitive tasks. But this can not satisfy all situations on the production lines. For example, the robot cannot see, so it does not know where the machine part is, it can not distinguish one part from another either. Because it has no force feeling, the robot may crush the bottle which it wants to pick up. This has led the robot research and design group to incorporate several kinds of sensors on robots. The second generation of industrial robots may be equipped with vision sensors so it can see, with proximate sensors so it knows its hand is now near some object, with the force and torque sensors it knows how much force/torque it applies on the object. Some of these second generation industrial robots are already used on the assembly lines of aircraft and auto-industries.

Now, the industrial robots are designed to do more than humanbeing can do, and even can do something that we cannot. Not only can the robots work three shifts a day without coffee breaks, but also they work in noisy, hot, fumy, radiative places without any complaints. They will never go on strike; they will never ask for promotions. Their productivity and quality are quite stable These are some of the reasons why an industry prefers the use of robots on production lines. Especially in the last decade, when oil crises, economic depression and high inflation have hit the whole world. Robot seem to be providing paratial answers to some of the productivity problems in U.S. industries. The one who can increase the productivity and decrease the cost at the same time may survive. Incorporating the use of robots on assembly line appears to be one of the best solutions. It appears that the Robot Revolution is just begining [17].

1.2 Kinematics in Robots and Manipulators

Robotics is a big interdisciplinary science and calls for people with expertise in different fields of engineering. It needs a kinematician to study the geometry and classification of robots, needs dynamics and control people to study the forces and control the motion of robots. It needs people to develop different kinds of sensors to make the robot see the object and feel the forces/torques applied on it. Computer specialists are trying to develop new high level robot languages and increase the intelligence of robots. New computers with high computing speed, large memory storage but with a small size are in need for developing robots of new generation.

This study will concentrate on the kinematics of robots and manipulators only. Some of the significant problems solved in this area are:

1. Classification of robots and manipulators.
2. Joint displacement analysis.
3. Extreme distances.

4. Reachable working space.
5. Performance.
6. Path synthesis.
7. Link parameter synthesis.

1.2.1 Classification

Most robots/manipulators can be represented by an open loop chain. Theoretically, kinematic pairs such as revolute pair (R pair), prismatic pair (P pair), cylindrical pair (C pair), spherical pair (S pair), helical pair (H pair), . . . etc. , can be used for connecting links. But from a practical point of view R and P pairs have proved to be most useful. The C or S pair may be made kinematically equivalent to the combinations of R and P pairs. The type of pairs and non-zero link parameters can be used to classify the manipulators [36]. Degrees of freedom of the last link (end effector) can be used to classify a robot also. When the number of degrees-of-freedom of the last link is less than the number of joints we call it degeneracy. A robot with zero link parameters may cause degeneracy. Duffy listed all possible spatial mechanisms with R, P, and C pairs having one overall degree of freedom [11,13]. Pieper and Roth examined the solvability and orientation restrictions in 6R manipulators [36, 39]. Because R pairs describe the orientation of the robot hand (end effector) and P pair cannot, Makino used the rank as a function of the number of R pairs of a robot, to classify the robot [28].

1.2.2 Joint Displacement Analysis

Given the displacements at each joint, finding the location and orientation of the last link (end effector) of a given robot/manipulator is a quite straight forward work. It just involves the multiplication of displacement matrices. But given the locations and orientations of the last link, finding the displacements of each joint is quite different from the first kind of problems. The latter problem involves solving complex simultaneous algebric equations. Analysis of a single-loop spatial mechanism becomes a difficult subject when the mechanism has more than four links, some problems are yet to be solved [36]. Analysis of spatial five link mechanisms have been done by several authors like Duffy [6-8], Soni [44], Yang [53], and Yuan [54]. Duffy [14] finally worked out the displacement analysis of general spatial 7-link, 7R mechanisms. All of these are helpful for joint displacement analysis of robots and manipulators.

Pieper [36] used (4 x 4) matrices to solve the joint displacement of 6R manipulator containing three intersecting revolute axes. He also presented analytical solutions for manipulators with any three prismatic joints. Numerical methods can also be used to solve the robot joint displacement problem when explicit solutions are difficult to get. Pieper [36] developed a procedure based on velocity methods which proved to be superior to the widely used Newton-

Raphson technique. Melendovic [30] divided the robot into two parts: "major mechanism" and "wrist mechanism". Major mechanism provides the "gross motion" of the robot, and wrist mechanism, having rotary joints, provides the orientation and local motion of the robot hand. He also developed a rapidly converging iterative procedure, that is needed only during off-line editing of input data and can do real time motion synthesis employing closed-form solutions.

1.2.3 Extreme Distances

A study of extreme distances of the joints and hand of a robot is important. It will help us to understand how the types and number of joints, and dimensions of linkage are related to the work spaces of robots. Shimano and Roth [42, 43], using Plucker line coordinates, developed an iterative method for determining extreme distances between axes of a robot. Sugimoto and Duffy [47,48] presented an iterative method for searching extreme distances between a base point and the center point of the robot hand or extreme distances between the center point of robot hand and the first joint axis of the robot.

1.2.4 Reachable Working Space

The reachable working space is defined as the region (or volume) within which every point can be reached by a reference point on the robot (manipulator) hand. Sometimes the working spaces are associated with the specified orientation of the robot (manipulator) hand. The working space of a robot is one of the most important specifications to either robot designer or user. One must arrange the production line properly such that the working processes are within the working spaces of robots. And the working space can be used to measure the efficiency of a designed robot mechanism.

Roth [39] presented the working spaces of some robot mechanisms. Tsai and Soni [50] solved the accessible regions of robot arms for planar case in closed-form and provided design charts. Kumar and Waldron [25-26] has presented a numerical method to trace the bounded surface of the working space of manipulator with ideal R pairs. They applied a force to the robot hand and moved it. The method is also applied to a manipulator having prismatic joints with motion limits.

1.2.5 Performance

There are some criteria for judging the performance of robots/manipulators. If two manipulators have the same scales after normalization, the one with larger working space is superior [51]. Besides working space, the dexterity quotient and approach angle are also very important measures [34]. Some other promising measures are number of possible different configurations [40], quality index or service coefficient and volume of motion [18,20].

1.2.6 Path Synthesis

When we use a robot to do welding work, the welding path defines the path and orientation of the robot hand. But in many cases when a robot moves from one working station to next working station, there are no special requirements on the location and orientation of the robot hand between working stations. Therefore one could let the robot move arbitrarily between working stations, or let it move in some special manner such that it will satisfy one or some of the following criteria:

1. Minimum energy consumption [52].
2. Minimum time (maximum rate of speed) [18,52].
3. Minimum total joint displacement [18].
4. Smooth curve [20,30].
5. Minimize the maximum joint force.
6. Obstacle avoidance [18,27,36].
7. Compromise of 1 and 2 [18,52].

1.2.7 Link Parameter Synthesis

Shimano [43] has derived equations to synthesize link parameters for two and three link mechanisms with R joints when one or more extreme distances between axes of rotation are specified. He also extended his work to N-link mechanism. Tsai and Soni [50] has presented the synthesis of two and three link robot arms when working stations and approaching angles are specified.

Synthesis of link parameters of robots and manipulators is a very practical problem. It will help the robot and manipulator designer to design a more efficient and more versatile robot for complicated performance requirements. We need more work to be done on this area. A general algorithm for synthesizing link parameters of general robot mechanisms is not developed yet.

Table I is a brief survey of the kinematics in robots and manipulators.

1.3 Present Study

The objectives of the present study are to develop synthesis procedures to synthesize the link parameters of industrial robots from the simplest 2R planar case up to the most versatile 6R spatial cases for a prescribed working space. In general, the synthesis problems of robots/manipulators may consist of three levels, they are:

1. Link position analysis: Given all link parameters and joint displacements of a robot/manipulator, find the locations and directions of the end link or/and other links and joints.

2. Joint displacement analysis (synthesis): Given all link parameters of the robot/manipulator and the position and direction of the robot hand, find all joint displacements.

3. Link parameter synthesis: Given the directions and/or locations of the robot hand (or the working space

TABLE I

A BRIEF SURVEY OF KINEMATICS IN ROBOTS AND MANIPULATORS

Category	Author	Remark
Classification of Robots and Manipulators	Pieper, Roth	Present a robot by the types of kinematic pair and non-zero link parameters.
	Duffy	List all possible spatial mechanisms which can be converted to robots and manipulators.
	Makino	Classify the robots by the rank (the number of R joints) and the degrees-of-freedom.
Joint Displacement Analysis	Yang, Yuan, Duffy Soni, ... et al.	Analysis of spatial five-link mechanisms
	Duffy	Displacement analysis of general 7-link, 7R mechanisms.
	Pieper, Roth	Joint displacement of 6R manipulators containing three intersecting revolute axes, and numerical solution for general 6R robots.
	Milenkovic	Consider the robot as two parts, major mechanism and wrist mechanism. Solve for the displacement by using a partially iterative procedure and partially in a close-form.
Extreme Distances	Shimano, Roth	Using Plucker coordinates, developed an iterative method for determing extreme distances between axes of a robot.

TABLE I (Continued)

Category	Author	Remark
Extreme Distances (cont.)	Sugimoto, Duffy	Iterative methods for: 1. Extreme distances between a base point and the center point of the robot hand. 2. Extreme distances between the center point of robot hand and the first joint axis of the robot.
Working Spaces	Roth	Some observations about the working spaces of manipulators.
	Kumar, Waldron	1. By applying a virtual force at the robot hand, plot the boundary contour of the working spaces on a plane containing the base axis of the robot. (Iterative method.) 2. Restricted to manipulators having ideal revolute joints or prismatic joints with motion limits.
	Tsai, Soni	1. Closed-form equations and design charts for working spaces of 2R robot arms (planar case). 2. Will be extended to 6R general robots.
Performance	Vertut	Evaluate the performance of a robot by its working space.
	Flatau	Evaluate the performance of a robot by its dexterity angle.
	Roth	Evaluate the performance of a robot by its number of possible different configurations.

TABLE I (Continued)

Category	Author	Remark
Performance (cont.)	Kobrinsky	Evaluate the performance of a robot by its service coefficient and volume of motion.
Path Synthesis	Vukobratovic	Synthesize the path by, 1. Minimum energy consumption. 2. Minimum time or maximum rate of speed. 3. Compromise of 1 and 2.
	Frolov, Kobrinsky	Synthesize the path by, 1. Maximum rate of speed. 2. Minimum volume of motion. 3. Compromise of 1 and 2. 4. Obstacle avoidance.
	Mikenkovic	Use curve fitting method to generate the path.
	Pieper, Roth	Obstacle avoidance path synthesis.
	Loeff, Soni	Obstacle avoidance path synthesis.
Link Parameter Synthesis	Vukobratovic	Synthesize the link parameters for a given path.
	Pieper, Roth	Synthesize the link parameters for given extreme distances.
	Tsai, Soni	Synthesize the link parameters for given working stations.

which is accessible by the robot hand), find all link parameters of the robot/manipulator to be designed.

The first level of the link position analysis problems is a pretty straightforward work; the second level of the joint displacement analysis is not so easy when the robot consists of more than four joint in general case. The joint displacement analysis of a general 6R robot in closed-form was believed to be impossible for several decades. Only recently, Duffy [14] presented the solution for general 7-link, 7R closed loop spatial mechanisms. The last level of link parameter synthesis problems is even more difficult than other levels that could be expected in general case.

A n-R robot/manipulater has 3n time-independent link parameters (a_i, α_i and s_i, i = 1,2, . . . , n) and n time-dependent joint displacements (θ_i, i = 1,2, . . . , n). For a 6R robot it contains 24 parameters, but for a rigid-body motion it has only six degrees of freedom. In other words to syntheize a robot for a prescribed working space may become a optimization problem. One may solve this problem by optimizing all the parameters at the same time. But one may solve the problem in an alternative way, study the effects of the link parameters on the sub-structures of the robot first and arrive at the optimal values of some link parameters or some relationships among link parameters. In this way one may reduce the number of parameters in the final optimization problem and may simplify the original problem of synthesis significantly.

In order to achieve the final objectives, the present study:

1. develops algorithms to plot the contours of the working spaces of robots/manipulators which may consist of two, three or any number of revolute joints with or without limitations on the ranges of joint motions.
2. studies the effects of link parameters on the working space and dexterity of the regional structure and orientational structure of robots/manipulators.
3. develops the link parameter synthesis procedures for 2R, 3R, 4R, 5R and 6R robots.

Chapter II develops the algorithms to plot the contours of working spaces of 2R, 3R and nR (in general case) robots/manipulators. The revolute joints may have limitations on range of motion. The contour may be plotted on an arbitrary plane. The robot hand can be treated as a point, a line or a rigid-body. Chapter III presents the effects of the link parameters of general 3R robots on their workspaces and dexterities. Chapter IV develops the synthesis procedures for 2R, 3R, 4R, 5R and 6R robots. Finally, Chapter V presents summary and conclusions of present study.

CHAPTER II

WORKSPACES OF ROBOTS/MANIPULATORS

The workspace (accessible region) of a robot/manipulator is defined as the region (area or volume) within which every point can be reached by a reference point on the robot/manipulator hand (see Figure 1). Sometimes the workspaces of robots/manipulators are associated with the specified orientation of the robot hand. In other words, the robot hand not only can reach every point within its workspace, but also can reach these points in a specified direction. The workspace is one of the most important specifications of a robot/manipulator. One must arrange the production line properly such that the working stations are within the workspaces of robots. And the workspace can be used to measure the efficiency of a designed robot/ manipulator mechanism.

2.1 Workspaces of 2R Robot Arms

The planar 2R robots/manipulators are very useful in the packaging industries due to their simplicity, low price, and capability of doing high speed jobs. Figure 2 shows a 2R robot arm in a X-Z reference frame. The coordinates of point B can be written as

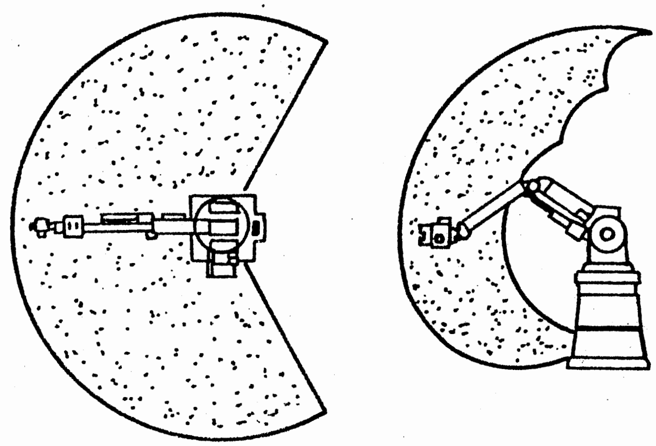


Figure 1. The Workspace of a Robot

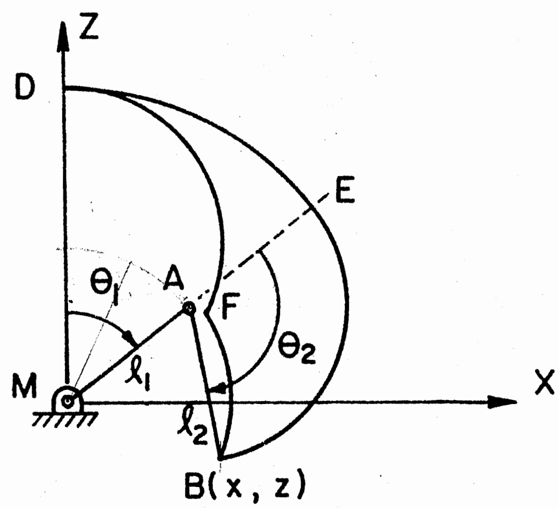


Figure 2. The Workspace of a 2R Robot

$$x = \ell_1 \mathrm{Sin}\theta_1 + \ell_2 \mathrm{Sin}(\theta_1 + \theta_2) \qquad (2.1)$$

$$z = \ell_1 \mathrm{Cos}\theta_1 + \ell_2 \mathrm{Cos}(\theta_1 + \theta_2) \qquad (2.2)$$

From Equation (2.1) and (2.2), one can get the following equations:

$$(x - \ell_1 \mathrm{Sin}\theta_1)^2 + (z - \ell_1 \mathrm{Cos}\theta_1)^2 = \ell_2^{\,2} \qquad (2.3)$$

$$x^2 + z^2 = \ell_1^{\,2} + \ell_2^{\,2} + 2\ell_1\ell_2 \mathrm{Cos}\theta_2 \qquad (2.4)$$

Obviously, both Equation (2.3) and Equation (2.4) represent loci of circles. More important, θ_1 and θ_2 are not interrelated, i.e. θ_1 only appears in Equation (2.3) and θ_2 in Equation (2.4). From Figure 7, it is easy to find that Equation (2.3) describes the circular arcs $\widehat{DF}$ and $\widehat{EB}$, and Equation (2.4) describes the circular arcs $\widehat{DE}$ and $\widehat{FB}$. Extending these results, one can use Equation (2.3) and Equation (2.4) to construct charts for general workspaces of 2R robots/manipulators. Figure 3 through Figure 5 demonstrate the workspaces of 2R robots with different link ratios. And one can easily get the workspace of a 2R robot when the ranges of motion $(\theta_1)_{min}$, $(\theta_1)_{max}$, $(\theta_2)_{min}$, and $(\theta_2)_{max}$ are specified. The shaded region in Figure 3 demonstrates the workspace of a 2R robot with θ_1 varying from 0° to 60° and θ_2 varying from 30° to 120°.

The workspaces of spatial 2R manipulators will be torus surfaces [15].

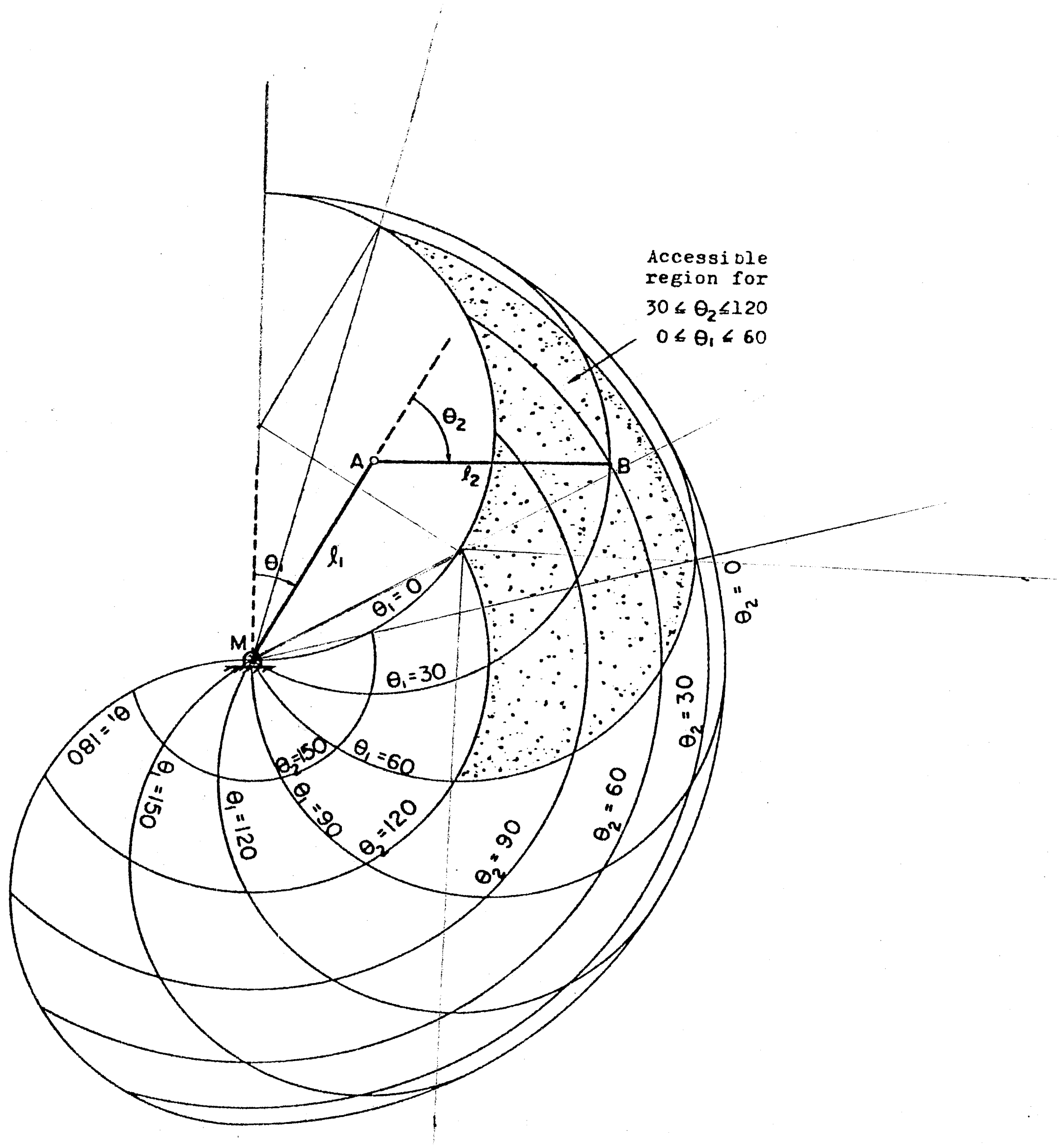


Figure 3. The Accessible Region of the 2R Robot arm ($\ell_2/\ell_1 = 1.0$)

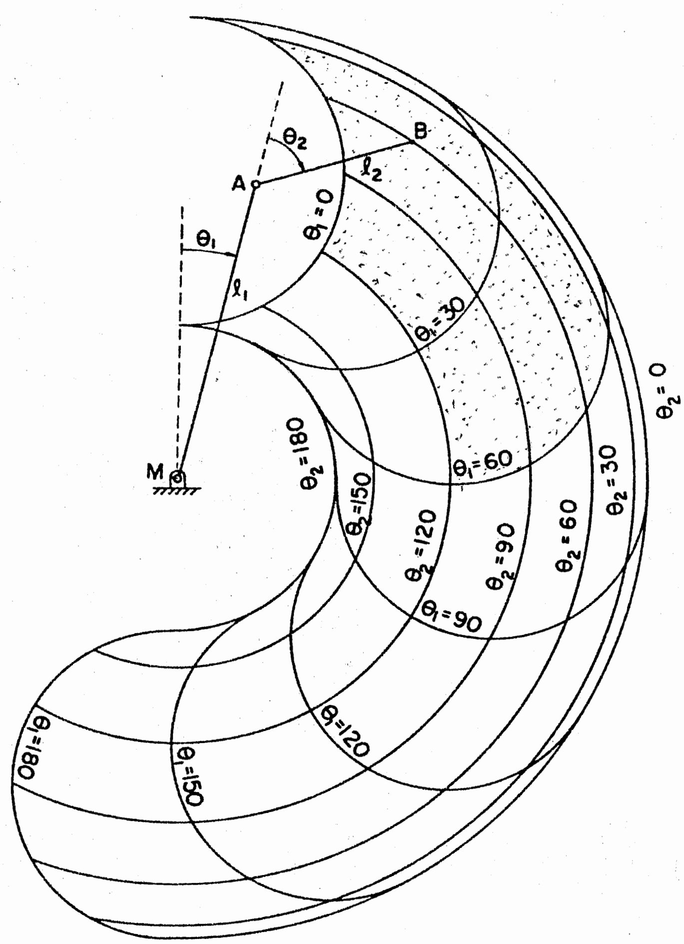


Figure 4. The Accessible Region of the 2R Robot arm ($\ell_2/\ell_1 = 0.5$)

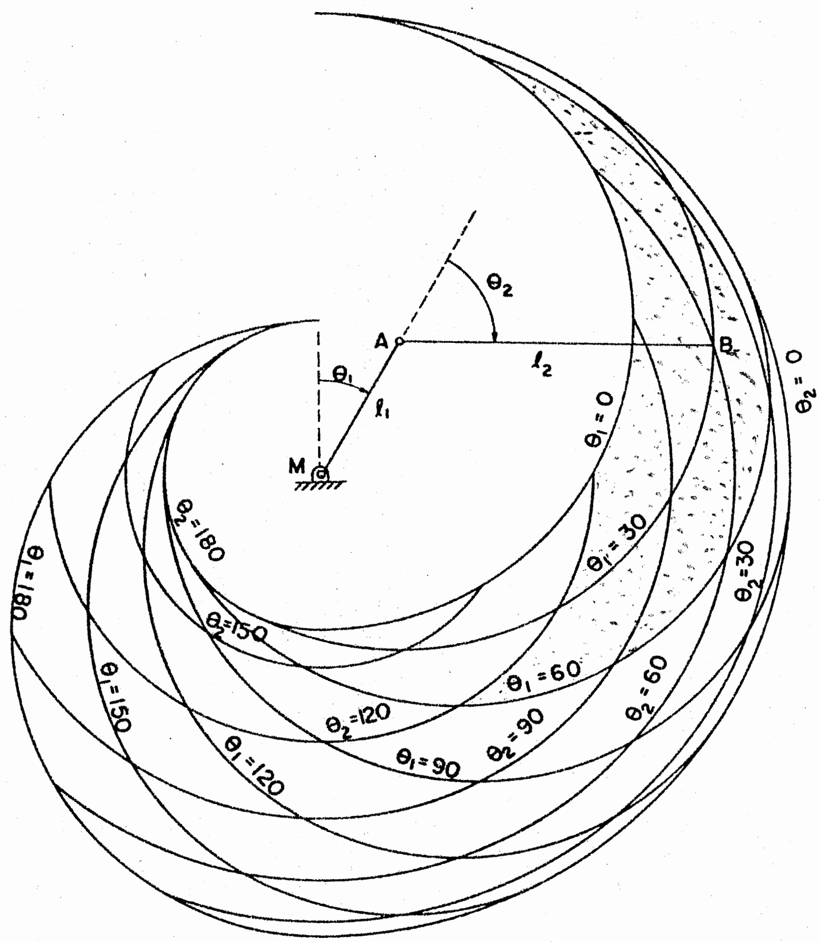


Figure 5. The Accessible Region of the 2R Robot arm (ℓ_2/ℓ_1 = 2.0)

2.2 Workspaces of 3R Robots

2.2.1 The Most Popular 3R Industrial Robots

Figure 6 shows the most popular 3R robot. The coordinates of the robot end can be expressed as,

$$x = \mathrm{Cos}\theta_0[\ell_1\mathrm{Sin}\theta_1 + \ell_2\mathrm{Sin}(\theta_1 + \theta_2)] \quad (2.5)$$

$$y = \mathrm{Sin}\theta_0[\ell_1\mathrm{Sin}\theta_1 + \ell_2\mathrm{Sin}(\theta_1 + \theta_2)] \quad (2.6)$$

$$z = \ell_0 + [\ell_1\mathrm{Cos}\theta_1 + \ell_2\mathrm{Cos}(\theta_1 + \theta_2)] \quad (2.7)$$

The contour of the robot/manipulator can be described by following equations:

$$x = y\mathrm{Cot}\theta_0 \quad (2.8)$$

$$[(x^2 + y^2)^{\frac{1}{2}} - \ell_1\mathrm{Sin}\theta_1]^2 + (z - \ell_1\mathrm{Cos}\theta_1)^2 = \ell_2^{\,2} \quad (2.9)$$

$$x^2 + y^2 + (z-\ell_0)^2 = \ell_1^{\,2} + \ell_2^{\,2} + 2\ell_1\ell_2\mathrm{Cos}\theta_2 \quad (2.10)$$

When $\theta_0 = 0$, i.e. on a cross-section plane containning Z axis, the contour of the workspace of the 3R robot will be the same as that of a 2R robot. In other words, the workspace of the 3R robot mentioned above can be obtained by simply rotating the workspace of the 2R robot about the Z

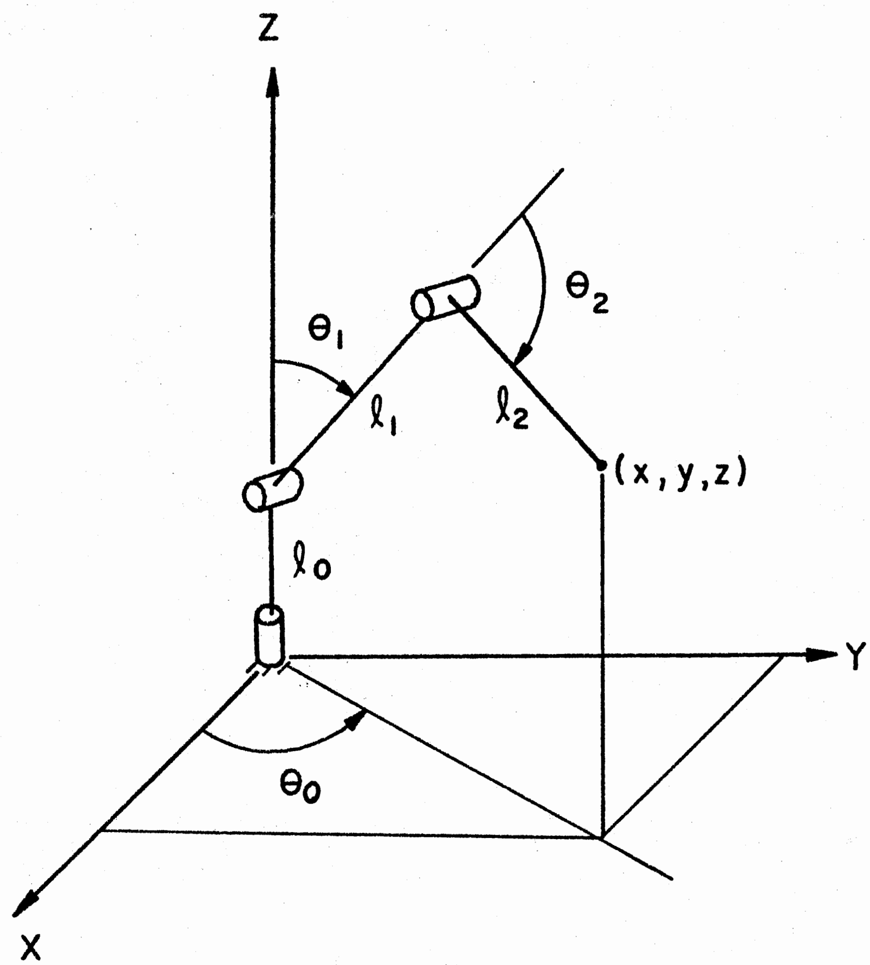


Figure 6. The Most Popular 3R Robot Mechanism

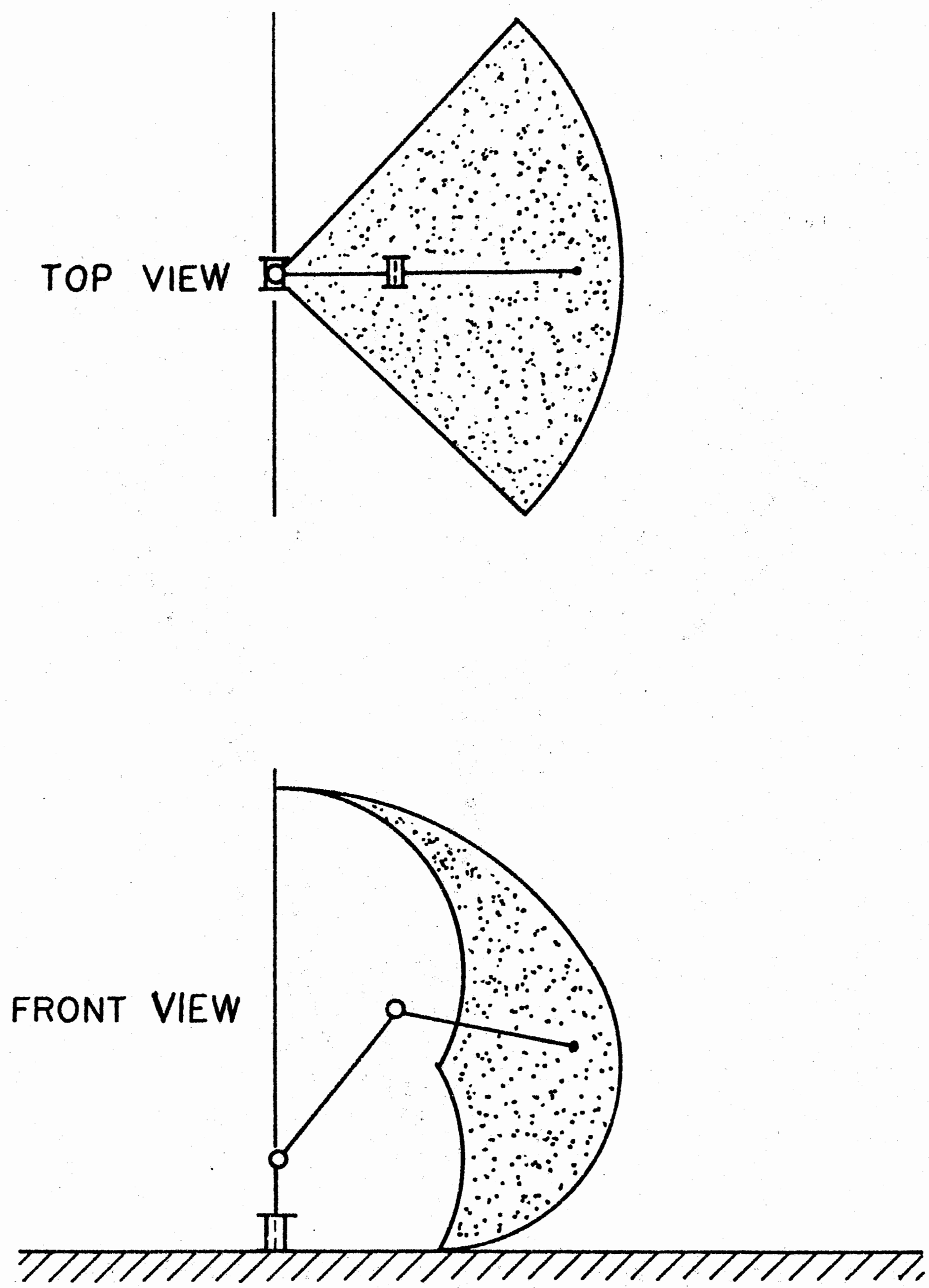


Figure 7. The Workspace of a 3R Industrial Robot

axis through a range from $(\theta_o)_{min}$ to $(\theta_o)_{max}$ (see Figure 7).

2.2.2 General 3R Robots/Manipulators

The workspace of a general 3R robot (see Figure 8) is difficult to be described by simple equations. It is convenient to plot the workspace on a plane containing the first joint axis. And then rotate the plane about the first joint axis througth the entire range of motion to get the workspace of the 3R robot/manipulator in the space.

From Kinematic points of view, links and their associated joints can be defined by four parameters a_i, α_i, s_i, and θ_i as shown in Figure 9.

a_i: The length of the common normal between two joint axes Z_i and Z_{i+1}

α_i: The angle between axes Z_i and Z_{i+1} measured in a right-handed sense, along the axis X_{i+1}. This is called the skew angle or the twist angle of the i^{th} link.

s_i: The distance along the joint axis Z_i from the common normal between X_{i-1} and X_i.

θ_i: The angle measured, in a right-handed sense along Z_i axis, from a_{i-1} to a_i. This is the angle of rotation of the i^{th} link with respect to the $(i-1)^{th}$ link.

The relative change in the orientation and position from $(X_{i+1}, Y_{i+1}, Z_{i+1})$ to (X_i, Y_i, Z_i) due to the joint i can

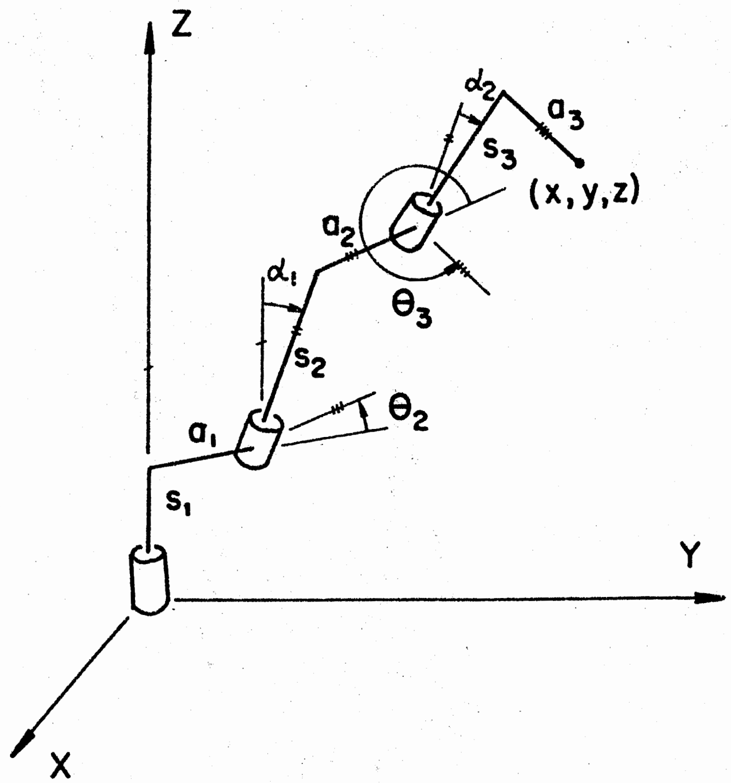


Figure 8. The General 3R Robot Mechanism

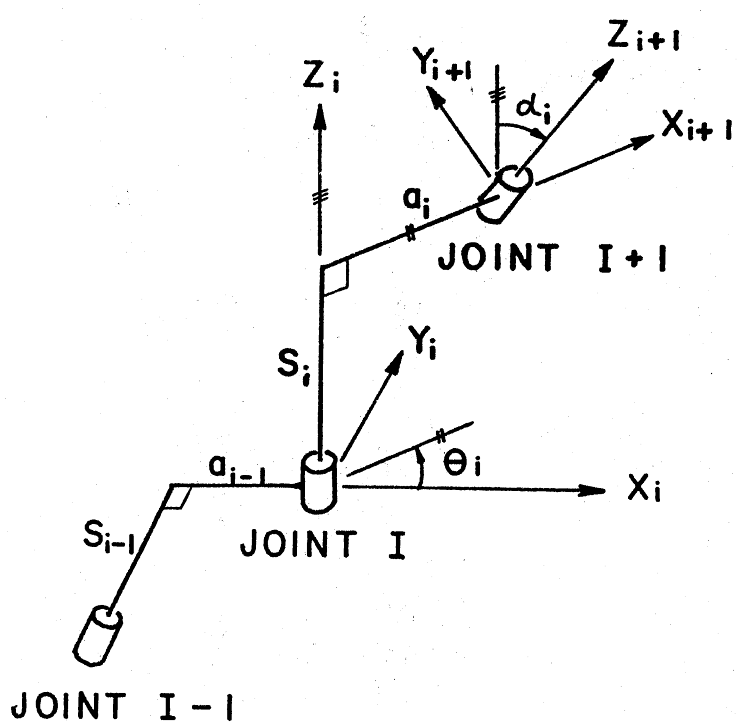


Figure 9. Link Parameters of Joints

be stated as:

$$\{\underline{X}_i\} = A_i \{\underline{X}_{i+1}\} \tag{2.11}$$

where:

$$\{\underline{X}_i\} = (X_i, Y_i, Z_i, 1)^T$$

$$\{\underline{X}_{i+1}\} = (X_{i+1}, Y_{i+1}, Z_{i+1}, 1)^T$$

and

$$A_i = \begin{bmatrix} \text{Cos}\theta_i & -\text{Sin}\theta_i\ \text{Cos}\alpha_i & \text{Sin}\theta_i\ \text{Sin}\alpha_i & a_i\ \text{Cos}\theta_i \\ \text{Sin}\theta_i & \text{Cos}\theta_i\ \text{Cos}\alpha_i & -\text{Cos}\theta_i\ \text{Sin}\alpha_i & a_i\ \text{Sin}\theta_i \\ 0 & \text{Sin}\alpha_i & \text{Cos}\alpha_i & s_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

For n link robot or minaipulator, the coordinate systems will be:

$$\{\underline{X}_1\} = A_{1n} \{\underline{X}_{n+1}\} \tag{2.12}$$

where

$$A_{1n} = A_1 A_2 \ . \ . \ . \ A_n$$

Using Equation (2.11) and Equation (2.12), one can get the coordinates of the end point of the third link in (X_1, Y_1, Z_1) reference frame,

$$x_1 = a_3\ \text{Cos}\theta_2\ \text{Cos}\theta_3 - a_3\ \text{Sin}\theta_2\ \text{Cos}\alpha_2\ \text{Sin}\theta_3$$

$$+ S_3 \operatorname{Sin}\theta_2 \operatorname{Sin}\alpha_2 + a_2 \operatorname{Cos}\theta_2 + a_1 \qquad (2.13)$$

$$y_1 = a_3\operatorname{Cos}\alpha_1\operatorname{Sin}\theta_2\operatorname{Cos}\theta_3 + a_3\operatorname{Cos}\alpha_1\operatorname{Cos}\theta_2\operatorname{Cos}\alpha_2\operatorname{Sin}\theta_3 - a_3\operatorname{Sin}\alpha_1\operatorname{Sin}\alpha_2\operatorname{Sin}\theta_3 - S_3\operatorname{Cos}\alpha_1\operatorname{Cos}\theta_2\operatorname{Sin}\alpha_2 - S_3\operatorname{Sin}\alpha_1\operatorname{Cos}\alpha_2 + a_2\operatorname{Cos}\alpha_1\operatorname{Sin}\theta_2 - S_2\operatorname{Sin}\alpha_1 \qquad (2.14)$$

$$z_1 = a_3\operatorname{Sin}\alpha_1\operatorname{Sin}\theta_2\operatorname{Cos}\theta_3 + a_3\operatorname{Sin}\alpha_1\operatorname{Cos}\theta_2\operatorname{Cos}\alpha_2 \operatorname{Sin}\theta_3 + a_3\operatorname{Cos}\alpha_1\operatorname{Sin}\alpha_2\operatorname{Sin}\theta_3 - S_3\operatorname{Sin}\alpha_1\operatorname{Cos}\theta_2\operatorname{Sin}\alpha_2 + S_3\operatorname{Cos}\alpha_1\operatorname{Cos}\alpha_2 + a_2\operatorname{Sin}\alpha_1\operatorname{Sin}\theta_2 + S_2\operatorname{Sin}\alpha_1 + S_1 \qquad (2.15)$$

The polar projection of (x_1, y_1, z_1) on to the $X_1 - Z_1$ plane will be $(x_1^*, 0, z_1^*)$, where $x_1^* = (x_1^2 + y_1^2)^{\frac{1}{2}}$ and $z_1^* = z_1$.

By increasing θ_2 and θ_3 a small quantity at a time through the entire ranges of motions, one can get the workspace of the 3R robot on X_1-Z_1 plane. One must note that the cross-section area of the workspace obtained by this method is correct only when the cutting plane is away from the motion limits of the first joint. This method will be used in Chapter III to study the effects of the link parameters on the workspaces of 3R regional structure of robot/manipulators.

2.3 Workspaces of n-R Robots

The boundary contour of the workspaces of some robots/ manipulators having three or less links may be described by explicit equations [50]. But in general case, the boundaries of workspaces of n-link robots are very difficult to be described by equations in an explicit way. It is convenient to use an arbitrary plane which cut the workspace and plot the boundary contour on this specified plane (see Figure 10). By rotating or moving the plane, one can determine the three dimensional workspace of the robot. In practical application, the workspace on a specified plane is very important when the robot works on an inclined working surface as shown in Figure 11.

In order to plot the contour of the workspace on an arbitrarily specified plane, the following steps have been used :

1. Locate the robot hand on the specified plane.

2. Move the robot hand on the specified plane until it reaches the boundary of the workspace.

3. Move the robot hand on the specified plane from one position to its neighboring position along the boundary of the workspace.

2.3.1 Locate the Robot Hand on the Specified Plane

To locate the robot hand on the specified plane belongs

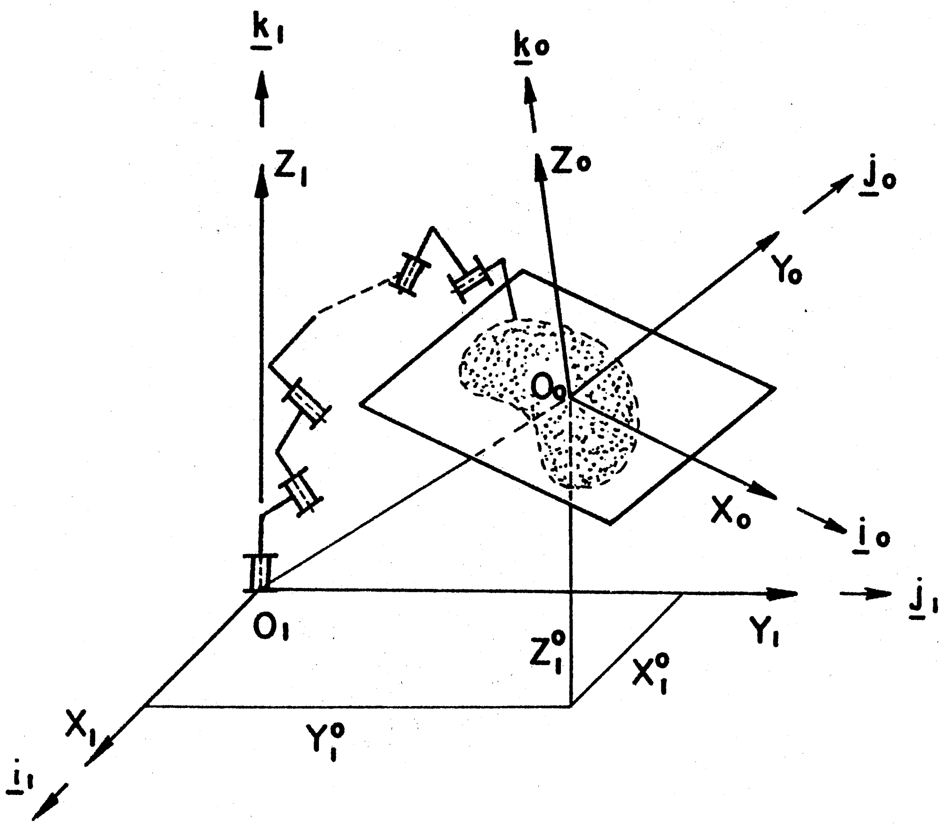


Figure 10. The Workspace on an Arbitrary Plane

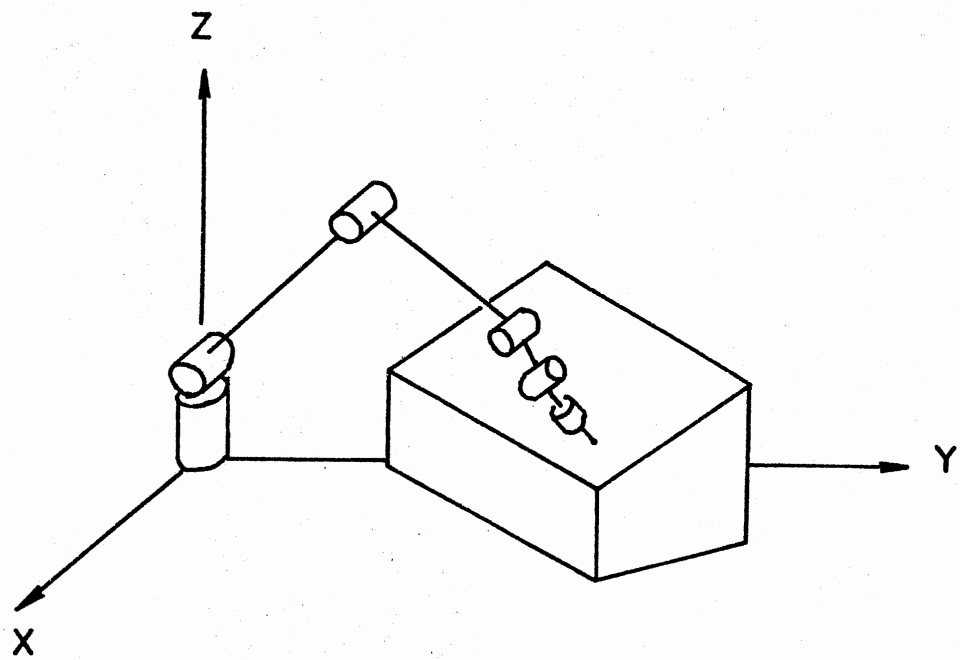


Figure 11. An Example of the Workspace on an Arbitrary Plane

to a joint displacement analysis problem. When two subsequential rigid-body positions of the robot hand of a 6R robot are given, one can use either Newton-Raphson method or velocity method [36] to solve the joint displacements of the robot. But when the robot hand is treated as a point or a line, or when the robot consists of more than six revolute joints, then the number of unknowns (joint displacements) will be more than the number of independent equations. In this case one may need to find some other methods.

Now consider there is a small increment on the i^{th} joint displacement,

$$\theta_i = \theta_{io} + \delta\theta_i \tag{2.16}$$

where θ_{io} is the old joint displacement and $\delta\theta_i$ is the small increment. Substitute Equation (2.16) into (2.11), and expand the terms Sin $(\theta_{io} + \delta\theta_i)$ and Cos $(\theta_{io} + \delta\theta_i)$. For small $\delta\theta_i$, let Sin $\delta\theta_i = 0$ and Cos $\delta\theta_i = 1$, it yields:

$\sin\delta\theta_i = \delta\theta_i$

$$A_i = A_i^o + \delta\theta_i \; B_i^o \tag{2.17}$$

where:

$$A_i^o = \begin{bmatrix} \cos\theta_{io} & -\sin\theta_{io}\cos\alpha_i & \sin\theta_{io}\sin\alpha_i & a_i\cos\theta_{io} \\ \sin\theta_{io} & \cos\theta_{io}\cos\alpha_i & -\cos\theta_{io}\sin\alpha_i & a_i\sin\theta_{io} \\ 0 & \sin\alpha_i & \cos\alpha_i & S_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and

$$B_i^o = \begin{bmatrix} -\mathrm{Sin}\theta_{io} & -\mathrm{Cos}\theta_{io}\ \mathrm{Cos}\alpha_i & \mathrm{Cos}\theta_{io}\ \mathrm{Sin}\alpha_i & -a_i\ \mathrm{Sin}\theta_{io} \\ \mathrm{Cos}\theta_{io} & -\mathrm{Sin}\theta_{io}\ \mathrm{Cos}\alpha_i & \mathrm{Sin}\theta_{io}\ \mathrm{Sin}\alpha_i & a_i\ \mathrm{Cos}\theta_{io} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

If one is interested in coordinates $\{\underline{X}_o\}$ on an arbitrarily specified plane (see Figure 10) rather than $\{\underline{X}_1\}$ on the base joint of robot. Equation (2.16) should be modified as,

$$\{\underline{X}_o\} = A_{on}\ \{\underline{X}_{n+1}\} \tag{2.18}$$

where:

$$A_{on} = A_o\ A_1\ A_2\ \cdot\ \cdot\ \cdot\ A_n$$

and

$$A_o = \begin{bmatrix} \underline{i}_o \cdot \underline{i}_1 & \underline{i}_o \cdot \underline{j}_1 & \underline{i}_o \cdot \underline{k}_1 & -X_1^o \\ \underline{j}_o \cdot \underline{i}_1 & \underline{j}_o \cdot \underline{j}_1 & \underline{j}_o \cdot \underline{k}_1 & -Y_1^o \\ \underline{k}_o \cdot \underline{i}_1 & \underline{k}_o \cdot \underline{j}_1 & \underline{k}_o \cdot \underline{k}_1 & -Z_1^o \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.19}$$

Substitute Equation (2.17) into (2.18) and simplify, to get

$$A_{on} = A_{on}^{o} + \delta\theta_1 (A_o B_1^o A_2^o A_3^o \ldots A_n^o)$$
$$+ \delta\theta_2 (A_o A_1^o B_2^o A_3^o \ldots A_n^o)$$
$$+ \ldots\ldots\ldots\ldots\ldots \tag{2.20}$$
$$+ \delta\theta_n (A_o A_1^o A_2^o A_3^o \ldots B_n^o)$$

Define:

$$\delta A_{on} = A_{on} - A_{on}^{o}$$

$$= \begin{bmatrix} \delta L_1 & \delta M_1 & \delta N_1 & \delta X_o \\ \delta L_2 & \delta M_2 & \delta N_2 & \delta Y_o \\ \delta L_3 & \delta M_3 & \delta N_3 & \delta Z_o \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.21}$$

then

$$\begin{bmatrix} \delta L_1 & \delta M_1 & \delta N_1 & \delta X_o \\ \delta L_2 & \delta M_2 & \delta N_2 & \delta Y_o \\ \delta L_3 & \delta M_3 & \delta N_3 & \delta Z_o \\ 0 & 0 & 0 & 0 \end{bmatrix} = \delta\theta_1 (A_o B_1^o A_2^o A_3^o \ldots A_n^o)$$
$$+\delta\theta_2 (A_o A_1^o B_2^o A_3^o \ldots A_n^o)$$
$$+ \ldots\ldots\ldots\ldots\ldots$$
$$+\delta\theta_n (A_o A_1^o A_2^o A_3^o \ldots B_n^o)$$

(2.22)

Where δL_1, δL_2, δL_3, δM_1, δM_2, δM_3, δN_1, δN_2, and δN_3 are referred to the change of orientation of the last coordinate from $\{\underline{X}_{n+1}\}$ with respect to the reference frame $\{\underline{X}_o\}$. And δX_o, δY_o, and δZ_o are referred to the change of position. Since there are only six degrees of freedom for a rigid body motion, one can get only six independent equations from twelve nontrivial equations in Equation (2.20). One can choose the following combination without losing the ginerality,

position

$$\delta X_o = f_1\ (\delta\theta_1, \delta\theta_2, \ldots \delta\theta_n) \tag{2.23}$$

$$\delta Y_o = f_2\ (\delta\theta_1, \delta\theta_2, \ldots \delta\theta_n) \tag{2.24}$$

$$\delta Z_o = f_3\ (\delta\theta_1, \delta\theta_2, \ldots \delta\theta_n) \tag{2.25}$$

$$\delta L_3 = f_4\ (\delta\theta_1, \delta\theta_2, \ldots \delta\theta_n) \tag{2.26}$$

$$\delta M_1 = f_5\ (\delta\theta_1, \delta\theta_2, \ldots \delta\theta_n) \tag{2.27}$$

$$\delta N_3 = f_6\ (\delta\theta_1, \delta\theta_2, \ldots \delta\theta_n) \tag{2.28}$$

Of course it is not necessary to choose δL_3, δM_1, and δN_3 in Equation (2.26), (2.27) and (2.28) all the time. One may choose some other combinations, according to his convenience, like

1. δL_3, δM_1 and δN_1 when the robot hand is perpendicular to the axis of last joint axis and it (the robot hand) is treated as a rigid-body.

2. δN_3, δL_1 and δM_1 when the robot hand is along the

axis of the last joint axis and it is treated as a rigid-body.

3. δL_3, δL_1 (δL_2) when the robot hand is perpendicular to the axis of last joint axis and it is treated as a line.

4. δN_3, δN_1 (δN_2) when the robot hand is along the axis of the last joint axis and it is treated as a ~~rigid-body~~ line.

Because Equation (2.23) through (2.28) held only when $\delta\theta_1, \delta\theta_2, \ldots \delta\theta_n$ are small, one must put upper and lower bounds to them. In this way it makes sure that every $\delta\theta_i$ will be small and every $\theta_i + \delta\theta_i$ will be within the limit positions of that joint. But at the same time we want the robot hand move to the specified plane as fast as possible. This leads to a linear programming problem with equality constraints and bounded variables. It may be stated as,

maximize (or minimize) $\delta Z_o = f_3\ (\delta\theta_1, \delta\theta_2, \ldots, \delta\theta_n)$

subject to:

$$\delta L_3 = f_4\ (\delta\theta_1, \delta\theta_2, \ldots, \delta\theta_n) \tag{2.29}$$

$$\delta M_1 = f_5\ (\delta\theta_1, \delta\theta_2, \ldots, \delta\theta_n) \tag{2.30}$$

$$\delta N_3 = f_6\ (\delta\theta_1, \delta\theta_2, \ldots, \delta\theta_n) \tag{2.31}$$

and

$$(\delta\theta_i)_{min} \leq \delta\theta_i \leq (\delta\theta_i)_{max} \tag{2.32}$$

$$i = 1, 2, \ldots, n$$

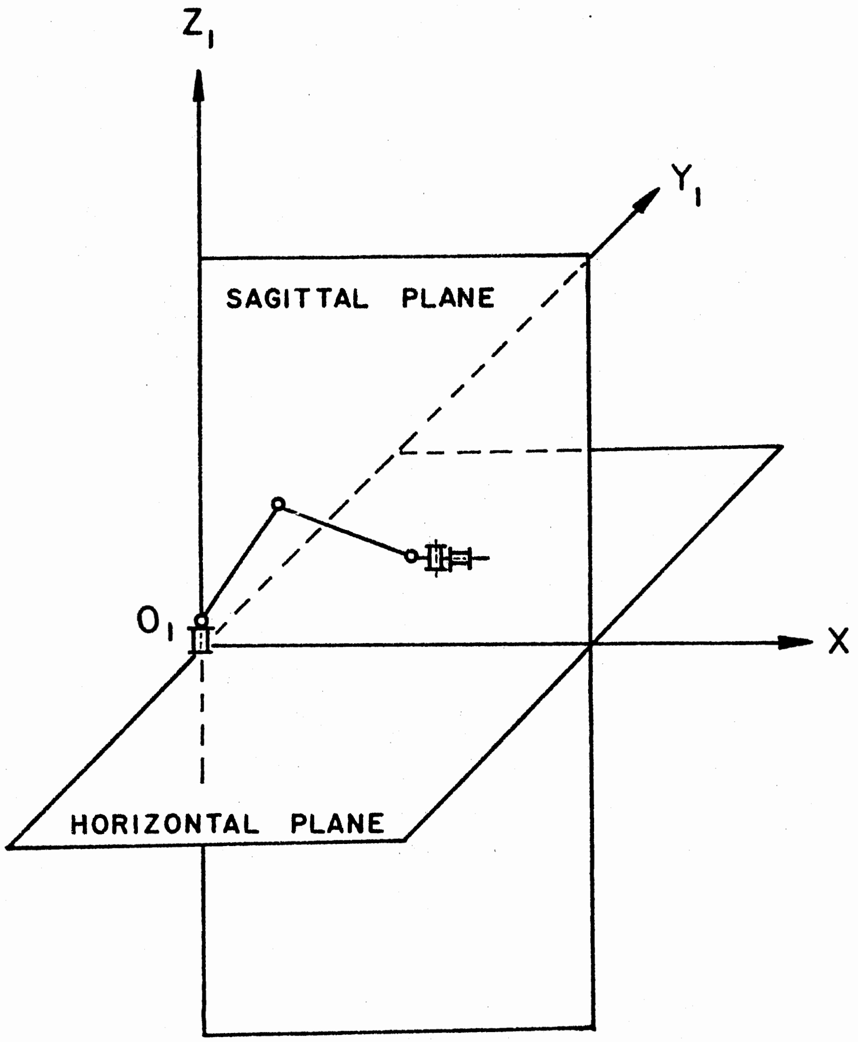


Figure 12. Typical Plane for Plotting Workspace

where δL_3, δM_1 and δN_3 are the required changes in orientations of the robot hand from some initial state to the specified state. If the required changes in the orientations are big, it may lead the problem to have no feasible solution. In this case one should break it into some intermediate steps and make δL_3, δM_1, and δN_3 small. If the robot hand is treated as a point, the equality constraints (2.29), (2.30) and (2.31) should be deleted.

2.3.2 Move the Robot Hand to the Boundary of the Workspace

After moving the robot hand on the specified plane, the next step is to move the robot hand on the plane along X_o axis. It means $\delta Y_o = \delta Z_o = 0$. If the orientation of the robot hand is kept unchanged also, then $\delta L_3 = \delta M_1 = \delta N_3 = 0$. Of course we also want the robot hand to move toward the boundary of the workspace as fast as possible while all $\delta\theta_i$ are kept small. It yields,

$$\text{maximize } \delta X_o = f_1\ (\delta\theta_1, \delta\theta_2, \ldots \delta\theta_n)$$

subject to:

$$f_2\ (\delta\theta_1, \delta\theta_2, \ldots \delta\theta_n) = \delta Y_o = 0 \quad (2.33)$$

$$f_3\ (\delta\theta_1, \delta\theta_2, \ldots \delta\theta_n) = \delta Z_o = 0 \quad (2.34)$$

$$f_4\ (\delta\theta_1, \delta\theta_2, \ldots \delta\theta_n) = \delta L_3 = 0 \quad (2.35)$$

$$f_5\ (\delta\theta_1, \delta\theta_2, \ldots \delta\theta_n) = \delta M_1 = 0 \quad (2.36)$$

$$f_6\ (\delta\theta_1,\ \delta\theta_2,\ .\ .\ .\ \delta\theta_n) = \delta N_3 = 0 \qquad (2.37)$$

and

$$(\delta\theta_i)_{min} \leq \delta\theta_i \leq (\delta\theta_i)_{max} \qquad (2.38)$$

$$i = 1,\ 2,\ .\ .\ .\ n$$

$0 \leq \delta\phi_i \leq U_i$
where $\delta\phi_i = \delta\theta_i - (\delta\theta_i)_{min}$
$U_i = (\delta\theta_i)_{max} - (\delta\theta_i)_{min}$

In case the orientation of the robot hand is not required to be held constant, the constraint Equations (2.35), (2.36) and (2.37) should be deleted.

2.3.3 Trace the Contour of the Workspace

Once the robot hand has reached the boundary of the workspace, the last step is to trace the contour of the workspace. One may use Mason's contour method [29] or its modified method by Cordray (Appexdix B) to plot the contour.

Before one can apply Cordray's method he needs to detect if a adjacent meshpoint is inside or outside the boundary. Let X, Y, Z, L_3, M_1, and N_3 are the position and orientation of the robot hand which is inside the boundary in the last step. And the robot hand wants to move to the adjacent position (X+δX, Y+δY, Z) without changing its orientation. From Equation (2.23) through (2.28), we get

$$f_1\ (\delta\theta_1,\ \delta\theta_2,\ .\ .\ .\ .\ ,\ \delta\theta_n) = \delta X_o \qquad (2.39)$$

$$f_2\ (\delta\theta_1,\ \delta\theta_2,\ .\ .\ .\ .\ ,\ \delta\theta_n) = \delta Y_o \qquad (2.40)$$

$$f_3\ (\delta\theta_1,\ \delta\theta_2,\ .\ .\ .\ .\ ,\ \delta\theta_n) = 0 \qquad (2.41)$$

$$f_4\ (\delta\theta_1,\ \delta\theta_2,\ .\ .\ .\ .\ ,\ \delta\theta_n) = 0 \qquad (2.42)$$

$$f_5\ (\delta\theta_1,\ \delta\theta_2,\ .\ .\ .\ .\ ,\ \delta\theta_n) = 0 \qquad (2.43)$$

$$f_6\ (\delta\theta_1,\ \delta\theta_2,\ .\ .\ .\ .\ ,\ \delta\theta_n) = 0 \qquad (2.44)$$

If the robot consists of six revolute joints, i.e. $n = 6$, then there are six unknowns and six independent equations. One can solve $\delta\theta_1$, $\delta\theta_2$, ... , $\delta\theta_6$ and check $\theta_1 + \delta\theta_1$, $\theta_2 + \delta\theta_2$, ... , $\theta_6 + \delta\theta_6$ if all of them are within their motion ranges. In this way one can tell if the robot hand can move to this adjacent point or not. But in general case the number of unknowns may be more than the number of independent equations. For example, if the robot hand of a 6R robot is treated as a point then there are six unknowns $\delta\theta_1$, $\delta\theta_2$, ... , $\delta\theta_6$, and only three independent equations, Equation (2.39), (2.40) and (2.41). It leads to an optimization problem. In general case, it yields

maximize the objective function $af_1 + bf_2$

subject to:

$$f_1\ (\delta\theta_1,\ \delta\theta_2,\ .\ .\ .\ .\ ,\ \delta\theta_n) = \delta X_o \qquad (2.45)$$

$$f_2\ (\delta\theta_1,\ \delta\theta_2,\ .\ .\ .\ .\ ,\ \delta\theta_n) = \delta Y_o \qquad (2.46)$$

$$f_3\ (\delta\theta_1,\ \delta\theta_2,\ .\ .\ .\ .\ ,\ \delta\theta_n) = 0 \qquad (2.47)$$

$$f_4 (\delta\theta_1, \delta\theta_2, \ldots, \delta\theta_n) = 0 \quad (2.48)$$

$$f_5 (\delta\theta_1, \delta\theta_2, \ldots, \delta\theta_n) = 0 \quad (2.49)$$

$$f_6 (\delta\theta_1, \delta\theta_2, \ldots, \delta\theta_n) = 0 \quad (2.50)$$

and

$$(\delta\theta_i)_{min} \leq \delta\theta_i \leq (\delta\theta_i)_{max} \quad (2.51)$$

$i = 1, 2, \ldots, n$

a, b are constants.

If the adjacent position $(X+\delta X, Y+\delta Y, 0)$ is inside the boundary and δX, δY are small then the above equations should have feasible solutions. Otherwise no feasible solution exists. In this way one can tell if the adjacent position inside or outside the contour. Then apply Cordray's method (see Appendix B) to plot the contour of the accessible workspace.

2.4 Verification and Limitation

A popular 6R robot has following kinematic parameters,

$a_1 = 0.0$	$a_2 = 0.4$	$a_3 = 0.4$
$a_4 = 0.08$	$a_5 = 0.0$	$a_6 = 0.0$
$s_1 = 0.0$	$s_2 = 0.0$	$s_3 = 0.0$
$s_4 = 0.0$	$s_5 = 0.0$	$s_6 = 0.12$
$\alpha_1 = 90°$	$\alpha_2 = 0°$	$\alpha_3 = 0°$
$\alpha_4 = -90°$	$\alpha_5 = 90°$	$(\alpha_6 = 0°)$

θ_1 = -120° to 120°

θ_2 = 0° to 90°

θ_3 = -120° to 0°

θ_4 = -120° to 120°

θ_5 = -30° to 210°

θ_6 = -240° to 240°

Computerize the algorithm developed in Section 2.3 and plot the workspace of this robot on following planes,

1. The sagittal plane of the robot, i.e. the X_1- Z_1 plane of the robot.
2. A Horizontal plane at the base joint, i.e. the X_1- Y_1 plane of the robot.

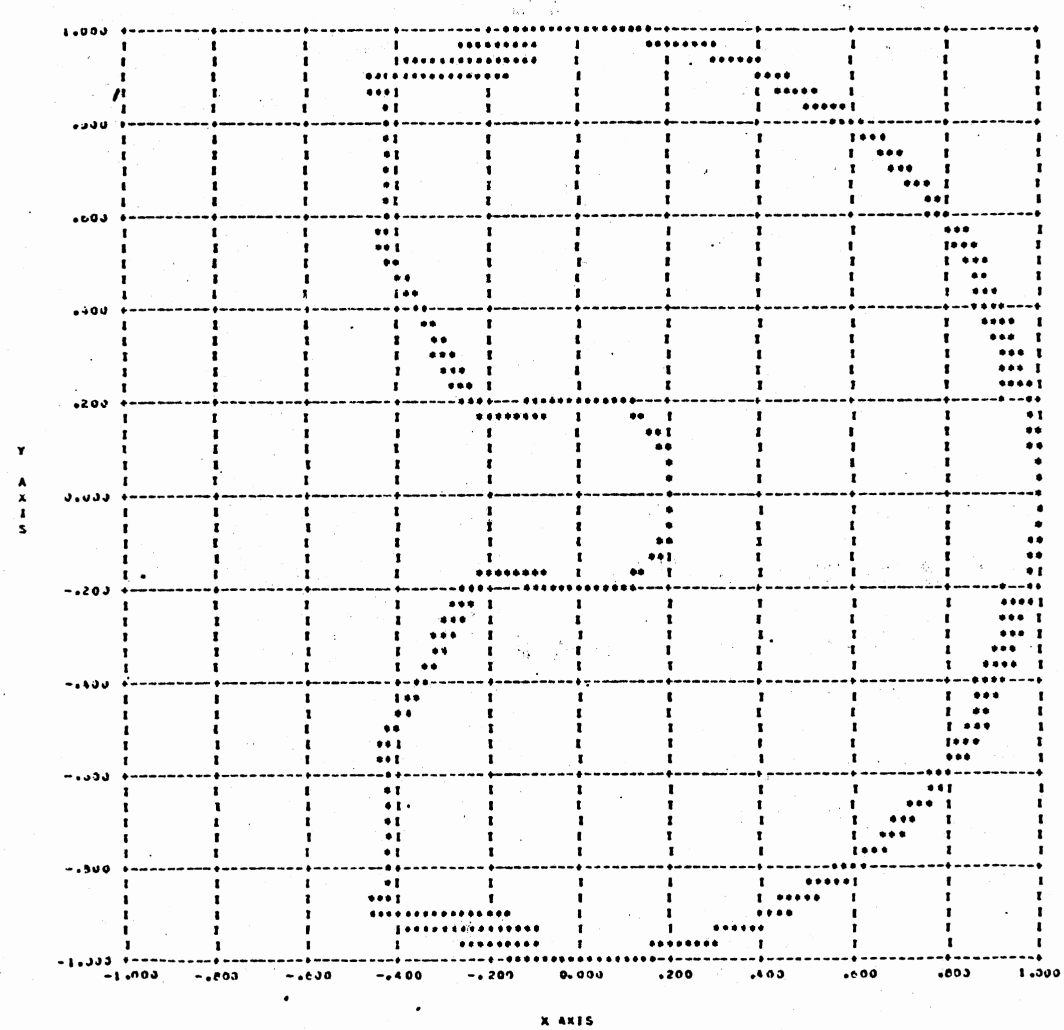
Figure 13 shows the accessible workspace of this robot at these planes.

The algorithm developed in Section 2.3 is based upon the hypothesis of small increment of joint displacement at each joint. This leads to the following limitations:

1. In order to get good results, one should select small step size for δX, δY, δZ, $\delta\theta_1$, $\delta\theta_2$, ... , $\delta\theta_n$. For a normalized robot, like the robot in the above example which has unit total link length, one can choose δx, δy, δz equal to 0.002 and $\delta\theta_1$, $\delta\theta_2$, . . . $\delta\theta_n$ equal to 2.5° as the starting values and make adjustment later on if necessary.

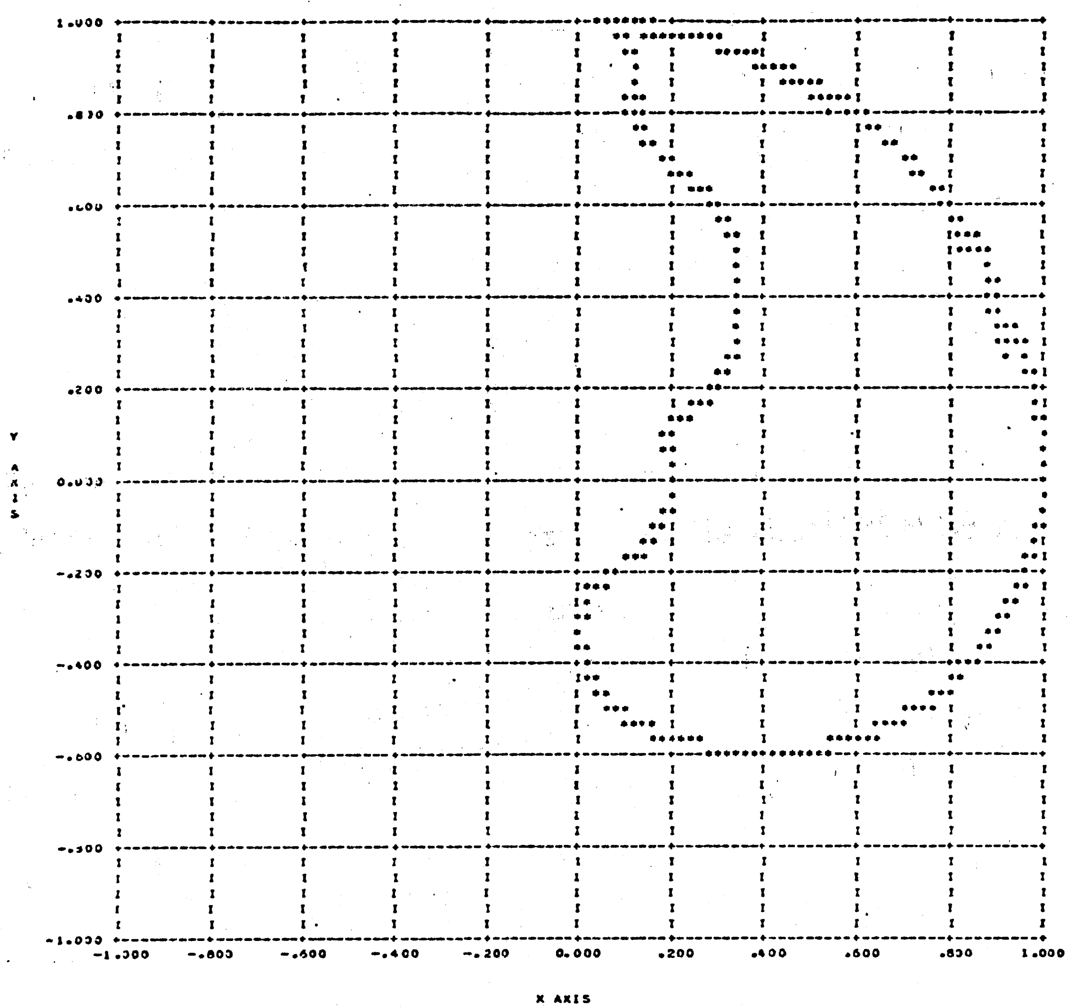
2. If the robot consists of joints which can rotate from negative range of motion to positive range of motion,

the workspace of this robot will probably consist of several sub-workspaces corresponding to different configurations of joint motions. Figure 14 shows a 2R robot of this case. In order to move from one sub-workspace to the other one, the robot links should be fully extend out then fold into other configurations. In other words, the joint displacements change from some negative values to zero first, then to positive values. This also true for a n-R robot. It is suggested to plot the contour of workspace from fully extended position in counter clockwise direction first then in clockwise direction. Combine these two sub-workspaces to get the workspace of the robot. For more complicated cases it also suggested to asign different values (say 1, 0, -1,) to the constants a and b of the objective function in Section 2.3.3. In this way one could hope to find the entire workspace of a given robot with complicated joint displacements.



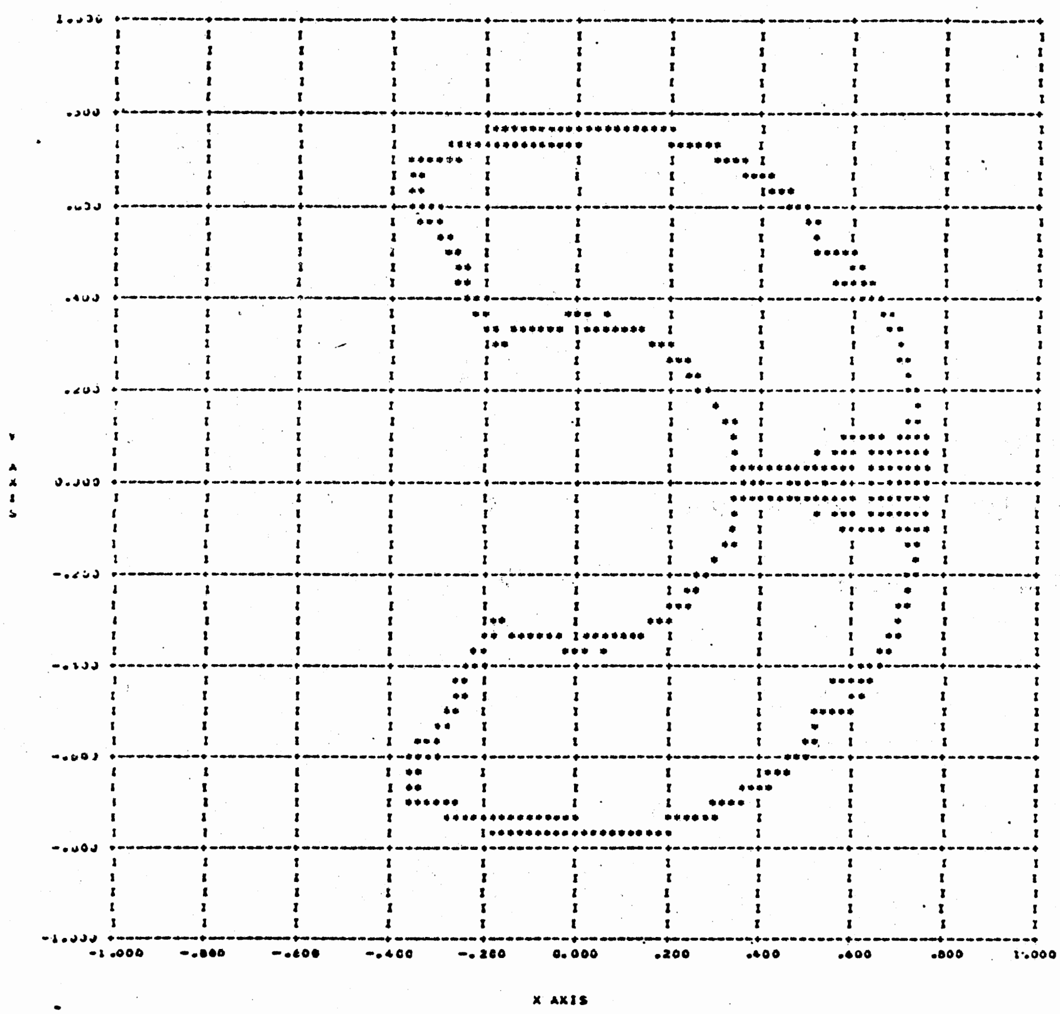
(a) On the Horizontal Plane (as a Point)

Figure 13. The Workspace of 6R Robot on the Specified Plane



(b) On the Sagittal Plane (as a Point)

Figure 13. (Continued)



(c) On the Horizontal Plane (as a Rigid Body)

Figure 13. (Continued)

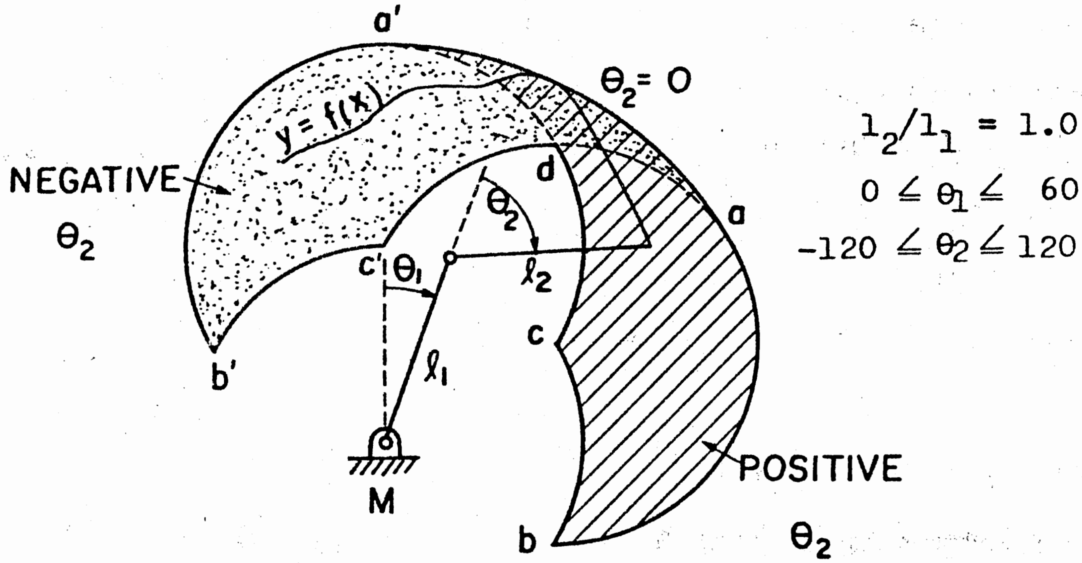


Figure 14. The Workspace of a 2R Robot With Positive and Negative Joint Displacements

CHAPTER III

EFFECTS OF LINK PARAMETERS ON THE WORKSPACES AND DEXTERITIES OF ROBOTS/MANIPULATORS

From a kinematics point of view, each revolute joint has three time-independent link parameters (a, α, and s) and one time-dependent joint variable θ. For the general robot having six revolute joints, it has 18 link parameters and 6 joint variables. Because the workspace is affected by so many parameters and variables, it is very difficult to analyze the effects of all link parameters on the workspaces at the same time. Fortunately, the structures of industrial robots can be divided into two parts: "regional structure" and "orientational structure." The regional structure, consisting of shoulder and arms, contributes the gross motion to the robot hand. The orientational structure, consisting of the wrist and hand, contributes the orientation to the robot hand. When there is a load applied on the robot hand, the torques existing on the joints near to the shoulder (base joint) are much larger than that on the joints near to the hand most of the time. From minimum energy consumption point of view, one wants the joint displacements of the regional structure as small as possible when the robot hand works around some working stations. So it is better to have a large regional structure and small orientational structure. Considering the primary and secondary working space of robots, Gupta and Roth [19] also point out the same conclusion. The working space of a robot, with a large regional structure and small orientational structure, is dominated by the regional structure. In other words one may almost understand the characteristics of the workspace of the robot by studying the workspace of its regional structure. And one can check the dexterity, the capability of reach the working station in any direction, of the robot by studying its orientational structure.

3.1 Effects of Link Parameters on the Workspaces of Robots/Manipulators

In order to study the effects of link parameters on the workspaces of general 3R regional structures of robots, one may compare the shapes and volumes of workspaces of robots with different link paremeters. Before comparing two robots, one should let these robots have the same total link length and same joint displacements. For convenience let the total of lengths of all the links be equal to one, and let all the joints make a complete rotation. Using the method described in Section 2.2, one can write computer programs to plot the contours of the workspaces and calculate the volumes of the workspaces of general 3R regional structures of robots/manipulators.

The effects of link parameters (a, α and s) on the workspaces of spatial 3R robots have been studied in the following sections.

3.1.1 The Effect of Link Parameters a_1, a_2, and a_3

Tables II, III and IV show the effect of a_1, a_2, and a_3 on shapes of the workspace, and Figure 15, 16, and 17 present some illustrative plots of the normalized volume of workspace (volume of workspace of a robot with unit total link length) as a function of a_1, a_2, and a_3.

From Table II and Figure 15, the following results are observed:

1. The link ratio a_3/a_2 equals to one is the necessary condition for 3R robot arms to get workspaces without voids.

2. Let $(a_3/a_2)_1 = K_1(> 1)$ and $(a_3/a_2)_2 = K_2(< 1)$, If $K_1\ K_2 = 1$ then the volumes and shapes of the workspaces of both cases are equal.

3. When $a_1 = 0$, and $a_2 = a_3$, the working space will be maximum.

3.1.2 The Effect of Link Parameters α_1 and α_2

The effect of α_1 and α_2 on the shape of workspace is shown in Table III through Table VI. For different combinations of α_1 and α_2 one may get working spaces with or without voids as shown in Table III. Figure 16 to 20 show how the α_1 and α_2 affect the volume of workspace of the 3R robot arms. Following observations are made.

1. In order to get the maximum working space and a workspace without voids, it is better to choose $\alpha_2=0°$ or $\pm 180°$ and $\alpha_1 = \pm 90°$.

2. When $a_2=a_3$ and $s_2=s_3=0$, the shapes and volumes of workspaces are symmetrical about $\alpha_1=0°$, $90°$ and $\alpha_2=0°$, $90°$.

3. When $a_2=a_3$, $s_2=s_3=0$, $\alpha_1=90°$ and $(a_1/\ell) \leq \frac{1}{2}$, the value of α_2 does not affect the shape and volume of the workspace (see Table V and Figure 16).

4. For 3R robot arms with $a_1=0$, $a_2=a_3$, $s_2=s_3=0$ and $\alpha_1 \approx 60°$, the volume of the workspace does not change much when α_2 changes (see Figure 19 and 20).

5. When $a_1=0$ and $\alpha_1=0$, the 3R robot arms will be degenerated. The working space of such a robot will become a torus similar to that generated by a spatial 2R robot arm.

6. Increasing the value of α_2 from $0°$ to $90°$ will increase the void in the workspace. Keep $\alpha_2=0°$ or $180°$ and $a_2=a_3$ will provide workspaces without voids for all values of α_1.

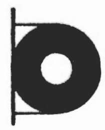
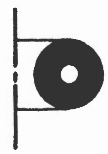
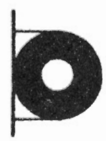
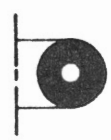
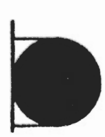
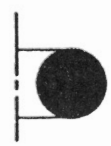
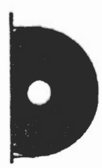
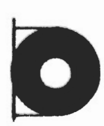
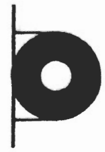
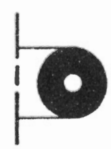
3.1.3 The Effect of Link Parameters

s_1, s_2, and s_3

The link parameter s_1 has no effect on the shape of workspace; it just shift the workspace up or down. The

TABLE II

EFFECT OF a_1, a_2 AND a_3 ON THE SHAPE OF CROSS-SECTION OF THE WORKSPACE OF 3R ROBOT ARMS WITH $S_1 = S_2 = S_3 = 0$, $\alpha_1 = 90$, $\alpha_2 = 0$

a_3/a_2 \ a_1/ℓ		< 0.5	= 0.5	> 0.5
< 1	$< 1-2\frac{a_1}{\ell}$			
	$\geqslant 1-2\frac{a_1}{\ell}$			
= 1				
> 1	$\leqslant \frac{1}{1-2\frac{a_1}{\ell}}$			
	$> \frac{1}{1-2\frac{a_1}{\ell}}$			

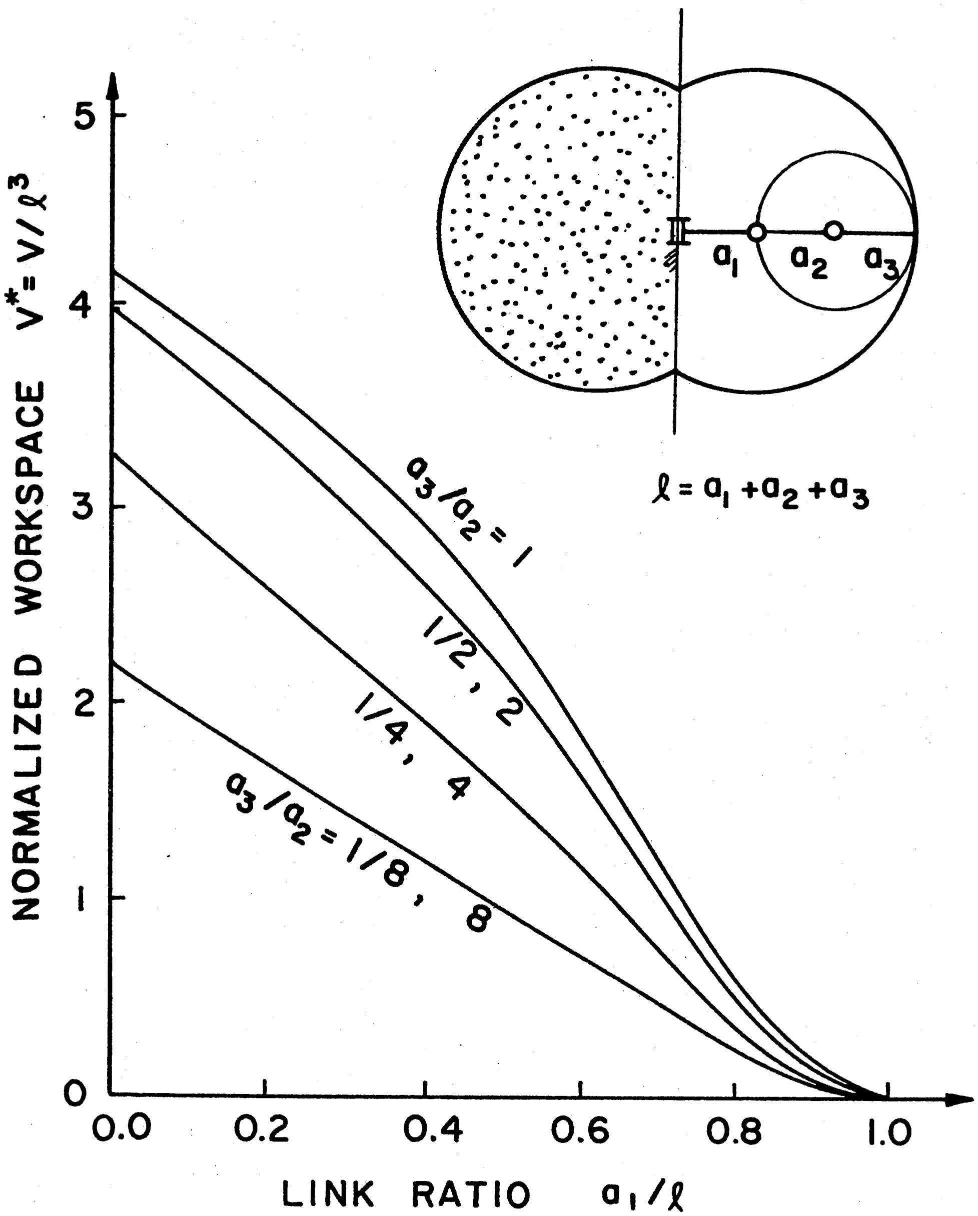


Figure 15. Effect of Link Length Ratio on the Working Space of 3R Regional Structure of Robots

TABLE III

EFFECT OF α_1 AND α_2 ON THE SHAPE OF CROSS-SECTION OF THE WORKSPACE OF 3R ROBOT ARMS WITH $a_1 = 0$, $a_2 = a_3$, $S_1 = S_2 = S_3 = 0$

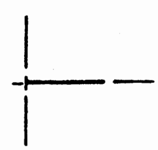
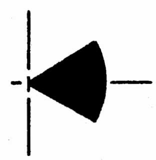
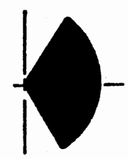
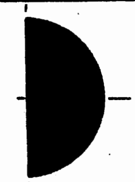
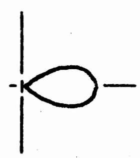
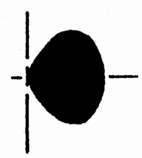
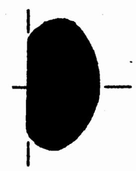
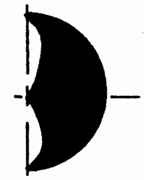
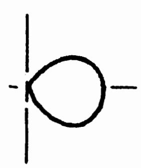
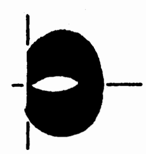
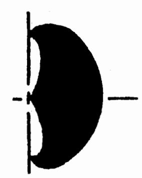
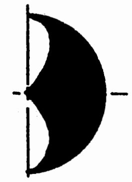
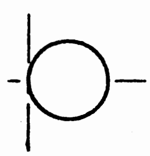
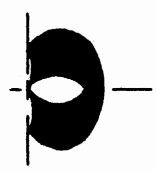
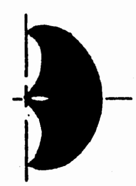
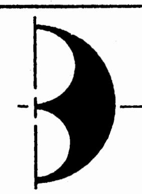
α_2 \ α_1	0, ± 180	± 30, ± 150	± 60, ± 120	± 90
0, ± 180				
± 30, ± 150				
± 60, ± 120				
± 90				

TABLE IV

EFFECT OF α_1 AND α_2 ON THE VOID OF THE WORK-SPACE OF 3R ROBOT ARMS WITH $a_1 = 0$, $a_2 = a_3$, $S_1 = S_2 = S_3 = 0$

<table>
<tr><th>α_1 / α_2</th><th>0 , ± 180</th><th>± 30, ± 150</th><th>± 60, ± 120</th><th>± 90</th></tr>
<tr><td>0 , ± 180</td><td colspan="4" rowspan="2">SOLID</td></tr>
<tr><td>± 30, ± 150</td></tr>
<tr><td>± 60, ± 120</td><td colspan="2" rowspan="2">WITH ONE VOID</td><td colspan="2">WITH TWO VOIDS</td></tr>
<tr><td>± 90</td><td colspan="2">WITH THREE VOIDS</td></tr>
</table>

TABLE V

EFFECT OF a_1 AND α_2 ON THE SHAPE OF CROSS-SECTION OF THE WORKSPACE OF 3R ROBOT ARMS WITH $\alpha_1 = 90$, $a_2 = a_3$, $S_1 = S_2 = S_3 = 0$

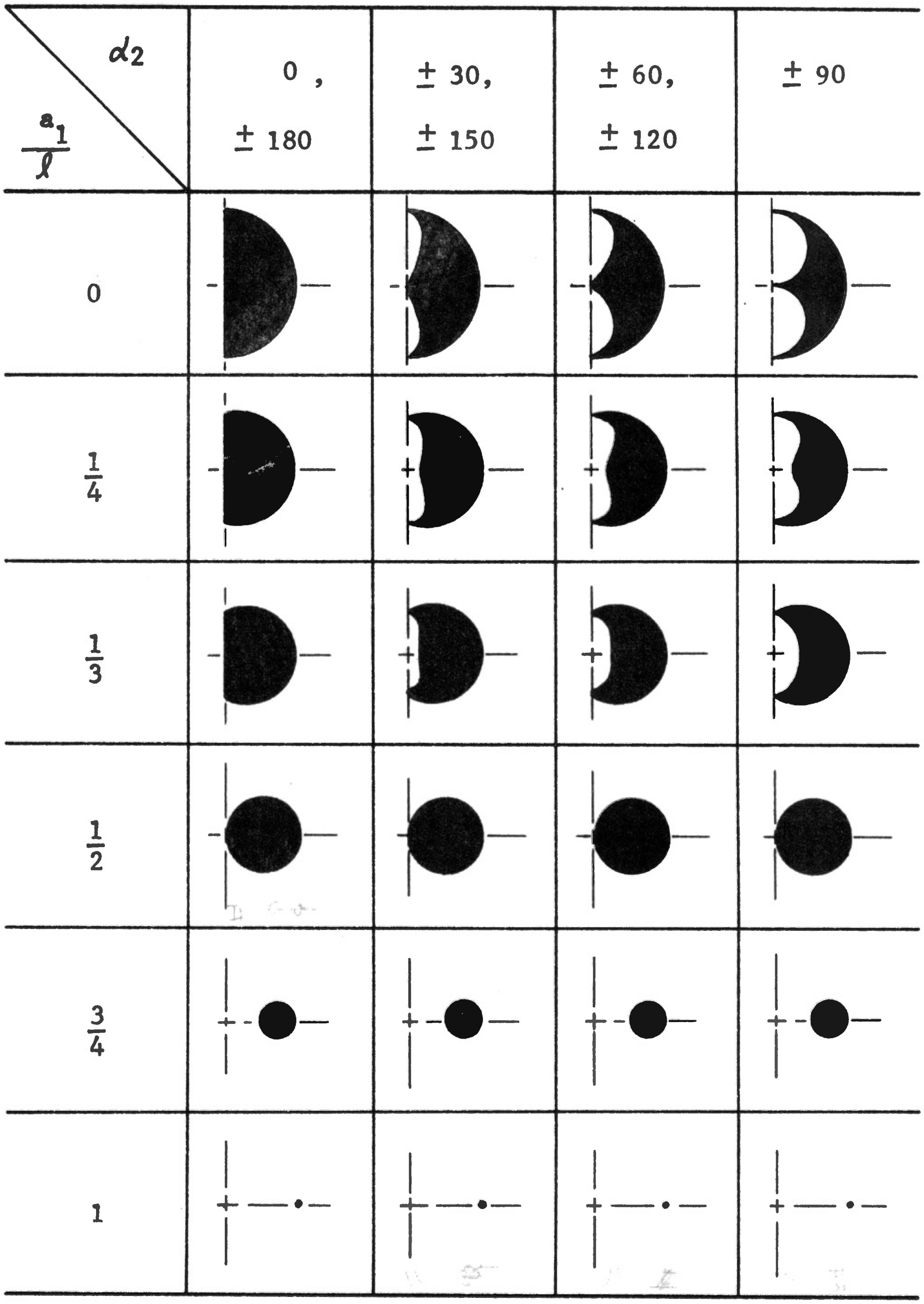
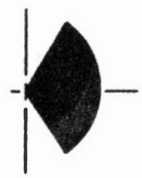
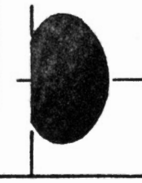
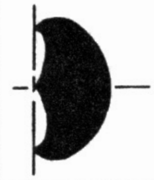
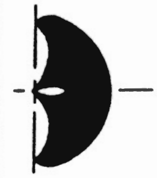
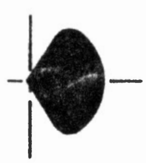
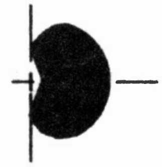
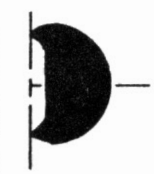
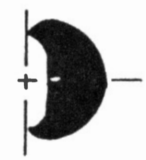
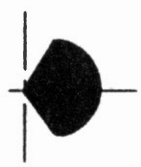
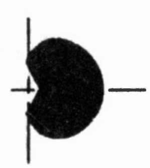
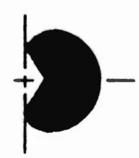
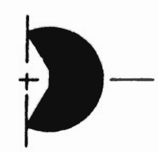
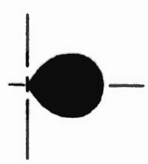
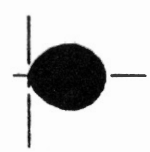
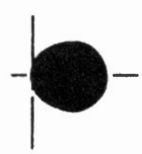
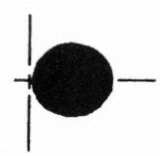
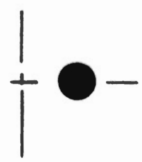
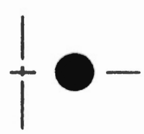
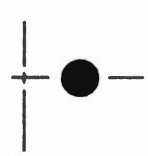
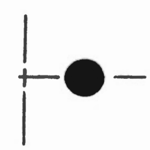
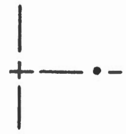
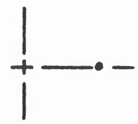
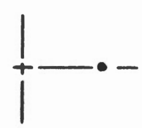
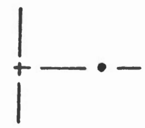


TABLE VI

EFFECT OF a_1 AND α_2 ON THE SHAPE OF CROSS-SECTION OF THE WORKSPACE OF 3R ROBOT ARMS WITH $\alpha_1 = 60$, $a_2 = a_3$, $S_1 = S_2 = S_3 = 0$

α_2 / $\frac{a_1}{\ell}$	0, ± 180	± 30, ± 150	± 60, ± 120	± 90
0				
$\frac{1}{4}$				
$\frac{1}{3}$				
$\frac{1}{2}$				
$\frac{3}{4}$				
1				

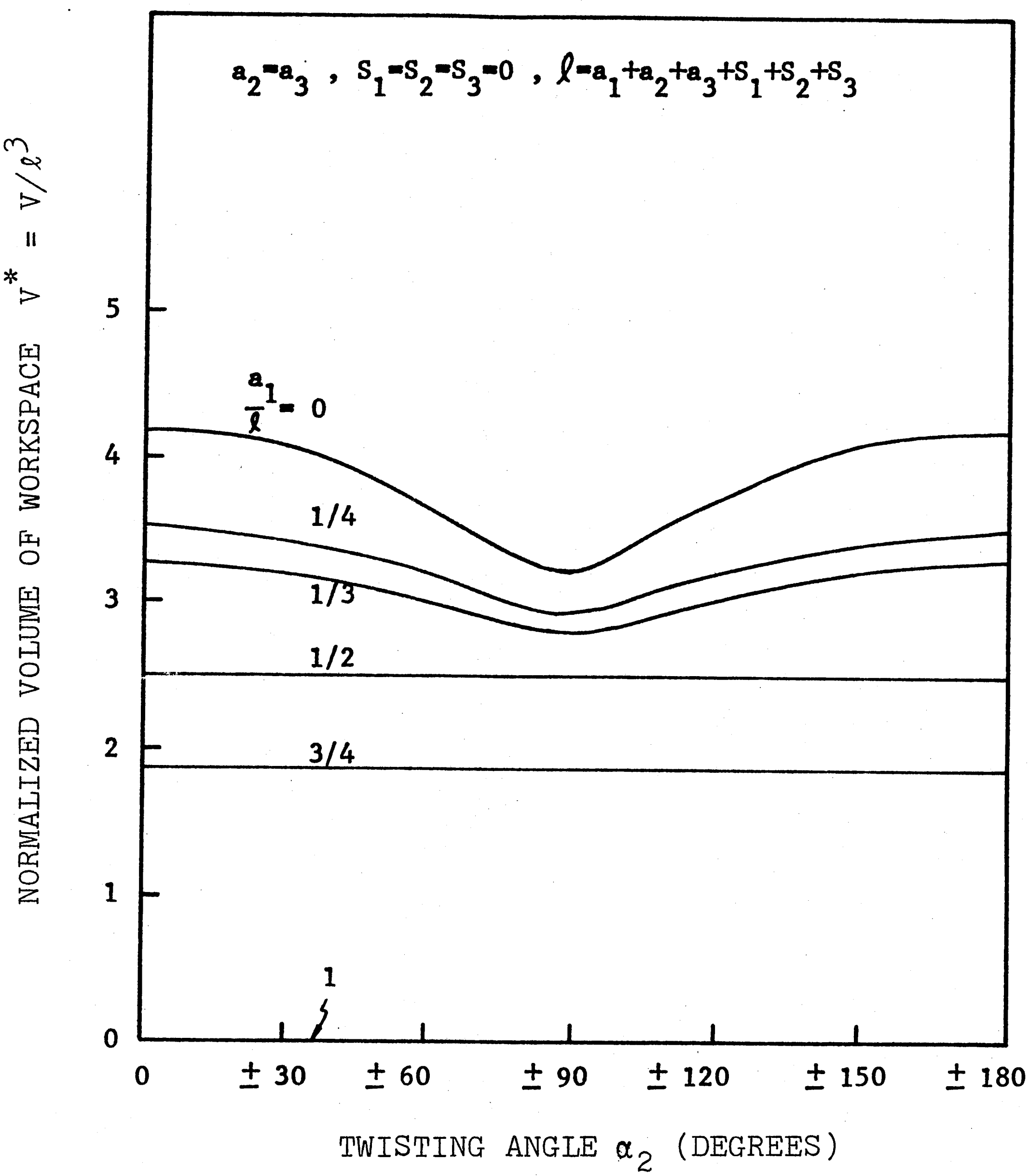


Figure 16. Effect of a_1 and α_2 on the Workspace of 3R Robot Arms (α_1 = 90)

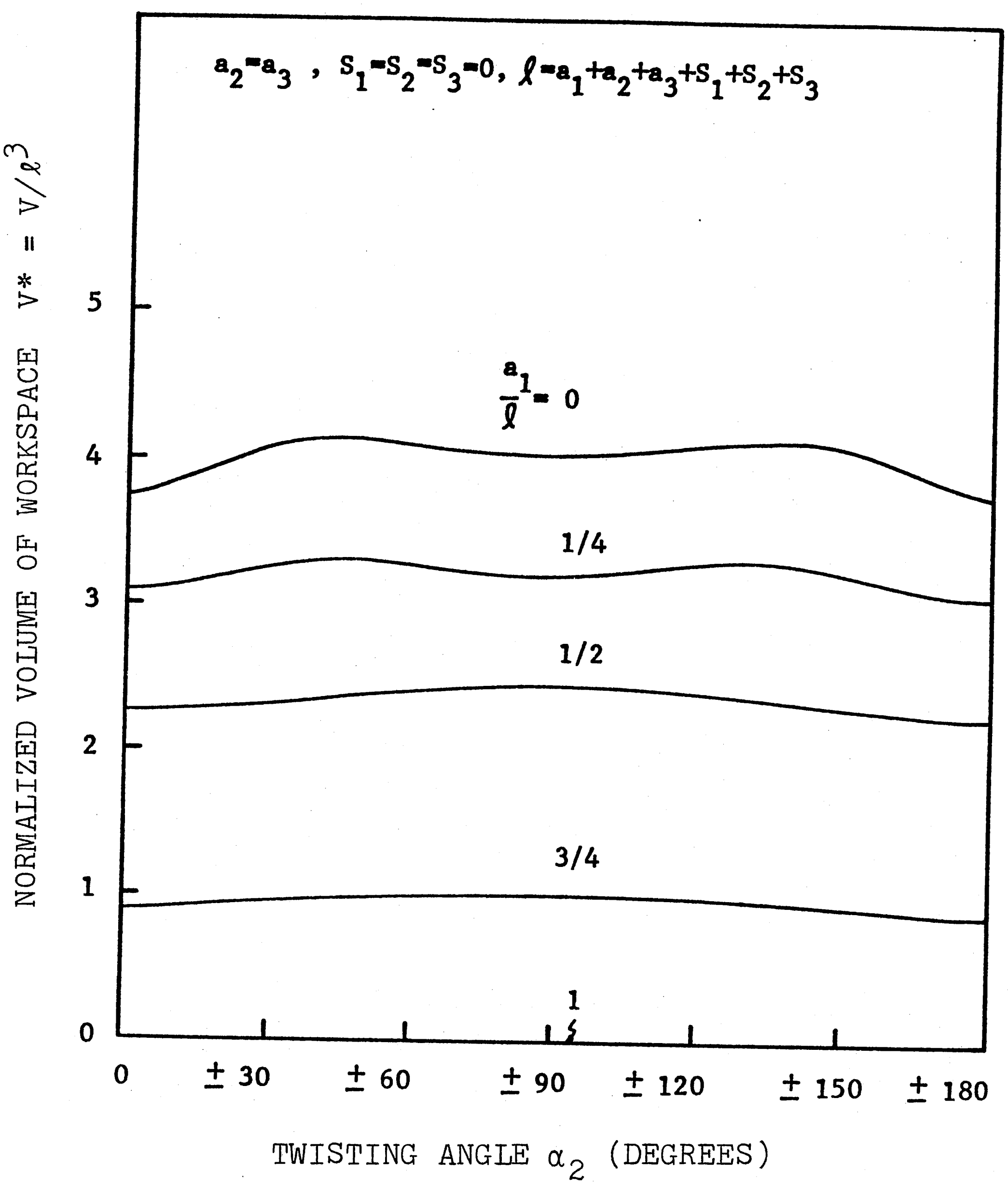


Figure 17. Effect of a_1 and α_2 on the Workspace of 3R Robot Arms (α_1 = 60)

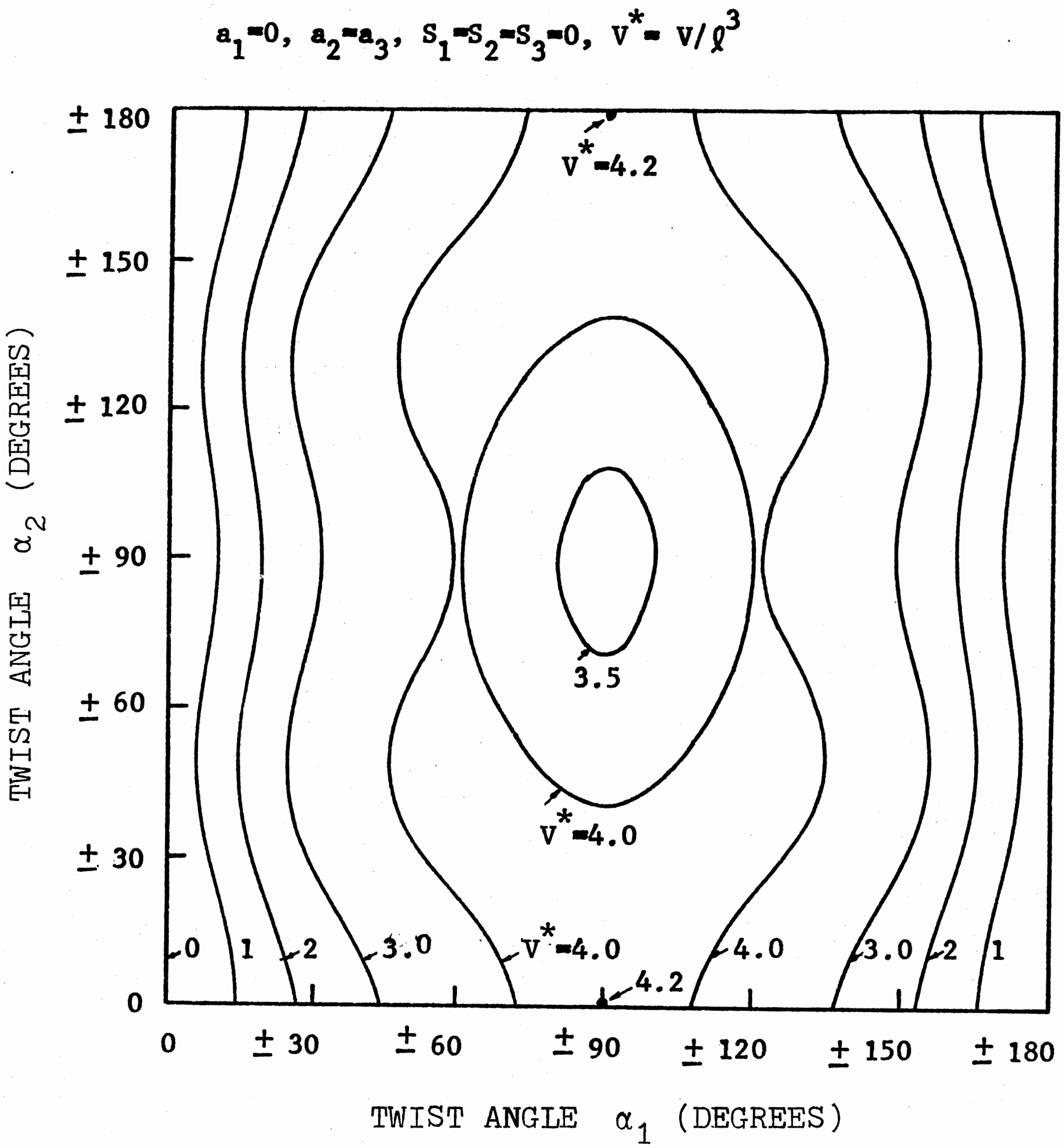


Figure 18. Effect of α_1 and α_2 on the Volume of Workspace of 3R Robot Arms (α_1 vs. α_2)

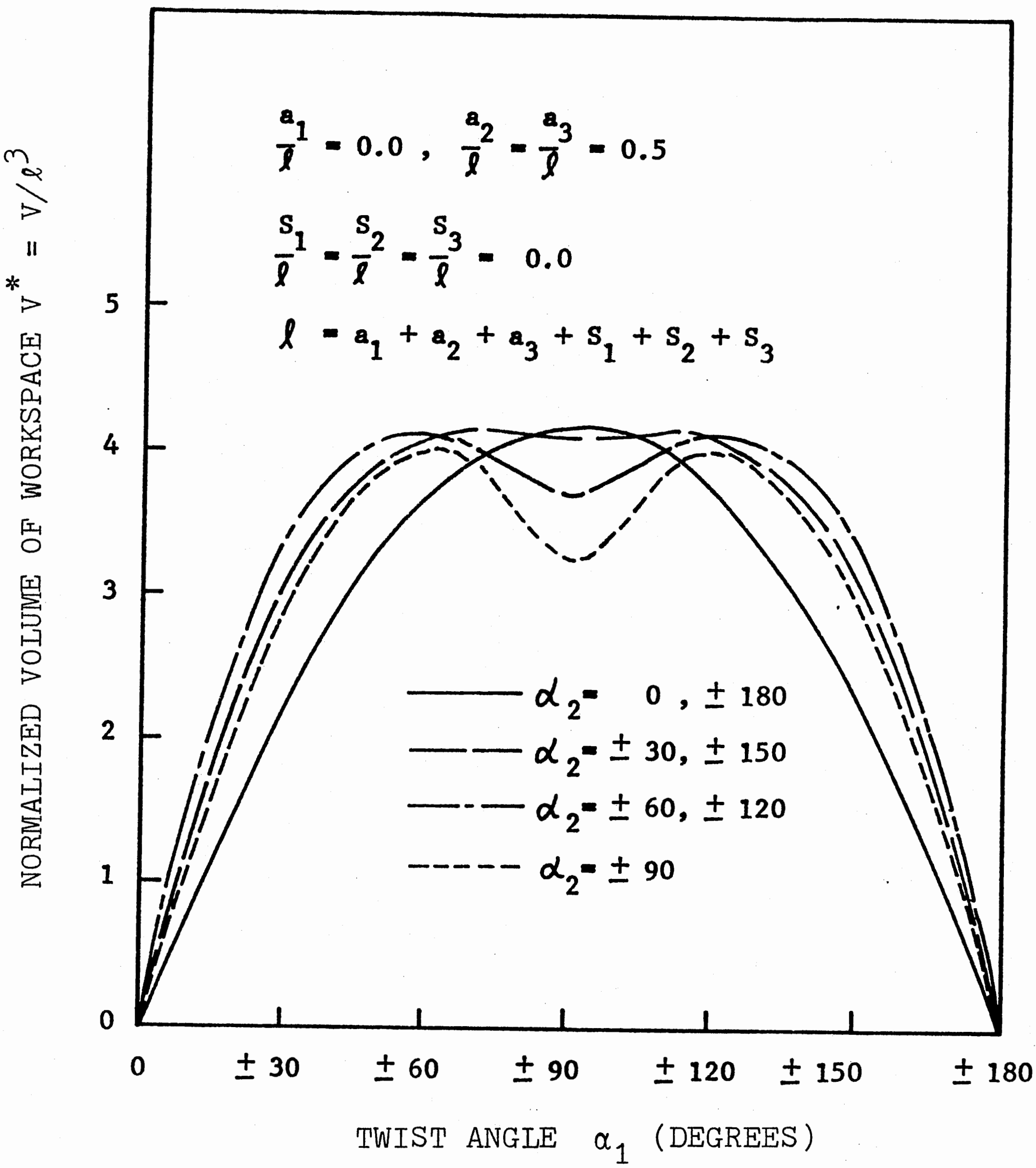


Figure 19. Effect of α_1 and α_2 on the Workspace of 3R Robot Arms (α_1 vs. V^*)

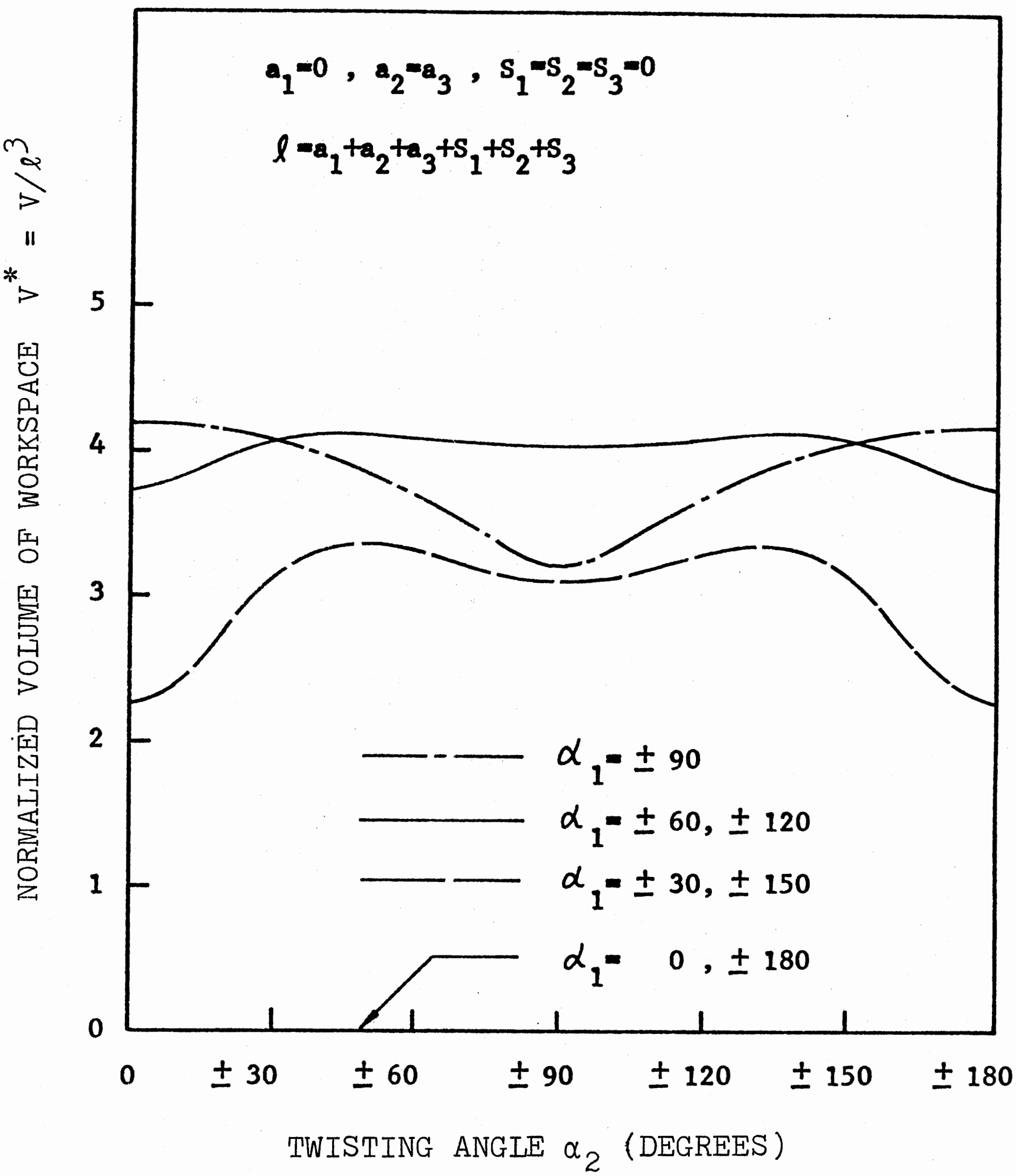


Figure 20. Effect of α_1 and α_2 on the Workspace of 3R Robot Arms (α_2 vs. V^*)

TABLE VII

EFFECT OF S_2, S_3 AND α_2 ON THE SHAPE OF CROSS-SECTION OF THE WORKSPACE OF 3R ROBOT ARMS WITH $\alpha_1 = 90$, $S_1 = a_1 = 0$, $a_2 = a_3$

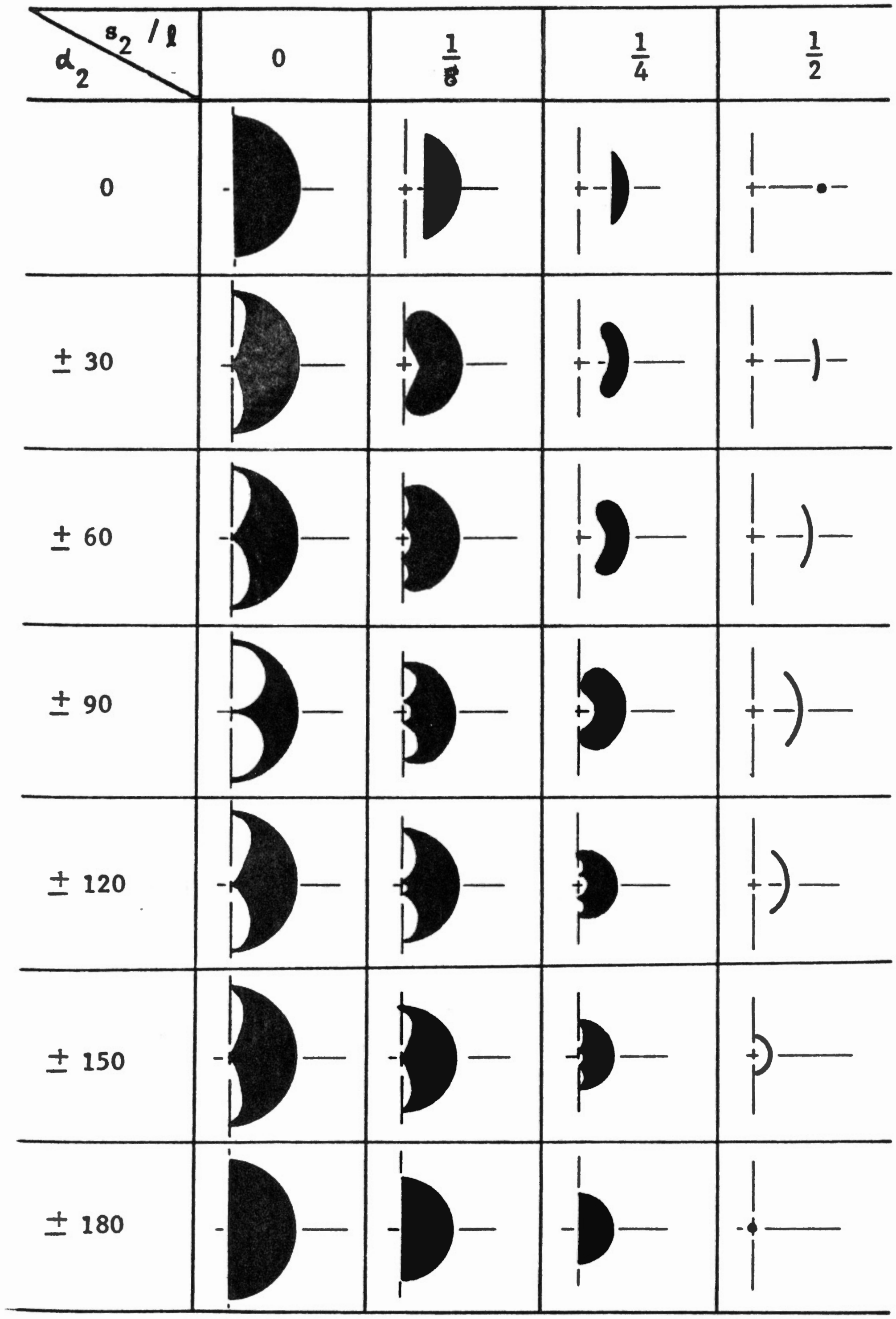
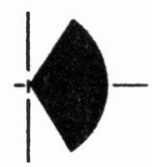
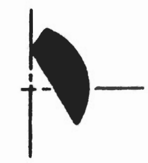
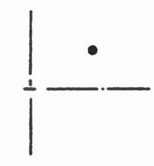
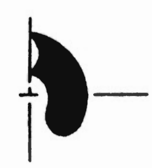
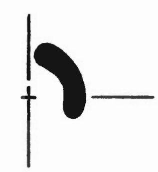
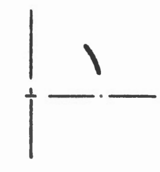
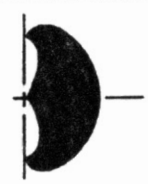
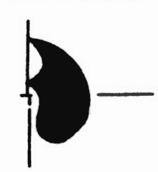
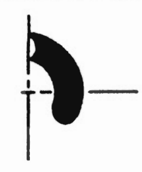
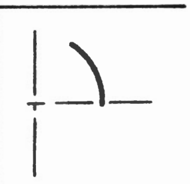
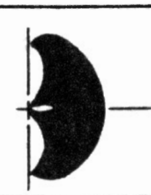
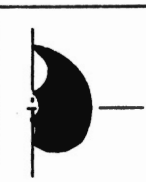
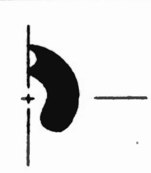
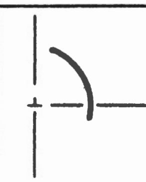
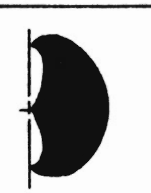
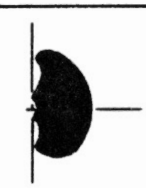
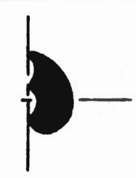
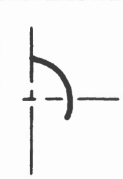
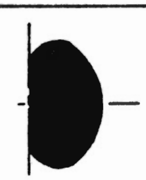
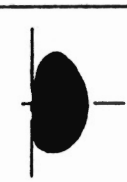
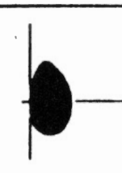
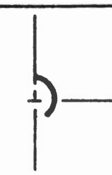
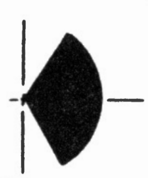
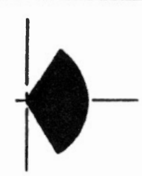
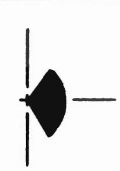
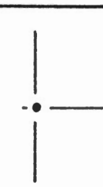


TABLE VIII

EFFECT OF S_2, S_3 AND α_2 ON THE SHAPE OF CROSS-SECTION OF THE WORKSPACE OF 3R ROBOT ARMS WITH $\alpha_1 = 60$, $S_1 = a_1 = 0$, $a_2 = a_3$

α_2 \ S_2/ℓ	0	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{2}$
0				
± 30				
± 60				
± 90				
± 120				
± 150				
± 180				

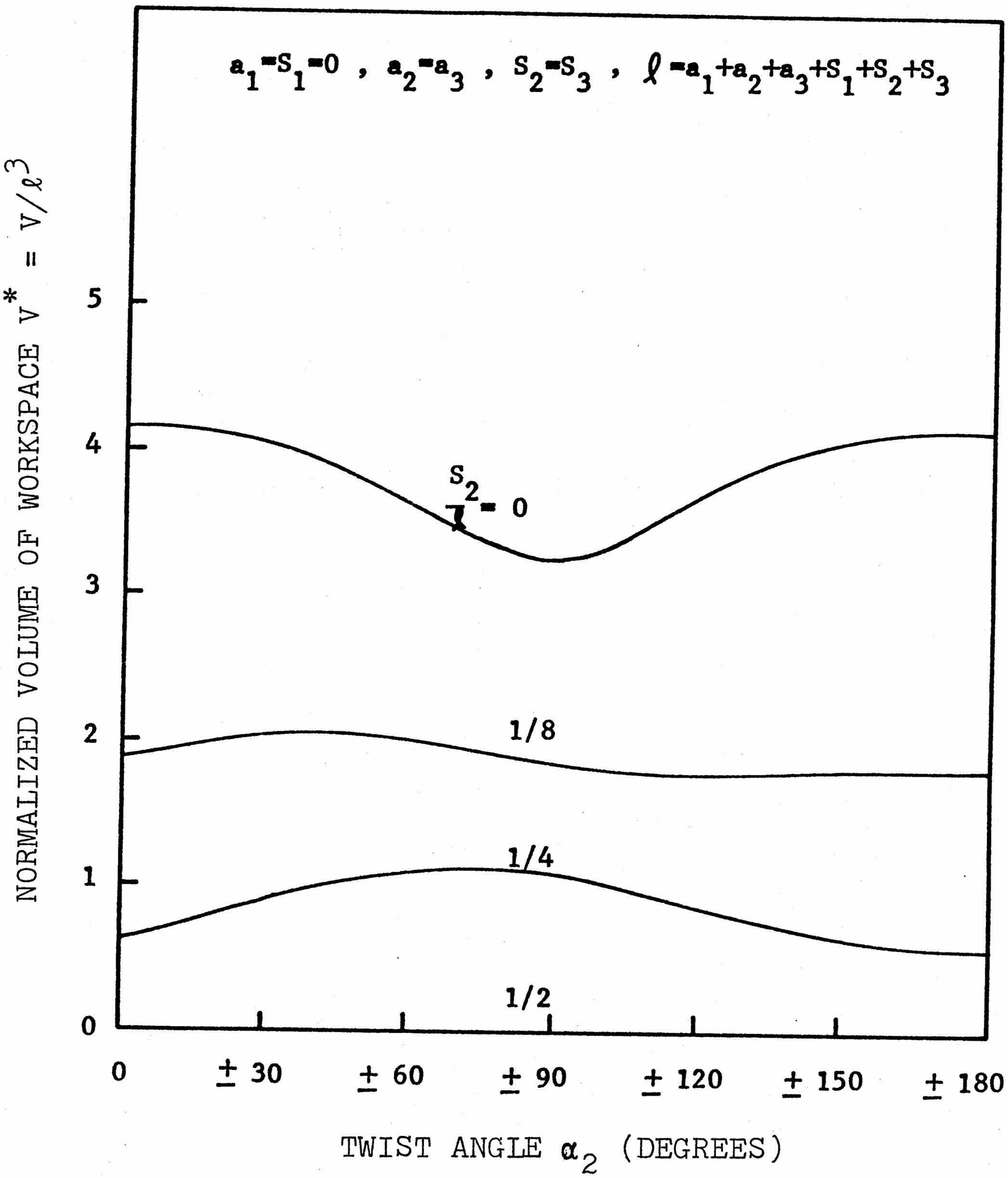


Figure 21. Effect of S_2 and α_2 on the Workspace of 3R Robot Arms ($\alpha_1 = \pm 90$)

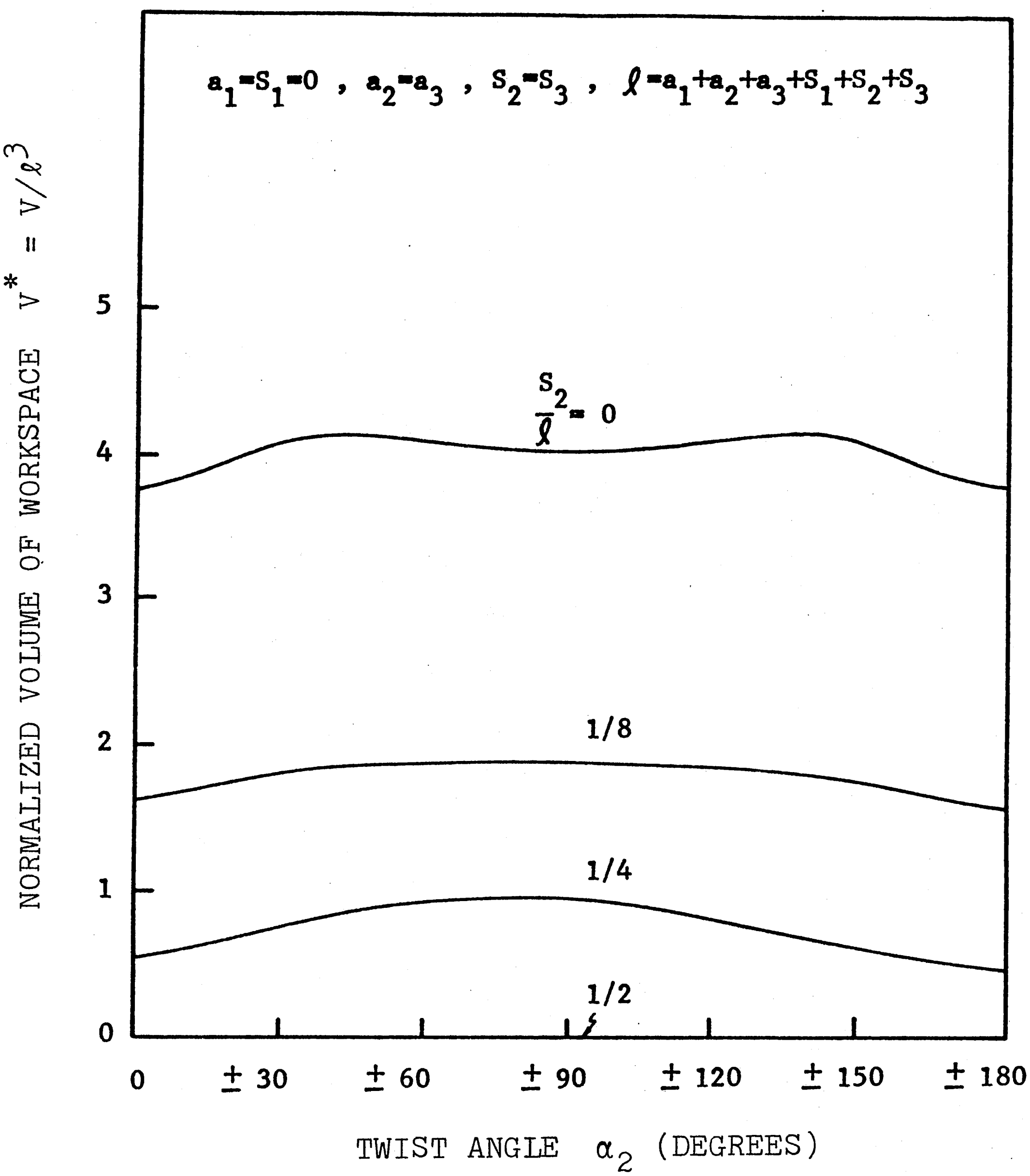


Figure 22. Effect of S_2 and α_2 on the Workspace of 3R Robot Arms (α_1 = 60)

existance of s_2 and s_3 may help the robot arm to be able to make complete rotation about the revolute joint. But s_2 and s_3 will affect the workspace in both shape and volume. Tables VII and VIII and Figures 21 and 22 demonstrate the effect of s_2 and s_3 on the workspace of 3R robot arms. Some observations from these tables and figures are:

1. Increasing s_2 and s_3 may increase the voids or hole in the workspace and reduce the normalized volume of workspace.

2. Existing of s_2 and s_3 will make the workspace unsymmetrical about $\alpha_2 = 90°$ (see Figures 21 and 22).

3.2 Effects of Link Parameters on the Dexterities of Robots

In a general case, the orientational structure consists of at least three joints such that the end link of this structure may rotate about three independent axis. Let the direction cosines of the third link along s_3 and a_3 be L_{s_3}, M_{s_3}, N_{s_3}, L_{a_3}, M_{a_3}, and N_{a_3} respectively, then

$$\begin{bmatrix} L_{s_3} \\ M_{s_3} \\ N_{s_3} \\ 0 \end{bmatrix} = A_1 A_2 \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \tag{3.1}$$

and

$$\begin{bmatrix} L_{a_3} \\ M_{a_3} \\ N_{a_3} \\ 0 \end{bmatrix} = A_1 A_2 A_3 \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad (3.2)$$

where

$$A_i = \begin{bmatrix} \mathrm{Cos}\theta_i & -\mathrm{Sin}\theta_i \mathrm{Cos}\alpha_i & \mathrm{Sin}\theta_i \mathrm{Sin}\alpha_i & a_i \mathrm{Cos}\theta_i \\ \mathrm{Sin}\theta_i & \mathrm{Cos}\theta_i \mathrm{Cos}\alpha_i & -\mathrm{Cos}\theta_i \mathrm{Sin}\alpha_i & a_i \mathrm{Sin}\theta_i \\ 0 & \mathrm{Sin}\alpha_i & \mathrm{Cos}\alpha_i & s_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$i = 1, 2, 3$

Expanding Equation (3.1) and (3.2) , one can get

$$L_{s_3} = \mathrm{Cos}\theta_1 \mathrm{Sin}\theta_2 \mathrm{Sin}\alpha_2 + \mathrm{Sin}\theta_1 \mathrm{Cos}\theta_2 \mathrm{Cos}\alpha_1 \mathrm{Sin}\alpha_2 + \mathrm{Sin}\theta_1 \mathrm{Sin}\alpha_1 \mathrm{Cos}\alpha_2 \qquad (3.3)$$

$$M_{s_3} = \mathrm{Sin}\theta_1 \mathrm{Sin}\theta_2 \mathrm{Sin}\alpha_2 - \mathrm{Cos}\theta_1 \mathrm{Cos}\theta_2 \mathrm{Cos}\alpha_1 \mathrm{Sin}\alpha_2 - \mathrm{Cos}\theta_1 \mathrm{Sin}\alpha_1 \mathrm{Cos}\alpha_2 \qquad (3.4)$$

$$N_{s_3} = -\mathrm{Cos}\theta_2 \mathrm{Sin}\alpha_1 \mathrm{Sin}\alpha_2 + \mathrm{Cos}\alpha_1 \mathrm{Cos}\alpha_2 \qquad (3.5)$$

and

$$L_{a_3} = \mathrm{Cos}\theta_1 \mathrm{Cos}\theta_2 \mathrm{Cos}\theta_3 - \mathrm{Cos}\theta_1 \mathrm{Sin}\theta_2 \mathrm{Sin}\theta_3 \mathrm{Cos}\alpha_2 - \mathrm{Sin}\theta_1 \mathrm{Sin}\theta_2 \mathrm{Cos}\theta_3 \mathrm{Cos}\alpha_1 + \mathrm{Sin}\theta_1 \mathrm{Sin}\theta_3 \mathrm{Sin}\alpha_1 \mathrm{Sin}\alpha_2 - \mathrm{Sin}\theta_1 \mathrm{Cos}\theta_2 \mathrm{Sin}\theta_3 \mathrm{Cos}\alpha_1 \mathrm{Cos}\alpha_2 \qquad (3.6)$$

$$M_{a_3} = \mathrm{Sin}\theta_1\mathrm{Cos}\theta_2\mathrm{Cos}\theta_3 - \mathrm{Sin}\theta_1\mathrm{Sin}\theta_2\mathrm{Cos}\alpha_2$$
$$+ \mathrm{Cos}\theta_1\mathrm{Sin}\theta_2\mathrm{Cos}\theta_3\mathrm{Cos}\alpha_1 - \mathrm{Cos}\theta_1\mathrm{Sin}\theta_3\mathrm{Sin}\alpha_1\mathrm{Sin}\alpha_2$$
$$+ \mathrm{Cos}\theta_1\mathrm{Cos}\theta_2\mathrm{Sin}\theta_3\mathrm{Cos}\alpha_1\mathrm{Cos}\alpha_2 \qquad (3.7)$$

$$N_{a_3} = \mathrm{Sin}\theta_2\mathrm{Cos}\theta_3\mathrm{Sin}\alpha_1 + \mathrm{Cos}\theta_2\mathrm{Sin}\theta_3\mathrm{Sin}\alpha_1\mathrm{Cos}\alpha_2$$
$$+ \mathrm{Sin}\theta_3\mathrm{Cos}\alpha_1\mathrm{Sin}\alpha_2 \qquad (3.8)$$

Obviously, there are θ's and α's in the above equations. These equations are independent of s_1, s_2, s_3, a_1, a_2 and a_3. In other words, only θ_1, θ_2, θ_3, α_1 and α_2 contribute to the dexterity, the capability of changing directions, of the last link of the robot. And the prismatic pair will contribute nothing to the dexterity of the robot. Therefore, the use prismatic pairs in the orientational structure should be avoided.

3.2.1 The Effect of α_1 and α_2 on the Dexterities of 3R Robots

In order to study the effects of link parameters α_1 and α_2 on the dexterity of a general 3R robot, it is convenient to assume that all revolute pairs can make a complete rotation. Let $\overline{U}_{s_3}$ and $\overline{U}_{a_3}$ represent the unit vectors attached to link s_3 and a_3 respectively. In the ideal case it is desired that both $\overline{U}_{s_3}$ and $\overline{U}_{a_3}$ can be in any direction, i.e., they can make 4π solid angle in space. But in some cases, if α_1 and α_2 are not specified properly, $\overline{U}_{s_3}$ and $\overline{U}_{a_3}$ may sweep only part of the entire solid angle.

Let $D_{s_3} = \phi_{s_3}/4\pi$ (3.9)

$D_{a_3} = \phi_{a_3}/4\pi$ (3.10)

$D_{hand} = D_{s_3} \cdot D_{a_3}$

where

D_{s_3} : Dexterity of link s_3 as a line.

D_{a_3} : Dexterity of link a_3 as a line.

D_{hand} : Dexterity of robot hand as a rigid-body.

ϕ_{s_3} : The solid angle swept by link s_3.

ϕ_{a_3} : The solid angle swept by link a_3.

Using Equation (3.3) through Equation (3.8), one may plot the cross-sections of the region swept by $\overline{U}_{s_3}$ and $\overline{U}_{a_3}$ (Table IX and Table X), and calcuate the dexterities of link s_3 and link a_3 (Figure 23, and Figure 24).

From Table IX, Table X and Figure 23 through Figure 26, the following observations have been made,

1. For each given value of twisting angle α_2,

the dexterity of link s_3 will reach its maximum only when the twisting angle α_1 is equal to $\pm 90°$ (see Figure 23).

2. Only when both α_1 and α_2 are equal to $\pm 90°$, the dexterity of link s_3 will be equal to one, the maximum value, and link s_3 can point to any direction (see Table IX and Figure 23).

3. When the twisting angle α_1 is equal to $\pm 90°$, the dexterity of link a_3 will always be equal to one, the maximum value, and the value of α_2 has no effect to the dexterity.

4. Let α_1* be α_1 or its supplementary angle whichever is an acute angle, and α_2* be α_2 or its supplementary angle whichever is an acute angle. If the sum of $|\alpha_1^*|$ and $|\alpha_2^*|$ is equal to or greater than 90°, then the dexterity of link a_3 will be equal to one.

5. The dexterity of link a_3 is always better than or equal to that of link s_3 for each give value of α_1 as shown in Figure 25.

6. The dexterity of the robot hand, considering it as a rigid body, can reach its maximum value only when both α_1 and α_2 are equal to $\pm 90°$ (see Figure 26).

TABLE IX

THE CROSS-SECTION OF THE REGION SWEPT BY UNIT VECTOR $\bar{U}_{a_3}$

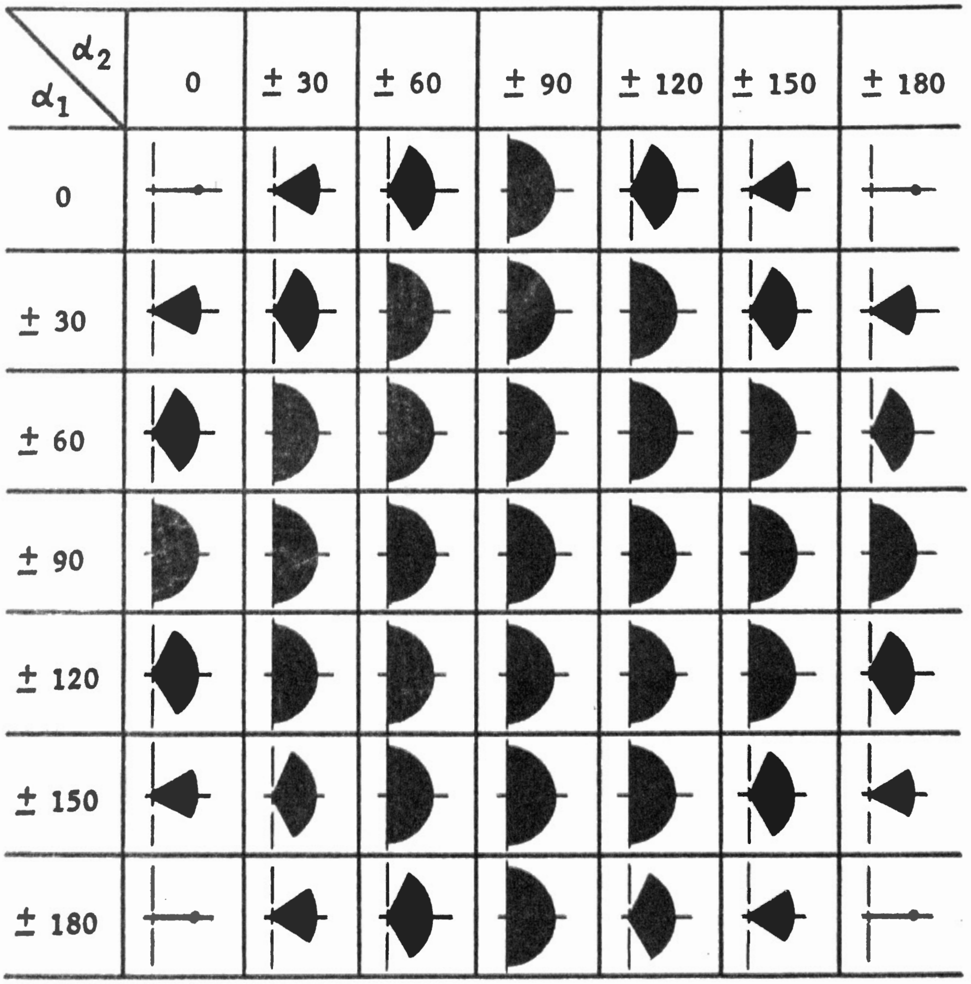
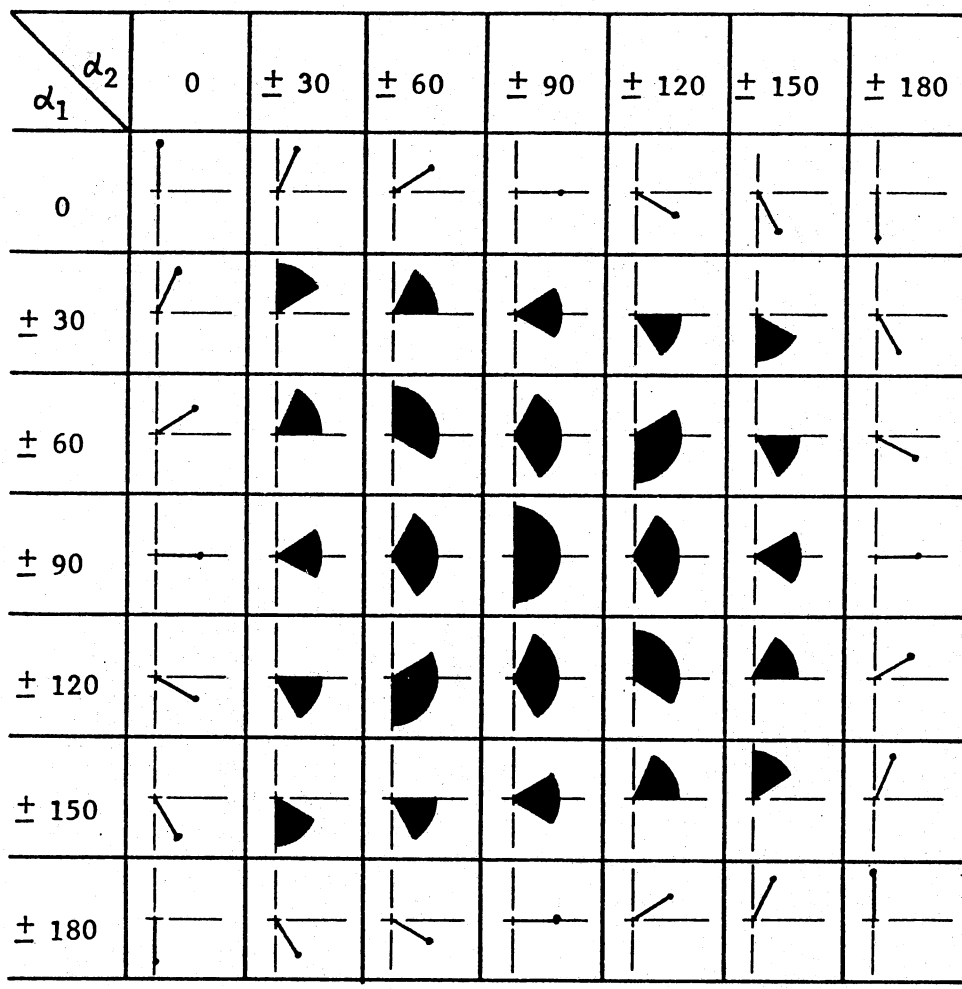


TABLE X

THE CROSS-SECTION OF THE REGION SWEPT
BY UNIT VECTOR $\bar{U}_{S_3}$



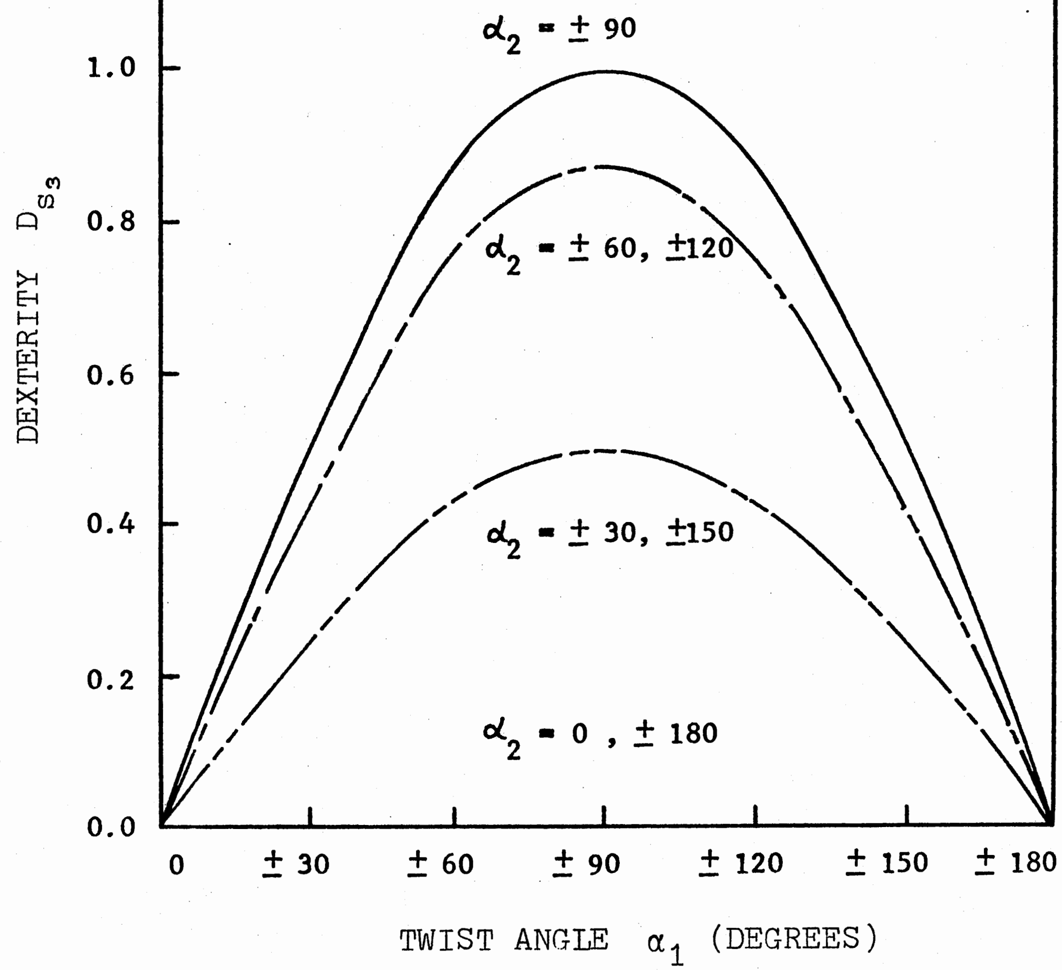


Figure 23. The Effects of α_1 and α_2 on the Dexterity of Link S_3

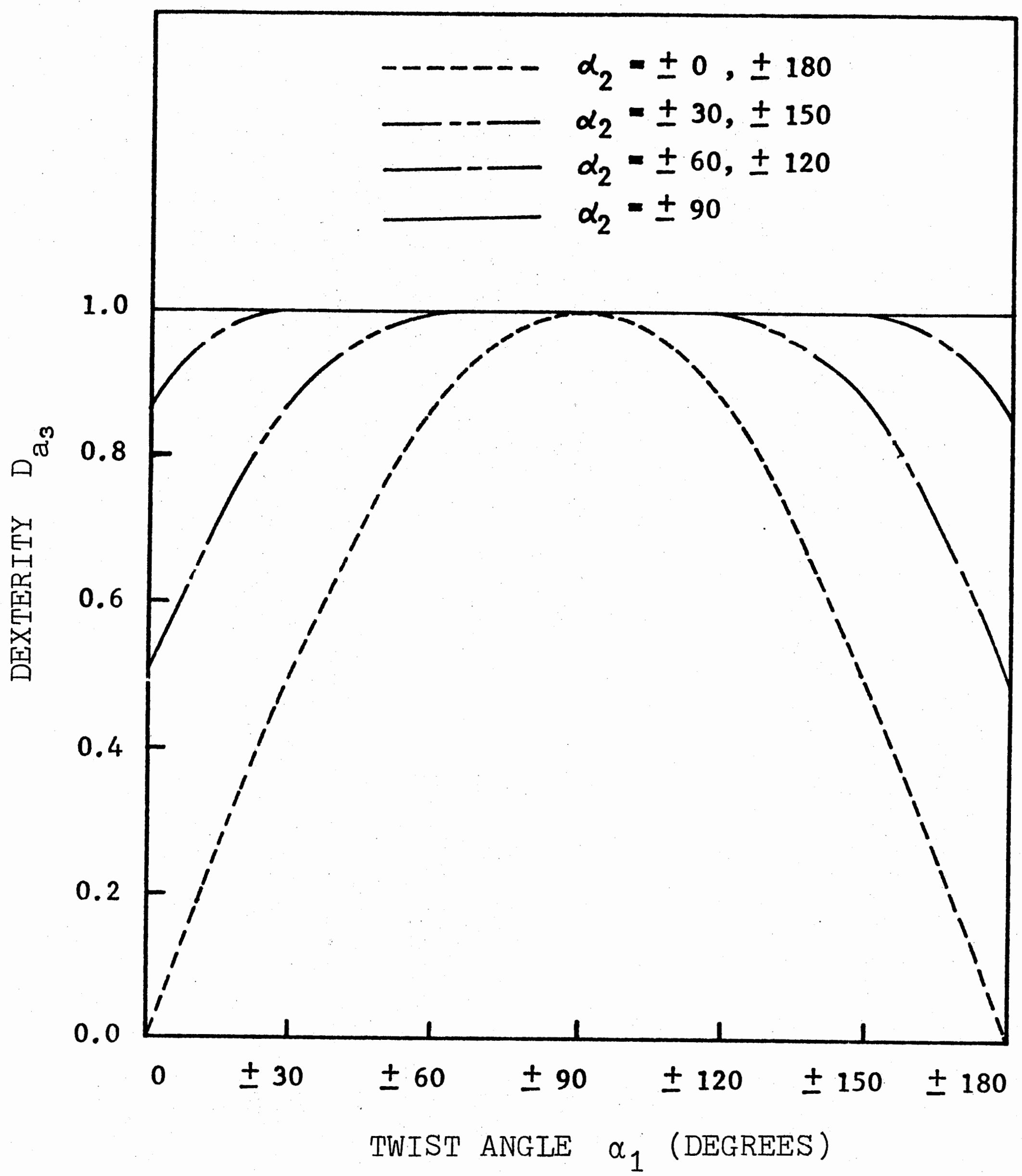


Figure 24. The Effects of α_1 and α_2 on the Dexterity of Link a_3

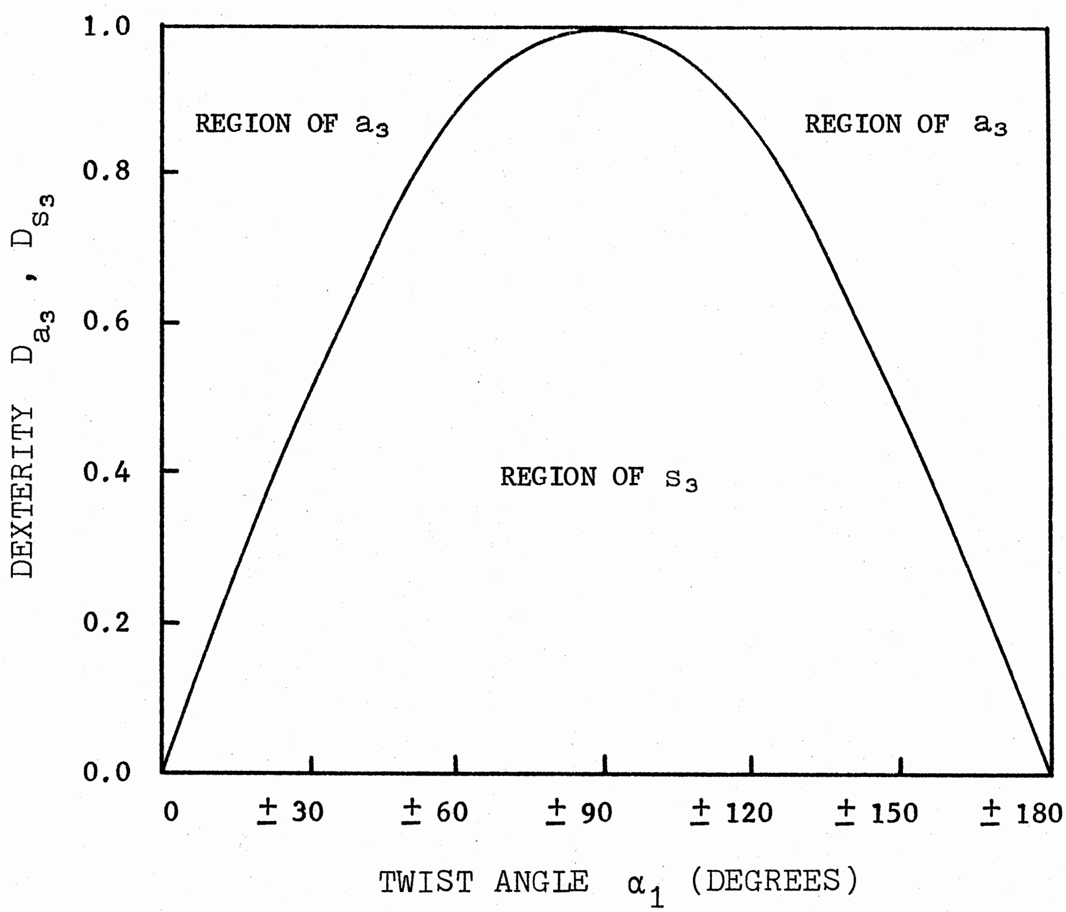


Figure 25. The Possible Region of the Dexterity of Link a_3 and Link s_3

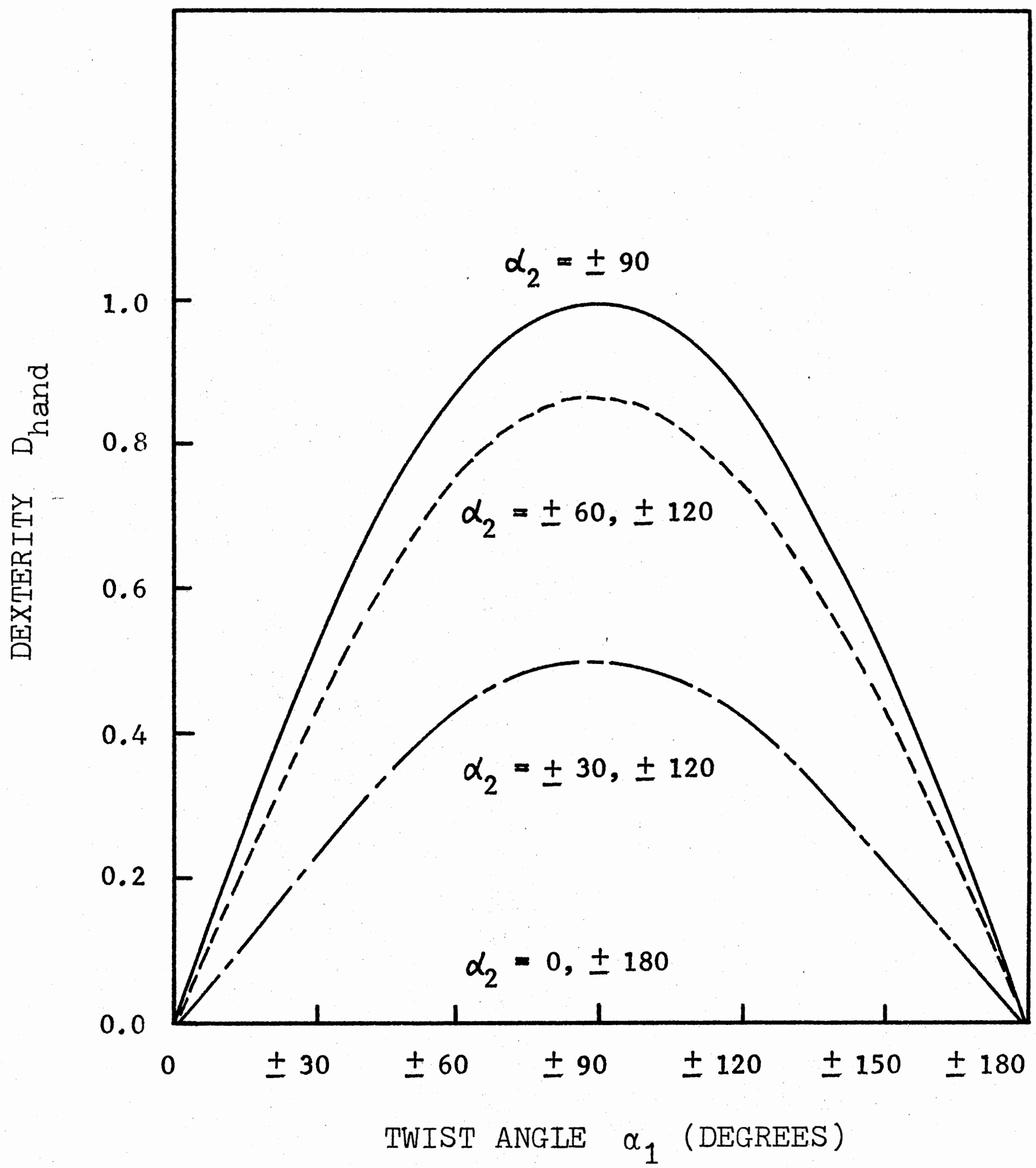


Figure 26. The Effect of α_1 and α_2 on the Dexterity of the Robot Hand

CHAPTER IV

SYNTHESIS OF ROBOTS/MANIPULATORS FOR A PRESCRIBED WORKING SPACE

A n-R robot/manipulator has 3n time-independent link parameters and n time-dependent joint variables. To synthesize a robot for a prescribed working space may become an optimization problem. One may solve this problem by optimizing all of the parameters at the same time. But one may solve the problem in an alternative way, study the effects of link parameters on the sub-structures of robots first, then use the results to find out the optimal values of some link parameters or some relationships among link parameters. In this way one may reduce the number of parameters in the final optimization problem and may simplify the original problem quite a lot. The effects of link parameters on the workspace of the regional structures and on the dexterity of orientational structures have been studied in Chapter III. And some optimal values have been determined under the criterion of maximizing the working space or the dexterity. So it is preferred to use the results from Chapter III, and use the latter method mentioned above to synthesize the robots in this chapter.

If there is no constraint on the values of link parameters, then one will be able to select the optimal value for each link parameter. According to Chapter III, a regional structure with the link parameters: $a_1 = 0$, $a_2 = a_3$, $s_1 = s_2 = s_3 = 0$, $\alpha_1 = \pm 90°$, and $\alpha_2 = 90°$, can get the maximum working space. And an orientational structure with the link parameters: $\alpha_1 = \pm 90°$ and $\alpha_2 = 0°$ or $180°$ is capable of having maximum dexterity. These data will be used in the following sections to synthesize the industrial robots/manipulators. In the industrial area, one may need manipulators as simple as 2R mechanisms to serve jobs requiring to excute simple function of a high speed. On the other end one needs versatile robots having six degrees of freedom to do complicated jobs on the production lines. Syntheses of 2R, 3R, 4R, 5R, and 6R robots have been studied in the following sections.

4.1 Synthesis of 2R Planar Robots

From Equations (2.3) and (2.4) one can obtain

$$\theta_1 = \mathrm{Cos}^{-1} \frac{z}{\sqrt{x^2 + z^2}} - \mathrm{Cos}^{-1} \frac{(x^2 + z^2) + (\ell_1^2 - \ell_2^2)}{2\,\ell_1 \sqrt{x^2 + z^2}} \tag{4.1}$$

$$\theta_2 = \mathrm{Cos}^{-1} \frac{(x^2 + z^2) - (\ell_1^2 + \ell_2^2)}{2\ell_1 \ell_2} \tag{4.2}$$

But Equations (4.1) and (4.2) are obtained under the assumption that the base point of a robot is located at the origin (0,0) of the reference coordinate X-Z. If the location of the base point is not at (0,0) but somewhere at (x_o, z_o), then Equations (4.1) and (4.2) must be modified as,

$$\theta_{1i} = \mathrm{Cos}^{-1} \frac{z_i - z_o}{\sqrt{(x_i - x_o)^2 + (z_i - z_o)^2}} - \mathrm{Cos}^{-1} \frac{(x_i - x_o)^2 + (z_i - z_o)^2 + \ell_1^{\,2} - \ell_2^{\,2}}{2\ell_1 \sqrt{(x_i - x_o)^2 + (z_i - z_o)^2}} \qquad (4.3)$$

$$\theta_{2i} = \mathrm{Cos}^{-1} \frac{(x_i - x_o)^2 + (z_i - z_o)^2 - (\ell_1^{\,2} + \ell_2^{\,2})^2}{2\ell_1\ell_2} \qquad (4.4)$$

In Synthesis of 2R planar robot which can reach a set of specified working positions (x_i, z_i), $i = 1, 2, 3, \ldots n$ (defined in the first quadrant for convenience), the following procedure is proposed.

1. Find x_{max}, x_{min}, z_{max}, and z_{min} from the given set of data (x_i, z_i).

2. If the location of the base point of robot is not specified, one may assume x_o and z_o to have some arbitrary values.

3. Find the maximum value of L_i, $i = 1, 2, 3, \ldots n$. Where $L_i = \sqrt{(x_i - x_o)^2 + (z_i - z_o)^2}$. Let the length of

robot link $\ell_1 = \ell_2 = \frac{1}{2}(L_i)_{max}$.

4. Compute θ_{1i} and θ_{2i} from Equations (4.3) and (4.4) corresponding to each working point (x_i, z_i) and find the maximum and minimum values of them, i.e. $\theta_{1,max}$, $\theta_{1,min}$, $\theta_{2,max}$ and $\theta_{2,min}$.

5. Compute the area of the accessible region of the robot with its link lengths computed in Step 3 and 4.

$$A' = F'(\theta_{1,max} - \theta_{1,min})(\ell_1 + \ell_2)^2 \tag{4.5}$$

where

$$F' = \frac{\frac{\ell_2}{\ell_1}(\cos\theta_{2,min} - \cos\theta_{2,max})}{[1 + \frac{\ell_2}{\ell_1}]^2}$$

6. Use suitable optimization method and repeat Step 2 to 5 to find optimum values of x_o and z_o such that A' is minimum. The optimum values of x_o, z_o, ℓ_1, ℓ_2, $\theta_{1,min}$, $\theta_{1,max}$, $\theta_{2,min}$, and $\theta_{2,max}$ will provide the necessary data for the synthesized robot.

The above procedure can be computerized. The example presented below demonstrates the technique of synthesis of a two link manipulator.

EXAMPLE: Design a two-link robot which has an accessible region as close as possible to the following working regions:

p_1(100.00, 15.00), p_2(110.00, 15.00), p_3(109.39, 10.61)

p_4(105.00, 15.00), p_5(105.00, 15.00), p_6(109.39, 55.61)

$p_7(130.00,\ 60.00)$, $p_8(140.00,\ 60.00)$

Using the procedure described in this section, the following solution has been obtained.

Location of robot base: (38.23, 54.64)

Link lingth: $L_1 = 50.9572$, $L_2 = 50.9572$

Extreme position of θ_1 and θ_2:

$(\theta_1)_{max} = 86.99$; $(\theta_1)_{min} = 43.51$

$(\theta_2)_{max} = 91.41$; $(\theta_2)_{min} = 0.00$

Figure 27 shows the working position as located by a two link robot and its accessible region.

4.2 Synthesis of 3R Robots

The 3R robots (Figure 28) with $a_1 = 0$, $a_2 = a_3$, $s_2 = s_3 = 0$, $\alpha_1 = \pm 90°$, and $\alpha_2 = 0°$ are capable of accessing the maximax working space as studied in Chapter III. And this kind of robots has been widely used as the regional structure of the industrial robots.

Let $(x_e,\ y_e,\ z_e)_{j,i}$ represents the coordinates of the i^{th} given position of the end point of the robot arm in the reference frame $\{ X_j,\ Y_j,\ Z_j \}$ which is attached to joint j. In Figure 28 for a given location of robot end $(x_e,\ y_e,\ z_e)_{1,i}$, the joint displacements $(\theta_1')_i$, $(\theta_2')_i$ and $(\theta_3')_i$ can be calculated by

$$(\theta_1')_i = \tan^{-1} (y_e/x_e)_{1,i} \qquad (4.6)$$

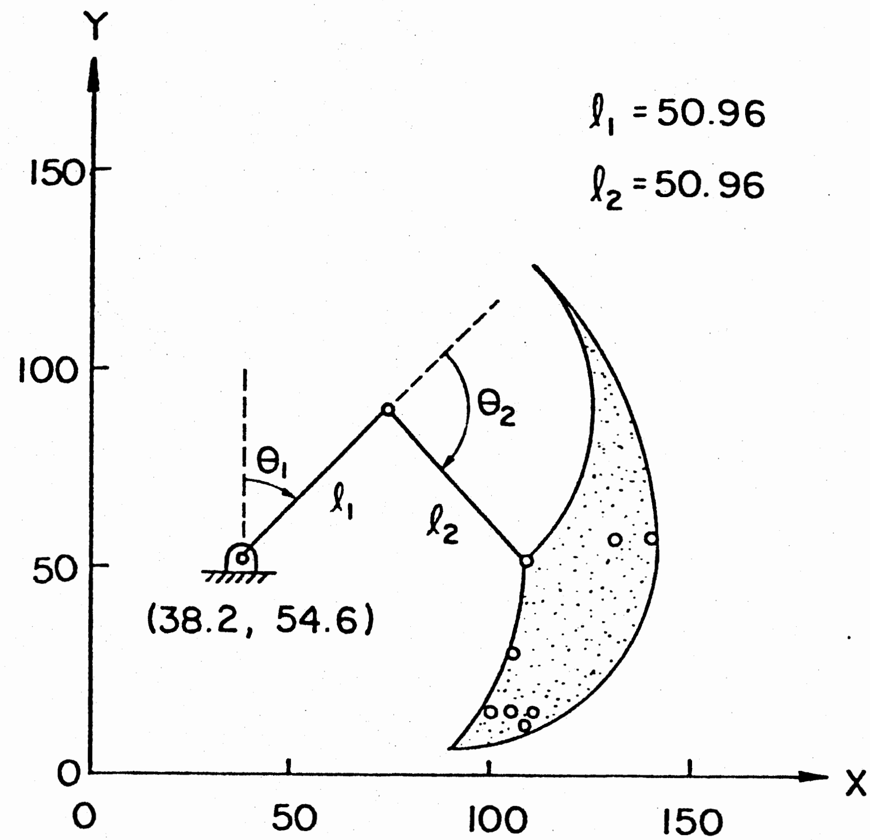


Figure 27. Working Positions and Synthesized 2R Robot

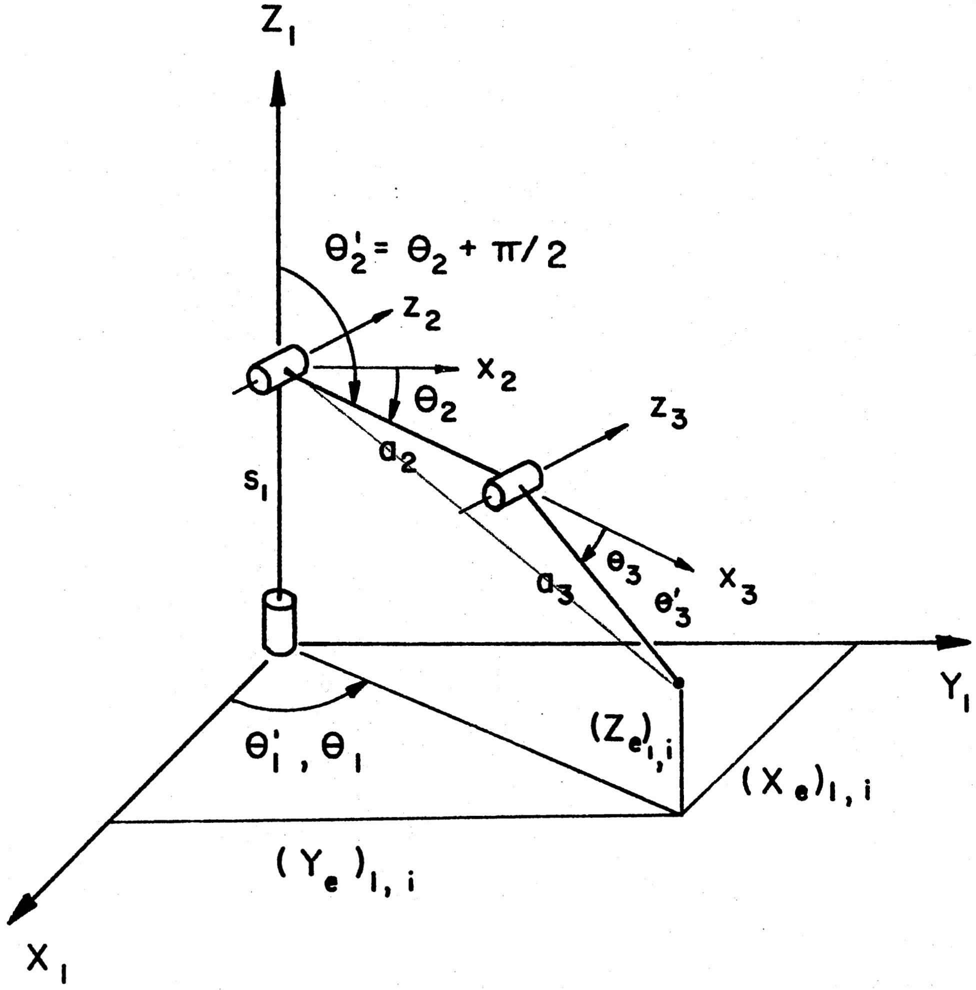


Figure 28. The Popular 3R Industrial Robot

$$(\theta_2')_i = \cos^{-1} \frac{(z_e)_{1,i}-s_1}{\ell_i} - \cos^{-1} \frac{\ell_i^{\,2}+(a_2^{\,2}-a_3^{\,2})}{2a_2\ \ell_i}$$

(4.7)

$$(\theta_3')_i = \cos^{-1} \frac{\ell_i^{\,2} - (a_2^{\,2} + a_3^{\,2})}{2a_2\ a_3} \tag{4.8}$$

where

$$\ell_i = \{(x_e)_{1,i}^{2} + (y_e)_{1,i}^{2} + [(z_e)_{1,i} - s_1]^2\}^{\frac{1}{2}}$$

Synthesis of this kind of 3R robots for a prescribed working space $(x_e, y_e, z_e)_i$, i=1,2,.....n, can be carried out by the following steps.

1. Let x_b, y_b, and z_b (assume $s_1 = 0$) represent the location of the robot base joint (joint 1). Assign them some values.

2. The coordinates of robot-end in the reference frame $\{X_1, Y_1, Z_1\}$, which is attached to joint 1, become

$$(x_e)_{1,i} = (x_e)_i - x_b$$

$$(y_e)1,i = (x_e)_i - y_b \tag{4.9}$$

$$(z_e)_{1,i} = (z_e)_i - z_b$$

3. Polar project each $(x_e, y_e, z_e)_{1,i}$ on the $X_1 - Z_1$ plane, and get $(x_e^*, 0, z_e)_{1,i}$, where $x_e^* = (x_e^{\,2} + y_e^{\,2})^{\frac{1}{2}}$. Then let $a_2 = a_3 = \frac{1}{2} (\ell_i^*)_{max}$, where $\ell_i^*=[(x_e^*)_{1,i}^2+(z_e)_{1,i}^2]^{\frac{1}{2}}$.

4. The joint displacements $(\theta_1')_i$, $(\theta_2')_i$ and $(\theta_3')_i$

corresponding to the robot end location $(x_e, y_e, z_e)_{1,i}$ can be calculated by using Equation (4.6), (4.7) and (4.8). Find $(\theta_1')_{min}$, $(\theta_1')_{max}$, $(\theta_2')_{min}$, $(\theta_2')_{max}$, $(\theta_3')_{min}$ and $(\theta_3')_{max}$.

5. Calculate the cross section area A on the X_1Z_1 plane and the solid of revolution volume V (generated by A) of the workspace.

$$A = [\text{Cos}(\theta_3')_{min} - \text{Cos}\ (\theta_3')_{max}]\Delta\theta_2'\ (a_2)^2$$

$$\begin{aligned} X_{cg} = \Delta\theta_2'\ (a_2)^3\ \{&\text{Sin}\ \theta_2*[\text{Cos}\ (\theta_3')_{min} - \text{Cos}\ (\theta_3')_{max}] \\ &-\text{Sin}\ \theta_2*[\text{Sin}^2(\theta_3')_{min} - \text{Sin}^2(\theta_3')_{max}]/2 \\ &-\text{Cos}\ \theta_2*[2(\theta_3')_{min} - 2(\theta_3')_{max} - \text{Sin}\ 2\ (\theta_3')_{min} \\ &+\text{Sin}\ 2(\theta_3')_{max}]\}/A \end{aligned}$$

$$V = (\Delta\theta_1')X_{cg}\ A \tag{4.10}$$

where

$$\Delta\theta_1' = (\theta_1')_{max} - (\theta_1')_{min} \quad \text{(rad.)}$$
$$\Delta\theta_2' = (\theta_2')_{max} - (\theta_2')_{min} \quad \text{(rad.)}$$
$$\theta_2^* = [(\theta_2')_{max} + (\theta_2')_{min}]/2$$

6. Use a suitable optimization method and repeat step 1 to step 5 to find the optimal values for x_b, y_b and z_b such that V is minimum. And these optimal values of x_b, y_b, z_b, and corresponding values of a_2, a_3, $(\theta_1')_{min}$, $(\theta_1')_{max}$, $(\theta_2')_{min}$, $(\theta_2')_{max}$, $(\theta_3')_{min}$ and $(\theta_3')_{max}$ provide the necessary data for the synthesized robot.

The procedure of synthesizing 3R robots developed above will be rsed repeatly in the following sections to synthesize 4R, 5R and 6R industrial robots.

4.3 Synthesis of 4R Robots

Adding one more link to the 3R regional structure of robots described in Section 4.2, it becomes a 4R industrial robot as shown in Figure 29. This kind of 4R robots is good for a job which requires to locate the robot hand (the tool) in a three dimensional space and to keep the robot hand in vertical direction (a direction paralled to the axis of the first joint of the robot) at the same time. To drill vertical holes in a machine part is one of the examples.

If some locations along each working process (say initial position, final position and some positions in between) are specified. The corresponding location of the joint 4 $(x_4, y_4, z_4)_i$ for each given location of the tip of robot hand $(x_t, y_t, z_t)_i$ can be calculated by,

$$\begin{aligned} (x_4)_i &= (x_t)_i \\ (y_4)_i &= (y_t)_i \qquad (4.11) \\ (z_4)_i &= (z_t)_i + a_4 \end{aligned}$$

$$i = 1,2,.....n$$

where a_4 is determined by the machine tool to be used.

The coordinates $(x_4,\ y_4,\ z_4)_i$ described here are the same as $(x_e,\ y_e,\ z_e)_i$ described in Section 4.2. Now it becomes a problem of synthesizing a 3R robot for a pre-

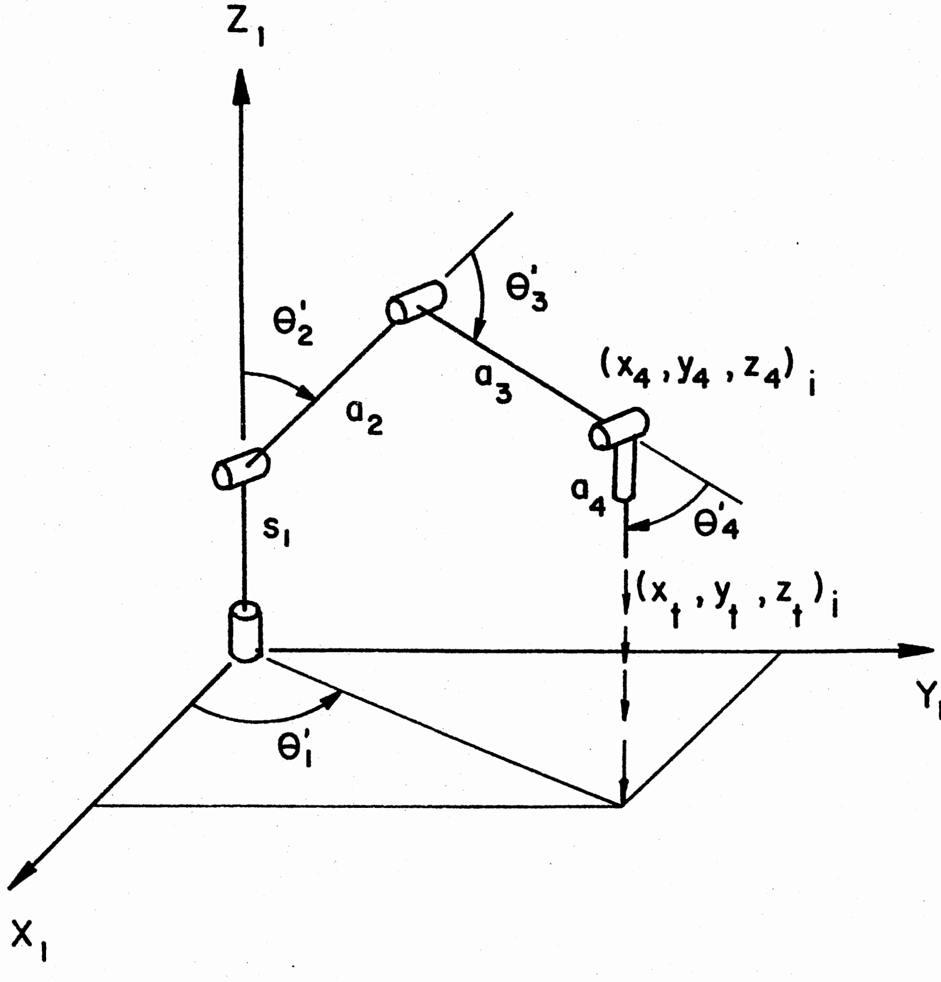


Figure 29. The 4R Industrial Robot

scribed workspace $(x_e, y_e, z_e)_i$, i = 1,2,...n. And it has been studied in Section 4.2.

4.4 Synthesis of 5R Robots

There is quite a lot of work like welding, flame cutting, spray painting, drilling, assembling, . . etc., that can be done by 5R industrial robots. All these jobs mentioned above require the robots to hold the tool at some specified locations and in the specified directions. In other words these jobs require five degrees of freedom, three translations and two rotations, so they need robots consisting of at least five joints.

In order to get the maximum working space and maximum dexterity at the same time, the kinematic paremeters of the 5R robots should be properly selected. The effects of kinematic parameters on the working space and the dexterity have been studied in Chapter III. And optimal values of the link parameters for 5R robots have been arrived as:

1. Regional structure: $a_1 = 0$, $a_2 = a_3$, $s_2 = s_3 = 0$, $\alpha_1 = \pm 90°$, and $\alpha_2 = \alpha_3 = 0°$.
2. Orientational structure: $\alpha_5 = \pm 90°$ and $a_5 \neq 0$.

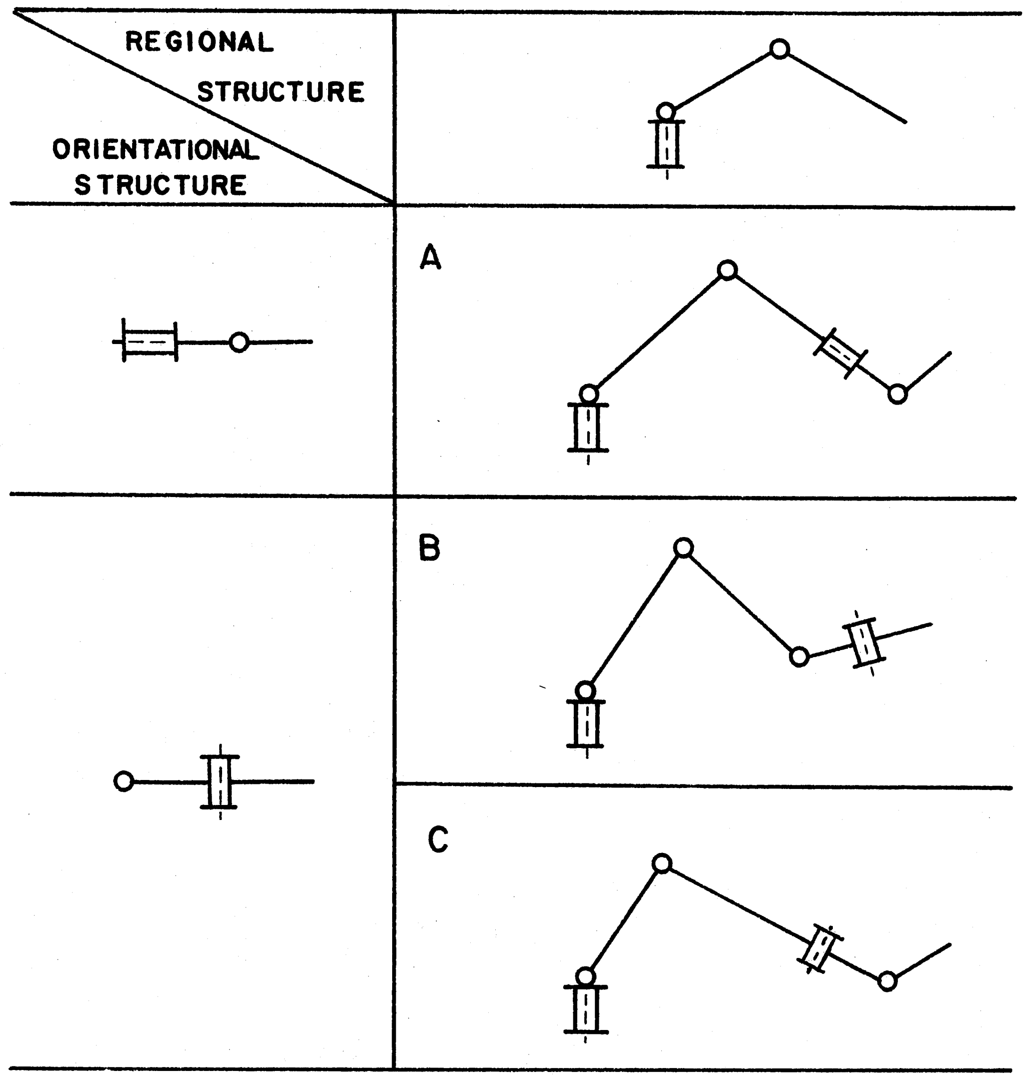
Table XI shows the best combinations of 5R industrial robots with link parameters mentioned above.

4.4.1 The 5R Type A Robots

Considering the robot hand as a line, one cannot specify the angle of rotation along the line axis when it

TABLE XI

STRUCTURE OF 5R INDUSTRIAL ROBOTS



approached a working position. In other words, one can put one imaginary revolute joint along the line axis of a robot hand (see Figure 30) during the synthesis procedure. After placing on imaginary revolute joint on the robot hand, the orientational structure of the robot can be treated as an equivalent link with a spherical joint as shown in Figure 30. By cutting the robot from the equivalent spherical joint, the robot has a 3R regional structure and an orientational structure. Because the spherical joint can rotate to any direction independent of the joint location, that the locations of the assembling joint calculated from both parts must have the same value is the only compatibility condition of assembling. And the compatibility condition of same direction from both parts will be satisfied all the time. In this way the given problem of synthesizing a 5R robot can be decomposed into two parts : to find the location of an equivalent spherical joint and to synthesize a 3R robot. After the 3R robot has been synthesized, one can determine the motion ranges of joint 4 and 5.

For each given $(x_e, y_e, z_e)_i$, the joint displacement θ'_{1i}, θ'_{2i}, θ'_{3i} can be calculated by using Equation (4.6), (4.7) and (4.8). The unit vector of z_4 axis will be

$$(\overline{U}_4)_i = (L_4)_i \, \overline{I} + (M_4)_i \, \overline{J} + (N_4)_i \, \overline{K}$$

$$(L_4)_i = \text{Sin}\,(\theta'_{2i} + \theta'_{3i})\,\text{Cos}\theta'_{1i}$$

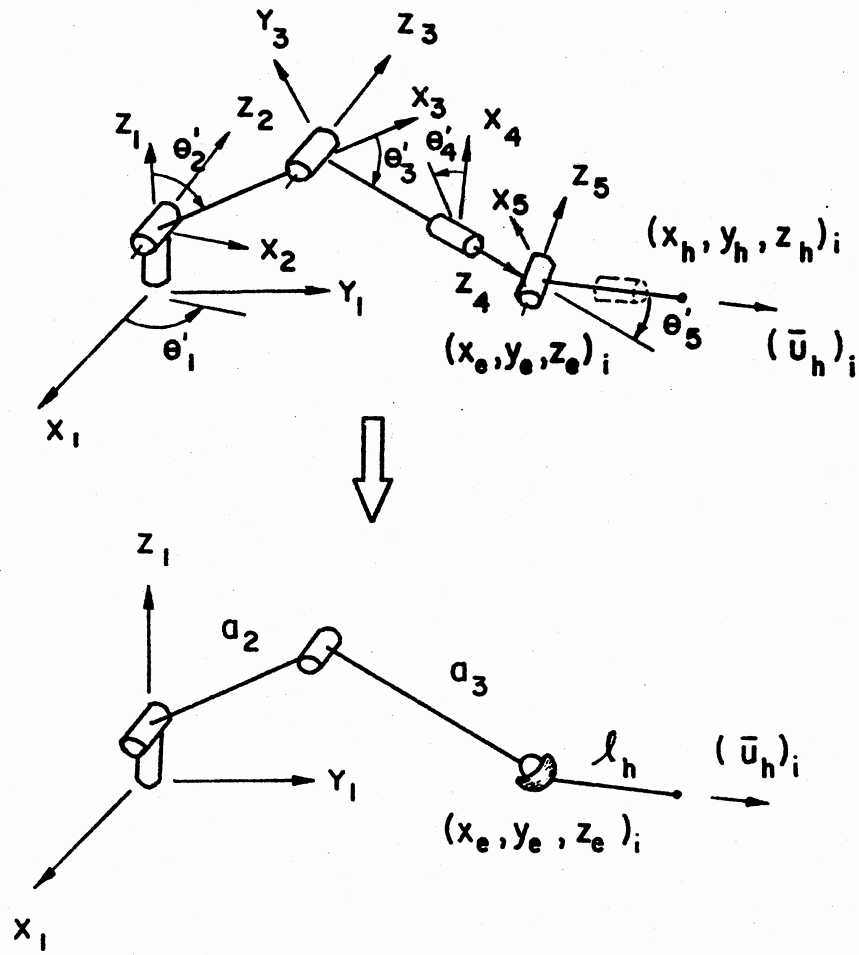


Figure 30. The 5R Type A Robot With one Imaginary R Joint

$$(M_4)_i = \mathrm{Sin}\,(\theta'_{2i} + \theta'_{3i})\,\mathrm{Sin}\theta'_{1i} \qquad (4.12)$$

$$(N_4)_i = \mathrm{Cos}\,(\theta'_{2i} + \theta'_{3i})$$

Then from Figure 30, it is obviously that

$$\theta'_{5i} = \mathrm{Cos}^{-1}[(\overline{U}_4)_i \cdot (\overline{U}_h)_i] \qquad (4.13)$$

The unit vector along Z_5 axis can be obtained by

$$(\overline{U}_5)_i = -\,(\overline{U}_4)_i \times (\overline{U}_h)_i / |(\overline{U}_4)_i \times (\overline{U}_h)_i| \qquad (4.14)$$

and the unit vector along Z_3 is

$$(\overline{U}_3)_i = -\,\mathrm{Sin}\theta'_{1i}\,\overline{I} + \mathrm{Cos}\theta'_{1i}\,\overline{J} \qquad (4.15)$$

From Figure 30, it yields

$$\theta'_{4i} = \mathrm{Cos}^{-1}[(\overline{U}_3)_i \cdot (\overline{U}_5)_i] \qquad (4.16)$$

The synthesizing procedure is summarized below,

1. For a given length of robot hand ℓ_h, specified working positions $(x_h, y_h, z_h)_i$ and approaching directions $(L_h, M_h, N_h)_i$, the corresponding locations of the equivalent spherical joint $(x_e, y_e, z_e)_i$ can be obtained.

$$(x_e)_i = (x_h)_i - \ell_h\,(L_h)_i$$

$$(y_e)_i = (y_h)_i - \ell_h\,(M_h)_i \qquad (4.17)$$

$$(z_e)_i = (z_h)_i - \ell_h\,(N_h)_i$$

2. After getting the location $(x_e, y_e, z_e)_i$, it becomes a problem of synthesizing a 3R robot with a prescribed working space. The same procedure developed in Section 4.2 can be applied here to get the optimal location of the robot base (first joint) x_b, y_b, z_b, link parameters a_2, a_3 and motion ranges of joints $(\theta_1')_{min}$, $(\theta_1')_{max}$, $(\theta_2')_{min}$, $(\theta_2')_{max}$, $(\theta_3')_{min}$ and $(\theta_3')_{max}$.

3. Using Equation (4.12) to (4.16), find $|\theta_4'|_{max}$ and $|\theta_5'|_{max}$. And motion ranges of joint 4 and 5 are

$$\begin{aligned} |\theta_4'| &\leq |\theta_4'|_{max} \\ |\theta_5'| &\leq |\theta_5'|_{max} \end{aligned} \tag{4.18}$$

4.4.2 The 5R Type B Robots

For the same reason put an imaginary revolute joint along the robot hand as discussed in Section 4.4.1. It becomes a 6R robot as shown in Figure 31. Similar to last section divide the robot from joint 4 into two parts, and degenerate the problem from synthesis of 6R robot to synthesis of 3R robot. But for a 5R type B robot the corresponding location of joint 4 $(x_4, y_4, z_4)_i$ should be determined according to the compatibility condition of assembling, i.e.,

$$x_4 = x_4'$$

$$y_4 = y_4'$$

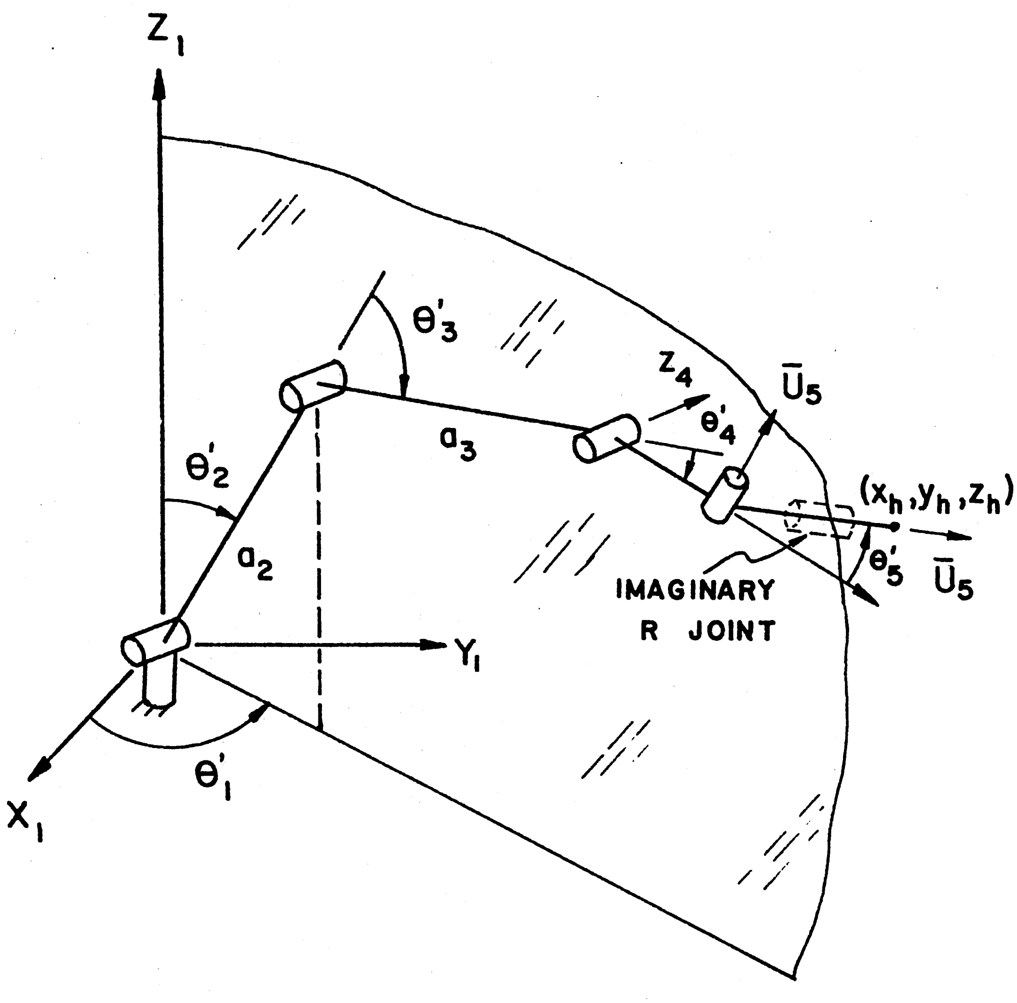


Figure 31. The 5R Type B Robot With one Imaginary R Joint

$$z_4 = z_4' \qquad \bar{U}_4 = \bar{U}_4' \tag{4.19}$$

where x_4, y_4, z_4 and $\bar{U}_4$ are the joint location and unit vector of joint axis Z_4 calculated from the 3R regional structure, and x_4', y_4', z_4' and $\bar{U}_4'$ are those calculated from the orientational structure. From Figure 31, it is obvious that s_1(axis Z_1), a_2, a_3, a_4, and $\bar{U}_5$(axis Z_5) are coplanar. Then the synthesis procedure can be written as following steps:

Step 1: Choose proper values for a_4 and s_6. These values depend upon the kind of driving motor to be used. Let a_4 and s_6 be as small as possible.

Step 2: Assume the robot base (first joint) is located at x_b, y_b.

Step 3: Calculate the location and direction of joint 5.

For each given working station $(x_h, y_h, z_h)_i$ and $(\bar{U}_h)_i = (L_h)_i \bar{I} + (M_h)_i \bar{J} + (N_h)_i \bar{K}$, we can calculate the location of joint 5 by following equation,

$$\begin{aligned} (x_5)_{1,i} &= (x_h)_i - s_6 \cdot (L_h)_i - x_b \\ (y_5)_{1,i} &= (y_h)_i - s_6 \cdot (M_h)_i - y_b \\ (z_5)_{1,i} &= (z_h)_i - s_6 \cdot (N_h)_i - z_b \end{aligned} \tag{4.20}$$

Because $\alpha_5 = 90°$, i.e., $(\bar{U}_5)_i \perp (\bar{U}_h)_i$, it yields,

$$(\overline{U}_5)_i \cdot (\overline{U}_h)_i = 0$$

or

$$(L_5)_i(L_h)_i + (M_5)_i(M_h)_i + (N_5)_i(N_h)_i = 0 \qquad (4.21)$$

and

$$(L_5)_i^2 + (M_5)_i^2 + (N_5)_i^2 = 1 \qquad (4.22)$$

Let $\{L_5, M_5, N_5, P_5, Q_5, R_5\}$ and $\{0,0,1,0,0,0\}$ represent the Plücker coordinates of the axes of joint 5 and joint 1 respectively. From the conditions of assembly, axis of joint 5 and axis of joint 1 are coplanar. The condition of coplanar for these two lines presented by Plücker coordinate (see Appendix A) yields

$$(R_5)_i = 0 \qquad (4.23)$$

By definition

$$(R_5)_i = (x_5)_{1,i}(M_5)_i - (y_5)_{1,i}(L_5)_i \qquad (4.24)$$

We have four unknowns $(L_5)_i$, $(M_5)_i$, $(N_5)_i$, $(R_5)_i$ and four independent Equations (3.21) through (4.24). Solve for $(L_5)_i$, $(M_5)_i$, $(N_5)_i$ to get:

Case 1: $(x_5)_{1,i} \neq 0$ and $(N_h)_i \neq 0$

$$(L_5)_i = \pm \left[\frac{1}{1 + A^2 + B^2}\right]^{\frac{1}{2}}$$

$$(M_5)_i = A(L_5)_i \qquad (4.25a)$$

$$(N_5)_i = -B(L_5)_i$$

where

$$A = \frac{(y_5)_{1,i}}{(x_5)_{1,i}}$$

$$B = \frac{(L_h)_i}{(N_h)_i} + \frac{(M_h)_i}{(N_h)_i} \cdot A$$

Case 2: $(x_5)_{1,i} = 0$ and $(N_h)_i \neq 0$

$$(L_5)_i = 0$$

$$(M_5)_i = \pm \left[\frac{1}{1 + B^2}\right]^{\frac{1}{2}} \quad (4.25b)$$

$$(N_5)_i = - B(M_5)_i$$

where

$$B = (M_h)_i \ / \ (N_h)_i$$

Case 3: $(N_5)_i = 0$

$$(L_5)_i = (M_5)_i = 0$$

$$(N_5)_i = -1 \quad (4.25c)$$

Step 4: Calculate the location and direction of joint 4. Let $\{L_4, M_4, N_4, P_4, Q_4, R_4\}$ represent the Plucker coordinates of link a_4. Because $\alpha_4 = 90°$, it yields

$$(L_4)_i(L_5)_i + (M_4)_i(M_5)_i + (N_4)_i(N_5)_i = 0 \quad (4.26)$$

and

$$(L_4)_i^2 + (M_4)_i^2 + (N_4)_i^2 = 1 \qquad (4.27)$$

Again link a_4 and joint 1 are coplanar,

$$(R_4)_i = (x_5)_{1,i}(M_4)_i - (y_5)_{1,i}(L_4)_i = 0 \qquad (4.28)$$

Solve $(L_4)_i$, $(M_4)_i$, and $(N_4)_i$ from Eq. (4.26) through (4.28)

Case 1: $(x_5)_{1,i} \neq 0$ and $(N_5)_i \neq 0$

$$(L_4)_i = \pm \left[\frac{1}{1 + C^2 + D^2}\right]^{\frac{1}{2}}$$

$$(M_4)_i = C \cdot (L_4)_i \qquad (4.29a)$$

where $$(N_4)_i = - D(L_4)_i$$

$$C = \frac{(y_5)_{1,i}}{(x_5)_{1,i}}$$

$$D = \frac{(L_5)_i}{(N_5)_i} + \frac{(M_5)_i}{(N_5)_i} \cdot C$$

Case 2: $(x_5)_{1,i} \neq 0$ and $(N_5)_i = 0$

$$(L_4)_i = 0$$

$$(M_4)_i = \pm \left[\frac{1}{1 + C^2}\right]^{\frac{1}{2}} \qquad (4.29b)$$

$$(N_4)_i = - C(M_4)_i$$

where $$C = (M_5)_i \,/\, (N_5)_i$$

Case 3: $$(N_4)_i = 0$$

$$(L_4)_i = (M_4)_i = 0 \qquad (4.29c)$$

$$(N_4)_i = -1$$

After solving $(L_4)_i$, $(M_4)_i$, and $(N_4)_i$, the location of joint 4 can be calculated as

$$(x_4)_{1,i} = (x_5)_{1,i} \pm (L_4)_i \cdot a_4$$

$$(y_4)_{1,i} = (y_5)_{1,i} \pm (M_4)_i \cdot a_4 \qquad (4.30)$$

$$(z_4)_{1,i} = (z_5)1,i \pm (N_4)_i \cdot a_4$$

One should note that there are two possible locations for joint 4. And it is proposed to synthesize a robot which can reach both of these two possible locations. In other words the synthesized robot can reach the given working stations in two different ways, and it is helpful for obstacle avoidance in the practical sense.

Step 5: Repeat Step 3 and 4 until we get all $(x_4)_{1,i}$, $(y_4)_{1,i}$, and $(z_4)_{1,i}$, $i = 1,2,3, . . n$ corresponding to each given working station. Now we simplify the problem from the synthesis of a 6R robot to the synthesis of a 3R robot.

Step 6: Use the procedure described in Section 4.2 to synthesize the 3R robot. And get the link lengths a_2, a_3, the location of robot base x_b, y_b, and z_b, and ranges of joint mation $(\theta_1')_{min}$, $(\theta_1')_{max}$, $(\theta_2')_{min}$, $(\theta_2')_{max}$, $(\theta_3')_{min}$ and $(\theta_3')_{max}$.

Step 7: Calculate the joint motion limits. In practical applications, joint motions of the orientational structure, θ_4' and θ_5' , are larger than that of the joints of the regional structure. And most of the time, it is convenient to design that it has the same motion range in the negative direction as the positive direction (e.g., $-120° \leq \theta_4' \leq +120°$). In this sense, the motion ranges of θ_4' and θ_5' cab be calculated.

$$(\theta_4')_i = \text{Cos}^{-1} (L_3L_4 + M_3M_4 + N_3N_4)_i \qquad (4.31)$$

$$(\theta_5')_i = \text{Cos}^{-1} (L_4L_h + M_4M_h + N_4N_h)_i$$

where $(L_4, M_4, N_4)_i$ are calculated by using Equation (4.29) and $(L_3, M_3, N_3)_i$ by

$$(L_3)_i = \frac{(x_4)_{1,i} - (x_3)_{1,i}}{a_3}$$

$$(M_3)_i = \frac{(y_4)_{1,i} - (y_3)_{1,i}}{a_3} \qquad (4.32)$$

$$(N_3)_i = \frac{(z_4)_{1,i} - (z_3)_{1,i}}{a_3}$$

And

$$(x_3)_{1,i} = a_2 \text{ Sin } (\theta_2')_i \cdot \text{Cos } (\theta_1')_i$$

$$(y_3)_{1,i} = a_2 \text{ Sin } (\theta_2')_i \cdot \text{Sin } (\theta_1')_i \qquad (4.33)$$

$$(z_3)_{1,i} = a_2 \text{ Cos } (\theta_2')_i$$

$$i = 1,2,3, \ldots n$$

Find $|\theta_4'|_{max}$ and $|\theta_5'|_{max}$ from Equation (4.31) and let

$$|\theta_4'| \leq |\theta_4'|_{max}$$
$$|\theta_5'| \leq |\theta_5'|_{max} \tag{4.34}$$

4.4.3 The 5R Type C Robots

Because the direction of joint axis 4 depends on the location of robot base (x_b, y_b, z_b) and the link length a_2 and a_3, that the 5R type C robots cannot be directly decomposed into two parts and degenerated to a problem of synthesis of 3R robot.

However for a given robot hand position $(x_h, y_h, z_h)_i$ and approaching direction $(\bar{U}_h)_i$, the location of joint 5 $(x_5, y_5, z_5)_i$ can be calculated

$$(x_5)_i = (x_h)_i - (L_h)_i\ a_5$$
$$(y_5)_i = (y_h)_i - (M_h)_i\ a_5 \tag{4.35}$$
$$(z_5)_i = (z_h)_i - (N_h)_i\ a_5$$

And the location of joint 4 must be located somewhere in a sphere which has a radius of a_4 and center location at $(x_5, y_5, z_5)_i$. If the regional structure is synthesisezed in such a way that its accessible region can cover every spherical region corresponding to every working station $(x_5, y_5, z_5)_i$, then it is possible to degenerate the original

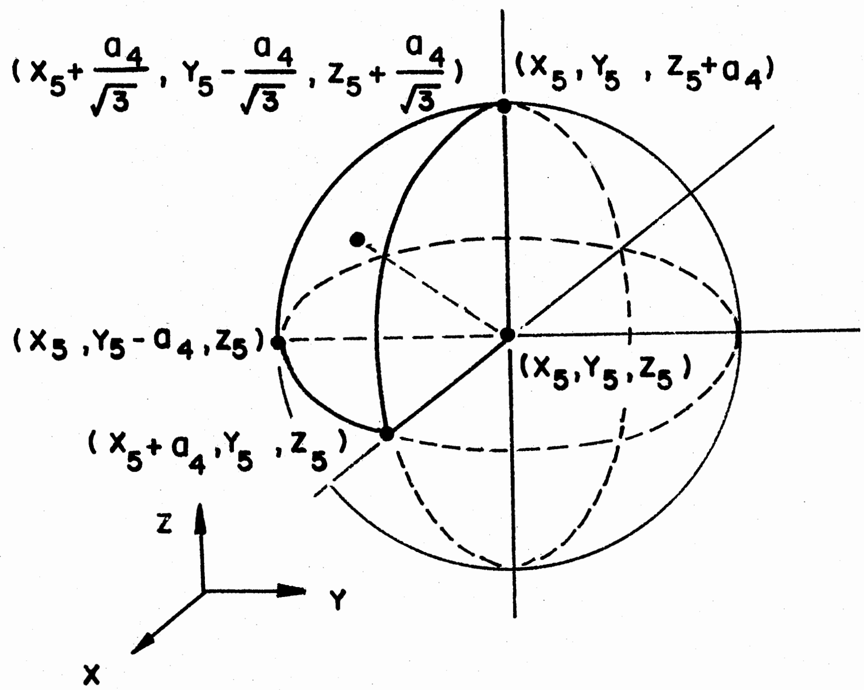


Figure 32. Possible Location of Joint 4

problem to a problem of synthesizing 3R robot.

For each given working position of the robot hand $(x_h, y_h, z_h)_{1,i}$, let the following fourteen points describe the spherical region (see Figure 32).

$$(x_5 \pm a_4 , \quad 0 \quad , \quad 0 \quad)_i$$

$$(\quad 0 \quad , y_5 \pm a_4 , \quad 0 \quad)_i$$

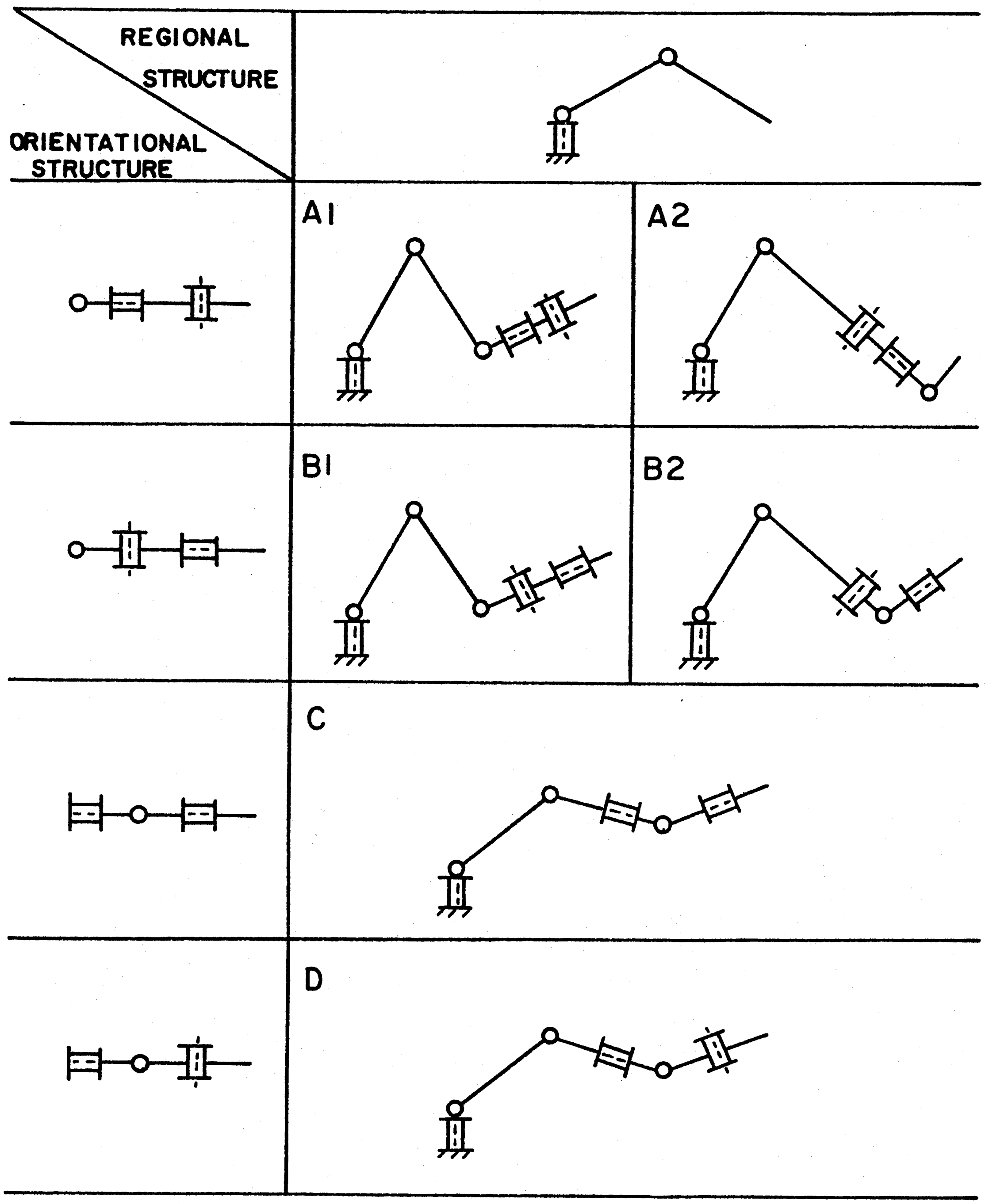
$$(\quad 0 \quad , \quad 0 \quad , z_5 \pm a_4)_i$$

$$(x_5 \pm a_4/\sqrt{3} , y_5 \pm a_4/\sqrt{3} , z_5 \pm a_4/\sqrt{3})_i$$

Then synthesize a 3R robot such that its accessible region can cover all of the points described above. After the 3R robot has been synthesized, the rest of the work is to determine the motion ranges of joint 4 and joint 5. And this is belong to a joint displacement analysis problem of 6R robot. For this type of robots, it is not easy to express the joint displacements in simple closed-form solutions. Either the method developed by Duffy [14] or the numerical method descried in Section 2.3 can be used to solve this problem. It may take a longer time to compute the joint displacements than that of other robots discussed above, and this kind of structure is seldom used among the popular industrial robots.

TABLE XII

STRUCTURE OF 6R INDUSTRIAL ROBOTS



4.5 Synthesis of 6R Robots

From a kinematical point of view, the general 6R robots have six degrees of freedom and should be able to handle rigid-body motion. There are lots of 6R industrial robots in the market right now. Because they can do so many things and can be used in so many working places, sometimes people even called them "universal robots." Now, the 6R industrial robots are widely used in many industries.

To synthesize a 6R robot is not an easy job, because there are eighteen time-independent kinematic parameters and six time-dependent joint displacement to be synthesized. In this section the 6R robots, will be treated as two parts, regional structures and orientational structures, as discussed before. In Chapter III the effects of link parameters on the workspace or dexterity of regional structures and orientational structures have been studied. In order to get the maximum workspace or maximum dexterity, some values or relationships among the kinematic parameters have been arrived. From the results of Chapter III the best combinations of the general purpose 6R robots are shown in Table XII.

Similar to 5R robots, the 6R robots listed in Table XII can be divided into three groups:

Group 1: Like Type C in Table XII, an equivalent spherical joint can be found at the conjunction of the regional structure and orientational structure. Because the spherical joint can rotate to any direction in space, the compatibility condition of assembling about joint direction will automatically be satisfied all the time. The locations of the conjunction joint can be determined independent of the link parameters and location of the regional structure. The original synthesis problem can be degenerated into a problem of synthesizing 3R robots. MA-23 and Unimation Puma 250 series robots belong to this group.

Group 2: Type A1 and B1 in Table XII belong to this group. The direction of the conjunction joint axis is independent of link length a_2 and a_3 of the regional structure. Assume the robot is located at (x_b, y_b) then use the compatibility conditions of assembling (compatibilities of both direction and location) to determine the locations of conjunction joints. Then it degenerates to a problem of synthesizing a 3R robot. Cincinnati Milacron T^3 and Polar 6000 robots belong to type B1 and ASEA IRS6 and Nordson robots belong to type A1.

Group 3: In this group, the direction of the conjunction joint axis depends on all the link parameters and the location of the base of the regional structure. Synthesizing a robot of this group cannot be directly degenerated into a problem of synthesizing a 3R robots. Type A2, B2, and D in Table XII belong to this group.

4.5.1 The 6R Type A1 Robots

Figure 33 shows the structure of the 6R type A1 robot.

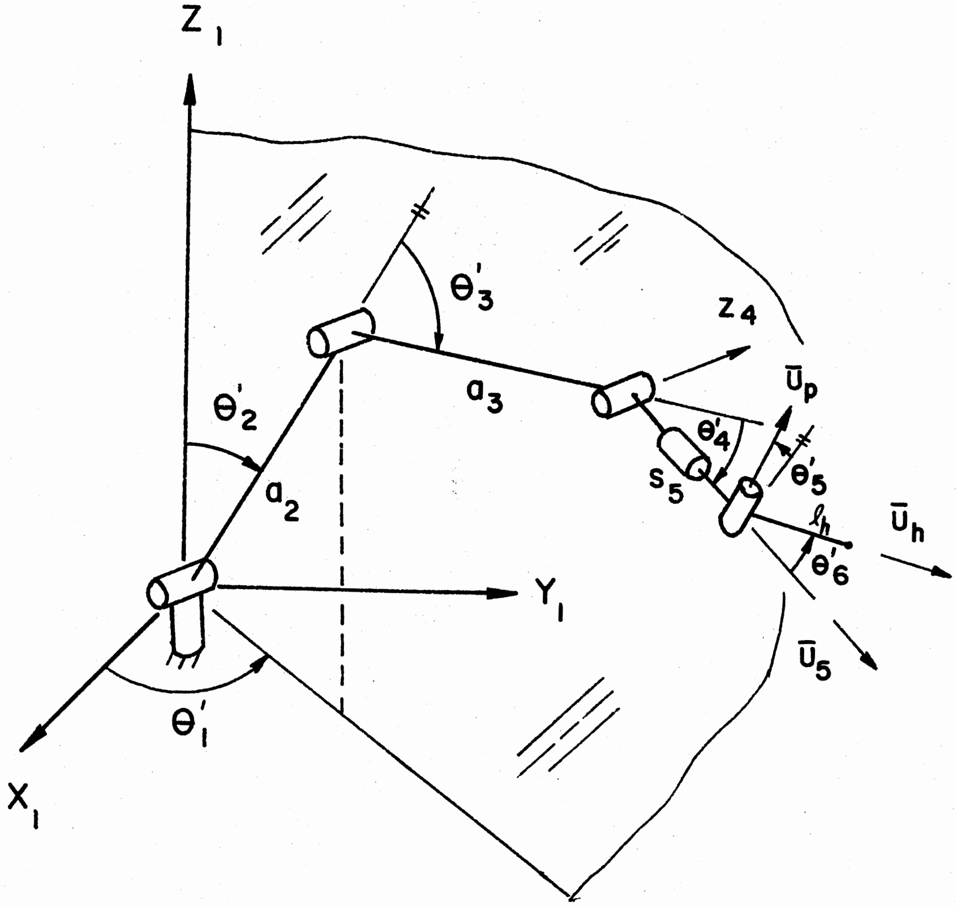


Figure 33. The 6R Type A1 Robot

Given a robot hand length ℓ_h (depends on the length of tool to be used), hand position $(x_h, y_h, z_h)_i$, and the direction vector of the robot hand $(\overline{U}_h)_i$ and $(\overline{U}_p)_i$, i = 1,2, . . n, the location of joint 6 is defined by

$$
\begin{aligned}
(x_6)_{1,i} &= (x_h)_i - \ell_h(L_h)_i - x_b \\
(y_6)_{1,i} &= (y_h)_i - \ell_h(M_h)_i - y_b \\
(z_6)_{1,i} &= (z_h)_i - \ell_h(N_h)_i - z_b
\end{aligned}
\qquad (4.36)
$$

The Plucker line coordinates of axis z_5 and z_1 are $\{L_5, M_5, N_5, P_5, Q_5, R_5\}$ and {0, 0, 1, 0, 0, 0} respectively. From the geometry configuration and compatibility condition, the unit vector $(\overline{U}_5)_i$ of joint axis Z_5 must pass through $(x_6, y_6, z_6)_{1,i}$ and be coplanar with axis Z_1. Similar to that in Section 4.42, the above condition yields

$$(R_5)_i = (x_6)_{1,i}(M_5)_i - (y_6)_{1,i}(L_5)_i = 0 \qquad (4.37)$$

But $(\overline{U}_5)_i$ perpendicular to $(\overline{U}_p)_i$, it yields

$$(L_5)_i(L_p)_i + (M_5)_i(M_p)_i + (N_5)_i(N_p)_i = 0 \qquad (4.38)$$

and

$$(L_5)_i^2 + (M_5)_i^2 + (N_5)_i^2 = 1 \qquad (4.39)$$

Solve Equation (4.37), (4.38) and (4.39) for $(L_5)_i$, $(M_5)_i$ and $(N_5)_i$ and get:

Case 1: $(x_6)_{1,i} \neq 0$ and $(N_p)_i \neq 0$

$$(L_5)_i = \pm \left[\frac{1}{1 + E^2 + F^2}\right]^{\frac{1}{2}}$$

$$(M_5)_i = E(L_5)_i \qquad (4.40a)$$

$$(N_5)_i = - F(L_5)_i$$

where

$$E = \frac{(y_6)_{1,i}}{(x_6)_{1,i}}$$

$$F = \frac{(L_p)_i}{(N_p)_i} + \frac{(M_p)_i}{(N_p)_i} \cdot E$$

Case 2: $(x_6)_{1,i} = 0$ and $(N_p)_i \neq 0$

$$(L_5)_i = 0$$

$$(M_5)_i = \pm \left[\frac{1}{1 + E^2}\right]^{\frac{1}{2}} \qquad (4.40b)$$

$$(N_5)_i = - B(M_5)_i$$

where

$$E = (M_p)_i \,/\, (N_p)_i$$

Case 3: $(N_5)_i = 0$

$$(L_5)_i = (M_5)_i = 0$$

$$\qquad (4.40c)$$

$$(N_5)_i = -1$$

Then the coordinates of joint 4 will be

$$(x_4)_{1,i} = (x_6)_{1,i} - (L_5)_i \cdot s_5$$

$$(y_4)_{1,i} = (y_6)_{1,i} - (M_6)_i \circ s_5 \qquad (4.41)$$

$$(z_4)_{1,i} = (z_6)_{1,i} - (N_6)_i \cdot s_5$$

In this way for given $(x_h, y_h, z_h)_i$, and $(\overline{U}_h)_i$, $(\overline{U}_p)_i$, one can easily get the corresponding location of joint 4 $(x_4, y_4, z_4)_{1,i}$. And the coordinates $(x_4, y_4, z_4)_{1,i}$ here are the same as $(x_e, y_e, z_e)_{1,i}$ given in Section 4.2. Now it becomes a synthesis problem of 3R robots. Use the procedure described in Section 4.2 to get the location of robot base (x_b, y_b, z_b), the link length a_2, a_3 and the joint motion limits $(\theta_1')_{min}$, $(\theta_1')_{max}$, $(\theta_2')_{min}$, $(\theta_2')_{max}$, $(\theta_3')_{min}$ and $(\theta_3')_{max}$.

After designing the 3R robot, the unit vector along link a_3 will be determined using the following procedure.

$$(\overline{U}_3)_i = (L_3)_i \, \overline{I} + (M_3)_i \, \overline{J} + (N_3)_i \, \overline{K}$$

$$(L_3)_i = \text{Sin}(\theta_{2i}' + \theta_{3i}') \, \text{Cos}\theta_{1i}'$$

$$(M_3)_i = \text{Sin}(\theta_{2i}' + \theta_{3i}') \, \text{Sin}\theta_{1i}' \qquad (4.42)$$

$$(N_3)_i = \text{Cos}(\theta_{2i}' + \theta_{3i}')$$

And the unit vector along the axis of joint 4,

$$(\overline{U}_4)_i = - \text{Sin}\theta_{1i}' \, \overline{I} + \text{Cos}\theta_{1i}' \, \overline{J} \qquad (4.43)$$

From Figure 33, it is obviously that,

$$(\theta_4')_i = \pm \text{Cos}^{-1}[\, (\overline{U}_3)_i \cdot (\overline{U}_5)_i \,] \qquad (4.44)$$

where the sign "+" stands when $(\overline{U}_3)_i \times (\overline{U}_5)_i \cdot (\overline{U}_4)_i \geq 0$ and $(\overline{U}_5)_i$ are determined by Equation (4.40).

$$(\theta_5')_i = \pm \text{Cos}^{-1}[(\bar{U}_4)_i \cdot (\bar{U}_p)_i] \qquad (4.45)$$

where the sign "+" holds when $(\bar{U}_4)_i \times (\bar{U}_p)_i \cdot (\bar{U}_5)_i \geq 0$

$$(\theta_6')_i = \pm \text{Cos}^{-1}[(\bar{U}_5)_i \cdot (\bar{U}_h)_i] \qquad (4.46)$$

where the sign "+" holds when $(\bar{U}_5)_i \times (\bar{U}_h)_i \cdot (\bar{U}_p)_i \geq 0$

The synthesis procedure can be summarized into the following steps:

1. Given the following information:
   a. robot hand position $(x_h, y_h, z_h)_i$, direction $(\bar{U}_h)_i$ and $(\bar{U}_p)_i$, $i = 1,2,3,\ldots,n$.
   b. robot hand length ℓ_h, it depends on the length of the tool to be used.
   c. the link length s_4, it depends on the driving motor to be used for joint 4.

2. Assume the location (x_b, y_b) of the robot base (the first joint).

3. Use Equation (4.41) to find the corresponding location $(x_4, y_4, z_4)_{1,i}$ of joint 4.

4. Use the procedure descried in Section 4.2 to synthesize the 3R robot which can access all $(x_4, y_4, z_4)_{1,i}$, $i = 1,2,3\ldots,n$.

5. Find the motion limits for joint 4, 5 and 6 by using Equations (4.44), (4.45) and (4.46).

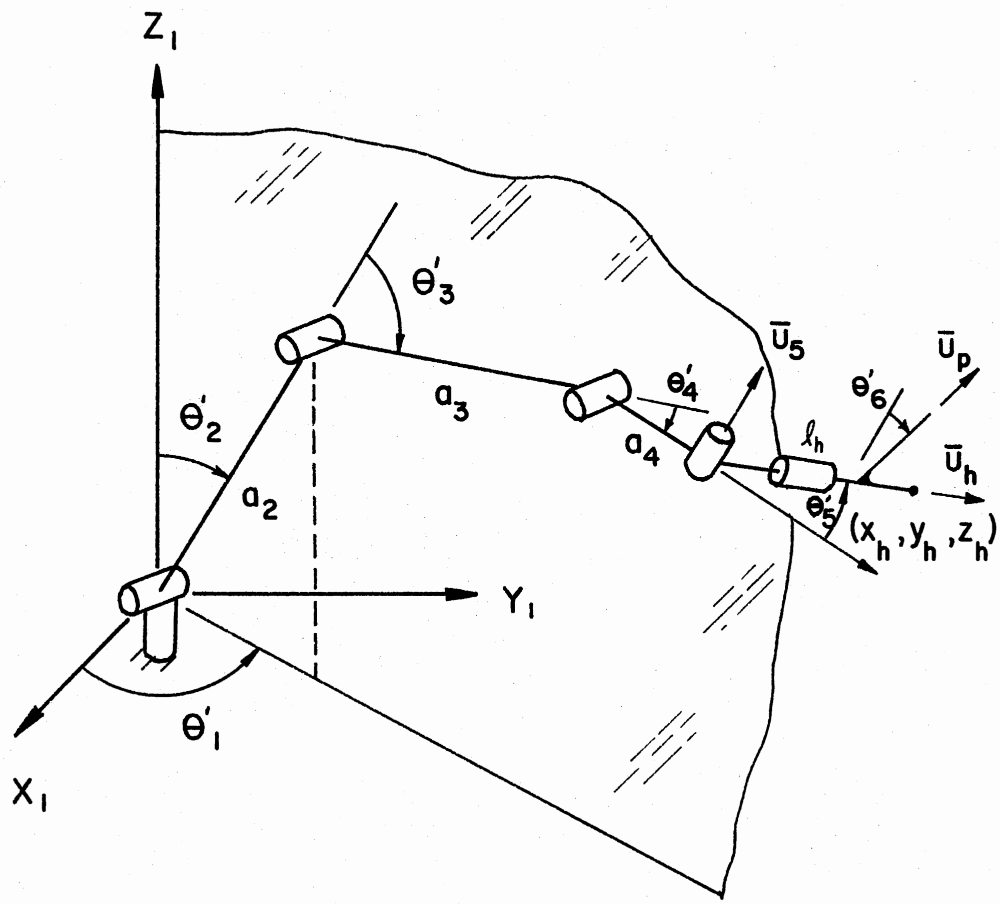


Figure 34. The 6R Type B1 Robot

4.5.2 The 6R Type B1 Robots

In Section 4.4.2, an imaginary revolute joint has been placed along the robot hand. Then the 5R type B robot becomes a 6R type B1 robot. Hence the synthesis procedure described in that section can be applied in this section.

Treating the robot hand as a rigid body, one should specify the location of the robot hand $(x_h, y_h, z_h)_i$, the direction along the robot hand axis $(\overline{U}_h)_i = (L_h, M_h, N_h)_i$ and the direction perpendicular to the robot hand axis $(\overline{U}_p)_i = (L_p, M_p, N_p)_i$, where i = 1,2,3,...,n. Because this kind of robot has a revolute joint (joint 6) along the axis of robot hand, the requirement of $(\overline{U}_p)_i$ can be easily achieved by simply rotating joint 6. Therefore the synthesizing procedure of the 6R type B1 robots can be carried out by following steps,

1. Ignore the requirement of $(\overline{U}_p)_i$ and treat the robot hand as a line. Then use the procedure described in Section 4.4.2 to synthesize a 5R type B robot.

2. After the 5R type B robot has been synthesized, the remaining work is to determine the $(\theta_6')_{min}$ and $(\theta_6')_{max}$, For each given robot hand position $(x_h, y_h, z_h)_i$, $(\theta_6')_i$ can be computed by

$$(\overline{U}_6)_i = (\overline{U}_5)_i \times (\overline{U}_h)_i \qquad (4.47)$$

$$i = 1,2,...n$$

where $(\overline{U}_h)_i$ given and $(\overline{U}_5)_i$ can be calculated by using

Equation (4.25). Then, compute the value of $(\theta_6')_i$,

$$(\theta_6')_i = \pm \operatorname{Cos}^{-1}[(\overline{U}_5)_i \cdot (\overline{U}_p)_i] \qquad (4.48)$$

where the sign "+" holds when $(\overline{U}_5)_i \times (\overline{U}_p)_i \cdot (\overline{U}_h)_i \geq 0$

After finding $(\theta_6')_{min}$ and $(\theta_6')_{max}$ in Equation (4.48), the synthesis procedure is completed.

4.5.3 The 6R Type C Robots

An imaginary R joint was put along the axis of robot hand before synthesizing the 5R type A robots in Section 4.4.1. After putting the imaginary R joint, the original 5R type A robot becomes a 6R type C robot. Therefore the synthesis procedure of the 6R type C robots is the same as that of the 5R type A robots, except the joint displacement of the imaginary joint (joint 6) was ignored in Section 4.4.2.

Now given the robot hand positions $(x_h, y_h, z_h)_i$, $i = 1,2,3 \ldots n$, and directions $(\overline{U}_h)_i$ along the robot hand and $(\overline{U}_p)_i$ perpendicular to the robot hand, the synthesis procedure for the 6R type C robots can be carried out by ignoring the robot hand $(\overline{U}_p)_i$ first. Use the procedure described in Section 4.4.1, to get all the data of the synthesized 6R robot except the range of motion of joint 6. To get the range of motion of joint 6, use the following equation,

$$(\theta_6')_i = \pm \operatorname{Cos}^{-1}[(\overline{U}_5)_i \cdot (\overline{U}_p)_i] \qquad (4.49)$$

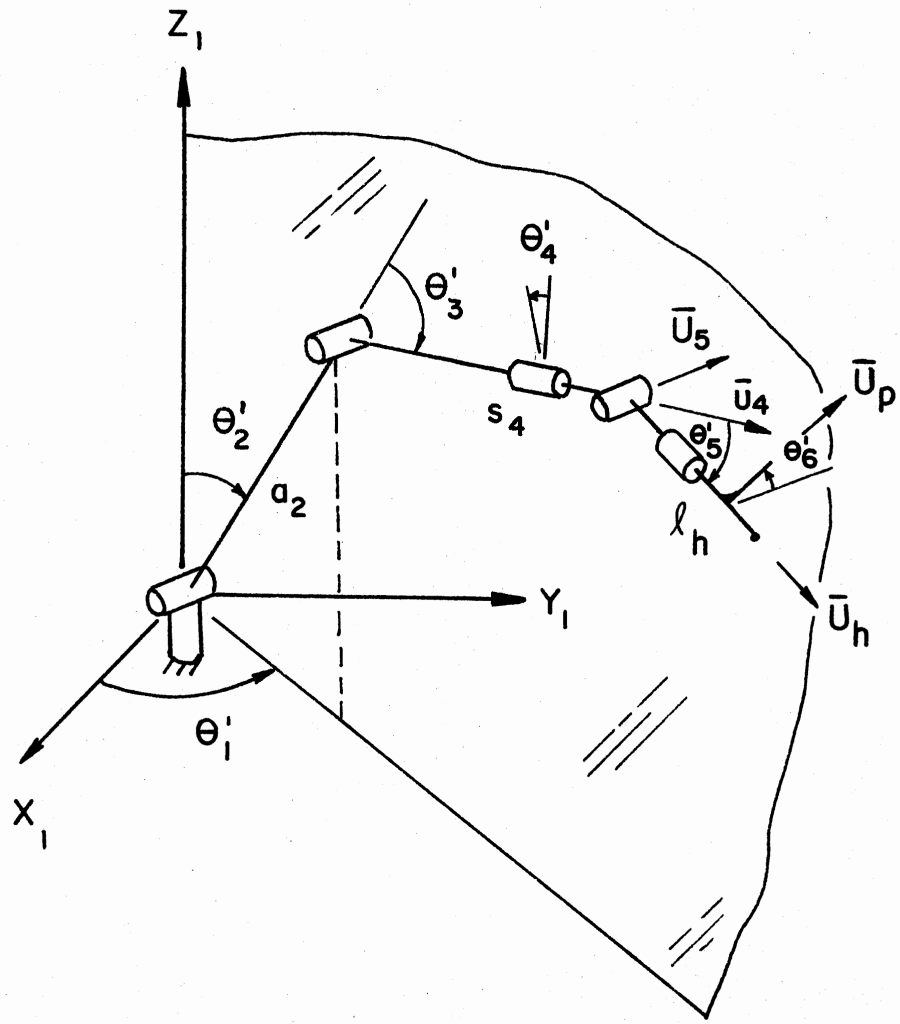


Figure 35. The 6R Type C Robot

where $(\overline{U}_5)_i$ is determined by Equation (4.12) and i = 1,2, . . , n.

Find $(\theta_6')_{min}$ and $(\theta_6')_{max}$ in Equation (4.49), and the synthesis procedure of the 6R type C robots is completed.

4.5.4 The 6R Type A2, B2 and D Robots

The 6R robot of type A2, B2 and D described in Table XII cannot be directly decomposed into two parts and degenerated to a problem of synthesis of 3R robots as has been discussed in Section 4.4.3. The same procedure described in Section 4.4.3 will be applied to synthesize the 6R robot of these kinks. The synthesis procedure may be summarized as,

1. Given the robot hand position $(x_h, y_h, z_h)_i$ and direction $(\overline{U}_h)_i$, $(\overline{U}_p)_i$, i = 1,2,3 . . .n, find the location of joint 5 by Equation (4.35).

2. Find the spherical regions that joint 4 may be located in (see Section 4.4.3).

3. Use the procedure in Section 4.2 to synthesize the 3R robot which can reach all the spherical regions described in step 2, and determine the motion ranges of all joints.

4.6 Example

A specific job needs a robot to work on eighteen working stations as shown in Figure 36. The location and orientation of each working station are listed in Table

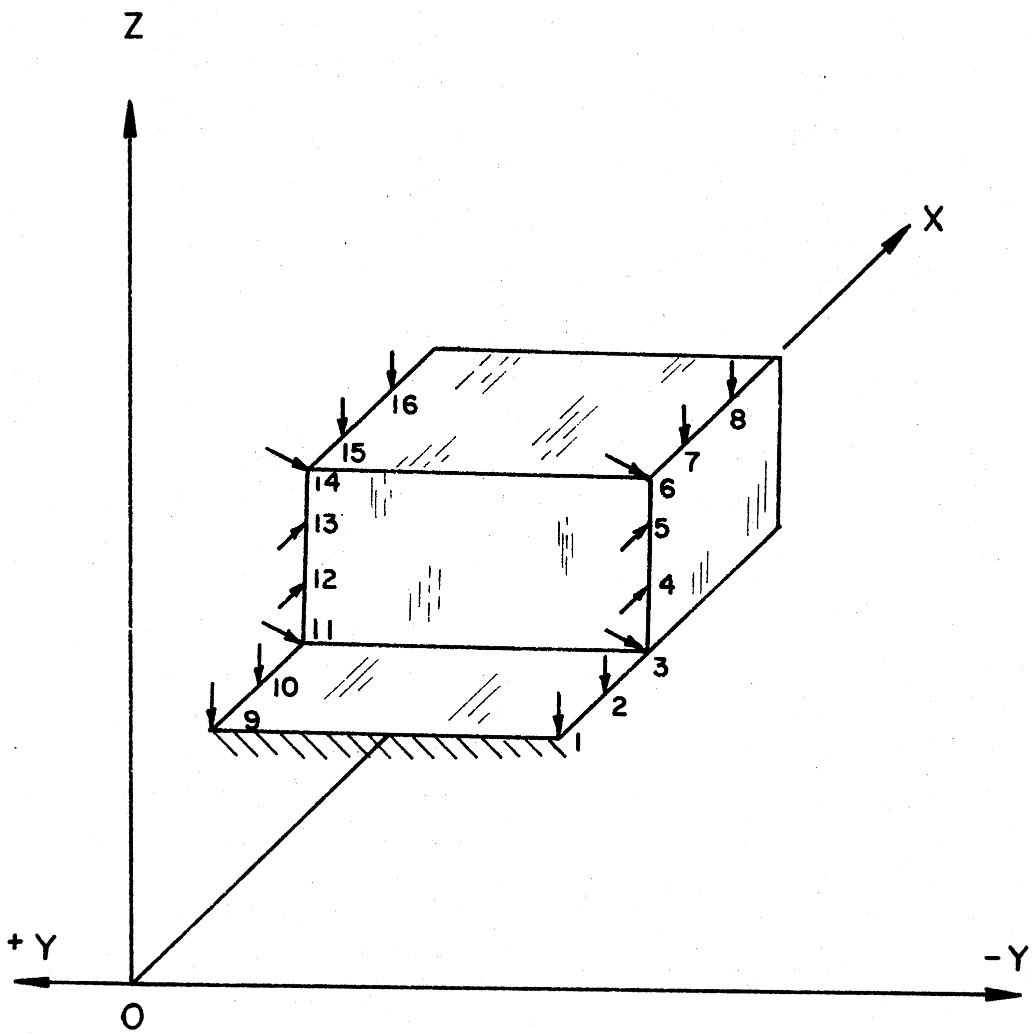


Figure 36. Working Stations of a Specified job

TABLE XIII

LOCATIONS AND ORIENTATIONS OF WORKING STATIONS

No.	X	Y	Z	L_h	M_h	N_h	L_p	M_p	N_p
1	100	-25	0	0.000	0.000	-1.000	0.000	1.000	0.000
2	110	-25	0	0.000	0.000	-1.000	0.000	1.000	0.000
3	120	-25	0	0.707	0.000	-0.707	0.000	1.000	0.000
4	120	-25	15	1.000	0.000	0.000	0.000	1.000	0.000
5	120	-25	30	1.000	0.000	0.000	0.000	1.000	0.000
6	120	-25	45	0.707	0.000	-0.707	0.000	1.000	0.000
7	130	-25	45	0.000	0.000	-1.000	0.000	1.000	0.000
8	140	-25	45	0.000	0.000	-1.000	0.000	1.000	0.000
9	100	25	0	0.000	0.000	-1.000	0.000	1.000	0.000
10	110	25	0	0.000	0.000	-1.000	0.000	1.000	0.000
11	120	25	0	0.707	0.000	-0.707	0.000	1.000	0.000
12	120	25	15	1.000	0.000	0.000	0.000	1.000	0.000
13	120	25	30	1.000	0.000	0.000	0.000	1.000	0.000
14	120	25	45	0.707	0.00	-0.707	0.000	1.000	0.000
15	130	25	45	0.000	0.000	-1.000	0.000	1.000	0.000
16	140	25	45	0.000	0.000	-1.000	0.000	1.000	0.000

XIII.

Obviously this job requires rigid-body motion on the robot hand. In other words it requires a 6R robot to do this job. Assume the minimum required space for mounting the driving motor is 5 inches. Computerize the procedures developed in previous sections (see Appendix D), and get the synthesized robots of different types as the following.

1. Type A1 robot
   a. Base location: (75.0, 0.0, 45.0)
   b. Link lingths: $a_2 = a_3 = 35.18$
   c. Joint Displacements: $\theta_1 = -45° \sim 45°$
      $\theta_2 = 35° \sim 97°$
      $\theta_3 = 0° \sim 94°$
      $\theta_4 = 1° \sim 99°$
      $\theta_5 = -45° \sim 45°$
      $\theta_6 = -90° \sim 0°$
2. Type B1 robot
   a. Base location: (92.1, 0.0, 22.5)
   b. Link lengths: $a_2 = a_3 = 31.52$
   c. Joint Displacements: $\theta_1 = -73° \sim 73°$
      $\theta_2 = 0° \sim 59°$
      $\theta_3 = 0° \sim 126°$
      $\theta_4 = -122° \sim 122°$
      $\theta_5 = -48° \sim 48°$
      $\theta_6 = -163° \sim -18°$
3. Type C robot

a. Base location: (75.0, 0.0, 42.5)

b. Link lengths: $a_2 = s_4 = 35.03$

c. Joint Displacements:

$$\theta_1 = -45° \sim 45°$$

$$\theta_2 = 38° \sim 102°$$

$$\theta_3 = 0° \sim 92°$$

$$\theta_4 = -65° \sim 65°$$

$$\theta_5 = -73° \sim 97°$$

$$\theta_6 = -86° \sim 86°$$

CHAPTER V

SUMMARY AND CONCLUSIONS

The main objective of the present investigation is to develop the synthesis procedures to synthesize the link parameters of robots/manipulators which are required to access a prescribed working space. The robot hand may be treated as a point, a line or a rigid body.

In order to achieve the objective, the present work has developed an algorithm to plot the contour of robots/ manipulators. The robot may consist of two, three, up to any number of revolute joints. The revolute joints may be able to make a complete rotation or may just be able to rotate within some range. The contour of the workspace may be plotted on a plane which contains the first joint axis or on an arbitrary plane in space. The robot hand may be treated as a point, a line or a rigid body in the algorithm of plotting the workspace.

Next, the present work has studied the effects of link parameters (a, α and s) on the workspace and dexterity of the general 3R robot. Considering a robot to be divided into two substructures, the regional structure which contributes the gross motion of the robot hand and the orientational structure which contributes the orientation of the robot hand, the present investigation has found the optimal combination of link parameters for the regional structure and orientational structures. A 3R regional structure with $a_1 = 0$, $a_2 = a_3$, $s_1 = s_2 = s_3 = 0$, $\alpha_1 = \pm 90°$ and $\alpha_2 = 0°$ or $180°$ can access the maximum workspace in compare with all other 3R structures when the summation of the lengths of all links is constant. And a 3R orientational structure with $\alpha_1 = \pm 90°$ and $\alpha_2 = \pm 90°$ can get the maximum dexterity. And these results have been applied to synthesize robots/manipulators for a prescribed working space. The synthesis procedures for 2R, 3R, 4R, 5R and 6R robots/manipulators have been presented in this work.

In summary, the contributions of the present investigation are:

1. To provide a new algorithm which can plot the contour of the working space of a n-R robot on an arbitrary plane. The revolute joints of the robot may make complete rotations or may just rotate between motion limits. The robot hand can be treated as a point, a line or a rigid-body.

2. To study the effects of link parameters on the workspace and dexterity of the general 3R robot structure. Valuable design charts have been presented. And the optimum regional structure and orientational structure of robots/manipulators have been found.

3. To develop synthesis procedures for 2R, 3R, 4R, 5R and 6R industrial robots/manipulators with well defined accessible working space. The best combinations of the regional structure and orienatational structures are also presented.

The algorithm for plotting the workspaces of robots provide a useful tool for people to evalute the performance of a robot or to select suitable robots and arrange them properly on the production lines. The results from the study of the effect of link parameters provide very useful data to robot designers. The synthesis procedures of robots may be applied to design robots/manipulator for special tasks. The robots used in outer space, undersea, underground, nuclear power plants, different kind of industries, . . etc, may need different robots/manipulator for working requirements.

An extension of this work is to consider that the robots have some prismatic pairs along with revolute pairs. From a kinematics point of view the existence of prismatic pairs will simplify the present problem, and required a minor modification of the present work. The further extension of this work may consider the space constraints on the motion of robot links. The application of robots in outer space, the available space for the robots may be limited. In that case, the space constraints on the robot links may become significant. Finally it is expected that more work along this line will be done in the near future.

BIBLIOGRAPHY

1. Bevridge, G. S. and R. S. Schechter. Optimization: Theory and Practics. New York: McGraw Hill, 1970.

2. Chen, P. and B. Roth. "Design Equations for the Finitely and Infinitesimally Seperated Position Synthesis of Binary Links and Combined Link Chains." Journal of Engineering for Industry, Trans. ASME, Vol. 91 (1969), pp. 209-219.

3. Collatz, L. and W. Wetterling. Optimization Problems. New York: Springer-Verlag, 1975.

4. Dantzig, G. B. Linear Programing and Extensions. New Jersey: Princeton University Press, 1963.

5. Dantzig, G. B. and R. M. Van Slyke. "Generalized Upper Bounding Techniques." Journal of Computer System Sci., Vol. 1 (1967), pp. 213-226.

6. Duffy, J. and H. Y. Habib-olahi. "A Displacement Analysis of Spatial Five-Link 3R-2C Mechanisms. Part 1. On the Closures of the RCRCR Mechanisms." Journal of Mechanisms, Vol. 6 (1971), pp. 289-301.

7. Duffy, J. and H. Y. Habib-Olahi. "A Displacement Analysis of Spatial Five-Link 3R-2C Mechanisms. Part 2. Analysis of RRCRC Mechanisms." Journal of Mechanisms, Vol. 6 (1971), pp. 463-473.

8. Duffy, J. and H. Y. Habib-Olahi. "A Displacement Analysis of Spatial Five-Link 3R-2C Mechanisms. Part 3. Analysis of RCRRC Mechanisms." Journal of Mechanisms, Vol. 7 (1972), pp. 71-84.

9. Duffy, J. and R. Martinovic. "Kinematics of 3R-3P Computer Controlled Manipulators." Proceedings 7th International Symposium on Industrial Robots, Tokyo, 1977, pp. 67-78.

10. Duffy, J. "Displacement Analysis of Spatial Seven-Link 5R-2P Mechanism." Journal of Engineering for Industry, Trans. ASME, Vol. 99 (1977), pp. 692-701.

11. Duffy, J. "A Kinematic Analysis and Classification of Robot Arms." Workshop on the Impact on the Academic Commuty of Required Research Activity for Generalized Robotic Manipulators, University of Florida, 1978, pp. 43-64.

12. Duffy, J. and S. Derby. "Displacement Analysis of a Spatial 7R Mechanisms--A Generalized Lobster's Arms." Journal of Mechanics, Trans. ASME, Vol. 101 (1979), pp. 224-231.

13. Duffy, J. Analysis of Mechanisms and Robot. New York: John Wiley & Sons, 1980.

14. Duffy, J. and C. Crane. "A Displacement Analysis of the General Spatial 7-Link, 7R Mechanisms." Mechanism and Machine Theory, Vol. 15 (1980), pp. 153-169.

15. Fichter, E. F. and K. H. Hunt. "The Fecund Torus, its Bitangent-Circles and Derived Linkages." Mechanism and Machine Theory, Vol. 10 (1975), pp. 167-176.

16. Flatau, K., et al. "Some Preliminary Correlations Between Control Modes of Manipulators and Their Performance Indics." Proceedings 1st National Conference on Remotely Manned Systems, Washington, D. C., 1971, p. 189.

17. Friedrich, O. "The Robot Revolution." TIME, Dec. 8, 1980, pp. 72-83.

18. Frolov, K. V., A. A. Kobrinsky, and A. E. Kobrinsky. "A Computer Guidance of Manipulators." Proceedings 7th International Symposium on Industrial Robots, Tokyo, 1977, pp. 161-168.

19. Gupta, K. C. and B. Roth. "Design Considerations for Manipulator Workspace." Journal of Mechanical Design, Trans. ASME, 1982. (in press.)

20. Kobrinsky, A. E. "Performance Evaluation of ROMANSY." Performance Evaluation of Programmable Robots and Manipulators, National Bureau of Standards, 1976, pp. 63-68.

21. Kohli, D. "Synthesis and Analysis of Spatial, two-Loop, Six-Link Mechanisms." (Ph.D. dissertation, Oklahoma State University, 1973.)

22. Konstantinov, M. "Structural and Kinematic Analysis of Robots and Manipulators." Proceedings 4th

International Symposium on Industrial Robots, Tokyo, 1974, pp. 311-327.

23. Krabs, W. Optimization and Approximation. New York: John Wiley & Sons, 1979.

24. Kumar, A. "Characterization of Manipulator Geometry." (Ph.D. dissertation, University of Houston, 1980.)

25. Kumar, A. and K. J. Waldron. "The Workspace of a Mechanical Manipulator." Journal of Mechanical design, Trans. ASME, Vol. 103 (1981), pp. 665-672.

26. Kumar, A. and K. J. Waldron. "The Dextrous Workspace." ASME Paper No. 80-DET-108.

27. Loeff, L. A. and A. H. Soni. "An Algorithm for Computer Guidance of a Manipulator in Between Obstacles." Journal of Engineering for Industry, Trans. ASME, Vol. 97 (1975), pp. 836-842.

28. Makino, H. "A Kinematical Classification of Robot Manipulators." Proceedings 6th International Symposium on Industrial Robots, University of Nottingham, 1976, pp. F2.11-F2.24.

29. Mason, R. M. "The Digital Approximation of Contours." JACM, Vol. (1956), pp. 355-359.

30. Milenkovic, V. "Computer Synthesis of Continuous Robot Motion." Proceedings 5th World Congress on the Theory of Machines and Mechanisms, Montreal, 1979, pp. 1332-1335.

31. Mischke, C. R. An Introduction to Computer-aided-design. Englewood, Cliffs, N.J.: Prentice-Hall Inc., 1968.

32. Mital, K. V. Optimization Methods. New York: John Wiley & Sons, 1976.

33. Orlandea, N. and T. Bereny. "Dynamic Continuous Path Synthesis of Industrial Robots Using ADAMS Computer Program." Journal of Mechanical Design, Trans. ASME, Vol. 103 (1981), pp. 602-607.

34. Paul, R. L. "Modeling, Trajectory Calculation and Servoing of a Computer Controlled Arm." Stanford Artificial Intelligence Laboratory Memo AIM-177, Stanford University, 1972.

35. Pieper, D. E. and B. Roth. "The Kinematics of

Manipulator Under Computer Control." Proceedings 2nd International Congress on the Theory of Machines and Mechanisms, Vol. 2 (1969), pp. 159-168.

36. Pieper, D. E. "The Kinematics of Manipulators Under Computer Control." Stanford Artificial Intelligence Laboratory Memo AIM-72, 1968.

37. Roth, B. "On the Screw Axes and Other Special Lines Associated with Spatial Displacements of a Rigid Body." Journal of Engineering for Industry, Trans. ASME, Vol. 89 (1967), pp. 102-110.

38. Roth, B. J. Rastegar, and V. Scheinman. "On the Design of Computer Controlled Manipulators." On Theory and Practice of Robots and Manipulators, New York: Springer-Verlag, 1974, pp. 93-113.

39. Roth, B. "Performance Evaluation of Manipulators from a Kinematic Viewpoint." Performance Evaluation of Programmable Robots and Manipulators, National Bureau of Standards, ~~1976,~~ pp. 39-62. 1975 39-61

40. Roth, B. "Robots." Applied Mechanics Reviews, Vol. 31 (1978), pp. 1511-1519.

41. Scheinman, V. D. "Design of Computer Controlled Manipulator." Stanford Artificial Intelligence Project Memo No. 92, 1969.

42. Shimano, B. and B. Roth. "Ranges of Motion of Manipulators." On the Theory and Practice of Robots and Manipulators, Elsevier, 1977, pp. 18-27.

43. Shimano, B. "The Kinematic Design and Force Control of Computer Controlled Manipulators." (Ph.D. dissertation, Stanford University, 1978.)

44. Soni, A. H. and P. R. Pamidi. "Closed Form Displacement Relations of a Five-Link RRCCR Spatial Mechanism." Journal of Engineering for Industry, Trans. ASME, Vol. 93 (1971), pp. 221-238.

45. Starr, G. "A Comparison of Control Modes for Time-Delayed Remote Manipulation." (Ph.D. dissertation, Stanford University, 1978.)

46. Sugimoto, K. "An Approach to Structural Synthesis of Robots." Proceedings 9th International Symposium on Industrial Robots, Washington, D.C., 1979, pp. 641-655.

47. Sugimoto, K. and J. Duffy. "Determination of Extreme Distances of a Robot Hand--Part 1, A General Theory." Journal of Mechanical Design, Trans. ASME, Vol. 103 (1981), pp. 631-636.

48. Sugimoto, K. and J. Duffy. "Determination of Extreme Distances of a Robot Hand--Part 2, Robot Arms with Special Geometry." Journal of Mechanical Design, Trans. ASME, Vol. 103 (1981), pp. 776-783.

49. Tsai, L. W. and B. Roth. "Incompletely Specified Displacements: Geometry and Spatial Linkage Synthesis." Journal of Engineering for Industry, Trans. ASME, Vol. 95 (1973), pp. 603-611.

50. Tsai, Y. C. and A. H. Soni. "Accessible Region and Synthesis of Robot Arms." Journal of Mechanical Design, Trans. ASME, Vol. 103 (1981), pp. 803-811.

51. Vertut, J. "Contribution to Analyses Manipulator Coverage and Dexterity." On Theory and Practice of Robots and Manipulators, New York: Springer-Verlag, 1974, pp. 277-289.

52. Vukobratovic, M., V. Potkonjak, and D. Hristic. "Dynamic Method for the Evaluation and Choice of Industrial Manipulator." Proceedings 9th International Symposium on Industrial Robots, Washington, D.C., 1979, pp. 549-561.

53. Yang, A. T. "Displacement Analysis of Spatial Five-Link Mechanisms Using (3 x 3) Matrices with Dual-Number Elements." Journal of Engineering for Industry, Trans. ASME, Vol. 93 (1971), pp. 221-226.

54. Yuan, M. S. C. "Displacement Analysis of the RCRCR Five-Link Spatial Mechanism." Journal of Mechanisms, Vol. 6 (1971), pp. 119-134.

APPENDIXES

APPENDIX A

PLUCKER'S LINE COORDINATES

A straight line in space can be represented by several ways,

1. The intersection of two planes

$$ax + by + cz + d = 0$$
$$a'x + b'y + c'z + d' = 0 \qquad (A.1)$$

2. Specifying two points, (x_1, y_1, z_1) and (x_2, y_2, z_2) which lie on the line. And the equation of the line passing through these two points will be:

$$\frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_1} = \frac{z - z_1}{z_2 - z_1} \qquad (A.2)$$

3. Specifying a point (x_1, y_1, z_1) lying on the line and the direction of the line (L, M, N). The line equation is:

$$\frac{x - x_1}{L} = \frac{y - y_1}{M} = \frac{z - z_1}{N} \qquad (A.3)$$

All Equations (A.1), (A.2), and (A.3) can be transformed to the form as:

$$x = Ay + B$$
$$y = Cz + D \qquad (A.4)$$

where A, B, C, and D are four independent parameters. Hence a straight line in space has four independent parameters and there are ∞^4 line in space.

In addition to the above three methods, the Plücker's line coordinates are widely used by kinematician in recent years. The Plücker coordinates are defined as {L, M, N, P, Q, R}. Where $\overline{U} = \{L, M, N\}$ represents the direction vector of the line and $\overline{T} = \{P, Q, R\}$ represents the moment of the line with respect to the origin as shown in Figure
And P, Q, R can be calculated by:

$$
\begin{aligned}
P &= y_1N - z_1N \\
Q &= z_1L - x_1N \qquad (A.5) \\
R &= x_1M - y_1L
\end{aligned}
$$

Because a straight line in space has only four independent parameters, only four of L, M, N, P, Q, and R are independent and they have the following relationships,

$$L^2 + M^2 + N^2 = 1 \qquad (A.6)$$

$$LP + MQ + NR = 0 \qquad (A.7)$$

or

$$\overline{U} \cdot \overline{T} = 0$$

If $\{L_1, M_1, N_1, P_1, Q_1, R_1\}$ and $\{L_2, M_2, N_2, P_2, Q_2, R_2\}$ represent the plücker coordinates of line ℓ_1 and ℓ_2 respectively. The condition for these two lines to be coplanar can be written as [43],

$$L_1P_2 + M_1Q_2 + N_1R_2 + L_2P_1 + M_2Q_1 + N_2R_1 = 0 \tag{A.8}$$

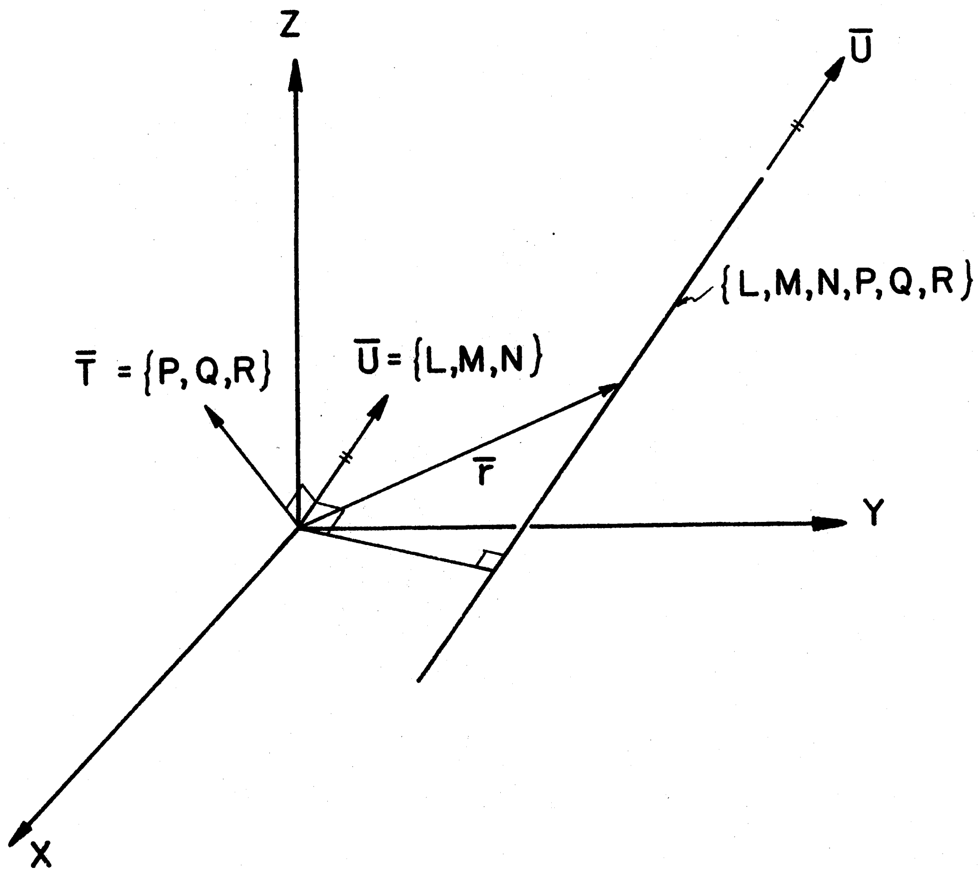


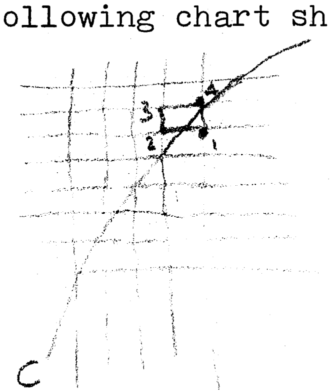
Figure 37. Physical Interpretation of Plucker's Line Coordinates

APPEXDIX B

CORDRAY'S CONTOURING METHOD

Let two adjacent meshpoints on opposite sides of a contour be known. Number these points 1 and 4, and number meshpoints 2 and 3 so that the numbered points are in order going clockwise around a unit square of the mesh. Next compute point 2 or point 3 as sign S is plus or minus. S is arbitrary to begin, but in general is determined by the location at which the contour C left the previous unit square of the mesh. S is plus if C left the previous unit square on side 3-4, minus if it left on side 1-2, and the same as on entering the previous unit square if C left on side 2-3.

If C leaves by side 1-2 when S is plus, or by side 3-4 when S is minus, the computation of an unwanted point (3 or 2 respectively) is avoided. When C leaves by side 2-3 both points 2 and 3 are wanted. The following chart shows the details of the procedure.



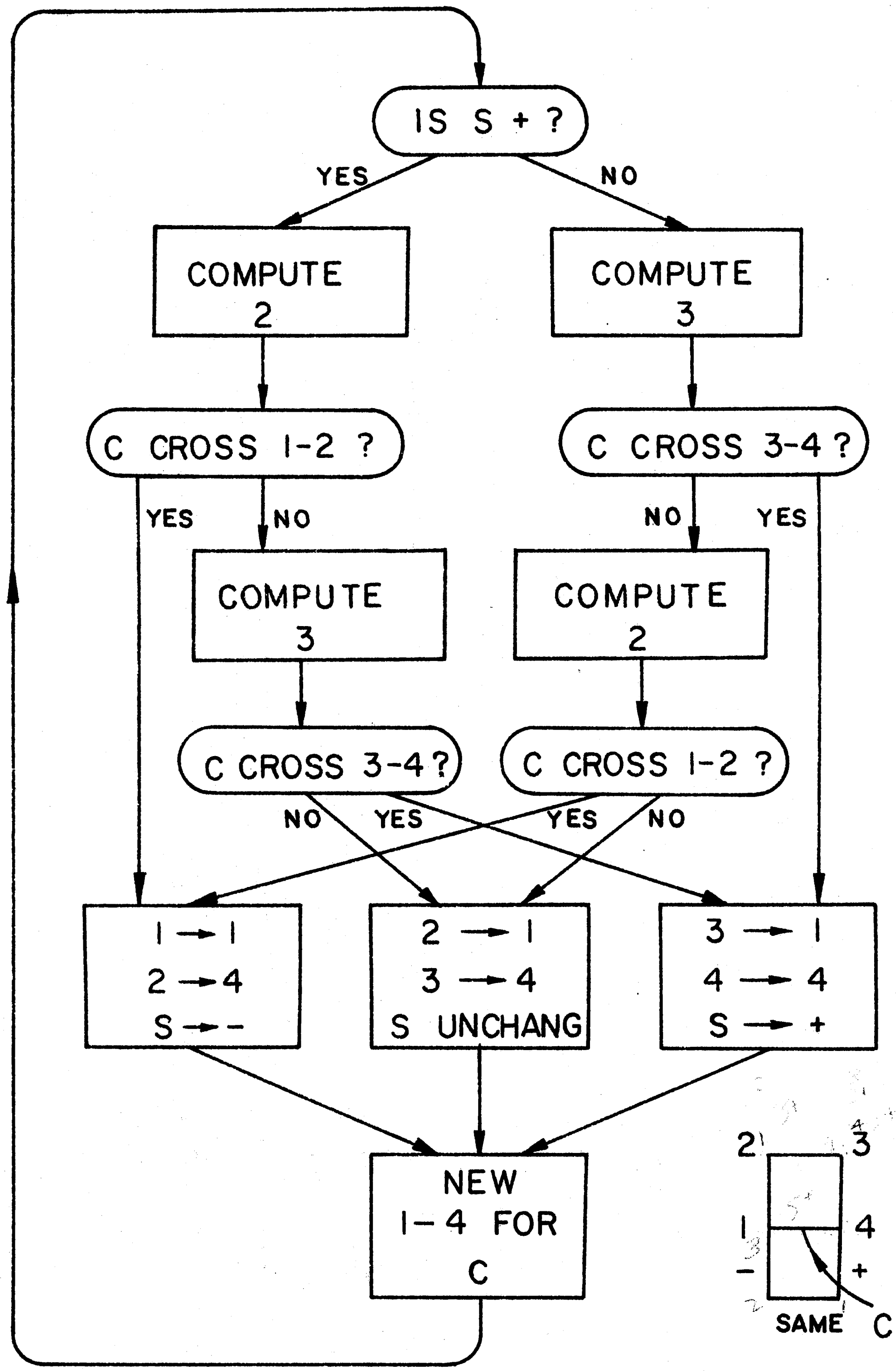


Figure 38. Flow Chart of Cordray's Method

APPENDIX C

COMPUTER PROGRAMS IN CHAPTER II

```
C=====================================================================
C
C    THIS IS AN INTERACTIVE PROGRAM TO PLOT THE ACCESSIBLE WORKSPACE OF
C    A GENERAL ROBOT WITH N-REVOLUTE JOINTS  ON AN ARBITRARILY SPECIFIED
C    PLANE IN SPACE.  THE ROBOT HAND CAN BE TREATED AS A POINT, A LINE
C    OR A RIGID-BODY.  THIS POGRAM CAN ALSO BE SUBMITTED AS BATCH JOB.
C
C    SET TERMINAL LINESIZE(132) IF YOU USE DECKWRITER TERMINAL.
C
C    WHEN THIS PROGRAM IS SUBMITTED AS A BATCH JOB, ONE SHOULD SUPPLY
C    THE DATA CARDS IN THE FOLLOWING ORDER:
C
C    1.     N: THE NUMBER OF JOINTS OF THE ROBOT.
C    2.     KINK LENGTH A(I), I=1,N
C    3.     LINK LENGTH S(I), I=1,N
C    4.     TWIST ANGLE ALFA(I), I=1,N
C    5.     MOTION RANGE OF JOINT 1.   (THETA 1)MIN  THEN  (THETA 1)MAX
C    6.     MOTION RANGE OF JOINT 2.   (THETA 2)MIN  THEN  (THETA 2)MAX
C    .
C    .
C    4+N.   MOTION RANGE OF JOINT N.   (THETA N)MIN  THEN  (THETA N)MAX
C    5+N.   XO, YO, ZO: THE COORDINATES OF THE ORIGIN OF THE REFERENCE
C              FRAME ATTACHED TO THE SPECIFIED PLANE ON WHICH THE WORK-
C              ING SPACE WILL BE PLOTTED.
C    6+N.   LZ, MZ, NZ: THE UNIT VECTOR NORMAL TO THE SPECIFICED PLANE.
C    7+N.   LX, MX, NX: THE UNIT VECTOR OF X AXIS OF THE REFERENCE FRAME
C              ON THE SPECIFIED PLANE.
C    8+N.   IHANDT =  0   IF THE ROBOT HAND IS TREATED AS A POINT.
C                     2     "   "   "   "   "   "   "   " A LINE.
C                     3     "   "   "   "   "   "   "   " A RIGID-BODY.
C    9+N.   L3: THE DIRECTION COSINE BETWEEN X AXIS OF ROBOT HAND AND Z
C               AXIS OF THE REFERENCE FRAME ON THE SPECIFIED PLANE.
C   10+N.   M1: THE DIRECTION COSINE BETWEEN Y AXIS OF ROBOT HAND AND X
C               AXIS OF THE REFERENCE FRAME ON THE SPECIFIED PLANE.
C   11+N.   N3: THE DIRECTION COSINE BETWEEN Z AXIS OF ROBOT HAND AND Z
C               AXIS OF THE REFERENCE FRAME ON THE SPECIFIED PLANE.
C**NOTE:    IF IHANDT=0 , OMIT DATA CARDS 9+N, 10+N, AND 11+N.
C           IF IHANDT=2 , OMIT DATA CARD 11+N.
C   12+N.   ISTEP: THE MAXIMUM ALLOWABLE NUMBER OF STEPS IN POLTTING
C                  THE CONTOUR OF THE WORKSPACE.
C   13+N.   PLOT SIZE: ENTER  7,9  FOR 8 LINES PER INCH PRINTER.
C                             5,9   "  6   "     "    "     "
C                             3,5  FOR CRT TERMINAL.
C   14+N.   ENTER  1 : PLOT THE CONTOUR IN C.C.W. DIRECTION.
C                  2 : PLOT THE CONTOUR IN C.C.W DIRECTION FIRST,
C                      THEN PLOT IT IN C.W. DIRECTION.
C                         (IF YOU HAVE NO IDEA ENTER 1)
C   15+N.   ENTER  0 : FOR NORMAL PLOTTING.
C                  1 : MAKE THE PLOT SYMATRIC ABOUT THE X AXIS.
C                         (IF YOU HAVE NO IDEA ENTER 0)
C   16+N.   ENTER THE COEFFICIENT A AND B OF THE OBJECTIVE FUNCTION
C              FOR PLOTING THE CONTOURE OF WORKSPACE.  THEY CAN BE
C              0.0, 1.0, OR -1.0
C              (IF YOU HAVE NO IDEA, ENTER  1.0, 1.0 OR ENTER -1.0, 1.0)
C
C=====================================================================
C
C
      DIMENSION A(8),S(8),ALFA(8),THETA(8),THETAU(8),THETAL(8),
     &DTHETA(15),AO(4,4),AON(4,4),AI(4,4),AW(4,4),BW(4,4),CW(4,4),
     &AW4(4,4,2,8),AINC(4,4,8),XO(3),VNOR(3),VX(3),THETA1(8),THETA2(8),
     &XN(4),VA(4),VS(4),ALP(15,6),BLP(15),CLP(15),DXUPP(8),
     &DXLOW(8),DIRR(3),POSR(3),DSOL(15),RW(263),IW(38),
     &PLOTXI(10),PLOTYI(10),PLOTXO(10),PLOTYO(10),
     &IMAGE(9000),IMOVE(4),OTHETA(10,8),IODIR(10)
C
      DATA BCD/'*    '/
      DATA OTHETA/80*0.0/,IODIR/10*0/,EPSM/0.0001 /,TEST3/0.025/
```

```
      DATA XYINC/0.002/,ISTEP/ 3000/,IYPLOT,IXPLOT/7,9/,MAXTIM/300/
      DATA IWRIT1,IWRIT2,IWRIT3/0,0,0/,NOPLOT/1/,IFLIP/0/
      DATA ADIRCH, AANGCH/0.1E-02, 1.0/
C
C...READ THE INPUT DATA
C
    5 FORMAT(/8X,4G15.5)
      WRITE(6,10)
   10 FORMAT(/1X,'ENTER THE NUMBER OF JOINT OF THE ROBOT')
      READ(5,*) N
      WRITE(6,5) N
      WRITE(6,15) N
   15 FORMAT(/1X,'ENTER',1X,I2,2X,'KINK LINKLENGTH A(I)')
      READ(5,*) (A(I),I=1,N)
      WRITE(6,5) (A(I),I=1,N)
      WRITE(6,20) N
   20 FORMAT(/1X,'ENTER',1X,I2,2X,'LINK LENGTH S(I)')
      READ(5,*) (S(I),I=1,N)
      WRITE(6,5) (S(I),I=1,N)
      WRITE(6,25) N
   25 FORMAT(/1X,'ENTER',1X,I2,2X,'TWIST ANGLE ALFA(I)')
      READ(5,*) (ALFA(I), I=1,N)
      WRITE(6,5) (ALFA(I),I=1,N)
      DO 30 I=1,N
      WRITE(6,35) I
      READ(5,*) THETAL(I),THETAU(I)
      WRITE(6,5) THETAL(I), THETAU(I)
   30 CONTINUE
   35 FORMAT(/1X,'ENTER THE MOTION RANGE OF THE JOINT NUMBER ', I2,/
     *        1X,'VALUE OF (THETA)MIN THEN VALUE OF (THETA)MAX')
      WRITE(6,40)
   40 FORMAT(/1X,'ENTER THE X,Y,Z COORDINATES OF THE ORIGIN OF THE'/
     *1X,'REFERENCE FRAME ATTACHED TO THE SPECIFIED PLANE ON WHICH'/
     *1X,'YOU ARE GOING TO PLOT THE WORK SPACE OF THE GIVEN ROBOT')
      READ(5,*) (XO(I),I=1,3)
      WRITE(6,5) (XO(I),I=1,3)
      WRITE(6,45)
   45 FORMAT(/1X,'ENTER THREE COMPONENTS OF THE UNIT VECTOR OF'/
     *1X,'Z AXIS OF THE REFERENCE FRAME WHICH IS NORMAL TO THE '/
     *1X,'SPECIFIED PLANE ON WHICH YOU WANT TO PLOT THE WORKSPACE')
      READ(5,*) (VNOR(I),I=1,3)
      WRITE(6,5) (VNOR(I),I=1,3)
      WRITE(6,55)
   55 FORMAT(/1X,'ENTER THREE COMPONENTS OF THE UNIT VECTOR OF'/
     *1X,'X AXIS OF THE REFERENCE FRAME WHICH IS ON THE'/
     *1X,'SPECIFIED PLANE ON WHICH YOU WANT TO PLOT THE WORKSPACE')
      READ(5,*) (  VX(I),I=1,3)
      WRITE(6,5) ( VX(I),I=1,3)
      WRITE(6,60)
   60 FORMAT(/1X,'CHOOSE ONE OF THE FOLLOWING NUMBERS'/
     *1X,'0 : TREAT THE ROBOT HAND AS A POINT'/
     *1X,'2 : TREAT THE ROBOT HAND AS A LINE'/
     *1X,'3 : TREAT THE ROBOT HAND AS A RIGID-BODY')
      READ(5,*) IHANDT
      WRITE(6,5) IHANDT
      IF(IHANDT.EQ.0) GO TO 82
      WRITE(6,65)
   65 FORMAT(/1X,'ENTER THE FOLLOEING DATA:'/
     *1X,'THE DIRECTION OF BETWEEN X AXIS OF ROBOT HAND AND Z AXIS OF'/
     *1X,'    THE REFERENCE FRAME ATTACHED ON THE SPECIFIED PLANE')
      READ(5,*) DIRR(1)
      WRITE(6,5) DIRR(1)
      WRITE(6,75)
   75 FORMAT(/1X,'ENTER THE FOLLOWING DATA:'/
     *1X,'THE DIRECTION OF BETWEEN Y AXIS OF ROBOT HAND AND X AXIS OF'/
     *1X,'    THE REFERENCE FRAME ATTACHED ON THE SPECIFIED PLANE')
      READ(5,*) DIRR(2)
      WRITE(6,5) DIRR(2)
```

```
      IF(IHANDT.EQ.2) GO TO 82
      WRITE(6,80)
   80 FORMAT(/1X,'ENTER THE FOLLOWING DATA:'/
     *1X,'THE DIRECTION OF BETWEEN Z AXIS OF ROBOT HAND AND Z AXIS OF'/
     *1X,'    THE REFERENCE FRAME ATTACHED ON THE SPECIFIED PLANE')
      READ(5,*) DIRR(3)
      WRITE(6,5) DIRR(3)
   82 WRITE(6,85)
   85 FORMAT(/1X,'ENTER THE MAX. ALLOWABLE NUMBER OF STEPS IN'/
     *1X,'PLOTTING THE CONTOUR')
      READ(5,*) ISTEP
      WRITE(6,5) ISTEP
      WRITE(6,90)
   90 FORMAT(/1X,'ENTER  7,9   FOR A 8 LINES PER INCH PRINTER'/
     &        1X,'       5,9   FOR A 6 LINES PER INCH PRINTER'/
     &        1X,'       5,5   FOR CRT TERMINAL')
      READ(5,*) IYPLOT, IXPLOT
      WRITE(6,5) IYPLOT, IXPLOT
      WRITE(6,95)
   95 FORMAT(/1X,'ENTER  1 : PLOT THE CONTOUR IN C.C.W. DIRECTION.'/
     &        1X,'       2 : PLOT THE CONTOUR IN C.C.W THEN C.W.'/
     &1X,'IF YOU HAVE NO IDEA JUST ENTER1')
      READ(5,*) NOPLOT
      WRITE(6,5) NOPLOT
      WRITE(6,96)
   96 FORMAT(/1X,'ENTER  0 : FOR NORMAL PLOTTING.'/
     &        1X,'       1 : TO MAKE THE PLOT SYMATRIC ABOUT X AXIS'/
     &1X,'IF YOU HAVE NO IDEA JUST ENTER 0')
      READ(5,*) IFLIP
      WRITE(6,5) IFLIP
      WRITE(6,98)
   98 FORMAT(/1X,'ENTER COEFFICIENT A AND B OF THE OBJECTIVE FUNCTION'
     &/1X,'THEY CAN BE  0.0,  1.0,  OR -1.0'
     &/1X,'IF YOU HAVE NO IDEA, ENTER  1.0, 1.0  OR ENTER -1.0, 1.0')
      READ(5,*) COEFX, COEFY
      WRITE(6,5) COEFX,COEFY
C=========================================================================
C
C   MOVE THE ROBOT HAND ON TO THE SPECIFIED PLANE.
C=========================================================================
C
C...INITIAL SET UP AND NORMALIZE THE LINK LENGTH
C
      TLENG=0.0
      DO 100 I=1,N
      TLENG=TLENG + A(I) + S(I)
  100 CONTINUE
      DO 105 I=1,N
      A(I)=A(I)/TLENG
      S(I)=S(I)/TLENG
  105 CONTINUE
      IA=15
      DTR=3.141593/180.0
      DIRCH=0.05
      ANGCHO=20.0
      ICHS=0
      DPOS=0.0
      ICOUNT=0
      ANGCHI=ANGCHO
      DIRCH1=5.0*ADIRCH
      IF(IHANDT.EQ.0) ANGCHI=ANGCHO/2.0
      ITIME1=-1
      CALL TRANAO(XO,VNOR,VX,AO,IERR)
      IF(IERR.EQ.1) STOP
C
C...SET EVERY JOINT AT ITS MIDDLE POSITION OF MOTION RANGE
C
```

```
      DO 110 I=1,N
  110 THETA(I)=(THETAU(I) + THETAL(I) )*0.5
  115 ITIME1=ITIME1+1
      CALL POSINR(N,A,S,ALFA,THETA,AO,AON,AI,AW,XN,VA,VS)
      VM1=VS(2)*VA(3) - VA(2)*VS(3)
      IF(IWRIT1.EQ.0) GO TO 116
      WRITE(6,270)
      WRITE(6,275) (XN(I),I=1,3)
      WRITE(6,265) VA(3),VM1,VS(3)
      WRITE(6,280)((I,THETA(I)),I=1,N)
      WRITE(6,295) ITIME1
  116 IF(ITIME1.GE.MAXTIM) GO TO 170
C
C...COMPUTE THE REQUIRED INCREAMENT OF THE DISPLACEMENT IN THE Z
C   DIRECTION
C
      DPOSO=DPOS
      DPOS= -XN(3)
      IF((DPOSO*DPOS) .LT. 0.0) ICHS=1
C
C...COMPUTE THE REQUIRED CHANGES OF THE DIRECTION OF THE ROBOT HAND
C
      DL3=0.0
      DM1=DL3
      DN3=DL3
      IF(IHANDT.EQ.0) GO TO 120
      DL3=DIRR(1) - VA(3)
      IF(ABS(DL3).GT.DIRCH) DL3=SIGN(DIRCH,DL3)
      IF(IHANDT.EQ.1) GO TO 120
      DM1=DIRR(2) - VM1
      IF(ABS(DM1).GT.DIRCH) DM1=SIGN(DIRCH,DM1)
      IF(IHANDT.EQ.2) GO TO 120
      DN3=DIRR(3) - VS(3)
      IF(ABS(DN3).GT.DIRCH) DN3=SIGN(DIRCH,DN3)
  120 CONTINUE
C
C...COMPUTE THE SMALL CHANGE OF THE TRANSFORMATION MATRIX DUE TO THE
C   SMALL CHANGES OF JOINT DISPLACEMENTS
C
      CALL INCRA(N,A,S,ALFA,THETA,AO,AINC,AW4,AW,BW,CW,IER)
C
C...COMPUTE THE UPPER AND LOWER BOUNDS OF THE JOINT DISPLACEMENTS
C   FOR THE LINEAR PROGRAMMING PROBLEM
C
      DO 125 I=1,N
      DXUPP(I)=THETAU(I) - THETA(I)
      DXLOW(I)=THETA(I)  - THETAL(I)
      IF(DXUPP(I).GT.ANGCH) DXUPP(I)=ANGCH
      IF(DXLOW(I).GT.ANGCH) DXLOW(I)=ANGCH
      BLP(I)=DXUPP(I) + DXLOW(I)
  125 CONTINUE
C
C...COMPUTE THE COEFFICIENT MATRIX " A" AND "B" OF THE LINEAR PROG.
C
      DO 130 I=1,N
      DO 130 J=1,N
      ALP(I,J)=0.0
      IF(I.EQ.J) ALP(I,J)=1.0
  130 CONTINUE
      IF(IHANDT.EQ.0) GO TO 150
C
C...COEFFICIENT OF L3
C
      K=N+1
      BLP(K)=DL3/DTR
      DO 135 I=1,N
      ALP(K,I)=AINC(3,1,I)
      BLP(K)=BLP(K) + ALP(K,I)*DXLOW(I)
```

```
 135 CONTINUE
     IF (IHANDT.EQ.1) GO TO 150
C
C...COEFFICENT OF DELT M1
C
     K=N+2
     BLP(K)=DM1/DTR
     DO 136 I=1,N
     ALP(K,I)=AINC(1,2,I)
     BLP(K)=BLP(K)+ ALP(K,I)*DXLOW(I)
 136 CONTINUE
     IF(IHANDT.EQ.2) GO TO 150
C
C...COEFFICIENT OF DELTA N3
C
     K=N+3
     BLP(K)= DN3/DTR
     DO 137 I=1,N
     ALP(K,I)=AINC(3,3,I)
     BLP(K)=BLP(K) + ALP(K,I)*DXLOW(I)
 137 CONTINUE
 150 CONTINUE
C
C...COMPUTE THE COEFFICIENT OF "C"
C
     DZCORR=0.0
     SI=SIGN(1.0,DPOS)
     DO 155 I=1,N
     CLP(I)=AINC(3,4,I)*SI
     DZCORR= DZCORR + CLP(I)*DXLOW(I)
 155 CONTINUE
C
C...CALL THE LINEAR PROGRAMMING SUBROUTINE ZX3LP1
C
     CALL ZX3LP1(ALP,IA,BLP,CLP,N,N,IHANDT,DZ,DTHETA,DSOL,RW,IW,IER)
     IF(IER.EQ.133) ANGCHI=1.5*ANGCHI
     IF(IER.EQ.0) ICOUNT=ICOUNT + 1
     IFACT=1 + ICOUNT/10
     ANGCH=ANGCHI/FLOAT(IFACT)
     IF(ANGCH.LT.AANGCH) ANGCH=AANGCH
     DO 160 I=1,N
 160 THETA(I)=THETA(I) + DTHETA(I) - DXLOW(I)
     DZ=(DZ - DZCORR)*DTR
     IF(ICHS.EQ.0) GO TO 115
     IF(ANGCH.GT.5.0.AND.IHANDT.EQ.0) GO TO 115
     IF(ABS(XN(3)+DZ).GT.DIRCH1) GO TO 115
     IF(ABS(DL3).GT.DIRCH1) GO TO 115
     IF(ABS(DM1).GT.DIRCH1) GO TO 115
     IF(ABS(DN3).GT.DIRCH1) GO TO 115
 170 WRITE(6,295) ITIME1
C
C=====================================================================
C
C   MOVE THE ROBOT HAND ON THE SPECIFIED PLANE TOWARD ITS BOUNDARY.
C=====================================================================
C
 200 SI=1.0
     ICOUNT=0
     ANGCHI=ANGCHO/2.0
     ANGCH=ANGCHI
     CALL POSINR(N,A,S,ALFA,THETA,AO,AON,AI,AW,XN,VA,VS)
     YLAST=XN(2)
 210 ITIME2=0
 215 ITIME2=ITIME2+1
     CALL POSINR(N,A,S,ALFA,THETA,AO,AON,AI,AW,XN,VA,VS)
     VM1=VS(2)*VA(3) - VA(2)*VS(3)
     IF(IWRIT2.EQ.0) GO TO 298
 262 WRITE(6,263)
 263 FORMAT(///1X,'REQUIRED HAND POSITION')
```

```
      WRITE(6,265) (DIRR(I),I=1,3)
  265 FORMAT(//1X,'L3 =',G13.6,5X,'M1 =',G13.6,5X,'N3 =',G13.6)
      WRITE(6,270)
  270 FORMAT(////1X,'FINAL HAND POSITION')
      WRITE(6,275)(XN(I), I=1,3)
  275 FORMAT(//1X,' X =',G13.6,5X,' Y =',G13.6,5X,' Z =',G13.6)
      WRITE(6,265) VA(3),VM1,VS(3)
      WRITE(6,280) ((I,THETA(I)), I=1,N)
  280 FORMAT(//1X, 'THETA('  ,I1,  ') ='  , G13.6)
      WRITE(6,295) ITIME2
  295 FORMAT(///1X,'NUMBER OF ITERATION =', I6,////)
  298 IF(ITIME2.LT.MAXTIM) GO TO 350
      WRITE(6,300) MAXTIM
  300 FORMAT(1X,'NEED TO SET MAXTIM MORE THAN',I5 )
      STOP
C
C...THE REQUIRED CORRECTION IN THE DISPLACEMENT OF
C...THE Y ZND Z DIRECTIONS
C
  350 DY=YLAST - XN(2)
      DZ= -XN(3)
C
C...THE REQUIRED CHANGES OF THE DIRECTION OF THE ROBOT HAND
C
      DL3=0.0
      DM1=DL3
      DN3=DL3
      IF(IHANDT.EQ.0) GO TO 420
      DL3=DIRR(1) - VA(3)
      IF(ABS(DL3).GT.ADIRCH) DL3=SIGN(ADIRCH,DL3)
      IF(IHANDT.EQ.1) GO TO 420
      DM1=DIRR(2) - VM1
      IF(ABS(DM1).GT.ADIRCH) DM1=SIGN(ADIRCH,DM1)
      IF(IHANDT.EQ.2) GO TO 420
      DN3=DIRR(3) - VS(3)
      IF(ABS(DN3).GT.ADIRCH) DN3=SIGN(ADIRCH,DN3)
  420 CONTINUE
C
C...COMPUTE THE SMALL CHANGE OF THE TRANSFORMATION MATRIX DUE TO THE
C   SMALL CHANGES OF JOINT DISPLACEMENTS
C
      CALL INCRA(N,A,S,ALFA,THETA,AO,AINC,AW4,AW,BW,CW,IER)
      IF(IWRIT.EQ.0) GO TO 424
      DO 422 II=1,N
  422 WRITE(6,423) II,((AINC(JJ,KK,II),KK=1,4),JJ=1,4)
  423 FORMAT(//1X,'AINC(',I3,')', 4(/1X,4G15.5))
C
C...COMPUTE THE UPPER AND LOWER BOUNDS OF THE JOINT DISPLACEMENTS
C    FOR THE LINEAR PROGRAMMING PROBLEM
C
 424  CONTINUE
      DO 425 I=1,N
      DXUPP(I)=THETAU(I) - THETA(I)
      DXLOW(I)=THETA(I)  - THETAL(I)
      IF(DXUPP(I).GT.ANGCH) DXUPP(I)=ANGCH
      IF(DXLOW(I).GT.ANGCH) DXLOW(I)=ANGCH
      BLP(I)=DXUPP(I) + DXLOW(I)
  425 CONTINUE
C
C...COMPUTE THE COEFFICIENT MATRIX " A" AND "B" OF THE LINEAR PROG.
C
      DO 430 I=1,N
      DO 430 J=1,N
      ALP(I,J)=0.0
      IF(I.EQ.J) ALP(I,J)=1.0
  430 CONTINUE
C
C...COEFFICIENT OF DY
```

```
C
      K=N+1
      BLP(K)=DY/DTR
      DO 432 I=1,N
      ALP(K,I)=AINC(2,4,I)
      BLP(K)=BLP(K) + ALP(K,I)*DXLOW(I)
  432 CONTINUE
C
C...COEFFICIENT OF L2
C
      K=N+2
      BLP(K)=DZ/DTR
      DO 433 I=1,N
      ALP(K,I)=AINC(3,4,I)
      BLP(K)=BLP(K) + ALP(K,I)*DXLOW(I)
  433 CONTINUE
      IF(IHANDT.EQ.0) GO TO 450
C
C...COEFFICIENT OF L3
C
      K=N+3
      BLP(K)=DL3/DTR
      DO 435 I=1,N
      ALP(K,I)=AINC(3,1,I)
      BLP(K)=BLP(K) + ALP(K,I)*DXLOW(I)
  435 CONTINUE
      IF (IHANDT.EQ.1) GO TO 450
C
C...COEFFICENT OF DELT M1
C
      K=N+4
      BLP(K)=DM1/DTR
      DO 436 I=1,N
      ALP(K,I)=AINC(1,2,I)
      BLP(K)=BLP(K)+ ALP(K,I)*DXLOW(I)
  436 CONTINUE
      IF(IHANDT.EQ.2) GO TO 450
C
C...COEFFICIENT OF DELTA N3
C
      K=N+5
      BLP(K)= DN3/DTR
      DO 437 I=1,N
      ALP(K,I)=AINC(3,3,I)
      BLP(K)=BLP(K) + ALP(K,I)*DXLOW(I)
  437 CONTINUE
  450 CONTINUE
C
C...COMPUTE THE COEFFICIENT OF "C"
C
      DXCORR=0.0
      DO 455 I=1,N
      CLP(I)=AINC(1,4,I)*SI
      DXCORR= DXCORR + CLP(I)*DXLOW(I)
  455 CONTINUE
C
C...CALL THE LINEAR PROGRAMMING SUBROUTINE ZX3LP1
C
      IF(IWRIT.EQ.0) GO TO 525
      WRITE(6,470) (DXLOW(II),II=1,N)
  470 FORMAT(//1X,'DXLOW', 4(/1X,6G15.5))
      WRITE(6,500) (  CLP(II),II=1,N)
  500 FORMAT(//1X,'CLP  ', 4(/1X,6G15.5))
      K=N+IHANDT + 2
      WRITE(6,510) ((ALP(II,JJ),JJ=1,N),II=1,K)
  510 FORMAT(//1X,' ALP ',15(/1X,6G15.5))
      WRITE(6,520) (  BLP(II),II=1,K)
  520 FORMAT(//1X,' BLP ',15(/1X,6G15.5))
```

```
  525 M2=IHANDT + 2
      CALL ZX3LP1(ALP,IA,BLP,CLP,N,N,M2,DX,DTHETA,DSOL,RW,IW,IER)
      IF(IER.EQ.133) GO TO 570
      DX=(DX - DXCORR)*DTR
      IF(IWRIT2.EQ.0) GO TO 550
      WRITE(6,540) DX
  540 FORMAT(///1X,'DX =' , G15.5)
  550 IF(DX.LT.0.0) GO TO 570
      IF(IER.EQ.0) ICOUNT=ICOUNT + 1
      IFACT=1 + ICOUNT/10
      ANGCH=ANGCHI/FLOAT(IFACT)
      IF(ANGCH.LT.AANGCH) ANGCH=AANGCH
      IF(ABS(DZ).LT.0.001.AND.ANGCH.GT.5.0) ANGCH=5.0
      DO 555 I=1,N
      THETA(I)=THETA(I) + DTHETA(I) - DXLOW(I)
      IF(IER.EQ.0) THETA1(I)=THETA(I)
  555 CONTINUE
      DO 560 I=1,N
      IF(ABS(DTHETA(I)).GE.DXLOW(I).AND.ABS(DTHETA(I)).GE.DXUPP(I))
     *  GO TO 215
  560 CONTINUE
  570 IF(ITIME2.GT.1) GO TO 580
      ADIRCH=ADIRCH*0.75
      IF(ADIRCH.GE.EPSM) GO TO 210
      WRITE(6,575)
  575 FORMAT(1X,'NO FEASIBLE SOLUTION EXISTS')
      STOP
  580 WRITE(6,295) ITIME2
      DO  585 I=1,N
  585 THETA1(I)=THETA(I)
C
C======================================================================
C
C    PLOT THE CONTOUR OF THE WORKSPACE OF THE ROBOT
C======================================================================
C
      CALL PLOT1(0,11,IYPLOT,11,IXPLOT)
      CALL PLOT2(IMAGE,1.0,-1.0,1.0,-1.0)
      CCW=1.0
      NOPT=0
      DO 587 I=1,N
  587 OTHETA(1,I)=THETA1(I)
      XYINCO=XYINC
  590 ITIME=1
      IDIR=1
      IS=1
      ANGCH=2.5
      CALL POSINR(N,A,S,ALFA,THETA1,AO,AON,AI,AW,XN,VA,VS)
      PLOTXI(1)=XN(1)
      PLOTYI(1)=XN(2)
      PLOTXO(1)=XN(1) + XYINC
      PLOTYO(1)=XN(2)
      PLOTXI(10)=PLOTXI(1)
      PLOTYI(10)=PLOTYI(1)
      PLOTXO(10)=PLOTXO(1)
      PLOTYO(10)=PLOTYO(1)
      IF(IWRIT3.EQ.0) GO TO 595
      WRITE(6,593)
  593 FORMAT(///T5,'PLOTXI(I)',T20,'PLOTYI(I)',T35,'PLOTXO(I)',
     &T50,'PLOTYO(I)',T67,'IDIR',T75,'ITIME'//)
      WRITE(6,950) PLOTXI(1),PLOTYI(1),PLOTXO(1),PLOTYO(1),IDIR
  595 DO 598 I=1,4
  598 IMOVE(I)=0
  600 ITEST=0
      CALL INCRA(N,A,S,ALFA,THETA1,AO,AINC,AW4,AW,BW,CW,IER)
      IF(IS.EQ.1) ICOR=2
      IF(IS.NE.1) ICOR=3
  605 ITEST=ITEST + 1
```

```
      DZ=-XN(3)
      GO TO (610, 620, 630, 640),IDIR
 610  DY=XYINC*CCW
      DX=0.0
      IF(ICOR.EQ.3) DX=XYINC
      GO TO 650
 620  DX=-XYINC
      DY=0.0
      IF(ICOR.EQ.3) DY= XYINC*CCW
      GO TO 650
 630  DY=-XYINC*CCW
      DX=0.0
      IF(ICOR.EQ.3) DX=-XYINC
      GO TO 650
 640  DX=XYINC
      DY=0.0
      IF(ICOR.EQ.3) DY=-XYINC*CCW
C
C...COMPUTE THE UPPER AND LOWER BOUNDS OF THE JOINT DISPLACEMENTS
C    FOR THE LINEAR PROGRAMMING PROBLEM
C
 650  CONTINUE
      DO 660 I=1,N
      DXUPP(I)=THETAU(I) - THETA1(I)
      DXLOW(I)=THETA1(I)  - THETAL(I)
      IF(DXUPP(I).GT.ANGCH) DXUPP(I)=ANGCH
      IF(DXLOW(I).GT.ANGCH) DXLOW(I)=ANGCH
      BLP(I)=DXUPP(I) + DXLOW(I)
  660 CONTINUE
C
C...COMPUTE THE COEFFICIENT MATRIX " A" AND "B" OF THE LINEAR PROG.
C
      DO 665 I=1,N
      DO 665 J=1,N
      ALP(I,J)=0.0
      IF(I.EQ.J) ALP(I,J)=1.0
  665 CONTINUE
C
C...COFFICIENT OF DX
C
      K=N+1
      BLP(K)=DX/DTR
      DO 668 I=1,N
      ALP(K,I)=AINC(1,4,I)
      BLP(K)=BLP(K)+ALP(K,I)*DXLOW(I)
 668  CONTINUE
C
C...COEFFICIENT OF DY
C
      K=N+2
      BLP(K)=DY/DTR
      DO 670 I=1,N
      ALP(K,I)=AINC(2,4,I)
      BLP(K)=BLP(K) + ALP(K,I)*DXLOW(I)
  670 CONTINUE
C
C...COEFFICIENT OF DZ
C
      K=N+3
      BLP(K)=DZ/DTR
      DO 675 I=1,N
      ALP(K,I)=AINC(3,4,I)
      BLP(K)=BLP(K) + ALP(K,I)*DXLOW(I)
  675 CONTINUE
      IF(IHANDT.EQ.0) GO TO 690
C
C...COEFFICIENT OF L3
C
```

```
      K=N+4
      DL3=DIRR(1) - VA(3)
      BLP(K)=DL3/DTR
      DO 680 I=1,N
      ALP(K,I)=AINC(3,1,I)
      BLP(K)=BLP(K) + ALP(K,I)*DXLOW(I)
  680 CONTINUE
      IF (IHANDT.EQ.1) GO TO 690
C
C...COEFFICENT OF DELT M1
C
      K=N+5
      VM1=VS(2)*VA(3) - VA(2)*VS(3)
      DM1=DIRR(2) - VM1
      BLP(K)=DM1/DTR
      DO 685 I=1,N
      ALP(K,I)=AINC(1,2,I)
      BLP(K)=BLP(K)+ ALP(K,I)*DXLOW(I)
  685 CONTINUE
      IF(IHANDT.EQ.2) GO TO 690
C
C...COEFFICIENT OF DELTA N3
C
      K=N+6
      DN3=DIRR(3) - VS(3)
      BLP(K)= DN3/DTR
      DO 688 I=1,N
      ALP(K,I)=AINC(3,3,I)
      BLP(K)=BLP(K) + ALP(K,I)*DXLOW(I)
  688 CONTINUE
  690 CONTINUE
C
C...COMPUTE THE COEFFICIENT OF "C"
C
      KX=N+1
      KY=N+2
      DO 700 I=1,N
      CLP(I)= ALP(KX,I)*COEFX + ALP(KY,I)*COEFY
 700  CONTINUE
C
C...CALL THE LINEAR PROGRAMMING SUBROUTINE ZX3LP1
C
      M3=IHANDT+3
      CALL ZX3LP1(ALP,IA,BLP,CLP,N,N,M3,DC,DTHETA,DSOL,RW,IW,IER)
 730  IF(ICOR.NE.2) GO TO 750
      DO 740 I=1,N
 740  THETA2(I)=THETA1(I) + DTHETA(I) - DXLOW(I)
 750  IF(ITEST.EQ.1) GO TO 760
      IF(ICOR.EQ.3.AND.IER.EQ.0) GO TO 830
      IF(ICOR.EQ.2.AND.IER.EQ.133) GO TO 810
      GO TO 820
 760  IF(ICOR.EQ.2.AND.IER.EQ.133) GO TO 810
      IF(ICOR.EQ.2.AND.IER.NE.133)   GO TO 770
      IF(ICOR.EQ.3.AND.IER.NE.133)   GO TO 830
      ICOR=2
      GO TO 605
 770  ICOR=3
      GO TO 605
C
C...CROSS 1-2
C
 810  IDIR=IDIR + 1
      IF(IDIR.GT.4) IDIR=IDIR-4
      PLOTXO(10)=PLOTXI(10) + DX
      PLOTYO(10)=PLOTYI(10) + DY
      IS=-1
      GO TO 900
C
```

```
C...CROSS 2-3
C
 820  CALL POSINR(N,A,S,ALFA,THETA2,AO,AON,AI,AW,XN,VA,VS)
      PLOTXI(10)=XN(1)
      PLOTYI(10)=XN(2)
      GO TO (821,822,823,824),IDIR
 821  DX=XYINC
      DY=0.0
      GO TO 825
 822  DX=0.0
      DY= XYINC*CCW
      GO TO 825
 823  DX=-XYINC
      DY= 0.0
      GO TO 825
 824  DX= 0.0
      DY=-XYINC*CCW
 825  PLOTXO(10)=XN(1) + DX
      PLOTYO(10)=XN(2) + DY
      DO 826 I=1,N
 826  THETA1(I) = THETA2(I)
      GO TO 900
C
C...CROSS 3-4
C
 830  DO 840 I=1,N
 840  THETA1(I)=THETA1(I) + DTHETA(I) - DXLOW(I)
      CALL POSINR(N,A,S,ALFA,THETA1,AO,AON,AI,AW,XN,VA,VS)
      PLOTXI(10)=XN(1)
      PLOTYI(10)=XN(2)
      IS=1
      IDIR=IDIR-1
      IF(IDIR.LT.1) IDIR=IDIR+4
C
C...CHECK IF COMPLETE THE PLOTTING
C
 900  CONTINUE
      CALL PLOT3(BCD,PLOTXI(10),PLOTYI(10),1,4)
      IF(IFLIP .EQ.0) GO TO 920
      CALL PLOT3(BCD,PLOTXI(10),-PLOTYI(10),1,4)
 920  IF(IWRIT3.EQ.0) GO TO 960
      WRITE(6,950) PLOTXI(10),PLOTYI(10),PLOTXO(10),PLOTYO(10),IDIR,
     &                  ITIME
 950  FORMAT(1X,4G15.6,5X,I3,5X,I5)
 960  IF(ITIME.GT.ISTEP ) GO TO 1000
      ITIME=ITIME+1
      DO 970 J=1,N
      DO 965 I=2,9
 965  OTHETA(I,J)=OTHETA(I+1, J)
 970  OTHETA(10,J) = THETA1(J)
      DO 975 I=2,9
      IODIR(I)=IODIR(I+1)
      PLOTXI(I)=PLOTXI(I+1)
      PLOTYI(I)=PLOTYI(I+1)
      PLOTXO(I)=PLOTXO(I+1)
      PLOTYO(I)=PLOTYO(I+1)
 975  CONTINUE
      IODIR(10)=IDIR
      IF(ITIME.LT.500) GO TO 980
      TEST1=ABS(PLOTXI(10) - PLOTXI(1))
      TEST2=ABS(PLOTYI(10) - PLOTYI(1))
      IF(TEST1.LT.TEST3.AND.TEST2.LT.TEST3) GO TO 1000
 980  DO 982 I=1,3
 982  IMOVE(I)=IMOVE(I+1)
      IMOVE(4)=IDIR
      DO 983 I=1,4
      IF(IMOVE(I).NE.I) GO TO 990
 983  CONTINUE
```

```
      IF(ITIME.LT.10) GO TO 987
C...PERMUTATION OF IDIR EQUALS TO  1-2-3-4-1-2-3-4
      DO 985 I=1,N
  985 THETA1(I)=OTHETA(2,I)
      IDIR=IODIR(2)
      PLOTXI(10)=PLOTXI(2)
      PLOTYI(10)=PLOTYI(2)
      PLOTXO(10)=PLOTXO(2)
      PLOTYO(10)=PLOTYO(2)
      IF(XYINC.LT.EPSM) GO TO 1000
  987 XYINC=XYINC*0.75
      GO TO 595
C...PERMUTATION OF IDIR IS NORMAL
  990 DO 995 I=1,4
      IF((IMOVE(I)+I) .NE. 5) GO TO 600
  995 CONTINUE
C...PERMUTATION OF IDIR EQUALS TO  4-3-2-1-4-3-2-1
      XYINC=XYINC*1.25
      GO TO 595
C
C
C... PLOTTING ROUTINE
C
 1000 CONTINUE
      WRITE(6,1010) ITIME
 1010 FORMAT(1H1/////1X,'WORKSPACE OF THE GIVEN ROBOT'
     & ,5X,I10,' STEPS'///)
      CALL PLOT4(6,'Y AXIS')
      WRITE(6,1020)
      WRITE(6,1025) TLENG
 1020 FORMAT(//T60,'X AXIS')
 1025 FORMAT(///T5,'SCALE :    1 UNIT = ', G15.5)
      IF(NOPLOT.EQ.1) STOP
      IF(NOPT.EQ.2) STOP
      CCW=-1.0
      DO 1030 I=1,N
 1030 THETA1(I)=OTHETA(1,I)
      NOPT=2
      XYINC=XYINCO
      COEFY=-1.0
      GO TO 590
      END
C*********************************************************************
C*********************************************************************
C
C
      SUBROUTINE INCRA(N,A,S,ALFA,THETA,AO,AINC,AW4,AW,BW,CW,IER)
C
C.....................................................................
C
C..THIS SUBROUTINE IS FOR COMPUTING THE SMALL INCREMENT OF TRANSFORMAT-
C  ION MATRIX A
C
C  N         :(INPUT) NUMBER OF JOINTS
C  A         :(INPUT) LINK PARAMETER A, A VECTOR OF DIMENSION N
C  S         :(INPUT) LINK PARAMETER S, A VECTOR OF DIMENSION N
C  ALFA      :(INPUT) LINK PARAMETER ALFA (IN DEGREE) , A VECTOR OF
C             DIMENSION N
C  THETA     :(INPUT) LINK PARAMETER THETA (IN DEGREE) , VECTOR OF
C             DIMENSION N
C  AO        :(INPUT) THE TRANSFORMATION MATRIX AO, MATRIX OF DIMENSION
C             4 BY 4
C  AINC      :(OUTPUT) THE SMALL INCREMENT OF TRANSFORMATION   MATRIX
C             ARRY OF DIMENSION 4 BY 4 BY N
C  AW4       :(WORKING SPACE) ARRY OF DIMENSION 4 BY 4 BY 2 BY N
C  AW,BW,CW:(WORKING SPACE) MATRICES OF DIMENSION 4 BY 4
C  IER       :(OUTPUT) ERROR PARAMETER FROM IMSL SUBROUTINE VMULFF
C
```

differential transformation matrices

Δ

$dT = \Delta T$

$dT = T^T \Delta$

```
C.......................................................................
C
      DIMENSION A(1),S(1),ALFA(1),THETA(1)
      DIMENSION AO(4,4),AW(4,4),BW(4,4),CW(4,4)
      DIMENSION AINC(4,4,1), AW4(4,4,2,1)
C
C..INITIAL SET UP
C
      DO 100 I=1,4
      DO 100 J=1,4
      AW4(I,J,1,1)=AO(I,J)
      AW4(I,J,2,N)=0.0
      DO 100 K=1,N
      AINC(I,J,K)= 0.0
  100 CONTINUE
      DO 120 I=1,4
      AW4(I,I,2,N)=1.0
  120 CONTINUE
      IER=0
      DTR=3.14159265/180.0
C
C..COMPUTE THE TRANSFORMATION MATRICES A1,A2,...AN.
C
      DO 200 I=1,N
      SITH=SIN(DTR*THETA(I))
      COTH=COS(DTR*THETA(I))
      SIAL=SIN(DTR*ALFA(I))
      COAL=COS(DTR*ALFA(I))
      AINC(1,1,I)=COTH
      AINC(2,1,I)=SITH
      AINC(1,2,I)=-SITH*COAL
      AINC(2,2,I)= COTH*COAL
      AINC(3,2,I)=SIAL
      AINC(1,3,I)=SITH*SIAL
      AINC(2,3,I)=-COTH*SIAL
      AINC(3,3,I)=COAL
      AINC(1,4,I)=A(I)*COTH
      AINC(2,4,I)=A(I)*SITH
      AINC(3,4,I)=S(I)
      AINC(4,4,I)=1.0
  200 CONTINUE
C
C..COMPUTE THE TRANSFORMATION MATRICES AO*A1, AO*A1*A2, AO*A1*A2*A3,...
C
      N1=N-1
      DO 350 I=1,N1
      DO 310 K=1,4
      DO 310 L=1,4
      BW(K,L)=AINC(K,L,I)
      AW(K,L)=AW4(K,L,1,I)
  310 CONTINUE
      CALL VMULFF(AW,BW,4,4,4,4,4,CW,4,IER)
      IF(IER.NE.O) RETURN 1
      DO 320 K=1,4
      DO 320 L=1,4
      AW4(K,L,1,I+1)= CW(K,L)
  320 CONTINUE
  350 CONTINUE
C
C..COMPUTE THE TRANSFORMATION MATRICES A2*A3*A4*..AN,  A3*A4*..AN,
C  A4*..AN,  ....
C
      DO 450 I=2,N
      M=N-I+2
      DO 410 K=1,4
      DO 410 L=1,4
      AW(K,L)=AW4(K,L,2,M)
      BW(K,L)=AINC(K,L,M)
```

```
  410 CONTINUE
      CALL VMULFF(BW,AW,4,4,4,4,4,CW,4,IER)
      IF(IER.NE.0) RETURN 2
      DO 420 K=1,4
      DO 420 L=1,4
      AW4(K,L,2,M-1)=CW(K,L)
  420 CONTINUE
  450 CONTINUE
C
C..COMPUTE THE SMALL INCREMENT OF THE TRANSFORMATION MATRIX
C
      DO 600 I=1,N
      DO 510 K=1,4
      DO 510 L=1,4
      AW(K,L)=AW4(K,L,1,I)
      BW(K,L)=0.0
  510 CONTINUE
      SITH=SIN(DTR*THETA(I))
      COTH=COS(DTR*THETA(I))
      SIAL=SIN(DTR*ALFA(I))
      COAL=COS(DTR*ALFA(I))
      BW(1,1)=-SITH
      BW(2,1)=COTH
      BW(1,2)=-COTH*COAL
      BW(2,2)=-SITH*COAL
      BW(1,3)=COTH*SIAL
      BW(2,3)=SITH*SIAL
      BW(1,4)=-A(I)*SITH
      BW(2,4)= A(I)*COTH
      CALL VMULFF(AW,BW,4,4,4,4,4,CW,4,IER)
      IF(IER.NE.0) RETURN 3
      DO 520 K=1,4
      DO 520 L=1,4
      BW(K,L)=AW4(K,L,2,I)
  520 CONTINUE
      CALL VMULFF(CW,BW,4,4,4,4,4,AW,4,IER)
      IF(IER.NE.0) RETURN 4
      DO 530 K=1,4
      DO 530 L=1,4
      AINC(K,L,I)=AW(K,L)
  530 CONTINUE
  600 CONTINUE
      RETURN
      END
C
C
C***********************************************************************
C***********************************************************************
C
C
      SUBROUTINE TRANAO(XO,VNOR,VX,AO,IERR)
C
C.......................................................................
C
C    THIS SUBROUTINE SET UP THE TRANSFORMATION MATRIX AO FOR A
C    SPECIFIED PLANE
C
C
C      XO   : (INPUT) THE COORDINATES OF THE ORIGIN OF THE REFERENCE
C                     ATTACHED TO THE SPECIFIED PLANE;
C                     DIMENSION XO(3)
C
C      VNOR : (INPUT) THE UNIT VECTOR OF Z AXIS (NORMAL TO THE
C                     SPECIFIED PLANE).  DIMENSION VNOR(3)
C
C      VX   : (INPUT) THE UNIT VECTOR OF X AXIS (ON THE SPECIFIED
C                     PLANE). DIMENSION VX(3)
C
```

```
C       AO   : (OUTPUT) THE TRANSFORMATION MATRIX AO
C                       DIMENSION AO(4,4)
C
C       IERR : (OUTPUT) ERROR IMFORMATION.
C                       IERR=0  NO ERROR;   IERR=1 SOME ERRORS
C
C.......................................................................
C
C
        DIMENSION XO(3), VNOR(3), VX(3), AO(4,1)
C
C...CHECK THE INPUT DATA
C
        SRR=0.00001
        IERR=0
        TEST=VNOR(1)**2 +VNOR(2)**2 + VNOR(3)**2
        IF(ABS(TEST-1.0) .GT. ERR) IERR=1
        TEST=VX(1)**2 + VX(2)**2 + VX(3)**2
        IF(ABS(TEST-1.0) .GT. ERR) IERR=1
        TEST=VNOR(1)*VX(1) +VNOR(2)*VX(2) +VNOR(3)*VX(3)
        IF(ABS(TEST) .GT.ERR) IERR=1
        IF(IERR .EQ. 0) GO TO 20
        WRITE(6,10)
   10   FORMAT(1X,'INPUT DATA ERROR;  CHECH THE UNIT VECTORS'//)
        RETURN 1
C
C...FIT THE COEFFICIENT INTO THE TRANSFORMATION MATRIX AO
C
   20   DO 30 I=1,3
   30   AO(3,I)=VNOR(I)
        DO 40 I=1,3
   40   AO(1,I)=VX(I)
        DO 50 I=1,3
   50   AO(I,4)=-XO(I)
        DO 60 I=1,3
   60   AO(4,I)=0.0
        AO(4,4)=1.0
C
C...CALCULATE THE UNIT VECTOR ALONG THE Y AXIS AND
C   FIT IT INTO THE TRANSFORMATION MATRIX AO
C
        AO(2,1)=VNOR(2)*VX(3) - VNOR(3)*VX(2)
        AO(2,2)=VNOR(3)*VX(1) - VNOR(1)*VX(3)
        AO(2,3)=VNOR(1)*VX(2) - VNOR(2)*VX(1)
        RETURN
        END
C
C
C***********************************************************************
C***********************************************************************
C
C
C       SUBROUTINE POSINR(N,A,S,ALFA,THETA,AO,AON,AI,AW,XN,VA,VS)
C
C
C.......................................................................
C
C...THIS SUBROUTINE COMPUTES THE POSITION AND DIRECTIONS OF THE
C   LAST LINK OF A GIVEN N-R ROBOT, WHEN ALL THE JOINT DISPLACEMENTS
C   ARE GIVEN.
C
C       N    : (INPUT) THE NUMBER OF THE JOINTS OF THE GIVEN ROBOT
C
C       A    : (INPUT) LINK PARAMETERS A.   DIMENSION A(N)
C
C       S    : (INPUT) LINK PARAMETERS S.   DIMENSION S(N)
C
C       ALFA : (INPUT) LINK PARAMETERS ALFA. DIMENSION ALFA(N)
```

```
C
C     THETA: (INPUT) THE GIVEN JOINT POSITIONS.  DIMENSION THETA(N).
C     AO    : (INPUT) THE TRANSFORMATION MATRIX AO.  FROM THE COORDINATES
C                     COORDINATES ATTACHED TO THE FIRST JOINT TO THE REFE
C                     REFERENCE COORDNATE ATTACHED TO THE GIVEN PLANE
C                     DIMENSION AO(4,4)
C
C     AON   : (OUTPUT) THE TOTAL TRANSFORMATION MATRIX AON.
C                      FROM THE COORDINATES ATTACHED TO THE ROBOT
C                      HAND TO THE COORDINATES ATTACHED TO THE
C                      SPECIFICED PLANE.
C                      DIMENSION AON(4,4)
C
C     AI    : (WORKING SPACE) DIMENSION AI(4,4)
C
C     AW    : (WORKING SPACE) DIMENSION AW(4,4)
C
C     XN    : (OUTPUT) THE COORDINATES OF THE POINT ATTACHED TO THE
C                      END OF THE LADT LINK.  DIMENSION XN(4,4)
C                      XN=(XN, YN, ZN, 1)
C
C     VA    : (OUTPUT) THE UNIT VECTOR ALONG THE KINK LINK A OF THE
C                      LAST LINK.  DIMENSION VA(4,4)
C                      VA=(LA, MA, NA, 0)
C              OUTPUT
C     VS    : (INPUT) THE UNIT VECTOR ALONG THE JOINT AXIS OF THE
C                     LAST LINK ( ALONG S).  DIMENSION VS(4,4)
C
C                     VS=(LS, MS, NS, 0)
C
C
C.......................................................................
C
C
      DIMENSION A(1), S(1), ALFA(1), THETA(1), AO(4,4), AON(4,4)
      DIMENSION AW(4,4), AI(4,4), XN(4), VA(4), VS(4)
C
C...INITIAL SET UP
C
      DTR=3.141593/180.0
      DO 10 I=1,4
      DO 10 J=1,4
   10 AW(I,J)=AO(I,J)
C
C...CALCULATE THE TRANSFORMATION MATRICE AI & AON
C
      DO 30 I=1,N
      COTHI=COS(THETA(I)*DTR)
      SITHI=SIN(THETA(I)*DTR)
      COALI=COS(ALFA(I)*DTR)
      SIALI=SIN(ALFA(I)*DTR)
      AI(1,1)= COTHI
      AI(1,2)=-SITHI * COALI
      AI(1,3)= SITHI * SIALI
      AI(1,4)= A(I) * COTHI
      AI(2,1)= SITHI
      AI(2,2)= COTHI * COALI
      AI(2,3)=-COTHI * SIALI
      AI(2,4)= A(I) * SITHI
      AI(3,1)= 0.0
      AI(3,2)= SIALI
      AI(3,3)= COALI
      AI(3,4)= S(I)
      AI(4,1)= 0.0
      AI(4,2)= 0.0
      AI(4,3)= 0.0
      AI(4,4)= 1.0
C
```

```
C...CALCULATE AON=AO*A1*A2*...*AN
C
      CALL VMULFF(AW,AI,4,4,4,4,4,AON,4,IER)
      DO 20 J=1,4
      DO 20 K=1,4
   20 AW(J,K)=AON(J,K)
   30 CONTINUE
C
C...COMPUTE THE COORDINATES OF THE POINT ATTACHED TO THE END OF
C    THE LAST LINK OF A GIVEN ROBOT
C
      AW(1,1)= 0.0
      AW(2,1)= 0.0
      AW(3,1)= 0.0
      AW(4,1)= 1.0
      CALL VMULFF(AON,AW,4,4,1,4,4,XN,4,IER)
      IF(IER.NE.0) WRITE(6,60)
C
C...COMPUTE THE UNIT VECTOR ALONG THE KINK LINK A OF LAST LINK
C
      AW(1,1)= 1.0
      AW(2,1)= 0.0
      AW(3,1)= 0.0
      AW(4,1)= 0.0
      CALL VMULFF(AON,AW,4,4,1,4,4,VA,4,IER)
      IF(IER.NE.0) WRITE(6,60)
C
C...COMPUTE THE UNIT VECTOR ALONG THE LAST JOINT AXIS (ALONG S).
C
      AW(1,1)= 0.0
      AW(2,1)= 0.0
      AW(3,1)= 1.0
      AW(4,1)= 0.0
      CALL VMULFF(AON,AW,4,4,1,4,4,VS,4,IER)
      IF(IER.NE.0) WRITE(6,60)
   60 FORMAT(///1X,'ERROR MESSAGE FROM SUBROUTINE VMLFF'///
     *1X,'CHECK THE DIMENSION'///)
      RETURN
      END
C
C*********************************************************************
C*********************************************************************
C
C     SUBROUTINE ZX3LP1(A,IA,B,C,N,M1,M2,S,PSOL,DSOL,RW,IW,IER)
C
C.....................................................................
C
C   THIS IS A LINEAR PROGRAMMING SUBROUTINE FROM IMSL.  IT IS ALMOST
C   THE SAME AS SUBROUTINE ZX3LP EXCEPT SOME MINOR MODIFICATION TO
C   MEET THE NATURE OF THE MAIN PROGRAM.
C
C.....................................................................
C
      INTEGER IA,N,M1,M2,IW(1),IER
      REAL A(IA,1),B(1),C(1),S,PSOL(1),DSOL(1),RW(1)
      INTEGER IPHASE,ITMAX,I,IEND,IR,J,IBEG,INEXT,K,L,
     1        LIC,M1P2,IDES,IDES2,IENDIW,ICOPI,II,JJ,M12,M,
     2        JER,M1P1,M1P3,IEND1,IQ,M1MIQ,L1,IWK,LICSV,JI
      REAL ZERO,TEMP,ONE,EPS
      DATA  EPS/1.0E-3/
      DATA  ZERO/0.0/, ONE/1.0/
      DATA ITMAX/10000/
      IER=0
      JER=0
      IEND=M1+M2
      M12=IEND
C
C...TERMAL ERROR --IA IS LESS THAN M1+M2+2
```

```
C
      IF(IA.GE.M12+2) GO TO 5
      IER=130
      GO TO 9000
    5 M1P1=M1+1
      M1P2=M1+2
      M1P3=M1+3
      IEND1=IEND+1
C
C...MOVE A AND B DOWN 2 ROWS
C
      DO 15 I=1,IEND
      K=IEND1-I
      IR=K+2
      B(IR)=B(K)
      PSOL(I)=I
      DO 10 J=1,N
      A(IR,J)=A(K,J)
   10 CONTINUE
   15 CONTINUE
      IR=IEND+2
C
C...CHECK EQUALITY CONSTRAINTS FOR NEGATIVE RIGHT SIDE
C
      IF(M2.EQ.0) GO TO 30
      IBEG=M1+3
      DO 25 I=IBEG,IR
      IF(B(I).GE.ZERO) GO TO 25
      B(I)= - B(I)
      PSOL(I-2)= - PSOL(I-2)
      DO 20  J=1,N
      A(I,J)= - A(I,J)
   20 CONTINUE
   25 CONTINUE
C
C...REORDER OTHER CONSTRAINTSSO B(1) .GE. 0, I=1,2...M1-IG
C
   30 IQ=0
      IF(M1.EQ.0) GO TO 60
      IEND=M1P2
      INEXT=IEND
   35 IF( B(IEND) .GE. ZERO) GO TO 55
      IF(INEXT.EQ.IEND) GO TO 45
      TEMP=B(IEND)
      B(IEND)=B(INEXT)
      B(INEXT)= TEMP
      TEMP=PSOL(IEND-2)
      PSOL(IEND-2)= PSOL(INEXT-2)
      PSOL(INEXT-2)=TEMP
      DO 40 J=1,N
      TEMP=A(IEND,J)
      A(IEND,J)= A(INEXT,J)
      A(INEXT,J)= TEMP
   40 CONTINUE
   45 IQ=IQ+1
      PSOL(INEXT-2)= -PSOL(INEXT-2)
      B(INEXT)=- B(INEXT)
      DO 50 J=1,N
      A(INEXT,J)= -A(INEXT,J)
   50 CONTINUE
      INEXT= INEXT - 1
   55 IEND= IEND-1
      IF(IEND.NE.2) GO TO 35
C
C...COMPUTE ROW 1 AND 2 OF A AND B
C
   60 DO 65 J=1,N
      A(2,J)= - C(J)
```

```
      A(1,J)= ZERO
      DO 65  I=2,IR
      A(1,J)= A(1,J) - A(I,J)
   65 CONTINUE
      B(1) =ZERO
      B(2) =ZERO
      DO 70 I=3,IR
      B(1) = B(1) - B(I)
   70 CONTINUE
      M=M12 + 1
      IF(IA.EQ.IR) GO TO 80
C
C...PACK A
C
      K=0
      L=0
      DO 75 J=1,N
      DO 75 I=1,IR
      K=MOD(K,IA) + 1
      IF(K.EQ.1) L=L+1
      A(K,L) =A(I,J)
   75 CONTINUE
C
C...GET ICOLMS AND ROW
C
   80 LIC=IR+IQ
      M1MIQ=M1-IQ
      L1=M1P1-IQ
      DO 95 I=1,LIC
      IF(I.GE.M1P3) GO TO 90
      IF(I.GT.L1) GO TO 85
      RW(I) =-ONE
      IW(I)=I+1
      GO TO 95
   85 RW(I)= ONE
      IW(I)= -I - 1
      GO TO 95
   90 RW(I)=ZERO
      IW(I)= I - IQ
   95 CONTINUE
C
C...WORK STORAGE ASSIGNMENTS ICOLMS(1) = IW(I) , IDES(1) = IW(IDES),
C    COPI(1,1) = RW(ICOPI), ROW(1) = RW(1), WA(1) = RW(IWK),
C    X(1) = DSOL(1)
C
      IW(M1P2) = 1
C...GET IDES
      IDES=LIC + 1
      IW(IDES)= N+M1P2
      IDES2=IDES + 1
      IEND= IDES2 + M1MIQ
      IENDIW= IDES2 + M12
      K= N + 1
      DO 100 I=IDES2,IENDIW
      IW(I)=K
      IF(I.EQ.IEND) K=N+M1P2
      K=K + 1
  100 CONTINUE
C...GET COPI
      ICOPI = IDES
      IWK= IR*IR + ICOPI
      K= IWK - 1
      DO 105  I=ICOPI,K
      RW(I)= ZERO
  105 CONTINUE
      L= ICOPI
      J=M1MIQ + 1
      DO  110 I=1,J
```

```
      L =L + IR
      RW(L) = ONE
  110 CONTINUE
      J=0
      DO 115 I=ICOPI,K,IR
      RW(I+J)=ONE
      J=J+1
  115 CONTINUE
      K=1
C
C..SOLVE PHASE 1 PROBLEM
C
      IPHASE=1
      CALL ZXOLP(IPHASE,A,B,IW,RW,K,M,N,ITMAX,LIC,IR,RW(ICOPI),
     1           IW(IDES),DSOL,RW(IWK),IER)
      IF(IER.NE.0) GO TO 185
      K=M1P2 + N
      J=2
C
C...CHECK PHASE 1 SOLUTION FOR ARTIFICIAL VARIABLES
C
      DO 125 I=IDES2,IENDIW
      IF(IW(I).LE.K) GO TO 120
      IF(DSOL(J).GT.EPS) GO TO 180
C
C...ARTIFICIAL VARIABLES REMAIN IN THE ZERO LEVEL
C
      JER=70
  120 J=J+1
  125 CONTINUE
C
C...INTERCHANGE FIRST TWO ROWS OF THE A MATRIX AND NEGATE THE SECOND
C   ROW
C
      K=(N-1)*IR + 1
      DO 140 L=1,K,IR
      J=(L+IA-1)/IA
      I=L-(J-1)*IA
      TEMP=A(I,J)
      IF(I.LT.IA) GO TO 130
      II=1
      JJ=J+1
      GO TO 135
  130 II=I+1
      JJ=J
  135 A(I,J)=A(II,JJ)
      A(II,JJ)=-TEMP
  140 CONTINUE
C
C...INTERCHANGE FIRST TWO ELEMENTS OF THE RIGHT HAND SIDE AND
C   NEGATE THE SECOD ELEMENT
C
      B(2)=-B(1)
      B(1)= ZERO
C
C...INTERCHANGE FIRST TWO COLUMS OF COPI AND NEGATE THE SECOND COLUMN
C
      J=ICOPI + IR - 1
      DO 145 I=ICOPI,J
      TEMP=RW(I)
      RW(I)= RW(I+IR)
      RW(I+IR)=-TEMP
  145 CONTINUE
      J=ICOPI+IR*IR -1
      DO 150 I=ICOPI,J,IR
      TEMP= RW(I)
      RW(I)=RW(I+1)
      RW(I+1)=TEMP
```

```
 150  CONTINUE
      K=2
C...NEGATE ROW
      DO 155 I=1,LIC
      RW(I) = - RW(I)
 155  CONTINUE
C...ICOLMS(1)=1
      IW(1)=1
C...INTERCHANGE IDES(1) AND IDES(2)
      J=IW(IDES)
      IW(IDES)= IW(IDES2)
      IW(IDES2)= J
      IW(M1P2)= -2
      LICSV=LIC
C
C...REMOVE ARTIFICIAL VARIABLES FROM TABLEAU, IF POSSIBLE
C
      IF(JER.EQ.0) LIC=M1P2
C
C...SOLVE PHASE 2 PROBLEM
C
      IPHASE=2
      CALL ZXOLP(IPHASE,A,B,IW,RW,K,M,N,ITMAX,LIC,IR,RW(ICOPI),
     1             IW(IDES),DSOL,RW(IWK),IER)
      LIC=LICSV
      S=DSOL(1)
C...RE-ORDER THE PRIMAL SOLUTION
      DO 160 J=1,M12
      RW(J)= PSOL(J)
 160  CONTINUE
      DO 165 J=1,N
      PSOL(J)=ZERO
 165  CONTINUE
      DO 170 J=1,M
      K=IW(IDES+J)
      IF(K.GT.N) GO TO 170
      PSOL(K)=DSOL(J+1)
 170  CONTINUE
C...GET DUAL SOLUTION
      JI=LIC+1+IR
      TEMP=RW(JI)
      DO 175 I=1,M12
      J=ABS(RW(I))
      JI=JI+IR
      DSOL(J)=RW(JI) + TEMP
      IF(RW(I).LT.ZERO) DSOL(J) = -DSOL(J)
      IF(I.LE.M1) DSOL(J)=ABS(DSOL(J))
 175  CONTINUE
      GO TO 195
C
C...ARTIFICIAL VARIABLES ARE IN THE SOLUTION HENCE NO FEASIBLE
C   SOLUTION EXISTS
C
 180  IER=133
 185  DO 190 J=1,M12
      RW(J)=PSOL(J)
 190  CONTINUE
C...RESTORE A AND B
 195  IF(IA.EQ.IR) GO TO 210
C...UNRACK A
      K=MOD(N*IR,IA)
      IF(K.EQ.0) K=IA
      L=(N*IR+IA-1)/IA
      J=N
      DO 205 JJ=1,N
      I=IR
      DO 200 II=1,IR
      A(I,J)=A(K,L)
```

```
      I=I-1
      K=K-1
      IF(K.NE.0) GO TO 200
      K=IA
      L=L-1
 200  CONTINUE
      J=J-1
 205  CONTINUE
 210  DO 220 I=1,M12
      B(I)=B(I+2)
      DO 215 J=1,N
      A(I,J)=A(I+2,J)
 215  CONTINUE
 220  CONTINUE
C
C...PERMUTE ROWS OF A AND ELEMENTS OF B ACCORDING TO PERMUTATIONS
C   STORED IN RW
C
      DO 230 I=1,M12
      J=ABS(RW(I))
      IF(J.EQ.I) GO TO 230
      TEMP=RW(I)
      RW(I)=RW(J)
      RW(J)=TEMP
      TEMP=B(I)
      B(I)=B(J)
      B(J)=TEMP
      DO 225 K=1,N
      TEMP=A(I,K)
      A(I,K)= A(J,K)
      A(J,K)=TEMP
 225  CONTINUE
 230  CONTINUE
      DO 240 I=1,M12
      IF(RW(I).GT.ZERO) GO TO 240
      B(I)=-B(I)
      DO 235 J=1,N
      A(I,J)=-A(I,J)
 235  CONTINUE
 240  CONTINUE
      IF(IER.EQ.0) IER=JER
9000  CONTINUE
9005  RETURN
      END
```

APPENDIX D

COMPUTER PROGRAMS IN CHAPTER IV

```
C ***********************************************************************
C *                                                                     *
C *  THIS IS AN INTERACTIVE PROGRAM FOR SYNTHESIS OF 6R INDUSTRIAL      *
C *  ROBOTS FOR A PRESCRIBED WORKING SPACE.                             *
C *                                                                     *
C *  THE USER JUST INPUT THE REQUIRED DATA BY ANSWERING THE QUESTIONS   *
C *  SHOWEN ON THE TERMINAL.                                            *
C *                                                                     *
C *  PREFER TO SET THE TERMINAL LINESIZE(132), IF IT IS POSSIBLE.       *
C *                                                                     *
C *  THE MAXIMUM NUMBER OF GIVEN HAND POSITIONS IS 100.                 *
C *                                                                     *
C ***********************************************************************
      COMMON ITYPE,N,HANDLG,WRISTL,HXYZ(3,100),HLMN(3,100),PLMN(3,100),
     &       XYZ1(6,3,100),DIR(6,3,100),THETA(6,102),A(6)
      DIMENSION XYZMAX(3),XYZMIN(3),ATYPE(3),
     &       BXY(9),BXYL(9),BXYR(9),BXYLOW(9),BXYHIG(9)
      DATA MPRINT, F, R /0, 0.001, 0.90/
      DATA ATYPE/'A1  ', 'B1  ', 'C   '/
   10 FORMAT(1X,'ENTER THE COORDINATES OF THE ROBOT HAND, X,Y,Z.')
   15 FORMAT(1X,'ENTER THE DIRECTION COSINES L, M, N, ALONG THE ROBOT'/
     &       1X,'HAND AXIS')
   20 FORMAT(1X,'ENTER THE DIRECTION COSINES L, M, N, PERPENDICULAR TO'/
     &       1X,'THE ROBOT HAND AXIS')
   25 FORMAT(1X,'THE SQUARE SUMMATION OF L, M, N IS NOT EQUAL TO 1',
     & 1X,'RE-ENTER -  ')
   30 FORMAT(1X,'ENTER   0 : TO RE-ENTER THE LAST POSITION,'/
     &       1X,'        1 : TO ENTER THE NEXT POSITION,'/
     &       1X,'        9 : NO MORE POSITION TO BE ENTERED.')
   35 FORMAT(1X,'ENTER   1 : FOR 6R TYPE A1 ROBOT,'/
     &       1X,'        2 : FOR 6R TYPE B1 ROBOT,'/
     &       1X,'        3 : FOR 6R TYPE C  ROBOT,')
   40 FORMAT(1X,'ENTER  1. THE LENGTH OF ROBOT HAND, AND'/
     &       1X,'       2. THE MINIMUM SPACE REQUIRED FOR MOUNTING'/
     &       1X,'          THE DRIVING MOTOR ON THE WRIST JOINTS.')
C
C...INPUT DATA
C
   50 N=1
  100 WRITE(6,10)
      READ(5,*) (HXYZ(I,N),I=1,3)
      WRITE(6,15)
  105 READ(5,*) (HLMN(I,N),I=1,3)
      TEST= HLMN(1,N)**2 + HLMN(2,N)**2 + HLMN(3,N)**2
      IF(ABS(TEST-1.0) .LE. 0.00001) GO TO 110
      WRITE(6,25)
      GO TO 105
  110 WRITE(6,20)
  115 READ(5,*) (PLMN(I,N),I=1,3)
      TEST= PLMN(1,N)**2 + PLMN(2,N)**2 + PLMN(3,N)**2
      IF(ABS(TEST-1.0) .LE. 0.00001) GO TO 120
      WRITE(6,25)
      GO TO 115
  120 WRITE(6,30)
      READ(5,*) ICHK
      N=N+ICHK
      IF(N.GT.100) GO TO 124
      IF(ICHK.EQ.0 .OR. ICHK.EQ.1) GO TO 100
      IF(ICHK.EQ.9) GO TO 125
      GO TO 120
  124 N=109
  125 WRITE(6,40)
      READ(5,*) HANDLG,WRISTL
      N=N-9
  128 WRITE(6,35)
      READ(5,*) ITYPE
C
C...INITIAL VALUE FOR GRID4
```

```
C
      DO 130 I=1,3
      XYZMIN(I)=1.0E20
      XYZMAX(I)=-1.0E20
  130 CONTINUE
      DO 135 J=1,N
      DO 135 I=1,3
      IF(HXYZ(I,J).LT.XYZMIN(I)) XYZMIN(I)=HXYZ(I,J)
      IF(HXYZ(I,J).GT.XYZMAX(I)) XYZMAX(I)=HXYZ(I,J)
  135 CONTINUE
      RL=AMAX1((XYZMAX(1)-XYZMIN(1)),(XYZMAX(2)-XYZMIN(2)),
     &         (XYZMAX(3)-XYZMIN(3)))
      BXYL(1)=XYZMIN(1) - 0.5*RL
      BXYL(2)=XYZMIN(2)
      BXYL(3)=AMIN1(0.0, XYZMIN(3))
      BXYR(1)=XYZMIN(1)
      BXYR(2)=XYZMAX(2)
      BXYR(3)=XYZMAX(3)
C
C...CALL GRID4 TO SEARCH THE OPTIMUM COMBINATION OF ROBOT
C   LINK LENGTHS AND ROBOT BASE LOCATION
C
      CALL GRID4(3,MPRINT,BXYL,BXYR,F,R,VOLUME,BXY,BXYLOW,BXYHIG,NN)
C
C...PRINTOUT THE RESULT
C
      RTD=180.0/3.141593
      DO 140 I=1,6
      THETA(I,101)=THETA(I,101)*RTD
      THETA(I,102)=THETA(I,102)*RTD
      DO 140 J=1,N
      THETA(I,J)= THETA(I,J)*RTD
  140 CONTINUE
      WRITE(6,142)
  142 FORMAT(1H1///1X,'THE GIVEN DATA :'//
     &T6,'ROBOT HAND POSITIONS AND ORIENTATIONS :')
      WRITE(6,144)
  144 FORMAT(///1X,' NO',T10,'XH',T22,'YH',T34,'ZH',T46,'LH',
     &T58,'MH',T70,'NH',T82,'LP',T94,'MP',T106,'NP'/)
      WRITE(6,146) ( I,(HXYZ(J,I),J=1,3),(HLMN(K,I),K=1,3),
     &(PLMN(L,I),L=1,3) ,I=1,N)
  146 FORMAT((/1X,I3,9G12.4))
      WRITE(6,148) HANDLG, WRISTL
  148 FORMAT(/////T6,'ROBOT HAND  LENGTH :',G15.6
     &        //T6,'ROBOT WRIST LENGTH :',G15.6)
      WRITE(6,150) ATYPE(ITYPE)
  150 FORMAT(1H1///1X,'THE SYNTHESIZED 6R TYPE ',A2,' ROBOT IS FOUND')
      WRITE(6,155) (BXY(I),I=1,3)
  155 FORMAT(//T6,'ROBOT BASE LOCATION :  (',3G20.6,' )')
      WRITE(6,160) A(2), A(3)
  160 FORMAT(//T6,'ROBOT LINK LENGTH   :'
     &/T30,'A(2) = ',G20.6/ T30,'A(3) = ',G20.6)
      WRITE(6,165) ( I,THETA(I,101),THETA(I,102) , I=1,6)
  165 FORMAT(//T6,'JOINT MOTION RANGES :',
     &(/T30,'THETA(',I1,') : ',G20.6,'TO',G20.6, ' (DEG.)'))
      WRITE(6,180)
  180 FORMAT(///1X,'  NO',T8,'THETA1',T20,'THETA2',T32,'THETA3',
     &T44,'THETA4',T56,'THETA5',T68,'THETA6'/)
      WRITE(6,185) ( I,(THETA(J,I),J=1,6) ,I=1,N)
  185 FORMAT((/1X,I3,6G12.4))
      WRITE(6,190)
  190 FORMAT(////1X,
     &'ENTER 1 : FOR THE SAME WORKSPACE BUT DIFFERENTTYPE OF ROBOT'/1X,
     &'      2 : FOR DIFFERENT WORKSPACE'/1X,
     &'      3 : TO TERMINATE THE PROGRAM')
      READ(5,*) IEND
      GO TO (128,50,200), IEND
  200 STOP
      END
```

```
C
C
C=====================================================================
      SUBROUTINE MERIT4(BXY,VOLUME)
C
C=====================================================================
C
      DIMENSION BXY(3)
      COMMON ITYPE,N,HANDLG,WRISTL,HXYZ(3,100),HLMN(3,100),PLMN(3,100),
     &       XYZ1(6,3,100),DIR(6,3,100),THETA(6,102),A(6)
C
C
      DO 50 J=4,6
      THETA(J,101)= 1.0E20
      THETA(J,102)=-1.0E20
   50 CONTINUE
      GO TO (100,200,300), ITYPE
C.........................................
C===> ===> ===> 6R TYPE A1 ROBOT <=== <=== <===
C.........................................
C
C........COORDINATES OF JOINT 6
C
  100 DO 105 I=1,N
      DO 105 IXYZ=1,3
      XYZ1(6,IXYZ,I)=HXYZ(IXYZ,I)-HANDLG*HLMN(IXYZ,I)-BXY(IXYZ)
  105 CONTINUE
C
C...DIRECTION OF JOINT 5
C
      DO 110 I=1,N
      IF(PLMN(3,I).EQ.0.0) GO TO 108
      IF(XYZ1(6,1,I).EQ.0.0) GO TO 106
      E = XYZ1(6,2,I) / XYZ1(6,1,I)
      F = (PLMN(1,I) +  E*PLMN(2,I)) / PLMN(3,I)
      DIR(5,1,I) = 1.0/SQRT(1.0 + E*E + F*F)
      DIR(5,2,I) = E*DIR(5,1,I)
      DIR(5,3,I) =-F*DIR(5,1,I)
      GO TO 110
  106 E=PLMN(2,I)/PLMN(3,I)
      DIR(5,1,I)=0.0
      DIR(5,2,I)=1.0/SQRT(1.0+E*E)
      DIR(5,3,I)=-E*DIR(5,2,I)
      GO TO 110
  108 DIR(5,1,I)=0.0
      DIR(5,2,I)=0.0
      DIR(5,3,I)=-1.0
  110 CONTINUE
C
C...COORDINATES OF JOINT 4
C
      DO 115 I=1,N
      DO 115 IXYZ=1,3
      XYZ1(4,IXYZ,I)=XYZ1(6,IXYZ,I)-DIR(5,IXYZ,I)*WRISTL
  115 CONTINUE
C
C...CALL SUBROUTINE GET3R TO GET THE 3R ROBOT
C
      CALL GET3R(VOLUME)
C
C...FIND THE MOTION RANGES OF JOINT 4,5,AND 6
C
      DO 140 I=1,N
      RS= SIN(THETA(2,I) + THETA(3,I))
      DIR(3,1,I)= RS*COS(THETA(1,I))
      DIR(3,2,I)= RS*SIN(THETA(1,I))
      DIR(3,3,I)= COS(THETA(2,I) + THETA(3,I))
      DIR(4,1,I)=-SIN(THETA(1,I))
```

```
      DIR(4,2,I)= COS(THETA(1,I))
      DIR(4,3,I)= 0.0
C
C...JOINT DISP OF JOINT 4
C
      U35=0.0
      DO 120 J=1,3
      U35=U35 + DIR(3,J,I)*DIR(5,J,I)
 120  CONTINUE
      IF(ABS(U35).GT.1.0) U35=SIGN(1.0,U35)
      THETA(4,I)= ARCOS(U35)
      TEST=
     & (DIR(3,1,I)*DIR(5,2,I)-DIR(5,1,I)*DIR(3,2,I))*DIR(4,3,I)
     &+(DIR(3,2,I)*DIR(5,3,I)-DIR(5,2,I)*DIR(3,3,I))*DIR(4,1,I)
     &+(DIR(3,3,I)*DIR(5,1,I)-DIR(5,3,I)*DIR(3,1,I))*DIR(4,2,I)
      IF(TEST. LT. 0.0) THETA(4,I)= - THETA(4,I)
C
C...JOINT DISP OF JOINT 5
C
      U4P=0.0
      DO 125 J=1,3
      U4P=U4P + DIR(4,J,I)*PLMN(J,I)
 125  CONTINUE
      IF(ABS(U4P).GT.1.0) U4P=SIGN(1.0,U4P)
      THETA(5,I)= ARCOS(U4P)
      TEST=
     & (DIR(4,1,I)*PLMN(2,I)-PLMN(1,I)*DIR(4,2,I))*DIR(5,3,I)
     &+(DIR(4,2,I)*PLMN(3,I)-PLMN(2,I)*DIR(4,3,I))*DIR(5,1,I)
     &+(DIR(4,3,I)*PLMN(1,I)-PLMN(3,I)*DIR(4,1,I))*DIR(5,2,I)
      IF(TEST. LT. 0.0) THETA(5,I)= - THETA(5,I)
C
C...JOINT DISP OF JOINT 6
C
      U5H=0.0
      DO 130 J=1,3
      U5H=U5H + DIR(5,J,I)*HLMN(J,I)
 130  CONTINUE
      IF(ABS(U5H).GT.1.0) U5H=SIGN(1.0,U5H)
      THETA(6,I)= ARCOS(U5H)
      TEST=
     & (DIR(5,1,I)*HLMN(2,I)-HLMN(1,I)*DIR(5,2,I))*PLMN(3,I)
     &+(DIR(5,2,I)*HLMN(3,I)-HLMN(2,I)*DIR(5,3,I))*PLMN(1,I)
     &+(DIR(5,3,I)*HLMN(1,I)-HLMN(3,I)*DIR(5,1,I))*PLMN(2,I)
      IF(TEST. LT. 0.0) THETA(6,I)= - THETA(6,I)
C
C...FIND THE MOTION LIMITS OF JOINT 4,5,AND 6
C
      DO 135 J=4,6
      IF(THETA(J,I) .LT. THETA(J,101)) THETA(J,101)=THETA(J,I)
      IF(THETA(J,I) .GT. THETA(J,102)) THETA(J,102)=THETA(J,I)
 135  CONTINUE
 140  CONTINUE
      RETURN
C.........................................
C===> ===> ===> 6R TYPE B1 ROBOT <=== <=== <===
C.........................................
C
C........COORDINATES OF JOINT 5
C
 200  DO 205 I=1,N
      DO 205 IXYZ=1,3
      XYZ1(5,IXYZ,I)=HXYZ(IXYZ,I)-HANDLG*HLMN(IXYZ,I)-BXY(IXYZ)
 205  CONTINUE
C
C...DIRECTION OF JOINT 5
C
      DO 210 I=1,N
      IF(HLMN(3,I).EQ.0.0) GO TO 208
```

```
      IF(XYZ1(5,1,I).EQ.0.0) GO TO 206
      AA= XYZ1(5,2,I) / XYZ1(5,1,I)
      B = (HLMN(1,I) + AA*HLMN(2,I)) / HLMN(3,I)
      DIR(5,1,I) = 1.0/SQRT(1.0 +AA*AA+ B*B)
      DIR(5,2,I) =AA*DIR(5,1,I)
      DIR(5,3,I) =-B*DIR(5,1,I)
      GO TO 210
 206  B=HLMN(2,I)/HLMN(3,I)
      DIR(5,1,I)=0.0
      DIR(5,2,I)=1.0/SQRT(1.0+B*B)
      DIR(5,3,I)=-B*DIR(5,2,I)
      GO TO 210
 208  DIR(5,1,I)=0.0
      DIR(5,2,I)=0.0
      DIR(5,3,I)=1.0
 210  CONTINUE
C
C...DIRECTION OF JOINT 4
C
      DO 215 I=1,N
      IF(DIR(5,3,I).EQ.0.0) GO TO 214
      IF(XYZ1(5,1,I).EQ.0.0) GO TO 212
      C = XYZ1(5,2,I) / XYZ1(5,1,I)
      D = (DIR(5,1,I) + C*DIR(5,2,I))/DIR(5,3,I)
      DIR(4,1,I) = 1.0/SQRT(1.0 + C*C + D*D)
      DIR(4,2,I) = C*DIR(4,1,I)
      DIR(4,3,I) =-D*DIR(4,1,I)
      GO TO 215
 212  C=DIR(5,2,I)/DIR(5,3,I)
      DIR(4,1,I)=0.0
      DIR(4,2,I)=1.0/SQRT(1.0+C*C)
      DIR(4,3,I)=-C*DIR(4,2,I)
      GO TO 215
 214  DIR(4,1,I)=0.0
      DIR(4,2,I)=0.0
      DIR(4,3,I)=-1.0
 215  CONTINUE
C
C...COORDINATES OF JOINT 4
C
      DO 220 I=1,N
      DO 220 IXYZ=1,3
      XYZ1(4,IXYZ,I)=XYZ1(5,IXYZ,I)-DIR(4,IXYZ,I)*WRISTL
 220  CONTINUE
C
C...CALL SUBROUTINE GET3R TO GET THE 3R ROBOT
C
      CALL GET3R(VOLUME)
C
C...FIND THE MOTION RANGES OF JOINT 4,5,AND 6
C
      DO 260 I=1,N
      XYZ1(3,1,I)=A(2)*SIN(THETA(2,I))*COS(THETA(1,I))
      XYZ1(3,2,I)=A(2)*SIN(THETA(2,I))*SIN(THETA(1,I))
      XYZ1(3,3,I)=A(2)*COS(THETA(2,I))
      DIR(3,1,I)= (XYZ1(4,1,I)-XYZ1(3,1,I))/ A(3)
      DIR(3,2,I)= (XYZ1(4,2,I)-XYZ1(3,2,I))/ A(3)
      DIR(3,3,I)= (XYZ1(4,3,I)-XYZ1(3,3,I))/ A(3)
C
C...JOINT DISP OF JOINT 4
C
      U34=0.0
      DO 225 J=1,3
      U34=U34 + DIR(3,J,I)*DIR(4,J,I)
 225  CONTINUE
      IF(ABS(U34).GT.1.0) U34=SIGN(1.0, U34)
      THETA(4,I)= ARCOS(U34)
      TEST=
```

```
     & (DIR(3,1,I)*DIR(4,2,I)-DIR(4,1,I)*DIR(3,2,I))*(-SIN(THETA(1,I)))
     &+(DIR(3,2,I)*DIR(4,3,I)-DIR(4,2,I)*DIR(3,3,I))*( COS(THETA(1,I)))
      IF(TEST. LT. 0.0) THETA(4,I)= - THETA(4,I)
C
C...JOINT DISP OF JOINT 5
C
      U4H=0.0
      DO 230 J=1,3
      U4H=U4H + DIR(4,J,I)*HLMN(J,I)
 230  CONTINUE
      IF(ABS(U4H).GT.1.0) U4H=SIGN(1.0,U4H)
      THETA(5,I)= ARCOS(U4H)
      TEST=  (DIR(4,1,I)*HLMN(2,I)-HLMN(1,I)*DIR(4,2,I))*DIR(5,3,I)
     &      +(DIR(4,2,I)*HLMN(3,I)-HLMN(2,I)*DIR(4,3,I))*DIR(5,1,I)
     &      +(DIR(4,3,I)*HLMN(1,I)-HLMN(3,I)*DIR(4,1,I))*DIR(5,2,I)
      IF(TEST. LT. 0.0) THETA(5,I)= - THETA(5,I)
C
C...JOINT DISP OF JOINT 6
C
      U5P=0.0
      DO 235 J=1,3
      U5P=U5P + DIR(5,J,I)*PLMN(J,I)
 235  CONTINUE
      IF(ABS(U5P).GT.1.0) U5P=SIGN(1.0,U5P)
      THETA(6,I)= ARCOS(U5P)
      TEST=  (DIR(5,1,I)*PLMN(2,I)-PLMN(1,I)*DIR(5,2,I))*HLMN(3,I)
     &      +(DIR(5,2,I)*PLMN(3,I)-PLMN(2,I)*DIR(5,3,I))*HLMN(1,I)
     &      +(DIR(5,3,I)*PLMN(1,I)-PLMN(3,I)*DIR(5,1,I))*HLMN(2,I)
      IF(TEST. LT. 0.0) THETA(6,I)= - THETA(6,I)
C
C...FIND TPE MOTION LIMITS OF JOINT 4,5,AND 6
C
      DO 255 J=4,6
      IF(THETA(J,I) .LT. THETA(J,101)) THETA(J,101)=THETA(J,I)
      IF(THETA(J,I) .GT. THETA(J,102)) THETA(J,102)=THETA(J,I)
 255  CONTINUE
 260  CONTINUE
      RETURN
C..........................................
C===> ===> ===> 6R TYPE C  ROBOT <=== <=== <===
C..........................................
C
C...COORDINATES OF JOINT 4
C
 300  DO 305 I=1,N
      DO 305 IXYZ=1,3
      XYZ1(4,IXYZ,I)=HXYZ(IXYZ,I)-HANDLG*HLMN(IXYZ,I)-BXY(IXYZ)
 305  CONTINUE
C
C...CALL SUBROUTE GET3R TO GET THE 3R ROBOT
C
      CALL GET3R(VOLUME)
C
C...DIR OF JOINT 3
C
      DO 350 I=1,N
      DIR(3,1,I)= -SIN(THETA(1,I))
      DIR(3,2,I)=  COS(THETA(1,I))
      DIR(3,3,I)=  0.0
C
C...DIR OF JOINT 4
C
      RS=SIN(THETA(2,I)+THETA(3,I))
      DIR(4,1,I)= RS*COS(THETA(1,I))
      DIR(4,2,I)= RS*SIN(THETA(1,I))
      DIR(4,3,I)= COS(THETA(2,I) + THETA(3,I))
C
C...DIR OF JOINT 5
```

```
C
      DIR(5,1,I)=DIR(4,2,I)*HLMN(3,I)-HLMN(2,I)*DIR(4,3,I)
      DIR(5,2,I)=DIR(4,3,I)*HLMN(1,I)-HLMN(3,I)*DIR(4,1,I)
      DIR(5,3,I)=DIR(4,1,I)*HLMN(2,I)-HLMN(1,I)*DIR(4,2,I)
      RNORM=0.0
      DO 306 K=1,3
  306 RNORM=RNORM + DIR(5,K,I)**2
      RNORM=SQRT(RNORM)
      U35=0.0
      DO 307 K=1,3
      U35=U35 + DIR(3,K,I)*DIR(5,K,I)
  307 CONTINUE
      IF(U35.LT.0.0) RNORM= - RNORM
      DO 308 K=1,3
      DIR(5,K,I)= DIR(5,K,I)/RNORM
  308 CONTINUE
C
C...JOINT DISPLACEMENTS OF JOINT 4,5,AND 6
C
      U35=0.0
      DO 310 J=1,3
      U35=U35 + DIR(3,J,I)*DIR(5,J,I)
  310 CONTINUE
      IF(ABS(U35).GT.1.0) U35=SIGN(1.0,U35)
      THETA(4,I)=ARCOS(U35)
      TEST=  (DIR(3,1,I)*DIR(5,2,I)-DIR(5,1,I)*DIR(3,2,I))*DIR(4,3,I)
     &      +(DIR(3,2,I)*DIR(5,3,I)-DIR(5,2,I)*DIR(3,3,I))*DIR(4,1,I)
     &      +(DIR(3,3,I)*DIR(5,1,I)-DIR(5,3,I)*DIR(3,1,I))*DIR(4,2,I)
      IF(TEST. LT. 0.0) THETA(4,I)= - THETA(4,I)
C
      U4H=0.0
      DO 315 J=1,3
      U4H=U4H + DIR(4,J,I)*HLMN(J,I)
  315 CONTINUE
      IF(ABS(U4H).GT.1.0) U4H=SIGN(1.0,U4H)
      THETA(5,I)=ARCOS(U4H)
      TEST=  (DIR(4,1,I)*HLMN(2,I)-HLMN(1,I)*DIR(4,2,I))*DIR(5,3,I)
     &      +(DIR(4,2,I)*HLMN(3,I)-HLMN(2,I)*DIR(4,3,I))*DIR(5,1,I)
     &      +(DIR(4,3,I)*HLMN(1,I)-HLMN(3,I)*DIR(4,1,I))*DIR(5,2,I)
      IF(TEST. LT. 0.0) THETA(5,I)= - THETA(5,I)
C
      U5P=0.0
      DO 320 J=1,3
      U5P=U5P + DIR(5,J,I)*PLMN(J,I)
  320 CONTINUE
      IF(ABS(U5P).GT.1.0) U5P=SIGN(1.0,U5P)
      THETA(6,I)=ARCOS(U5P)
      TEST=  (DIR(5,1,I)*PLMN(2,I)-PLMN(1,I)*DIR(5,2,I))*HLMN(3,I)
     &      +(DIR(5,2,I)*PLMN(3,I)-PLMN(2,I)*DIR(5,3,I))*HLMN(1,I)
     &      +(DIR(5,3,I)*PLMN(1,I)-PLMN(3,I)*DIR(5,1,I))*HLMN(2,I)
      IF(TEST. LT. 0.0) THETA(6,I)= - THETA(6,I)
C
      DO 330 J=4,6
      IF(THETA(J,I) .LT. THETA(J,101)) THETA(J,101)=THETA(J,I)
      IF(THETA(J,I) .GT. THETA(J,102)) THETA(J,102)=THETA(J,I)
  330 CONTINUE
  350 CONTINUE
      RETURN
      END
C
C
C=====================================================================
C
      SUBROUTINE GET3R(VOLUME)
C
C=====================================================================
C
      COMMON ITYPE,N,HANDLG,WRISTL,HXYZ(3,100),HLMN(3,100),PLMN(3,100),
```

```
     &         XYZ1(6,3,100),DIR(6,3,100),THETA(6,102),A(6)
C
C
C
      DO 50 I=1,3
      THETA(I,101)=  1.0E20
      THETA(I,102)= -1.0E20
   50 CONTINUE
      RLMAX= -1.0E20
C
C...FIND THE MAXIMUM DISTANT AND GET A(2) AND A(3)
C
      DO 100 I=1,N
      RL=SQRT(XYZ1(4,1,I)**2 + XYZ1(4,2,I)**2 + XYZ1(4,3,I)**2)
      IF(RL .GT. RLMAX) RLMAX=RL
  100 CONTINUE
      A(2)=RLMAX*0.5
      A(3)= A(2)
C
C...FIND THE JOINT DISPLACEMENTS
C
      DO 120 I=1,N
      RL=SQRT(XYZ1(4,1,I)**2 + XYZ1(4,2,I)**2 + XYZ1(4,3,I)**2)
      THETA(1,I)=ATAN2(XYZ1(4,2,I) , XYZ1(4,1,I))
      THETA(2,I)=ARCOS(XYZ1(4,3,I)/RL) - ARCOS(RL/(2.0*A(2)))
      TEST= (RL*RL - A(2)**2 - A(3)**2) / (2.0*A(2)*A(3))
      IF(ABS(TEST) .GT. 1.0) TEST= SIGN(1.0, TEST)
      THETA(3,I)= ARCOS(TEST)
      DO 110 J=1,3
      IF(THETA(J,I) .LT. THETA(J,101)) THETA(J,101)=THETA(J,I)
      IF(THETA(J,I) .GT. THETA(J,102)) THETA(J,102)=THETA(J,I)
  110 CONTINUE
  120 CONTINUE
C
C... THE AREA OF WORKSPACE
C
      F= (COS(THETA(3,101)) - COS(THETA(3,102)) )/4.0
      AREA=F*(THETA(2,102) - THETA(2,101))*(A(2) + A(3))**2
      ANG2M=(THETA(2,101) + THETA(2,102))/2.0
      XCG=(COS(THETA(3,101)) - COS(THETA(3,102)))*SIN(ANG2M)
     &    -SIN(ANG2M)*(SIN(THETA(3,101))**2 - SIN(THETA(3,102))**2)/2.0
     &    -COS(ANG2M)*(THETA(3,101) - THETA(3,102))/2.0
     &    +COS(ANG2M)*(SIN(2.0*THETA(3,101)) - SIN(2.0*THETA(3,102)))/4.
      XCG=XCG*(THETA(2,102) - THETA(2,101))*A(2)**3/AREA
      VOLUME=(THETA(1,102) - THETA(1,101))*XCG*AREA
      VOLUME=-ABS(VOLUME)
      RETURN
      END
C
C======================================================================
C
      SUBROUTINE GRID4(N,MPRINT,XL,XR,F,R,Y,X,XLOW,XHIGH,NN)
C
C======================================================================
C
      DIMENSION XL(9),XR(9),X(9),XLOW(9),XHIGH(9),CENTER(9),SAVEX(9)
      COMMON ITYPE,NO,HANDLG,WRISTL,HXYZ(3,100),HLMN(3,100),PLMN(3,100),
     &       XYZ1(6,3,100),DIR(6,3,100),THETA(6,102),A(6)
C......................................................................
C     GRID SEARCH
C
C     THIS ROUTINE USES A GRID SEARCH TO MAXIMIZE Y(X)
C     WHERE THE DIMENSION ON X IS NOT GREAT THAN 8.
C
C     CALLING PROGRAM REQUIREMENTS
C
C     PROVIDE A DIMENSION STATEMENT AS FOLLOWING:
C
C         DIMENSION XL(9),XR(9),XLOW(9),XHIGH(9),X(9)
```

```
C
C     NOMENCLATURE
C
C     N=NUMBER OF INDEPENDENT VARIABLES
C     MPRINT = 0 CONVERGENCE MONITOR DOES NOT PRINT
C            = 1 CONVERGENCE MONITOR WILL PRINT
C     XL = ORIGINAL LOWER EXTREMITY OF INTERVAL OF UNCERTAINTY
C     XR = ORIGINAL UPPER EXTREMITY OF INTERVAL OF UNCERTAINTY
C     F = FRACTIONAL REDUCTION IN INTERVAL OF UNCERTAINTY DESIRED
C     R = FRACTIONAL GRID REDUCTION UTILIZED
C     Y = MAXIMUM ORDINATE TO MERIT SURFACE DISCOVERED BY GRID SEARCH
C     X = COLUMN VECTOR OF ABSCISSAS CORRESPONDING TO Y
C     XLOW = FINAL UPPER EXTREMITY OF INTERVAL OF UNCERTAINTY
C     XHIGH = FINAL UPPER EXTREMITY OF INTERVAL OF UNCERTAINTY
C     NN = NUMBER OF FUNCTION EVALUATIONS EXPENDED IN GRID SEARCH
C
C     NEED SUPPLY A SUBROUTINE MERIT4(X,Y) FOR GIVEN CALCULATE Y
C
C     ..... PROTECTIO .....
C
      IF(N-8) 11,11,12
   12 WRITE(6,13)N
   13 FORMAT(1X,'*****ERROR MESSAGE SUBROUTINE GRID4*****',
     1/T11,'N=',I3,' GREATER THAN 8.'/)
      RETURN
   11 IF(F-1.0)14,14,15
   15 WRITE(6,16)F
   16 FORMAT(1X,'*****ERROR MESSAGE SUBROUTINE GRID4*****',/
     1 T11,'F = ',E15.8,' GREATER THAN 1.')
      RETURN
   14 DO 50 I=1,N
      IF(XR(I)-XL(I)) 51,51,50
   51 XRR=XR(I)
      XLL=XL(I)
      WRITE(6,52) I,XLL,I,XRR
   52 FORMAT(1X,'*****ERROR MESSAGE SUBROUTINE GRID4*****',/
     1  T11,'XL(',I1,')=',E15.8,'GREATER THAN XR(',I1,')=',E15.8)
      RETURN
   50 CONTINUE
      IF(R-2.0/3.0) 53,54,54
   54 IF(R-1.0) 58,58,53
   53 WRITE(6,55) R
   55 FORMAT(1X,'*****ERROR MESSAGE SUBROUTINE GRID4*****',/
     1  T11,'R=',E15.8,'DOES NOT LIE BETWEEN 2/3 AND 1.0')
      RETURN
C
C     ..... INITILIZE .....
C
   58 NN=0
      SIDE=1.0
      DO 7 I=N,8
    7 CENTER(I)=0.0
      IF(MPRINT) 1,3,1
    1 WRITE(6,2)
    2 FORMAT(1X,'CONVERGENCE MONITOR SUBROUTINE GRIDE4',//
     1 T5,'NN',T11,'SIDE',T22,'Y',T33,'X(1)' ,T44,'X(2)',T55,'X(3)',T66,
     2 'X(4)',T77,'X(5)',T88,'X(6)',T99,'X(7)',T110,'X(8)',T119,
     3 'MAXIMUM Y'/)
    3 DO 4 I=1,N
      CENTER(I)=0.5
      SAVEX(I)=0.5
    4 CONTINUE
      JJ=0
C
C     .....DETERMINE CENTER MERIT ORDINATE.....
C
      CALL UNNORM(N,XL,XR,CENTER)
      CALL REGION(N,XL,XR,CENTER)
      CALL MERIT4(CENTER,YMID)
```

```
          NN=NN+1
          CALL NORMAL(N,XL,XR,CENTER)
          DO 5 I=1,N
          XLOW(I)=0.0
      5   CONTINUE
          YBIG=YMID
C
C         .....DETERMINE MERIT ORDINATES IN GRID, NOTE LARGEST.....
C
     10   STEP=SIDE/3.0
C
C         .....AT EVERY GRID REDUCTION OCCASION, ALTERNATE BETWEEN A......
C         .....SQUARE SURVEY PATTERN AND A STAR SURVEY PATTERN,     ......
C         .....DEPENDING ON ODDNESS OR EVENESS OF JJ.               ......
C
          IF(JJ/2*2-JJ) 600,510,600
C
C         .....SQURE GRID SURVEY.....
C
    510   DO 500 I=1,N
          X(I)=XLOW(I)
    500   CONTINUE
          GO TO (71,72,73,74,75,76,77,78),N
     78   I8=0
     88   I8=I8+1
          X(8)=X(8)+STEP
     77   I7=0
     87   I7=I7+1
          X(7)=X(7)+STEP
     76   I6=0
     86   I6=I6+1
          X(6)=X(6)+STEP
     75   I5=0
     85   I5=I5+1
          X(5)=X(5)+STEP
     74   I4=0
     84   I4=I4+1
          X(4)=X(4)+STEP
     73   I3=0
     83   I3=I3+1
          X(3)=X(3)+STEP
     72   I2=0
     82   I2=I2+1
          X(2)=X(2)+STEP
     71   I1=0
     81   I1=I1+1
          X(1)=X(1)+STEP
          CALL UNNORM(N,XL,XR,X)
          CALL REGION(N,XL,XR,X)
          CALL MERIT4(X,Y1)
          NN=NN+1
          CALL NORMAL(N,XL,XR,X)
          IF(Y1-YBIG) 171,171,6
      6   YBIG=Y1
          DO 30 K=1,N
          SAVEX(K)=X(K)
     30   CONTINUE
    171   IF(I1.EQ.1) GO TO 81
          X(1)=XLOW(1)
          IF(N.EQ.1) GO TO 501
          IF(I2.EQ.1) GO TO 82
          X(2)=XLOW(2)
          IF(N.EQ.2) GO TO 501
          IF(I3.EQ.1) GO TO 83
          X(3)=XLOW(3)
          IF(N.EQ.3) GO TO 501
          IF(I4.EQ.1) GO TO 84
          X(4)=XLOW(4)
          IF(N.EQ.4) GO TO 501
```

```
      IF(I5.EQ.1) GO TO 85
      X(5)=XLOW(5)
      IF(N.EQ.5) GO TO 501
      IF(I6.EQ.1) GO TO 86
      X(6)=XLOW(6)
      IF(N.EQ.6) GO TO 501
      IF(I7.EQ.1) GO TO 87
      X(7)=XLOW(7)
      IF(N.EQ.7) GO TO 501
      IF(I8.EQ.1) GO TO 88
      X(8)=XLOW(8)
      GO TO 501
C
C     .....STAR SURVEY PATTERN.....
C
  600 DO 601 I=1,N
      X(I)=CENTER(I)
  601 CONTINUE
      DO 620 I=1,N
      X(I)=CENTER(I)+STEP
      CALL UNNORM(N,XL,XR,X)
      CALL REGION(N,XL,XR,X)
      CALL MERIT4(X,YPLUS)
      NN=NN+1
      CALL NORMAL(N,XL,XR,X)
      IF(YPLUS-YBIG) 611,611,610
  610 YBIG=YPLUS
      DO 612 K=1,N
      SAVEX(K)=X(K)
  612 CONTINUE
  611 X(I)=CENTER(I)-STEP
      CALL UNNORM(N,XL,XR,X)
      CALL REGION(N,XL,XR,X)
      CALL MERIT4(X,YMINUS)
      NN=NN+1
      CALL NORMAL(N,XL,XR,X)
      IF(YMINUS-YBIG) 614,614,613
  613 YBIG=YMINUS
      DO 615 K=1,N
      SAVEX(K)=X(K)
  615 CONTINUE
  614 CONTINUE
      X(I)=CENTER(I)
  620 CONTINUE
C
C     .....CHECK TO SEE IF GRID SIZE IS SMALL ENOUGH.....
C
  501 JJ=JJ+1
      IF(F-SIDE)32,45,45
C
C     .....GRID DIZE NOT SUFFICIENT SMALL, SELECT LARGEST.....
C     .....ORDINATE LOCATION FROM GRID AND CENTER NEXT    .....
C     .....SMALLER GRID ABOUT THIS POINT.                 .....
C
   32 IF(YBIG-YMID) 44,44,33
   33 YMID=YBIG
      DO 40 K=1,N
      CENTER(K)=SAVEX(K)
   40 CONTINUE
C
C     .....IF PRINTING OF CONVERGENCE MONITOR IS REQUIRED, DO SO.....
C
   44 IF(MPRINT)41,43,41
   41 CALL UNNORM(N,XL,XR,CENTER)
      WRITE(6,42) NN,SIDE,YMID,(CENTER(I),I=1,8),YBIG
   42 FORMAT(1X,I5,11E11.3)
      CALL NORMAL(N,XL,XR,CENTER)
C
```

```
C         .....REDUCE SIZE OF GRIDE AND CONTINUE SEARCH.....
C
    43    SIDE=SIDE*R
          DO 502 I=1,N
          XLOW(I)=CENTER(I)-SIDE/2.0
          XHIGH(I)=CENTER(I)+SIDE/2.0
   502    CONTINUE
          CALL UNNORM(N,XL,XR,XLOW)
          CALL UNNORM(N,XL,XR,XHIGH)
          CALL REGION(N,XL,XR,XLOW)
          CALL REGION(N,XL,XR,XHIGH)
          CALL NORMAL(N,XL,XR,XLOW)
          CALL NORMAL(N,XL,XR,XHIGH)
          GO TO 10
C
C         .....GRIDE SIZE SUFFICIENT SMALL, EXIT FROM SEARCH.....
C
    45    CALL UNNORM(N,XL,XR,SAVEX)
          CALL REGION(N,XL,XR,SAVEX)
          CALL MERIT4(SAVEX,Y)
          NN=NN+1
          CALL NORMAL(N,XL,XR,SAVEX)
          DO 46 K=1,N
          X(K)=SAVEX(K)
          IF(CENTER(K)-SAVEX(K)) 60,61,62
    60    XLOW(K)=CENTER(K)
          XHIGH(K)=CENTER(K)+SIDE/2.0
          GO TO 46
    61    XLOW(K)=CENTER(K)-SIDE/2.0
          XHIGH(K)=CENTER(K)+SIDE/2.0
          GO TO 46
    62    XLOW(K)=CENTER(K)-SIDE/2.0
          XHIGH(K)=CENTER(K)
    46    CONTINUE
          CALL UNNORM(N,XL,XR,XLOW)
          CALL UNNORM(N,XL,XR,XHIGH)
          CALL UNNORM(N,XL,XR,SAVEX)
          CALL UNNORM(N,XL,XR,X)
          IF(MPRINT) 47,49,47
    47    FF=SIDE
          WRITE(6,48) Y,NN,FF
    48    FORMAT(/1X,
         1'MAXIMUM MERIT ORDINATE FOUND DURING SEARCH ..........',E15.8/1X,
         2'NUMBER OF FUNCTION EVALUATIONS USED DURING SEARCH ...',I15,/1X,
         3'FRACT. REDUCTION IN INTERVAL OF UNCERTAINTY EXTANT ..',E15.8,/)
          DO 100 I=1,N
          X1=XLOW(I)
          X2=SAVEX(I)
          X3=XHIGH(I)
          WRITE(6,101) I,X1,I,X2,I,X3
   101    FORMAT(1X,'XLOW(',I1,')=',E15.8,2X,
         1 'X(',I1,')=',E15.8,2X,'XHIGH(',I1,')=',E15.8)
   100    CONTINUE
    49    RETURN
          END
C
C======================================================================
C
          SUBROUTINE NORMAL(N,XL,XR,XNORM)
          DIMENSION XL(9),XR(9),XNORM(9)
          DO 1 I=1,N
          XNORM(I)=(XNORM(I)-XL(I))/(XR(I)-XL(I))
     1    CONTINUE
          RETURN
          END
```

```
C
C========================================================================
C
      SUBROUTINE UNNORM(N,XL,XR,EX)
      DIMENSION XL(9),XR(9),EX(9)
      DO 1 I=1,N
      EX(I)=XL(I)+EX(I)*(XR(I)-XL(I))
    1 CONTINUE
      RETURN
      END
C
C======================================================================
C
      SUBROUTINE REGION(N,XL,XR,X)
      DIMENSION XL(9),XR(9),X(9)
      DO 4 I=1,N
      IF(XL(I)-X(I)) 2,2,1
    1 X(I)=XL(I)
      GO TO 4
    2 IF(XR(I)-X(I)) 3,4,4
    3 X(I)=XR(I)
    4 CONTINUE
      RETURN
      END
```

VITA

YING-CHIEN TSAI

Candidate of the Degree of

Doctor of Philosophy

Thesis: SYNTHESIS OF ROBOTS/MANIPULATORS FOR A PRESCRIBED WORKING SPACE

Major Field: Mechanical Engineering

Biographical:

Personal Data: Born in Taiwan, Republic of China, April 29, 1947, the son of Pai-Ken and Ching-Luan Tsai.

Education: Graduated from Ma-Kung High school, Peng-Hu, Taiwan in 1965; received the Bachelor of Science and Engineering degree in Mechanical Engineering from Cheng Kung University, Tainan, Taiwan, in 1969; received the Master of Science and Engineering degree in Mechanical Engineering from National Cheng Kung University, Taiwan, Republic of China, in 1973; completed the requirements for the Doctor of Philosophy degree at Oklahoma State University in May, 1982.

Professional Experience: Teaching assistant at Tatung Institute of Technology, Taipei, Taiwan, 1970-71; associate investigator at Metal Industrial Research Laboratories (a division of Industrial Technology Research Institute), Kaohsiung, Taiwan, 1973-76; instructor in the Department of Mechanical Engineering, National Cheng Kung University, Tainan, Taiwan, 1976-1978; graduate assistant in the Department of Mechanical and Aerospace Engineering, Oklahoma State University, 1978-82.