

**STRUCTURAL ASPECTS
OF
ROBOT PERFORMANCE**

**M. Y. Zarrugh
Yihfang Jeng**

**Department of Mechanical Engineering and Applied Mechanics
The University of Michigan
Ann Arbor, Michigan 48107**

November 1983

CENTER FOR ROBOTICS AND INTEGRATED MANUFACTURING

Robot Systems Division

COLLEGE OF ENGINEERING

THE UNIVERSITY OF MICHIGAN

ANN ARBOR, MICHIGAN 48109

TABLE OF CONTENTS

1. INTRODUCTION	2
2. WORKSPACE	4
3. CUTTING PLANE ALGORITHM	8
4. ILLUSTRATIVE EXAMPLE	10
5. CONCLUSIONS	15
6. ACKNOWLEDGEMENTS	15
7. REFERENCES	15

ABSTRACT

There is a strong interrelationship between the kinematic structure of a manipulator and the nature of the tasks it can perform, particularly in terms of reach and orienting capabilities of the manipulator. It is the purpose of this report to discuss the effects of the manipulator's structural factors on performance. The performance aspects addressed are positioning and orienting capability of the hand, shape and size of reach envelope or workspace and the quality of the workspace as to the presence of holes and voids. In addition the paper presents an efficient algorithm based on optimization techniques for computing the workspace of any manipulator. The algorithm determines the boundary of the workspace on an arbitrary cutting plane. The procedure does not require small joint motion or ideal joint assumptions as customary in other existing algorithms.

1. INTRODUCTION

There is a strong interrelationship between the structure of a manipulator and its performance. One of the main aspects of a manipulator performance is the shape and size of its workspace which is the region in space containing all points that can be reached by the end point of the manipulator. The shape of the workspace dictates how the work environment is arranged to place manipulated objects and points to be visited within reach. Thus, the workspace serves as an important criterion for evaluating performance.

Roth [1] was the first to point out the possibility of predicting aspects of manipulator performance from its geometry and discussed the effects of the geometry on position of the end point, coupling between this position and orientation of the hand and workspaces. Since this early work, much interest has been generated in issues related to workspace, such as algorithms for computing workspace [2,3,4] and criteria for evaluating the work space [5,6,7]. Kumar and Waldron [2] developed an algorithm for tracing the boundaries of workspaces of manipulators. The closed form algorithm was based on the fact that when a manipulator assumes maximum extension under the action of a force applied at the end point, the line of action of the force (wrench axis) must intersect all joint axes. Some difficulties were encountered in implementing the algorithm numerically. Namely the high sensitivity of the solution to the angle of the applied force and the limited range of this angle over which real solutions exist.

Tsai and Soni [3] dealt with the problem of determining the workspace of an nR robot as a linear program of computing the extreme distance that can be reached by the end point on a specified cutting plane. In this algorithm, the end point is gradually moved from a known initial configuration to the specified plane, while each joint undergoes a small rotation. The end point is then moved on the designated plane using linear programming until the boundary of the

workspace is reached without violating the accuracy requirements that joint rotations must not exceed certain limits. The last step in the Tsai and Soni [3] algorithm is to trace the contour of the workspace by searching on the cutting plane for points of intersection with the workspace.

Another algorithm was presented by Lee and Yang [4] for outlining the workspace of a manipulator containing an nR ideal joints (i.e. having unlimited rotations). Their method considers the relative displacement between two consecutive joints at a time in a recursive manner starting with the joints closest to the free end. At first, the end point is rotated about the n th (last) axis to form a closed curve which is then rotated about the $n-1$ axis forming the first relative workspace. Subsequently in each step k ($k=2$ to $n-1$) the workspace found in the previous step is rotated about axis $n-k$ to generate a new relative workspace accounting for motions at joints n to $n-k+1$ while the rest of the joints are held fixed. This formulation offers only an implicit analytical definition of the workspace. But computation of the workspace itself is accomplished via detecting the curve of intersection between the workspace and a particular cutting plane. Such a procedure involves the scanning of all joint angles at small step sizes which renders the procedure inefficient and places limitation on computational precision.

Another aspect of geometric performance of manipulators which received some attention is the quality of workspace. One issue, hand dexterity is dealt with by Kumar and Waldron [5] and Gupta and Roth [6]. Kumar and Waldron presented the concept of dexterity as the ability of a manipulator to orient an object in any arbitrary direction at each point visited by the end point. In addition they discussed some of the kinematic arrangements of a $6R$ robot which produce maximum dexterity [5]. For example, they asserted that a wrist consisting of three mutually perpendicular and concurrent axes is especially superior in dexterity. Gupta and Roth [6] discussed the effects of hand link length on dexterity and observed that the largest dextrous (or primary) workspace is achieved when the hand link is as short as possible.

In order to assess and simulate these and other geometrical aspects of robot performance, there is a need for more efficient methods for determining robot workspaces. Such methods will permit evaluation of performance under different conditions of kinematic and structural parameters. Ultimately, the comprehensive evaluation will lead to the development of meaningful performance criteria for dexterity and accuracy within the workspace and to the use of the measures to predict improved kinematic arrangements for given tasks. It is the purpose of this paper to outline an efficient procedure for determining the workspace of any robot containing any number of real joints (joints with motion limits). The proposed method does not necessitate small joint motion assumptions and thus it is fast and does not have computational accuracy limitations.

2. WORKSPACE

The workspace of a manipulator has been defined as "the aggregate of all possible positions of a point attached to the free end of the manipulator," [6]. If a manipulator contains n single-degree of freedom joints (Fig. 1) the workspace can be computed point-by-point through an n -step coordinate transformation process involving the familiar homogeneous transformation \mathbf{A}_k relating coordinates of frame k, \mathbf{x}_k to those of \mathbf{x}_{k+1} , as depicted in Fig. 2. This can be represented by

$$\mathbf{x}_k = \mathbf{A}_k \mathbf{x}_{k+1} \quad k = 1, \dots, n \quad (1)$$

Here the coordinate transformation matrix \mathbf{A}_k takes the following form for revolute joint:

$$\mathbf{A}_i = \begin{bmatrix} C_i & -S_i \cos \alpha_i & S_i \sin \alpha_i & a_i C_i \\ S_i & C_i \cos \alpha_i & -C_i \sin \alpha_i & a_i S_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

where $C_i = \cos \theta_i$,

$$S_i = \sin\theta_i$$

$$\theta_i = \underline{ith} \text{ joint rotation}$$

$$\alpha_i = \underline{ith} \text{ joint axis twist angle (Fig. 2)}$$

$$a_i = \underline{ith} \text{ joint axis lateral offset (Fig. 2)}$$

$$d_i = \underline{ith} \text{ joint axis axial offset (Fig. 2)}$$

The transformation matrix \mathbf{A}_i for a prismatic (sliding) joint is as follows:

$$\mathbf{A}_i = \begin{bmatrix} C_i & -S_i \cos\alpha_i & S_i \sin\alpha_i & 0 \\ C_i & C_i \cos\alpha_i & -C_i \sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

All symbols have the same meaning as in equation (2) except d_i , which here designates linear displacement along the axis.

The most common kinematic structure of a manipulator consists of six joints divided into two groups: the arm proper group and the wrist. The wrist generally consists of three orthogonal and concurrent revolute axes to provide maximum hand dexterity [5]. The arm contains the other three joints which can consist of any combination of revolute and prismatic joints. This set of joints is to a large extent responsible for the shape, size and geometric characteristic of the workspace. Although the algorithm we are proposing is applicable to any combination of revolute and prismatic joints, only robots containing one or no sliding joints are considered since the presence of more than one sliding joint will only simplify the procedure of computing the workspace.

The workspace and particularly its bounding surfaces are of interest in robot applications. Manipulated objects must be within reach of the manipulator and the workspace can serve as a measure of suitability and effectiveness of a particular kinematic arrangement for a particular application. Thus, it is of interest to determine the workspace or its bounding surface as quickly

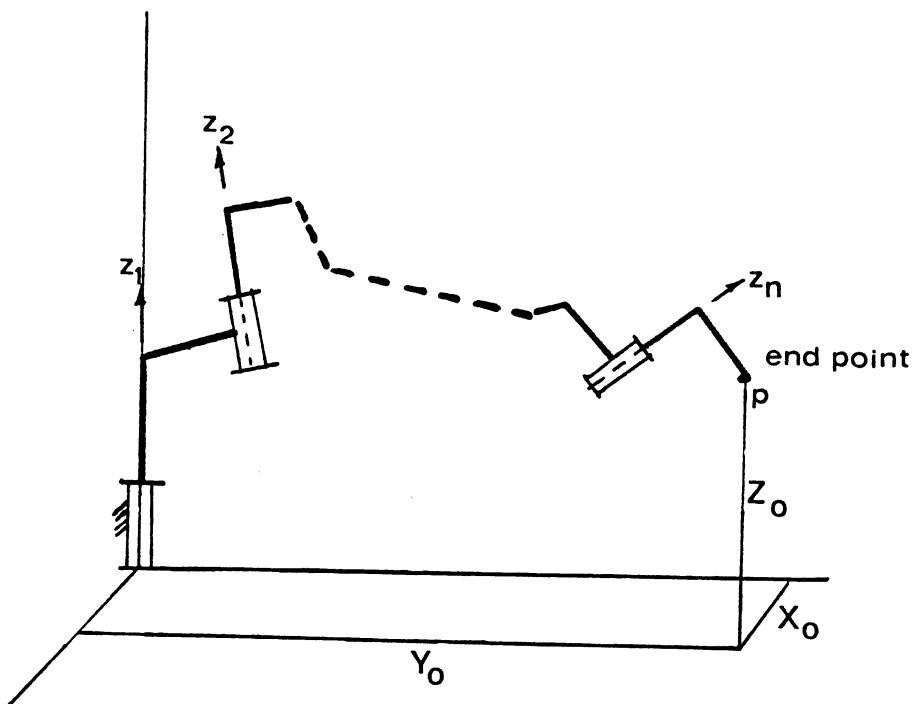


Figure 1. A Manipulator with n -revolute joints.

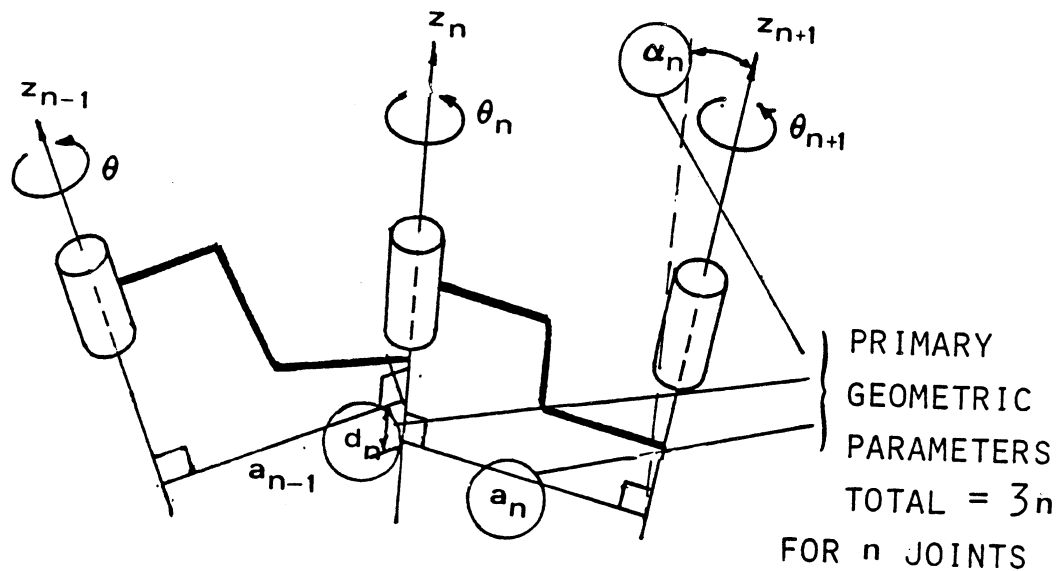


Figure 2. Parameters of Relative Coordinate Transformation Between Two Successive Joints

as possible so that various kinematic arrangements are optimized for a desired workspace shape. The boundary of the workspace cannot be described analytically in explicit form, but it can be described geometrically in terms of its curves of intersection with arbitrary planes.

3. CUTTING PLANE ALGORITHM

The method, first described by Tsai and Soni [3], employs optimization technique to determine the intersection curve between an arbitrary cutting plane and the workspace. With sequential movement of the cutting plane, the entire workspace boundary can be determined as a family of intersection curves. As illustrated in Fig. 3, the algorithm begins with locating the end point p on the specified cutting plane via linear programming. Secondly, the hand is moved on the plane as far as it can reach i.e. to the boundary. Lastly, the end point is moved on the plane tracing the intersection curve. Tsai and Soni acknowledged several limitations in their algorithm, the most serious of which is the small joint displacement requirement for obtaining acceptable computational accuracy.

Our approach is similar to that of Tsai and Soni, in that the work surface is computed in terms of intersection curves with a cutting plane. However, our approach avoids the small angle limitation eliminating accuracy dependence on step size. In addition, the boundary tracing procedure is made more efficient by using polynomial extrapolation.

The coordinates of the end point (X_o, Y_o, Z_o) first expressed in terms of the unknown joint displacements θ_i or d_i using the coordinate transformation matrices of equations (1) and (2)

$$\begin{aligned} X_o &= f_1 (\theta_1, \theta_2, \dots, \theta_n) = f_1 (\theta) \\ Y_o &= f_2 (\theta_1, \theta_2, \dots, \theta_n) = f_2 (\theta) \\ Z_o &= f_3 (\theta_1, \theta_2, \dots, \theta_n) = f_3 (\theta) \end{aligned} \quad (3)$$

θ_i can be replaced with d_i for prismatic joints.

The problem of moving the end point to the boundary is formulated as a nonlinear programming problem to maximize or minimize distance on the specified cutting plane:

maximize or minimize

$$X_o = f_1 (\theta)$$

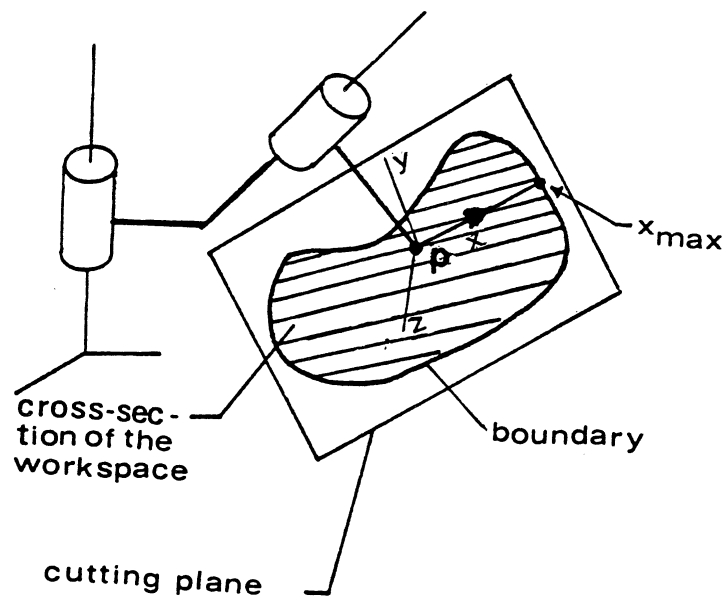


Figure 3. Schematic Representation of Intersection Plane Method for Computing Workspace

subject to

$$\begin{aligned}
 Y_o &= f_2(\theta) = a \\
 Z_o &= f_3(\theta) = b \\
 (\theta_i)_{\min} &\leq \theta_i \leq (\theta_i)_{\max} \quad i = 1, \dots, n
 \end{aligned}
 \tag{4}$$

Here a or b are constants defining the chosen cutting plane and a certain line along which X_o is extremized. Of course, any of the other coordinates Y_o and Z_o of the end point can be extrem-

ized here instead of X_o . The last set of constraint equations specify the range of motion of the joints.

Once the optimization problem is solved via any standard nonlinear programming package, the end point is now at the workspace boundary. The next step of the procedure is to trace the intersection curve. Typically these intersection curves are closed contours and two symmetric contours may exist on one cutting plane. Thus, a special care is necessary to devise an effective strategy to follow such double-valued and multiple curves.

4. ILLUSTRATIVE EXAMPLE

Consider the manipulator discussed in Tsai and Soni [3] consisting of six revolute joints. Table 1 shows the kinematic parameters of the manipulator and a schematic representation of the manipulator appears in Fig. 4. Using the parameters listed in Table 1, coordinate transformation matrices are written in the same form as equation (1). The end point p is given in the fixed coordinate system (ground) by successively transforming it from the origin of system 4:

$$p_o = A_1 A_2 A_3 A_4 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (5)$$

The coordinates X_o, Y_o and Z_o are the same as those defined in equation (3) which indicates that equations (5) and (3) are equivalent. Substitution for the transformation matrices and performing the matrix products yields

$$X_o = qC_1C_2 - rC_1S_2 + 0.4C_2C_1 \quad (6)$$

$$Y_o = qS_1C_2 - rS_1S_2 + 0.4C_2S_1 \quad (7)$$

$$Z_o = qS_2 + rC_2 + 0.4S_2 \quad (8)$$

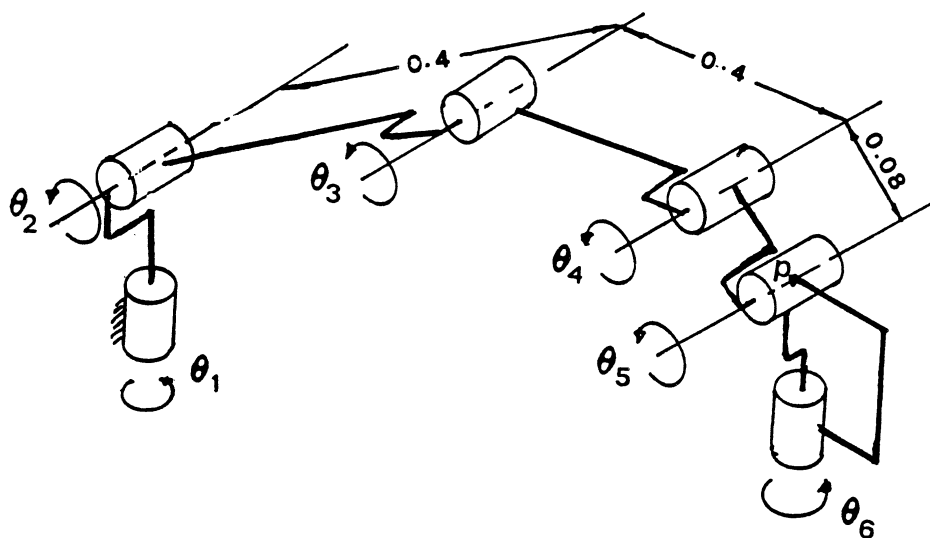


Figure 4. Manipulator Structure for the Example

i	a_i	d_i	α_i	θ_i range
1	0.0	0.0	90 deg	-120 to 120 deg
2	0.4	0.0	0 deg	0 to 90 deg
3	0.4	0.0	0 deg	-120 to 0 deg
4	0.08	0.0	-90 deg	-120 to 120 deg
5	0.0	0.0	90 deg	-30 to 210 deg
6	0.0	0.0	(0 deg)	-240 to 240 deg

Table 1. Kinematic Parameters of the 6R Manipulator shown in Fig. 4

In which $C_i = \cos\theta_i$,

$S_i = \sin\theta_i$,

$$q = 0.08(C_3C_4 + S_3S_4) + 0.4C_3$$

$$r = 0.08(S_3C_4 + C_3S_4) + 0.4S_3$$

Subsequently, the optimization problem is formulated and solved. Suppose we wish to determine the boundary of the workspace on the $X_o - Z_o$ plane for which $Y_o = 0$ (see Fig. 4). This implies that θ_i must equal 0 for equation (7) to be satisfied. Let us also assume that the first point to be determined is along a line for which $Z_o = \text{const}$. Thus the optimization problem is

maximize X_o

subject to $Y_o = 0$, $Z_o = Const.$

and the joint motion limits shown in Table 1, Column 4.

For $Z_o = 0$, the extreme value of X is 0.88. which is indicated as point A in Fig. 5 which shows the entire intersection curve on the $X_o - Z_o$ plane of the manipulator defined in Fig. 4. Results of the optimization can be verified very quickly by inspection of equation (6). Since $Y_o = 0$ requires that $S_1 = \sin\theta_1$ in equation (7) be equal to zero which means that $C_1 = \cos\theta_1 = 1.0$.

$$\text{Thus} \quad X_o = (q + 0.4)C_2 - rS_2$$

for which X_o is at its maximum value when C_2 is at its maximum value and S_2 is at its lowest value. Thus θ_2 must = 0 which implies $C_2 = 1.0$ and $S_2 = 0$. or

$$X_o = q + 0.4 = 0.08(C_3C_4 - S_3C_4) + 0.4C_3 + 0.4$$

Again X_o is maximized if C_4 is maximized subject to the motion limits. Using $C_4 = 1.0$ and $S_4 = 0$ we have

$$X_o = 0.48C_3 + 0.4$$

Since $C_3 = 1$ maximizes X_o , maximum $X_o = 0.88$ as obtained in the optimization solution.

Two more points are determined by incrementing Z_o twice and solving the optimization problem. Through the known three points a quadratic function is fitted. The function is used to predict a fourth point for a new value of Z_o . Then the second, third and fourth point are used to guess a fifth point. The process continues till all points on the boundary are determined, for increasing values of Z_o up to the maximum value of Z_o . The process is repeated by decrementing Z_o and minimizing X_o until minimum Z_o is obtained. This traces the intersection curve

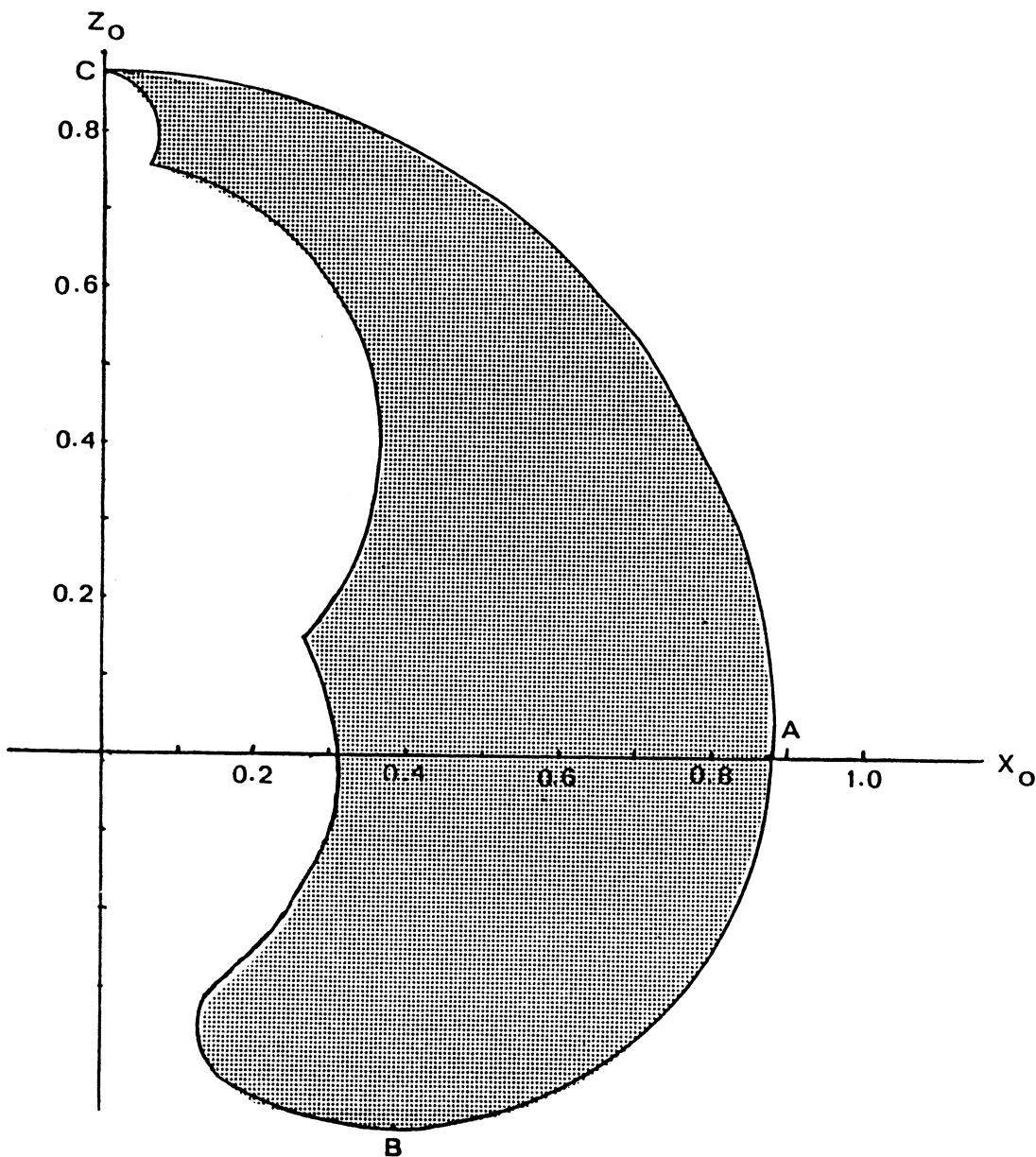


Figure 5. Workspace of the manipulator depicted in Fig. 4

from point C to point B in Fig. 5. The last segment from the bottom point B to the starting point A are computed by incrementing Z_0 and maximizing X_0 till point A is reached completing the entire contour curve.

5. CONCLUSIONS

Through the algorithm described in the report, the boundary of the workspace of any manipulator can be determined on any given plane. The algorithm, based on nonlinear programming converges rapidly because initial guesses are generated on the basis of extrapolation from previous points. The strategy of search along the intersection curve is simple and applicable to any manipulator. The algorithm does not require that the joints be ideal (i.e. without limits) and thus realistic joints with motion constraints can be considered. Unlike a similar algorithm of Tsai and Soni [3], this method does not necessitate small joint approximation.

6. ACKNOWLEDGEMENTS

The authors extend their appreciation to the Center for Robotics and Integrated Manufacturing at the University of Michigan for technical and financial assistance.

7. REFERENCES

- (1) Roth B., "Performance Evaluation of Manipulators from a Kinematic Viewpoint," National Bureau of Standards Special Publication No. 459, pp 39-61, 1976.
- (2) Kumar A. and K. J. Waldron, "The Workspaces of a Mechanical Manipulator," *Trans. ASME, J. Mech. Design*, Vol. 103, pp. 665-672, July 1981.

- (3) Tsai, Y. C. and A. M. Soni, "An Algorithm for the Workspace of a General n-R Robot," *Trans. ASME, J. Mech. Design, Vol. 105, pp. 52-57, March, 1983.*
- (4) Yang, D.C.H. and T. W. Lee, "On the Workspace of Mechanical Manipulators", *Trans. ASME, J. Mech. Design, Vol. 105, pp. 62-69, March 1983.*
- (5) Kumar, A. and K. J. Waldron, "The Dextrous Workspace," ASME Paper No. 80-DET-108, Sept. 1980.
- (6) Gupta, K.C. and B. Roth, "Design Consideration for Manipulator Workspace," *Trans. ASME, J. Mech. Design, Vol. 104, pp. 704-712, Oct. 1982.*
- (7) Lee, T.W. and D.C.H. Yang, "On the Evaluation of Manipulator Workspace," *Trans. ASME, J. Mechanisms, Transmissions and Automation in Design, Vol. 105, pp. 70-77, March 1983.*

UNIVERSITY OF MICHIGAN



3 9015 03527 6149