NSF Engineering Research Centers Program NSFD CDR 8803012.

References

1 Calson, J. H., 1985, "Harmonic Drives for Servomechanisms," *Machine Design*, Vol. 57, No. 1, pp. 102–106.

2 Chang, S. L., 1991, "Redundant-Drive Backlash-Free Robotic Mechanisms: Mechanisms Creation, Analysis, and Control," Ph.D. dissertation, Mechanical Engineering Department, The University of Maryland, College Park, MD 20742.

3 Chang, S. L., and Tsai, L. W., 1990, "Topological Synthesis of Articulated Gear Mechanisms," *IEEE Trans. on Robotics and Automation*, Vol. 6, No. 1, pp. 97–103.

4 Chang, S. L., and Tsai, L. W., 1993, "On the Redundant-Drive Backlash-Free Robotic Mechanisms," ASME Journal of Mechanical Design, Vol. 115, No. 2, pp. 247–254.

5 Dagalakis, N. G., and Myers, D. R., 1985, "Adjustment of Robot Joint Gear Backlash Using Robot Joint Test Excitation Technique," *The International J. of Robotics Research*, Vol. 4, No. 2.

6 Klein, C. A., and Huang, C. H., 1983, "Review of Pseudo-Inverse Control for Use with Kinematically Redundant Manipulators," *IEEE Trans. on System, Man, and Cybernetics*, Vol. SM513, pp. 245–250.

7 Michalec, G. W., 1966, *Precision Gearing: Theory and Practice*, John Wiley & Sons, Inc., New York.

8 Paul, R. P., 1981, *Robot Manipulators: Mathematics, Programming, and Control*, MIT Press, Cambridge, Mass.

9 Thomas, M., and Tesar, D., 1982, "Dynamic Modeling of Serial Manipulator Arms," ASME Journal of Mechanisms, Transmissions, and Automation in Design, Vol. 104, pp. 218–228.

10 Tsai, L. W., 1988, "The Kinematics of Spatial Robotic Bevel-Gear Trains," *IEEE J. of Robotics and Automation*, Vol. 4, No. 2, pp. 150–156.

General Algorithm for Automatic Generation of the Workspace for *n*-link Planar Redundant Manipulators

Sang-Joo Kwon[1], Youngil Youm[2], and Wan Kyun Chung[3]

An efficient algorithm to simulate the workspace for a n-link redundant manipulator with rotating base is developed. The unique feature of the algorithm is its ability to automatically generate the binary angle selection code for the given manipulator. This code system makes it possible to avoid repetitions in calculating positions and plotting arcs for the workspace and thus saves considerable computing and graphic time. Two types of workspace are possible using this algorithm. Numerical examples including the workspace of human arm motion and that of a 20-link system are demonstrated to illustrate the effectiveness of the algorithm.

1 Introduction

The concept of workspace is utilized as a useful index to evaluate the performance of robotic manipulators. The workspace characterizes different types of manipulators and also represents an important criterion for the optimum design of manipulators [1]. There were several approaches to obtain the boundary surfaces of manipulator workspace. Kumar and Waldron [2] developed a numerical technique for tracing the boundary surfaces. Lee and Yang [3] used the grid scanning technique and developed an algorithm called kinematic analysis of manipulators (KAM). Cwaikala and Lee [4] developed an efficient algorithm, based on optimum path search, for the grid scanning technique. Kohli and Spanos [5] proposed an analytical method using polynomial discriminants.

In this paper, the *angle selection code* to generate the workspace for *n*-link manipulator with rotating base is developed. Using this angle selection code, the workspace of the manipulator with any number of links can be generated.

[1]Researcher, Agency for Defence Development, P.O. Box 35, 305-600 Taejeon, Korea [2]Professor, [3]Associate Professor, Mechanical Engineering Department, Pohang University of Science and Technology, Pohang P.O. BOX 125, 790-600 Pohang, Korea.

Contributed by the Mechanisms Committee for publication in the Journal of Mechanical Design. Manuscript received Feb. 1991; revised Aug. 1993. Associate Technical Editor: B. S. Thompson.

Table 1 Angle selection code for a 4-link boundary workspace

link 1	0	0	0	1	2	2	2	1
link 2	0	0	1	2	2	2	1	0
link 3	0	1	2	2	2	1	0	0
link 4	1	2	2	2	1	0	0	0

2 Development of the Angle Selection Code

A binary angle selection code is introduced in this section. This code system makes it possible to avoid repetitions in calculating kinematic positions and plotting arcs for the workspace. This means that the maximum workspace of the reference point can be obtained using this code system with minimum computation time.

The angle selection code is categorized into two cases according to the workspace types. If only the outer boundary contour of the workspace is required, this type will be called the *boundary workspace* (BW). However, if the interior hierarchical arcs denoting the subworkspaces (i.e., $W_k(P)$ in [1]) are to be plotted as workspace contours, this type will be called the *hierarchical workspace* (HW).

2.1 Angle Selection Code for the Boundary Workspace. The general procedures to generate the boundary workspace can be stated as follows:

(*1*) Locate links at their minimum joint angles.

(*2*) Rotate links from the last link to the first link within their range of rotation sequentially, that is, from the minimum to the maximum joint angles.

(*3*) Conversely, rotate links from the last link to the first link within their range of rotation, that is, from the maximum to the minimum joint angles.

(*4*) In steps 2 and 3, the minimum and the maximum radius of arc from the current rotating joint to the reference point on the end-effector of the manipulator should be computed and their corresponding arcs are also to be plotted.

If the above procedures are coded, the generated angle selection code for BW of 4-link will be as shown in Table 1, where the row number corresponds to the link number and the number of columns is just two times the number of links. So the code system of BW can be represented by an ($n \times 2n$) matrix. Each column of this code is composed of the numbers 0, 1, and 2. The number 0 and 2 denote the minimum and the maximum rotation angle of a joint, respectively, and number 1 describes the rotation of the joint within a specified range. The column of this code is called sequentially in the computer program and the plotting procedures use the information in the column. When a column is called, an arc is plotted by rotating the joint corresponding to code value 1 within its range of rotation while the other joints are fixed at their specified joint angles according to their code values.

As illustrated in Table 1, the code systems have a *band* of 1 values in the diagonal lines. The physical meaning of this pattern is the orderly rotation of joints from their minimum to maximum (or maximum to minimum) joint angles. This tendency is simple and the code system for general *n*-link can be constructed easily using the above rule.

2.2 Angle Selection Code for the Hierarchical Workspace. The hierarchical workspace contains more informations about a manipulator in that it includes all the subworkspaces. The

Table 2 Angle selection code for a 4-link hierarchical workspace

link 4	1 0 1 2	0	1 0 1 2	2	0	1 0 1 2	0	1 0 1 2	2	2
link 3	0 1 2 1	0	0 1 2 1	2	0	0 1 2 1	0	0 1 2 1	2	2
link 2	0 0 0 0	1	2 2 2 2	1	0	0 0 0 0	1	2 2 2 2	1	2
link 1	0 0 0 0	0	0 0 0 0	0	1	2 2 2 2	2	2 2 2 2	2	1

procedures to generate the hierarchical workspace for general n-link system is as follows:

(*1*) Repeat the following steps 2 and 3 for all numbers of m as $2, 3, 4, \ldots, n-2, n-1$.

(*2*) Plot the workspace for the last m links (i.e., $W_{n-m+1}(P)$ in [1]) while the $(n-m)$th link is at its minimum joint angle (code value 0) and again at its maximum joint angle (code value 2). The other links 1 to $(n-m-1)$ remain as their minimum joint angles (code value 0).

(*3*) To connect the workspaces apart from each other in the above step, plot arcs when the $(n-m)$th link moves within a specified range of rotation (code value 1) while the last m links are at their minimum joint angles and again at their maximum joint angles. The other links 1 to $(n-m-1)$ remain as their minimum joint angles.

Based on the above procedures, the angle selection codes for the 4-link HW can be made out manually as shown in Table 2. From the example in Table 2, we can extract the following formula to calculate the number of columns:

$$size[n] = 2 \times size[n-1] + 2 \qquad (1)$$

where $size[n]$ defines the number of columns for an n-link system and $size[1]$ is defined as 1. Thus the code system for HW can be represented by a $(n \times size[n])$ matrix.

2.3 Generation of the Angle Selection Code for the Hierarchical Workspace. In the array of Table 2, the bold characters **1** are defined as the *starting points* to store code values. The first problem is to find out the logic to detect the position of these starting points.

When we consider each row (except the 1st row) of the code array in Table 2, we find out that the number of starting points in a row is equal to 2^{n-row} and the position of starting points can be expressed analytically as,

$$for(1 \le k \le 2^{n-row})(2k-1) \times hs[row] + plus[k] \qquad (2)$$

where $hs[n]$ is the half value of $size[n]$ of Eq. (1) and the $plus[k]$ gives the number added to the regular term $(2k-1) \times hs[row]$ of the kth starting point of a row. The $plus[k]$ goes in the sequence, 0, 1, 3, 4, 7, 8, 10, 11, 15, ..., which seems to be irregular at a glance. However, with careful examination, the generation rule of the sequence can be found. Considering the series of the function $plus[k]$ of each row, the following two facts are found out:

(*1*) There exists a *summation rule* in each row as: $plus[1] + plus[max\ k] = plus[2] + plus[max\ k-1] = plus[3] + plus[max\ k-2] = \ldots$ *etc.* where $max\ k$ is the maximum number of k for a certain row. This fact can explain why the imperfect regularity appears, which is hidden in the sequence which goes approximately as 0, 1, 2, 1 in each row of the angle selection code for HW.

(*2*) The series of $plus[k]$ for the 2nd row contains the array for the other remaining rows hierachically. This fact is physically reasonable when we consider the structure of HW and make it possible to generate all $plus[k]$ for n-link system in a recursive loop.

By combining the above two facts, the series of $plus[k]$ for an n-link system can be generated. Successively, the positions of all starting points can be determined. Finally, the angle selection for HW of the n-link system can be generated automatically by repeating the fill up process of Fig. 1 (where the number of 2's of the 3rd step is $size[i-1]$ for the ith row) from every starting point in each row after initializing the code array by storing zero values in all the places of the code array except the first row where value 1's are stored.

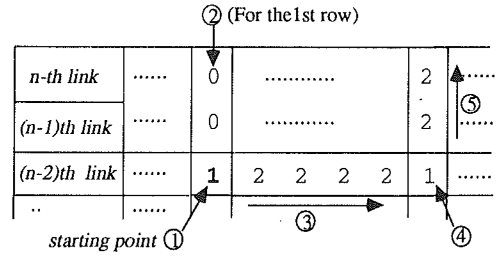


Fig. 1 Illustration of the storage process of code values for the hierarchical workspace

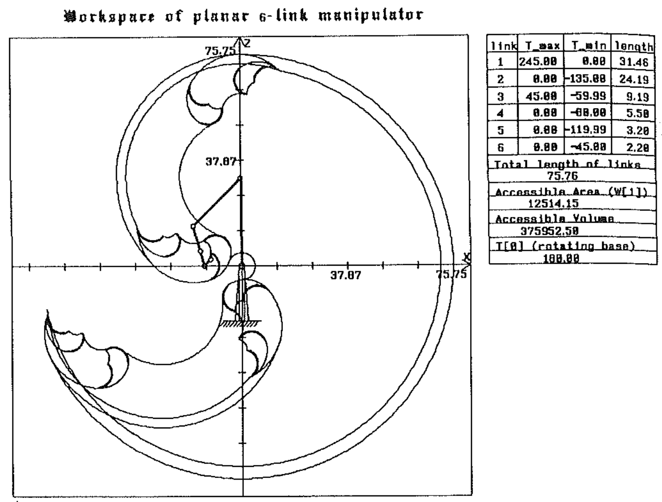


Fig. 2 The maximal workspace (HW) of human arms

3 Numerical Examples

The workspace simulator based on the generation of the angle selection code had been developed. The required computation time for workspace plotting was only a few seconds in most cases on the SUN 3 computer. For example, it took approximately 5 sec. and 30 sec. for BW of 6-link and 20-link, respectively. For HW, it took, for example, approximately 40 sec. for 6-link and 2 min. for 9-link. It is, therefore, considered to be fast enough to evaluate the kinematic performance of manipulators using the proposed binary code system.

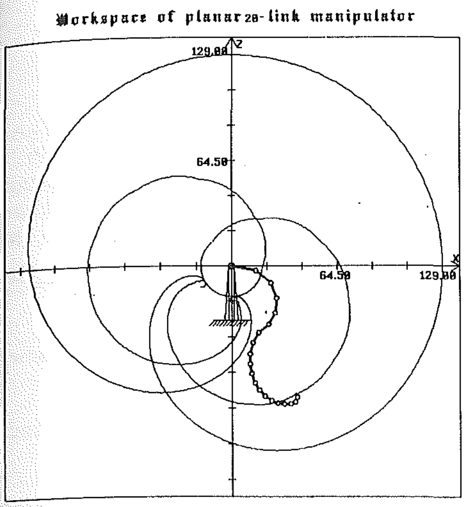
Followed by plotting, the *accessible area and volume* of workspace were calculated by using the concept of the equivalent area and volume [6].

Example 1: the maximal workspace of a human arm: Figure 2 represents the hierarchical workspace of a human arm in the sagittal plane, from the shoulder joint to the fingertip, as a planar 6-link open-loop system with approximate specifications shown in the right box of the figure.

Example 2: hypothetical 20-link manipulator: Figure 3 is the boundary workspace for the hypothetical manipulator with 20 links whose specifications were given arbitrarily. It is also possible to calculate the workspace of the system with infinite number of links.

4 Conclusions

A new algorithm generating the workspace of an n-link planar redundant manipulator with rotating base has been developed. We proposed a new approach of binary angle selection code system which is composed of a set of notations of joint variables to generate workspace. It was proved that



link	T_max	T_min	length
1	100.00	-45.00	15.00
2	29.99	-29.99	12.00
3	29.99	-29.99	10.00
4	25.00	-25.00	9.00
5	25.00	-25.00	8.00
6	20.00	-20.00	8.00
7	20.00	-20.00	7.00
8	14.99	-14.99	7.00
9	14.99	-14.99	6.00
10	10.00	-10.00	6.00
11	10.00	-10.00	6.00
12	10.99	-10.99	5.00
13	12.00	-12.00	5.00
14	13.00	-13.00	5.00
15	14.99	-14.99	4.00
16	10.00	-10.00	4.00
17	12.00	-12.00	4.00
18	20.00	-20.00	3.00
19	45.00	-45.00	3.00
20	135.00	-135.00	2.00
Total length of links			
129.00			
Accessible Area (W[1])			
30961.41			
Accessible Volume			
6004990.00			
T[0] (rotating base)			
300.00			

Fig. 3 The boundary workspace of the 20-link manipulator

the developed angle selection code system is very efficient in plotting workspaces, especially, for a system with a large number of degrees of freedom. Numerical examples were shown for the manipulators with revolute joints and it was shown that the developed algorithm works well for any n-link planar system.

References

1 Gupta, K. C., and Roth, B., 1982, "Design Consideration for Manipulator Workspace," ASME Journal of Mechanical Design, Vol. 104, Oct., pp. 704–711.

2 Kumar, A., and Waldron, K. J., 1981, "The Workspace of a Mechanical Manipulator," ASME Journal of Mechanical Design, Vol. 103, No. 3, July, pp. 665–672.

3 Yang, D. C. H., and Lee, T. W., 1983, "On the Workspace of Mechanical Manipulators," ASME Journal of Mechanisms, Transmissions, and Automation in Design, Vol. 105, Mar., pp. 62–69.

4 Cwaikala, M., and Lee, T. W., 1985, "Generation and Evaluation of a Manipulator Workspace Based on Optimum Path Search," ASME Journal of Mechanisms, Transmissions, and Automation in Design, Vol. 107, June, pp. 245–255.

5 Kohli, D., and Spanos, J., 1985, "Workspace Analysis of Mechanical Manipulators using Polynomial Discriminants," ASME Journal of Mechanisms, Transmissions, and Automation in Design, Vol. 107, June, pp. 209–215.

6 Youm, Y., and Yih, T. C., 1986, "The Kinematic Spaces of Planar n-R Open-Loop System with Rotating Base," ASME Paper, 86-DET-98, *The Design Engineering Conference*, Columbus, OH, October 5–8.

7 Kwon, S. J., 1990, "Self-Collision Avoidance and Workspace Generation for Redundant Manipulators," M.S. thesis, Department of Mechanical Engineering, Pohang Institute of Science & Technology, Dec.

Kinematic Analysis and Design of Articulated Manipulators with Joint Motion Constraints

T. A. Dwarakanath,[1] A. Ghosal,[1] and U. Shrinivasa[1]

For an articulated manipulator with joint rotation constraints, we show that the maximum workspace is not necessarily obtained for equal link lengths but is also determined by the range and mean positions of the joint motions. We present expressions for sectional area, workspace volume, overlap volume and work area in terms of link ratios, mean positions and ranges of joint motion. We present a numerical procedure to obtain the maximum rectangular area that can be embedded in the workspace of an articulated manipulator with joint motion constraints. We demonstrate the use of analytical expressions and the numerical plots in the kinematic design of an articulated manipulator with joint rotation constraints.

[1]Department of Mechanical Engr., Indian Institute of Science, Bangalore—560 012, India.

Contributed by the Mechanisms Committee for publication in the Journal of Mechanical Design. Manuscript received June 1992; Sept. 1993. Associate Technical Editor: G. L. Riuzel.

1 Introduction

The concept of a manipulator workspace and the workspace of the manipulator regional structures has been studied extensively and is well understood (Roth, 1975; Kumar and Waldron, 1980; Hansen et al., 1983; Sugimoto and Duffy, 1981a,b; Tsai and Soni, 1984). Most of the results assume unconstrained joint motions and are not directly applicable to industrial manipulators which usually have limited joint motions. Several researchers have discussed the design of manipulators based on workspace considerations (Tsai and Soni, 1984; Paden, 1986). Again, these are based on the assumption of complete joint rotation. For a manipulator with joint constraints, Rastegar and Deravi (1987) discussed the number of possible configurations as a function of the "overlap" and the "location" of the joint rotation range. Gupta (1986) pointed out the effect of rotation ranges of links on the number of possible configurations. Their treatment is, however, not analytical. Vijaykumar et al. (1985) have discussed the effect of joint limits on workspace and dexterity. However, the design problem is not addressed. In this paper, we present analytical expressions for sectional area, volume, work area and bounding surfaces for an articulated manipulator. We also present a method to choose link lengths, joint ranges and their mean positions to obtain a workspace of given size.

In manipulator applications, such as welding and painting, the total workspace is of less importance. It is more important to know the optimum location and orientation of the part in the workspace since this would allow the operator or task planner to compute how much painting or welding can be done in one setting of the part and the manipulator and thus maximize the use of the manipulator. Very little is known about embedding regular shapes in the workspace. We present an algorithm to obtain the maximum rectangle that can be embedded in the workspace of an articulated manipulator with joint motion constraints.

2 Articulated Manipulators with Joint Constraints

To study the workspace of an articulated manipulator, shown schematically in Fig. 1, we consider the so-called wrist point. Denoting the wrist point by $(x_w, y_w, z_w)^T$, and by using the well known 4×4 matrix transformations (Paul, 1981), we can write

$$(x_w, y_w, z_w) = \{ (a_2c_2 + a_3c_{23})c_1, \quad (a_2c_2 + a_3c_{23})s_1, \quad a_2s_2 + a_3s_{23} \} \quad (1)$$

where a_2, a_3 are the link lengths and $c_{(\cdot)}$, $s_{(\cdot)}$ denote cosine and sine of angle $(\cdot)$, respectively. We can eliminate two out of the three θ's from Eq. (1) to obtain equations of the form

$$4a_2^2(x_w^2 + y_w^2)c_2^2 - (x_w^2 + y_w^2 + z_w^2 + a_2^2 - a_3^2 - 2a_2s_2z_w)^2 = 0 \quad (2)$$

$$2a_2a_3c_3 + a_2^2 + a_3^3 - (x_w^2 + y_w^2 + z_w^2) = 0 \quad (3)$$

Equations (2) and (3) represent two families of surfaces but the solid region described by them is the same and is the workspace of the manipulator. The above equation can also be used to study the workspace of a manipulator which has joint motion constraints. If, for example, θ_2 is between $\theta_{2_{max}}$