12-1992

An Efficient Technique for Finding the Desired Global Optimum of Robotic Joint Displacement

Paul P. Lin
Cleveland State University, p.lin@csuohio.edu

An-Jen J. Yang
Cleveland State University

Publisher's Statement

This is the accepted version of the following article: Lin, P. P., & Yang, A. (1992). An Efficient Technique for Finding the Desired Global Optimum of Robotic Joint Displacement. Journal Of Robotic Systems, 9(8), 1043-1058., which has been published in final form at http://onlinelibrary.wiley.com/doi/10.1002/rob.4620090804/abstract

Original Citation

Lin, P. P., & Yang, A. (1992). An Efficient Technique for Finding the Desired Global Optimum of Robotic Joint Displacement. Journal Of Robotic Systems, 9(8), 1043-1058. .

An Efficient Technique for Finding the Desired Global Optimum of Robotic Joint Displacement

Paul P. Lin and An-Jen Yang
Department of Mechanical Engineering
Cleveland State University
Cleveland, OH 44115

1. INTRODUCTION

The kinematics of planar closed-loop mechanisms can be analyzed by using techniques such as graphical analysis, complex algebra method, and vector algebra. Pieper[1] used the theories of closed-link chains to obtain the kinematic solutions of simple manipulators. However, some of these methods are not applicable for analyzing the kinematic problems of spatial closed-loop or open-loop mechanisms. There is only a limited number of simple manipulators that can be solved analytically.

Apart from above conventional methods, many optimization methods were applied to analyze the kinematics of mechanisms. Denavit and Hartenberg[2] developed a new iterative technique for solving the kinematic equations of lower-pair mechanisms. The restriction of this technique is that it was designed for solving closed-loop linkages only. Later, Hall et al.[3] used a gradient optimization technique to solve the general spatial closed-loop mechanism. As the applications of robotics extend to spatial open-loop mechanisms, performing the analysis with the effective techniques becomes more and more important. The modified Newton–Raphson technique was applied to solving these types of problems by Turcic.[4] The major drawback was the necessity of having the starting vector close to the final solution vector.

In this study, the robot end effector's orientation is not specified, which makes it possible to reach the specified end effector's position through several different arm configurations. The objective here is to develop a technique for finding a global optimum set of joint solutions so that the joint displacements and the end effector's positional error are simultaneously minimized when a robot moves from one position to another.

The successful optimization of models that are likely to involve multiple local optimum requires that some strategies be employed to provide assurance that the global optimum has been found. A practical strategy for finding the global optima to a general problem is multistart with random sampling.[5] However, the strategy is very time consuming and therefore cannot be applied to on-line calculations. The complex optimization method has shown a tendency to find the global optimum.[6] However, this method sometimes fails. This article presents a modified searching algorithm based upon the complex method to avoid the searching from falling into a dead zone in which no solution can be found. In addition, a technique is developed for finding the desired global optimum more efficiently. The advantage of this technique is that the starting vector need not be close to the final solution. Furthermore, among many multiple kinematic solutions for an open-loop mechanism such as a robot manipulator the solution that is nearest to previous configuration of the mechanism can be found directly.

2. ROBOT KINEMATICS

2.1. Homogeneous Transformation

A local coordinate frame is attached to each robot link. The homogenous transformation describes the relative position and orientation between these coordinate frames. The $[A]$ matrix is generally used to present the relation between any two consecutive links.[7] $[A_1]$ describes the position and orientation of the first link. $[A_2]$ describes the position and orientation of the second link with respect to the first link. Similarly, $[A_3]$ describes the third link in terms of the second link. For an n degree of freedom (dof) manipulator, there will be n links and n joints.

2.2. Kinematic Equations

The general form of the $[T]$ matrix is

$$\begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

where n, o, a, and p are the normal vector, orientation vector, approach vector, and position vector, respectively.

$$\begin{aligned} \vec{n} &= \vec{o} \times \vec{a} \\ \vec{o} &= \vec{a} \times \vec{n} \\ \vec{a} &= \vec{n} \times \vec{o} \end{aligned} \tag{2}$$

3. GLOBAL OPTIMIZATION METHOD AND SEARCHING ALGORITHMS

3.1. Complex Optimization Method

The complex method developed by Box[8] is essentially a modified simplex direct-search method. This method is a sequential search technique that can solve problems with nonlinear objective functions subject to nonlinear inequality constraints. The method, if properly used, should tend to find the global maximum or minimum due to the fact that the initial set of points are randomly scattered throughout the feasible region. Here, the feasible region is defined as the set of all feasible solutions that satisfy all constraints. No derivatives are required in this method. Box proposed that the set of the K trial points be generated randomly and sequentially. Given the upper and lower bounds, H_i and G_i, of a variable X_i, the pseudorandom variable uniformly distributed on the interval (0, 1) is sampled. N samples are required to define a point in N dimensions. Each newly generated point is tested for feasibility, and if not feasible it is retracted toward the centroid of the previously generated points until it becomes feasible. The total number of points, K, to be used should be no less than $N + 1$. The objective function and constraints are as follows.

Minimize $F(X_1, \ldots, X_n)$ subject to constraints

$$G_t \leq X_t \leq H_t \qquad t = 1, 2, \ldots, m \tag{3}$$

where F is the objective function; $X_1, X_2, \ldots, X_n$ are called the explicit independent variables and $X_{n+1}, \ldots, X_m$ are called the implicit variables, which are the dependent functions of the explicit independent variables; and H_t and G_t are the upper and lower constraints, which can be either constants or functions of the independent variables. The total number of trial points (complex points), K, to be used in the complex method should be no less than $N + 1$ (N is the number of the independent variables) but can be larger.

Some numerical experiments have been performed by Box,[8] and $K \approx 2N$ was recommended based upon the experiments. On the other hand, good results with $K = n + 2$ were reported by Biles.[9] There is no specific way required to generate the initial set of points as long as they are uniformly distributed in the feasible region. The algorithm of the existing complex method is described step by step as follows.[10]

Step 1. The method takes $k \geq n + 1$ points to search for the optimum point. The first feasible starting point is chosen by the user. The other $k - 1$ initial points are generated from pseudorandom numbers.

$$X_{i,j} = G_j + R_{i,j}(H_j - G_j) \tag{4}$$

$$i = 2, 3, \ldots, k$$

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

and $R_{i,j}$ is the pseudorandom numbers between 0 and 1.

Step 2. The selected points must be in the feasible region. If the explicit constraints are violated, the point will be moved a small distance inside the violated limit. If the implicit constraints are violated, the point will be moved one half of the distance to the centroid of the remaining points. Thus,

$$X_{\text{new}} = (X_{\text{old}} + X_c)/2 \tag{5}$$

where X_c is the coordinates of the centroid of the remaining $k - 1$ points.

Step 3. Evaluate the function value at each point and find the point with the highest function value and the point with the lowest function value.

Step 4. Check the convergence of points. The convergence of the method is assured when the points are sufficiently close or the difference between the function values at the points becomes small enough.

Step 5. The point, X_h, having the highest function value is replaced by a point located at a distance α times as far from the centroid of remaining points as the distance of the rejected point on the line joining the rejected point and the centroid, where $\alpha = 1.3$ was suggested by Box[5,8,10].

$$X_{\text{new}} = X_{\text{center}} + \alpha(X_{\text{center}} - X_h) \tag{6}$$

$$X_{\text{center}} = \left(\sum_{i=1}^{k} X_i - X_h\right)/(k - 1) \tag{7}$$

where X_{new} must satisfy all constraints.

Step 6. If the point repeats in giving the highest function value on consecutive trials, it is moved one half the distance to the centroid of the remaining points.[10]

Step 7. The point is checked against the constraints and is adjusted as before if the constraints are violated.

Step 8. Convergence is assured when the objective function values at each point are within the prespecified tolerance for a certain number of iterations. Otherwise, go to step 3.

3.2. Searching Algorithms Used in Step 6

When a reflected point repeats in giving the highest function value on consecutive trials to find the point with minimum function value, it needs to be moved again until its function value is not the highest one among those of the complex points. The existing searching algorithms can be summarized in the following two categories.

Existing Algorithm A

Move the point one half the distance to the centroid of the remaining points. This is the original searching algorithm of Box's complex method,[8] which states that

$$X_n = (X_c + X_h)/2 \tag{8}$$

where X_n is the new point, X_c is the center of the remaining points, and X_h is the point having the highest function value.

Existing Algorithm B

Reduce the overreflection coefficient α to $\alpha/2$ and reflect the point again.[6]

$$X_n = (1 + \alpha')X_c - \alpha' X_h \tag{9}$$

where $\alpha' = \alpha/2$.

Recall that steps 5 and 6 in Section 3.1 showed how a point having the highest function value was replaced by another point and hopefully a point having the highest function values could be located eventually. A problem arises in using algorithms A or B when a point having the highest function value is close to the remaining points' centroid whose function value happens to be the highest among the remaining points. Under this circumstance, finding a global optimum or even a local optimum will fail because searching will remain at step 6 and will never go back to step 5 to continue the searching iterations.

Proposed Algorithm

To overcome this problem, it is proposed that the point is moved one half the distance to the point that has the lowest function value

$$X_n = (X_h + X_s)/2$$

where X_h and X_s are the points having the highest and lowest function values, respectively.

The proposed algorithm will continuously move the highest function value point to a point whose function value is the lowest among the remaining points. Thus, the searching will go back to the regular searching process (i.e., back to step 5). This algorithm differs from algorithms A and B, in which the repeated highest function value point moves toward the centroid of the remaining points, and it is likely that the centroid has a higher function value than those of the remaining points. With the proposed algorithm, the search for the global optimum becomes easier. The following example illustrates the aforementioned searching algorithms.

3.3. Illustrative Numerical Example in Optimization

To compare the optimum searching capability between the proposed algorithm and the existing algorithms in step 6, a modified Himmelblau's function[5] is used as the objective function to be minimized. The optimization problem is stated as follows.

Minimize the function

$$F(x, y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2 + 2x + 2y$$

subjected to constraints

$$-6.0 \leq x \leq 6.0$$

$$-6.0 \leq y \leq 6.0$$

Table I lists the current searching set (five points) when using the complex method.

Table I. One set of searching points.

Point	(x, y)		$F(x, y)$
1	−3.571	−3.571	−6.218 (lowest)
2	3.517	−2.143	4.360
3	2.857	1.429	14.972
4	−2.857	2.857	2.869
5	0.500	0.500	146.125 (highest)

A point whose function value is highest in the searching set will be reflected by the centroid of the remaining points. In this case, point 5 will be reflected by the centroid of points 1, 2, 3, and 4. This centroid can be easily calculated:

$$x_c = (x_1 + x_2 + x_3 + x_4)/4 = 0$$

$$y_c = (y_1 + y_2 + y_3 + y_4)/4 = -0.3571$$

Tables II, III, and IV show the sequential moves of point 5 based on existing algorithms A and B and the proposed algorithm. The contour plot and 3-D diagram of objective function are shown in Figures 1 and 2, respectively.

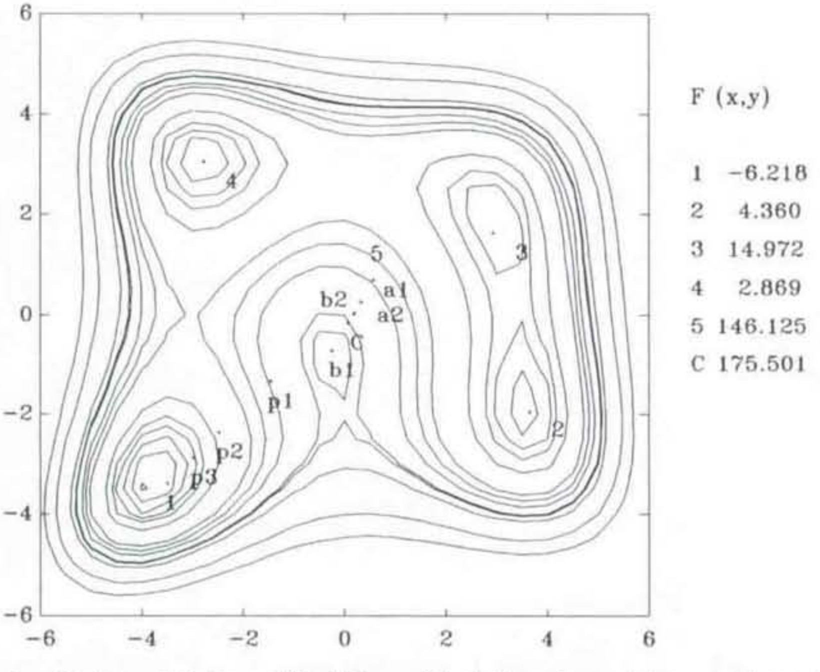

Figure 1. Contour plot of a modified Himmelblau's function and the searching points. Points 1–5: Searching points of a complex set (step 5). *C*, centroid of the remaining points (points 1–4); *a*1, *a*2, searching points using algorithm A (step 6); *b*1, *b*2, searching points using algorithm B (step 6); *p*1, *p*2, *p*3, searching points using the proposed algorithm (step 6).

Table II. Sequential moves of point 5 using algorithm *A*.

Point	(x, y)		$F(x, y)$
a_1	0.250	0.072	164.208
a_2	0.125	-0.143	170.765
.	.	.	.
.	.	.	.
.	.	.	.
C	0.000	-0.357	175.501 (centroid)

Table III. Sequential moves of point 5 using algorithm *B*.

Point	(x, y)		$F(x, y)$
b_1	-0.325	-0.914	179.074
b_2	0.106	-0.176	171.620
b_3	-0.017	-0.386	176.005
.	.	.	.
.	.	.	.
C	0.000	-0.357	175.501 (centroid)

Table IV. Sequential moves of point 5 using the proposed algorithm.

Point	(x, y)		$F(x, y)$
p_1	-1.536	-1.536	135.594
p_2	-2.554	-2.554	48.447
p_3	-3.062	-3.062	10.155
.	.	.	.
.	.	.	.
.	.	.	.

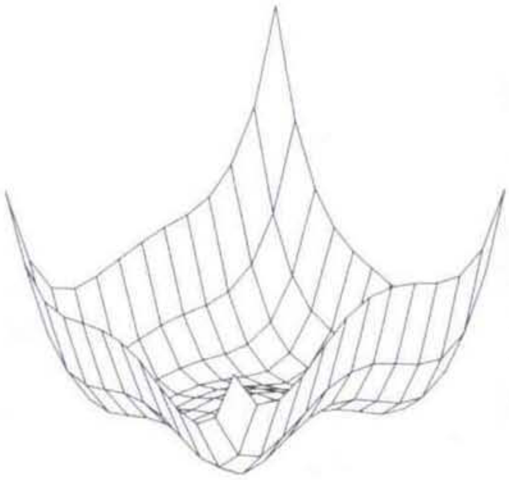

Figure 2. Three-dimensional diagram of a modified Himmelblau's function.

As can be seen in Tables II and III, when using algorithms A and B the searches all converged to point C (the centroid of points 1, 2, 3, and 4). During the searches, no points were found whose function values were smaller than those of points 1–4. As a result, the searching collapsed at step 6 in the complex method. Usually, a new set of trial searching points can be used[11] when the collapsing problem occurred, but there is a considerable possibility of the same problem occurring again and more wasted time may be spent in changing new trial points. In contrast to the result of using the proposed algorithm, at the third move the function value of point 5 ($F = 10.155$ in Table IV) is smaller than that of point 3 ($F = 14.972$ in Table I). Thus, the searching is returned to the main route of the complex method and the iterations continue. The proposed algorithm demonstrates its capability in handling the collapsing problem.

4. DESIRED GLOBAL OPTIMIZATION OF ROBOTIC JOINT DISPLACEMENTS

4.1. Objective Function Formulation for Achieving a Desired Goal

A general optimization problem is to minimize or maximize the objective function subject to some constraints. In the nonlinear displacement analysis of robotic manipulators, the main objective is to minimize the positional error and the secondary objective is to minimize the joint displacements of some specific joints when the robotic end effector moves from one position to another. The combined objective function is written as follows:

$$F(X_1, X_2, \ldots, X_n) = AF_1(X_1, \ldots, X_n) + B_1F_2(X_1) + \cdots + B_nF_{n+1}(X_n) \tag{10}$$

where

$$F_1(X_1, \ldots, X_n) = (p_x - p_x')^2 + (p_y - p_y')^2 + (p_z - p_z')^2 \tag{11}$$

is the position objective function derived from the kinematic equation, (p_x, p_y, p_z) and (p_x', p_y', p_z') are the actual and desired positions of the end effector, respectively, and $F_2, F_3, \ldots, F_{n+1}$ are the objective functions for minimizing joint displacements.

$$\begin{aligned} F_2(X_1) &= (X_1 - X_1')^2 \\ F_3(X_2) &= (X_2 - X_2')^2 \\ &\cdot \\ &\cdot \\ F_{n+1}(X_n) &= (X_n - X_n')^2 \end{aligned} \tag{12}$$

The $X_1, \ldots, X_n$, and $X'_1, \ldots, X'_n$ are the joint variables and previous joint positions, respectively. $B_1, \ldots, B_n$ are the weighting factors used to achieve a desired goal and A is a coefficient to be varied. The criterion in choosing the initial value of A is

$$A_{\text{initial}} = \text{Max}(F_2 + F_3 + \cdots + F_{n+1})/\text{Max}(F_1) \tag{13}$$

The coefficient A is varied according to a flag value that is defined as follows:

$$A_{\text{flag}} = 1/(A_1 + A_2 + \cdots + A_n) \tag{14}$$

where

$$\begin{gathered} A_1 = \text{Max}(|X_1 - X_{c1}|/|H_1 - G_1|) \\ A_2 = \text{Max}(|X_2 - X_{c2}|/|H_2 - G_2|) \\ \cdot \\ \cdot \\ A_n = \text{Max}(|X_n - X_{cn}|/|H_n - G_n|) \end{gathered} \tag{15}$$

and

$$\begin{gathered} X_{c1} = (X_1(1) + X_1(2) + \cdots + X_1(k))/k \\ X_{c2} = (X_2(1) + X_2(2) + \cdots + X_2(k))/k \\ \cdot \\ \cdot \\ X_{cn} = (X_n(1) + X_n(2) + \cdots + X_n(k))/k \end{gathered} \tag{16}$$

where k is the number of complex points.

The A_{initial} is an important value for obtaining a compromised solution for the end effector's position accuracy and the global optimum of the joint displacements. The value of A should be gradually increased to reach a better position accuracy when the A_{flag} is greater than a certain number such as 10. The value of A is not critical, but the larger A is the better position accuracy can be achieved.

4.2. Robot Manipulator as a Practical Example

A spatial three-link robot manipulator as shown in Figure 3 is used as a numerical example. The link lengths, L_1, L_2, and L_3, are 10 units each. The

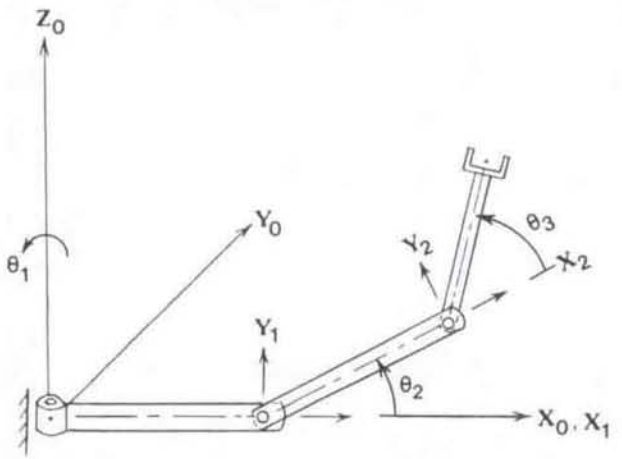

Figure 3. Spatial three-link robot manipulator.

previous three joint positions are (0°, 0°, 0°), which gives the end effector's position at (30, 0, 0) in Cartesian space.

The kinematic equations are described in terms of $[A]$ matrices as follows.

$$[T_3] = [A_1][A_2][A_3] \tag{17}$$

from which the end effector's position in Cartesian space can be expressed as

$$\begin{aligned} p_x &= L_3C_1C_{23} + L_2C_1C_2 + L_1C_1 \\ p_y &= L_3S_1C_{23} + L_2S_1C_2 + L_1S_1 \\ p_z &= L_3S_{23} + L_2S_2 \end{aligned} \tag{18}$$

where $C_1 = \cos \Theta_1$, $S_1 = \sin \Theta_1$, and $C_{23} = \cos(\Theta_2 + \Theta_3)$, etc. In this case, the following objective function is to be minimized:

$$F(\Theta_1, \Theta_2, \Theta_3) = AF_1(\Theta_1, \Theta_2, \Theta_3) + BF_2(\Theta_1) + CF_3(\Theta_2) + DF_4(\Theta_3) \tag{19}$$

subject to the rotational constraints of the three joints

$$\begin{aligned} 0^\circ &\le \Theta_1 \le 300^\circ \\ 0^\circ &\le \Theta_2 \le 270^\circ \\ 0^\circ &\le \Theta_3 \le 270^\circ \end{aligned} \tag{20}$$

where A is a coefficient to be varied, B, C, and D are the weighting factors for achieving a desired goal, and

$$F_1(\Theta_1, \Theta_2, \Theta_3) = (p_x - p_x')^2 + (p_y - p_y')^2 + (p_z - p_z')^2$$

where $p_x' = 4.6252$, $p_y' = 2.6704$, and $p_z' = 11.2485$ are the Cartesian components of the end effector's desired position and

$$F_2(\Theta_1) = (\Theta_1 - \Theta_1')^2$$

$$F_3(\Theta_2) = (\Theta_2 - \Theta_2')^2$$

$$F_4(\Theta_3) = (\Theta_3 - \Theta_3')^2$$

Note that $\Theta_1' = 0°$, $\Theta_2' = 0°$, and $\Theta_3' = 0°$ are the previous joint positions. The initial value of A used here is 100 and will be increased to a large number such as 10^6. Weighting factors B, C, and D are set to 1 if the degree of optimization preference is the same for all three joints.

Because only the end effector's position is specified, there exist three sets of possible joint solutions. They are

1. 30°, 60°, 105°
2. 30°, 165°, 255°
3. 210°, 125.76°, 35.97°

The comparisons of computational efficiency among different algorithms are listed in Tables V and VI. Note that the trial points indicated in the tables are essentially the starting vectors or initial guesses used as input in optimization. The objective is to globally minimize the joint displacements starting from the previous joint positions (0°, 0°, 0°) and minimize the error of positional function F_1.

Trial set (2) in Table V leads to a local minimum that indicates that there is no assurance for finding the global optimum when the value of A is fixed all the time. The other trial sets all lead to the same global minimum joint displacement, which can be described by $(30 - 0)^2 + (60 - 0)^2 + (105 - 0)^2$.

Table V. Solutions and comparison of computational efficiency in optimum searching when A is fixed at 10^6.

	Joint positions(°)			Optimum solutions(°)			Number of iterations by algorithm		
Trial	Θ_1	Θ_2	Θ_3	Θ_1	Θ_2	Θ_3	A	B	Proposed
(1)	0	0	0	30	60	105	—	845	388
(2)	210	120	50	210	125.7	35.97	—	388	175
(3)	30	150	250	30	165	255	—	—	152
(4)	30	50	100	30	60	105	—	346	223
(5)	100	30	90	30	60	105	—	664	428
(6)	80	250	150	30	60	105	—	—	255

Previous joint position: $\Theta_1 = 0°$, $\Theta_2 = 0°$, $\Theta_3 = 0°$. —, no solution could be found.

Table VI. Solutions and comparison of computational efficiency in optimum searching when A is gradually increased from 10^2 to 10^6.

	Joint positions(°)			Optimum solutions(°)			Number of iterations by algorithm		
Trial	Θ_1	Θ_2	Θ_3	Θ_1	Θ_2	Θ_3	A	B	Proposed
(1)	0	0	0	30	60	105	—	—	503
(2)	210	120	50	30	60	105	395	708	450
(3)	30	150	250	30	60	105	—	596	422
(4)	30	50	100	30	60	105	193	378	224
(5)	100	30	90	30	60	105	488	750	440
(6)	80	250	150	30	60	105	363	—	382

Previous joint position: $\Theta_1 = 0°$, $\Theta_2 = 0°$, $\Theta_3 = 0°$. —, no solution could be found.

When A is gradually increased as in Table VI, all five trial sets lead to the same global minimum.

Tables V and VI show a big difference between using a fixed value of A and varied values of A. With A varied (as in Table VI), it is more promising to find the global optimum solution. It is interesting to note that same global optimum solutions (listed in Table VI) were obtained when A was gradually increased from a small number to a large number, whereas a local optimum was found in Table V when A was fixed all the time.

To achieve a specific goal (i.e., to obtain the desired solution of a defined goal), the weighting factors B, C, and D should be properly distributed.

Goal 1. Minimizing the joint displacements of all three joints $B = C = D = 1$.

Goal 2. Minimizing the joint displacement of the first joint $B = 1$, $C = D = 0$.

Goal 3. Minimizing the joint displacement of the second joint $C = 1$, $B = D = 0$.

Goal 4. Minimizing the joint displacement of the third joint $D = 1$, $B = C = 0$.

Six trial joint positions (initial guesses) are used in each of the following tables. As can be seen in Table VII, it is obvious that different trial joint positions always lead to the same desired global optimum (the minimum displacements of three joints). Table VIII shows that the global optimum displacement of joint 1 can always be found when the weighting factors are properly distributed. Comparing trial set (4) in Tables VIII and IX, one will notice that a desired global optimum solution could be obtained to achieve a specific goal. In Table VIII both solutions (30, 60, 105) and (30, 165, 255) are considered the global minima because only the first joint displacement was minimized, whereas in Table IX only one global minimum was obtained because the goal was to minimize the second joint displacement.

Table VII. Solutions of minimum displacement for all three joints with weighting factors: $B = 1$, $C = 1$, and $D = 1$.

	Joint position(°)			Optimum solution(°)		
Trial	Θ_1	Θ_2	Θ_3	Θ_1	Θ_2	Θ_3
(1)	0	0	0	30	60	105
(2)	200	100	50	30	60	105
(3)	30	60	105	30	60	105
(4)	30	165	255	30	60	105
(5)	210	160	200	30	60	105
(6)	210	126	36	30	60	105

Previous joint position: $\Theta_1 = 0°$, $\Theta_2 = 0°$, $\Theta_3 = 0°$.

Table VIII. Solutions of minimum displacement for the first joint only with weighting factors: $B = 1$, $C = 0$, and $D = 0$.

	Joint position(°)			Optimum solution(°)		
Trial	Θ_1	Θ_2	Θ_3	Θ_1	Θ_2	Θ_3
(1)	0	0	0	30	60	105
(2)	200	100	50	30	60	105
(3)	30	60	105	30	60	105
(4)	30	165	255	30	165	255
(5)	210	160	200	30	60	105
(6)	210	126	36	30	60	105

Previous joint position: $\Theta_1 = 0°$, $\Theta_2 = 0°$, $\Theta_3 = 0°$.

Table IX. Solutions of minimum displacement for the second joint only with weighting factors: $B = 0$, $C = 1$, and $D = 0$.

	Joint position(°)			Optimum solution(°)		
Trial	Θ_1	Θ_2	Θ_3	Θ_1	Θ_2	Θ_3
(1)	0	0	0	30	60	105
(2)	200	100	50	30	60	105
(3)	30	60	105	30	60	105
(4)	30	165	255	30	60	105
(5)	210	160	200	30	60	105
(6)	210	126	36	30	60	105

Previous joint position: $\Theta_1 = 0°$, $\Theta_2 = 0°$, $\Theta_3 = 0°$.

Table X. Solutions of minimum displacement for the third joint only with weighting factors: $B = 1$, $C = 0$, and $D = 1$.

	Joint position(°)			Optimum solution(°)		
Trial	Θ_1	Θ_2	Θ_3	Θ_1	Θ_2	Θ_3
(1)	0	0	0	210	125.76	35.97
(2)	200	100	50	210	125.76	35.97
(3)	30	60	105	210	125.76	35.97
(4)	30	165	255	210	125.76	35.97
(5)	210	160	200	210	125.76	35.97
(6)	210	126	36	210	125.76	35.97

Previous joint position: $\Theta_1 = 0°$, $\Theta_2 = 0°$, $\Theta_3 = 0°$.

Note that the optimum solution for Θ_2 is 60 and 125.76° in Tables IX and X, respectively, whereas the optimum solution for Θ_3 is 105 and 35.97° in Tables IX and X, respectively. Tables VII–X show that the desired global optimum can always be found if a desired optimization goal is specified. It should be noted that coefficient A was gradually increased and the proposed searching algorithm was employed to make the global optimum search possible.

5. CONCLUSIONS

In this study, the searching algorithm in the existing complex optimization method was modified to avoid the optimum search from falling into a dead zone where no solution can be found. The article presents a technique for finding the desired global optimum solution more efficiently by properly distributing the weighting factors (B, C, and D) and more convergently toward the global optimum by gradually increasing the value of coefficient A in eq. (19). An algorithm to specify the initial value of A and its subsequent larger value is developed. The value of A depends upon the nature of the optimization problem and should not be fixed all the time; otherwise, the complex method may fail to find the global optimum solution.

In contrast to the tedious multistart with the random sampling technique, the proposed technique provides assurance and high efficiency to find the desired global optimum solution in the robotic joint displacement optimization. The presented technique shows promise for on-line global optimization of robot motions based upon the following three conclusions:

1. It is not necessary to have a starting point close to the final solution.
2. Among several multiple solutions (local optima), the solution that is nearest to the previous robot configuration can be found directly.
3. It is unlikely that the global optimum search will fail when using the proposed searching algorithm in conjunction with a proper assignment for the value of A.

References

1. D. L. Pieper, "The kinematics of manipulator under compute control," AIM 72, Stanford University, Stanford, CA, 1968.
2. S. Denavit and R. S. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," *ASME J. of Applied Mechanics*, **22,** 215–221 (1955).
3. A. S. Hall Jr., R. R. Root, and E. Sandgren, "A dependable method of solving matrix loop equation for the general three-dimensional analysis mechanisms," *ASME J. of Engineering for Industry*, 547–550 (1977).
4. D. Turcic, "*A Point-Oriented Kinematic Analysis Procedure for Unconstrained Mechanisms With Interactive Computer Graphics Application,*" M. S. thesis, The Pennsylvania State University, University Park, PA, 1979.
5. G. V. Reklaitis, A. Ravindran, and K. M. Ragsdell, *Engineering Optimization: Methods and Applications*, John Wiley and Sons, New York, 1983, pp. 268–277.
6. S. Mahalingam and A. M. Sharan, "The nonlinear displacement analysis of robotic manipulators using the complex optimization method," *Mechanisms and Machine Theory*, **22,** 89–95 (1987).
7. R. P. Paul, *Robot Manipulator: Mathematics, Programming, and Control*, MIT Press, Cambridge, MA, 1981, pp. 9–41.
8. M. J. Box, "A new method of constrained optimization and a comparison with other methods," *Computer J.*, **8,** 42–52 (1965).
9. W. E. Biles, "Optimization of multiple response simulation models," Final Report, ONR-Contract N00014-76-C-1021, University of Notre Dame, Notre Dame, IN 1978.
10. J. Luster and J. H. Mize, *Optimization Techniques With FORTRAN*, McGraw-Hill, New York, 1973.
11. P. E. Gill and W. Murray, *Numerical Methods for Constrained Optimization*, Academic Press, New York, 1974, pp. 208–213.