'89 KACC 1989. 10. 27~28

COLLISION-FREE TRAJECTORY PLANNING FOR DUAL ROBOT ARMS USING ITERATIVE LEARNING CONCEPT

Il Hong Suh[1], °Nak Young Chong[2], Donghoon Choi[2], and Kang G. Shin[3]

[1] Dept. of Electronics Eng., Hanyang University, Seoul, Korea
[2] Dept. of Machine Design & Production Eng., Hanyang University, Seoul, Korea
[3] Dept. of EECS, The Univ. of Michigan, Ann Arbor, MI48109

A collision-free trajectory planning algorithm using the iterative learning concept is proposed for dual robot arms in a 3-D workspace to accurately follow their specified paths with constant velocities. Specifically, a collision-free trajectory minimizing the trajectory error is obtained first by employing the linear programming technique. Then the total operating time is iteratively adjusted based on the maximum trajectory error of the previous iteration so that the collision-free trajectory has no deviation from the specified path and also the operating time is near-minimal.

1. Introduction

There are numerous industrial applications (e.g., assembly) that necessitate the use of multiple robots. A multi-robot system can be classified into three cases according to the way they are incorporated in the workstation ; 1) isolated case in which all robot arms do not overlap their workspaces, 2) loosely coupled case in which the robot arms execute independent tasks in a common workspace, 3) tightly coupled case in which the robot arms transfer an object by holding it from an initial position to a desired position. The robot arms in the isolated case never collide with each other, since they are physically apart from each other by an adequate distance. Thus, those trajectory planning techniques known to date may be sufficient for such a case. However, in loosely coupled and tightly coupled cases, most industrial robots may waste most of their time waiting to enter the shared workspace, unless their motions are effectively coordinated. Thus, it is important to coordinate robots' motions so as to minimize their idle time for productivity increase and cost saving.

There exist only a few path planning schemes concerning the coordination of a multi-robot system in terms of collision-free motion planning. Freund and Hoyer [1] proposed an on-line collision avoidance scheme for multi-robot systems by assuming a fictitious robot to detect a danger of collision. Its application, however, is restricted only to the robots of cylindrical type. Lee and Lee [2] developed the notion of collision map and time scheduling, and applied it to realize a collision-free motion planning for two robots. But their scheme is limited only to avoiding potential wrist collision, since it does not consider the collision among the links of robot manipulators. In [3], a potential collision area is defined to be the operating space of a robot from its current position to final position. If the end-effector of the other robot is expected to be in the potential collision area, then the trajectory is modified to follow the boundary of the potential collision area. However, the resulting trajectory is not guaranteed to be optimal, since this method prevent a robot from reaching a safe area where the risk of collision may not actually exist. Shin and Bien [4] employed a coordination chart to visualize all the collision-free coordinations of two specified trajectories for two planar robots where the concept of virtual obstacle is incorporated. But their scheme requires an excessive amount of computation for mapping the virtual obstacle into the coordination chart and thus suffers from the "curse of dimensionality."

In this paper, we will investigate an alternative scheme for effective motion coordination of multi-robot arms in the loosely coupled case. It is remarked that the loosely coupled case covers a number of industrial applications such as arc welding, spray coating, deburring of metal parts, assembly operations, and inspection. Such applications usually require continuous-path (CP) control while avoiding obstacles, since the robot arms interfere with one another as well as stationary or moving objects around them. In almost all applications of CP control the velocity of the end-effector of each robot should be constant. Thus a collision-free motion control algorithm for dual robot arms is required to ensure the maximum permissible constant velocities along their specified paths.

We propose a trajectory planning algorithm for dual robot arms in a common workspace using the iterative learning concept to accurately follow their specified paths with constant velocities. Specifically, a collision-free trajectory is obtained first by minimizing the trajectory error with the linear programming technique [9]. One of the two robots is designated to be the master and the other the slave by giving motion priority. The master is commanded to move along the specified trajectory and the increments of joint angles of the slave are computed at each time step of equal interval so as to track the desired nominal trajectory as closely as possible, while satisfying the joint angle/velocity constraints as well as the collision avoidance condition, i.e., given that the minimum distance between two robots is greater than the pre-specified allowable distance. Thus, the slave's optimum trajectory is determined by minimizing the weighted sum of joint motions and trajectory error subject to given constraints. Then the total operating time will be iteratively adjusted using the maximum trajectory error of the slave at the previous iteration. By such adjustments, the collision-free trajectory will not deviate from the specified path and also the operating time is near-minimal.

In the following section, a collision-free trajectory planning problem is formulated. In Section 3, a solution approach to this trajectory planning problem is proposed by employing the iterative learning concept. In Section 4, a condition for collision avoidance is derived. In Section 5, the problem is reformulated in a form suitable to utilize the linear programming technique. Simulation results are summarized in Section 6 and conclusions are drawn in the final section.

2. Problem Statement

We want to determine a sequence of optimal joint coordinate vectors of dual robot arms which move their end-effectors from an initial state to a final state along their pre-specified paths. The admissible configuration q should lie within the joint angle and velocity constraints and satisfy the condition for collision avoidance expressed in terms of the distance between the two robots.

Let n denote the number of joints of each robot, m be the dimension necessary to describe robots' hand position and orientation, and let the subscripts m and s denote the master and the slave, respectively. The hand positions and orientations of dual robot arms, $X_m(t)$ and $X_s(t)$, with reference to world coordinate are uniquely determined by their respective configurations $q_m(t)$ and $q_s(t)$ as follows :

$$X_m(t) = f_m(q_m(t)) \qquad (1\text{-}1)$$

$$X_s(t) = f_s(q_s(t)) \qquad (1\text{-}2)$$

where f_m and f_s : $R^n \Rightarrow R^m$ are nonlinear, continuous, and differentiable vector functions. In order to solve the equality constraints (1-1) and (1-2), we will use the following linearized kinematic model

$$\dot{X}_m = J_m(q_m)\dot{q}_m \qquad (2\text{-}1)$$

$$\dot{X}_s = J_s(q_s)\dot{q}_s \qquad (2\text{-}2)$$

where $J = \partial f/\partial q \in R^{m \times n}$ is the Jacobian matrix relating the joint velocities to Cartesian veolcities. The problem can then be stated as follows :

(Problem I) Let T be the unknown time required to complete the given task. Let $X_m^D(t)$ and $X_s^D(t)$, $0 \le t \le T$ be the given nominal trajectories for the master and the slave, respectively. Let $\|.\|_n$ be the n-dimensional maximum norm (i.e., max

$|\cdot|$) and d(.) be the minimum distance between dual robot arms. Also let $\varepsilon_m^* > 0$ and $\varepsilon_s^* > 0$ be given tolerance bounds for the master and the slave, respectively. Find a time sequence of joint angles $q_m(t)$ for the master such that the corresponding trajectory $X_m(t)$ satisfies

$$P_1(X_m(t)) \equiv \|X_m(t)-X_m^D(t)\|_n \le \varepsilon_m^*, 0 \le t \le T \quad (3\text{-}1)$$

and concurrently find another time sequence of joint angles $q_s(t)$ for the slave such that the corresponding trajectory $X_s(t)$ satisfies

$$P_1(X_s(t)) \equiv \|X_s(t)-X_s^D(t)\|_n \le \varepsilon_s^*, 0 \le t \le T \quad (3\text{-}2)$$

subject to

$$d(X_m, X_s) > 0. \quad (4)$$

This problem is new and useful in the sense that it includes a trajectory planning scheme for path-dependent tasks of dual robot arms as well as collision avoidance between them. However, since existing collision-free trajectory planning methods usually employ path modification schemes, they cannot give any solution to Eq. (3.1) and/or (3-2) [1], [3]. When velocity modification schemes are used, they cannot guarantee to satisfy the constant velocity requirement [2], [4].

Since it is difficult to simultaneously consider the inequality constraints (3-1), (3-2), and (4), we assume that the master should completely follow the assigned nominal trajectory and the slave is then commanded to move along its trajectory avoiding collision with the master. This motion priority may make the slave deviate from its desired trajectory. To overcome such deviation, the time-scaling concept is employed. Let t_m and t_s be the new time scales for the master and the slave, and α_m and α_s be scaling factors for the master and the slave, respectively. Now Problem I can be modified as follows :

(Problem II) Find a minimal α_m and $q_m(t_m)$ for the master such that the corresponding trajectory $X_m(t_m)$ satisfies

$$P_2(X_m(t_m)) \equiv \|X_m(t_m)-X_m^D(t_m)\|_n \le \varepsilon_m^*,$$
$$0 \le t_m \le \alpha_m T \quad (5\text{-}1)$$

and concurrently find another minimal α_s and $q_s(t_s)$ for the slave such that the corresponding trajectory $X_s(t_s)$ satisfies

$$P_2(X_s(t_s)) \equiv \|X_s(t_s)-X_s^D(t_s)\|_n \le \varepsilon_s^*,$$
$$0 \le t_s \le \alpha_s T \quad (5\text{-}2)$$

subject to

$$d(X_m, X_s) > 0. \quad (4)$$

In Problem II, α_s and the resultant joint trajectory $q_s(t_s)$ should be found such that $X_s(t_s)$ satisfies the inequality constraints (3-2) and (4). However, unfortunately, any of the conventional schemes in [1]~[9] cannot be applied to such a problem. Thus, we will employ the iterative learning concept to solve this problem.

3. Solution Approaches based on Iterative Learning

To solve Problem II, let τ be the number of iterations, and let ω_i and γ_j be weighting factors, respectively. To transform the problem of continuous time domain to that of discrete time domain with sampling interval Δt_s, let ΔX_s and Δq_s be the desired increments of the slave's end-effector in Cartesian coordinate and the corresponding increments of joint angles satisfying equation (2-2), respectively. Under the assumption that these increments are very small, the relationship $\Delta X_s = J_s \Delta q_s$ can be established. Then, the optimal joint configuration for the slave is derived by finding $\Delta {q_s}^\tau$ such that the performance index $I(\Delta q_s)^\tau$

$$I(\Delta q_s)^\tau = \max_{j=1,\dots,m}\{\omega_j |(\Delta X_s - J_s \Delta q_s)_j|\} + \sum_{i=1}^{n} \gamma_i |\Delta q_s^i| \quad (6)$$

is minimized at each time step subject to the joint angle and velocity constraints as well as the collision avoidance condition. But the optimal solution minimizing (6) may not satisfy the condition (5-2). Thus the total operating time is modified in terms of the maximum trajectory error of the slave at the previous iteration. Minimization of (6) iterates until (5-2) is satisfied.

To be more specific, let

$$E_{max} \equiv \sup P_2(X_s), \quad (7)$$

$$\Omega_j \equiv \{ t_s \mid P_2(X_s(t_s)) > \varepsilon_s^* \}, \quad (8)$$

$$T_{ef} \equiv \max \{ t_s \mid P_2(X_s(t_s)) > \varepsilon_s^* \}, \quad (9)$$

$$T_{ei} \equiv \min \{ t_s \mid P_2(X_s(t_s)) > \varepsilon_s^* \}, \quad (10)$$

$$\text{and } T_{et} = T_{ef} - T_{ei}. \quad (11)$$

If Ω_j is the union of disjoint sets, T_{ef} and T_{ei} are determined independently and thus T_{et} can be obtained by summing up their differences. Also let the critical distance d_{cr} be the minimum distance between dual robot arms at each iteration. Then α_s is given by

$$\alpha_s = 1 + \frac{E_{max} - (d_{cr} - d_{min})}{L} \frac{T_{et}}{T_{ei}}. \quad (12)$$

The above solution is expected to reduce trajectory error iteratively since the maximum error is minimized at each iteration. Thus the maximum trajectory error E_{max} can be chosen as a learning parameter with T_{ei} and T_{et} as the scale factors. Here T_{ei} and T_{et} are considered with the maximum error, since they have a great influence on the error shape on the whole even in the case the maximum error is identical.

Let the distance margin be defined as the difference between the critical distance and a pre-specified allowable minimum distance d_{min}. It is remarked that the distance margin can prevent the robot arm from falling behind too much to ensure the increase of productivity by minimizing the operating time [10]-[13]. Thus, in case the slave moves at a distance from the master with no trajectory error, the critical distance may be used as an accelerating factor of the slave. Thus we can adjust the velocity of the slave's end-effector in terms of a combination of the above factors. The algorithm can now be summarized as follows :

Algorithm

Let L be the total length of the slave's path, and ε_s^* be the tolerance bound for the termination test.

{initialize}

GET $T, d_{min}, L, \varepsilon_s^*$

{iterate}

DO WHILE $E_{max} \geq \varepsilon_s^*$

BEGIN

Minimize (6) subject to the joint angle/velocity constraints and collision avoidance condition

{get $E_{max}, T_{ei}, T_{et}, d_{cr}$}

{scale} $T^{\tau} = \alpha_s T^{\tau-1}$

{update} T

END

Note that if ε_s^* is set to a relatively large value this algorithm is confined only to point-to-point control systems, while if ε_s^* is set to an extremely small value this algorithm can even be applied effectively to the case requiring complete adherence to the specified paths. This algorithm is therefore applicable to most industrial tasks depending on the choice of ε_s^*. As we modify the velocity of the slave's end-effector based on the maximum trajectory error and the critical distance at the previous iteration, the error diminishes as iteration goes and the slave's velocity approaches an optimal value as this procedure repeats itself. Finally, we can find a near-optimum trajectory of constant velocity under which the deviation completely disappears and the operating time of two robots is reduced to a maximum possible extent. Note that in [14] an iterative learning control method is proposed for a class of linear periodic control systems, where a parameter estimator of the system together with an inverse system model is utilized to generate the control signal at each iteration. A mathematical condition for collision avoidance is given in the next section.

4. Collision Avoidance of Two Robots

Let S^k and C^k denote the subspaces at a time instant k in a common workspace occupied by the slave and the master, respectively, and let

$$d(x,C^k)\equiv\min_{y\in C^k}\|x-y\| \text{ and } d(S^k,C^k)\equiv\min_{x\in S^k}d(x,C^k).$$

Here $d(x,C^k)$ implies the minimum distance between a point x on the link of the slave and all the points on the links of the master [8]. $d(S^k,C^k)$ represents the minimum distance between the two robots. Then two robots avoid collision if the following condition is satisfied :

$$d(S^k,C^k) \geq d_{min}, \text{ for all k.} \quad (13)$$

Let us assume that at the k-th step in trajectory planning, the distance between link l of the slave and the subspace C occupied by the one-step-ahead configuration of the master is given by $d(x_l[k],C[k+1])$, where $x_l[k]$ denotes the point on the slave's link l which is closest to the subspace $C[k+1]$. Note that the next location, instead of the current location, of the master is used since we must consider the master's movement during the k-th step. Evaluation of the increment $\Delta q_s[k]$ of the slave's joint angles at the k-th step should be done in such a way that the displacement $\Delta x_l[k]$ of the critical point satisfies the inequality :

$$d(x_l[k],C[k+1])+\langle\nabla d_l[k],\Delta x_l[k]\rangle\geq d_{min} \quad (14)$$

where $\langle x,y\rangle$ is defined to be $\sum_{i=1}^{n} x_i y_i$ and $\nabla d_l[k]$ is the subgradient of the distance function $d(x,C[k+1])$ at the point $x_l[k]$. Assuming that the increments are very small, $\Delta x_l[k]$ can be approximated by $J_l[k]\Delta q_s[k]$, where $J_l[k]$ is the Jacobian matrix for the critical point $x_l[k]$. Thus the inequality equation (14) can now be rewritten as :

$$-\langle J_l^T[k]\nabla d_l[k],\Delta q_s[k]\rangle \leq d(x_l[k],C[k+1]) - d_{min}, \quad l=1,\ldots,n. \quad (15)$$

The conditions for collision avoidance between two robots have thus been expressed as linear functions of $\Delta q_s[k]$. At each optimization step k, one must calculate the distances $d(x_l,C[k+1])$ and subgradients, and the Jacobian matrices $J_l[k]$ for the critical points on the slave's links.

Assume that all the links of two robots are represented by line segments. Such an assumption may hold for almost all industrial robots if the midline of each link is taken and its width is included in the pre-specified allowable distance.

Denoting the minimum distance between a link l of the slave at the time step k and a link $\hat{l}$ of the master at the time step k+1 as $d(x_l[k],x_{\hat{l}}[k+1])$, the condition for collision avoidance between two robots can be obtained by expanding (15) as follows :

$$-\langle J_l^T[k]\nabla d_{l\hat{l}}[k],\Delta q_s[k]\rangle \leq d(x_l[k],x_{\hat{l}}[k+1]) - d_{min}, \quad \hat{l}=1,\ldots,n \text{ for } l=1,\ldots,n \quad (16)$$

where n is the number of links of the master and $\nabla d_{l\hat{l}}$ is the subgradient of the distance function which relates the l-th link of the slave to the $\hat{l}$-th link of the master at the point $x_l[k]$. Here we employed well-known quadratic programming software package ZXMWD in IMSL to obtain critical points and minimum distances between two links.

Stationary and moving obstacles of arbitrary shapes can also be easily included by modeling them as the closed chain of piecewise line segments. The number of conditions for collision avoidance will increase greatly when arbitrary-shaped obstacles are accurately modeled with line segments, but the computational complexity will not be a serious problem since the linear programming technique is to be employed.

5 Trajectory Planning by Linear Programming Approach

Now the optimization problem at step k can be stated formally as follows :

$$\text{Minimize}_{\Delta q_s[k]} \max_{j=1,\ldots,m} \{\omega_j |(\Delta x_s[k]-J[k]\Delta q_s[k])_j|\}$$

$$+\sum_{i=1}^{n}\gamma_i\left|\Delta q_s^i[k]\right| \quad (17)$$

subject to

$$-\langle J_l^T[k]\nabla d_{l\hat{l}}[k],\Delta q_s[k]\rangle \le d(x_l[k],x_{\hat{l}}[k+1]) - d_{min},\ \hat{l}=1,\dots,\hat{n} \text{ for } l=1,\dots,n \quad (18)$$

$$q_s^{imin} \le q_s^i[k+1] \le q_s^{imax} \quad i=1,\dots,n \quad (19)$$

$$\Delta q_s^{imin} \le \Delta q_s^i[k] \le \Delta q_s^{imax} \quad i=1,\dots,n \quad (20)$$

The optimality criterion (17) represents a weighted sum of joint increments and the error in the Chebishev sense of the solution to the system $\Delta x_s[k]=J_s[k]\Delta q_s[k]$. (18) is the linear constraints for collision avoidance, and (19) and (20) are joint angle and velocity constraints, respectively. To convert the above formulation to a standard linear programming problem [9], the following nonnegative variables are introduced :

$$x_i - x_{n+i} = \Delta q_i \quad i=1,\dots,n \quad (21)$$

$$x_{2n+1} = \max\left|\omega_j(\Delta x - J\Delta q)_j\right| \quad j=1,\dots,m \quad (22)$$

where $x_k \ge 0$, $k=1,\dots,2n+1$, and the symbols for time steps and the slave are omitted for notational simplicity. The optimization task (17) constrained by (18), (19), and (20) can now be rewritten as :

$$\text{Minimize} \quad x_{2n+1}+\sum_{i=1}^{n}\gamma_i x_i + \sum_{i=1}^{n}\gamma_i x_{n+i} \quad (23)$$

subject to

$$-\sum_{i=1}^{n}(J_l^T\nabla d_{l\hat{l}})_i\ (x_i - x_{n+i}) \le d(x_l,x_{\hat{l}}) - d_{min},\ \hat{l}=1,\dots,\hat{n} \text{ for } l=1,\dots,n \quad (24)$$

$$x_i - x_{n+i} \le q_{imax} - q_i\ , \ i=1,\dots,n \quad (25\text{-}1)$$

$$x_{n+i} - x_i \le q_i - q_{imin}\ , \ i=1,\dots,n \quad (25\text{-}2)$$

$$x_i - x_{n+i} \le \Delta q_{imax}\ , \ i=1,\dots,n \quad (26\text{-}1)$$

$$x_{n+i} - x_i \le -\Delta q_{imin}\ , \ i=1,\dots,n \quad (26\text{-}2)$$

$$-\omega_j\sum_{i=1}^{n}J_{ji}(x_i - x_{n+i}) - x_{2n+1} \le -\omega_j\Delta x_j\ , \ j=1,\dots,m \quad (27\text{-}1)$$

$$\omega_j\sum_{i=1}^{n}J_{ji}(x_i - x_{n+i}) - x_{2n+1} \le \omega_j\Delta x_j\ , \ j=1,\dots,m \quad (27\text{-}2)$$

$$x_k \ge 0,\ k=1,\dots,2n+1. \quad (28)$$

The following equivalence is established since x_{n+i} becomes zero for a positive Δq_i and x_i becomes zero for a negative Δq_i.

$$\text{Minimize } x_{2n+1} + \sum_{i=1}^{n}\gamma_i\left|x_i - x_{n+i}\right|$$

$$\Leftrightarrow \text{Minimize } x_{2n+1} + \sum_{i=1}^{n}\gamma_i(x_i + x_{n+i})$$

(24) is the condition for collision avoidance and (25) and (26) are the joint angle and velocity constraints, respectively. (27) is additional constraints due to the introduction of a new variable x_{2n+1}. For each time step, the above problem is solved for $x_1,\dots,x_{2n+1}$ by using the linear programming technique and the global optimum values $\Delta q_1,\dots,\Delta q_n$ are then obtained by using (21).

6 Simulation Results

Two 2-link planar robots whose initial configurations are shown in Fig. 1 are considered as an example to show the validity of the proposed algorithm. Motion priority is given to the robot on the left-hand side. The nominal path of the slave is set to be the arc of which the center point is (0.6, 0.2, 0.0) [m] and the starting and target points are (0.45, 0.35, 0.0) [m] and (0.45, 0.05, 0.0) [m], respectively, while the master is commanded to move along the arc of which the center point is (0.25, 0.20, 0.0) [m] and the starting and target points are (0.4, 0.35, 0.0) [m] and (0.4, 0.05, 0.0) [m], respectively with T = 0.417 seconds. The link lengths and the initial configurations of two robots are listed in Table 1. The joint angle constraints and the joint velocity constraints of the slave are given in Table 2. The trajectory of the slave's end-effector is shown in Fig. 2 with $\gamma_1=2.0$, $\gamma_2=1.0$, and $\omega_1=10.0$, $\omega_2=10.0$.

The computation time required was about 200 CPU seconds on a VAX-11/8700 with 600 time steps employed at the first iteration. ZX3LP in IMSL is used to solve this linear programming problem. Figs. 3 and 4 show that the trajectory of the slave's end-effector is gradually improved through the proposed learning process, indicating that the error diminishes as the time step increases.

7 Conclusion

A collision-free trajectory planning algorithm using the iterative learning concept has been proposed for dual robot arms working in a common three-dimensional workspace to ensure a maximum permissible constant velocity along the specified paths. The proposed algorithm turns out to be very effective in the sense that (i) collisions between all the links of two robots as well as their end-effectors are simultaneously considered and (ii) the algorithm does not require any mapping from one space to another. The proposed formulation allows for the use of the efficient linear programming technique which always guarantees a unique solution.

Extension to a three-dimensional multi-robot system will be straightforward, since the trajectory of each robot can be planned sequentially according to its motion priority. The sophisticated problem to prove the mathematical convergence of the proposed algorithm is now under way and the trajectory planning of the multi-robot system, taking its dynamics into account, is the subject of our future research.

References

[1] E. Freund, and H. Hoyer, "Pathfinding in Multi-Robot Systems : Solution and Applications," Proc. IEEE Int. Conf. Robotics Automat., San Francisco, pp. 103 - 111, 1986

[2] B. H. Lee and C. S. G. Lee, "Collision-Free Motion Planning of Two Robots," IEEE Trans. Syst. Man, and Cybern., Vol. SMC-17, No.1, Jan./Feb., pp. 21 - 32, 1987

[3] J. Yuh, "On-Line Adaptive Collision Avoidance Motion Control Strategy for Two Planar Robots," Proc. IEEE Int. Conf. Syst. Man, and Cybern., pp. 21 - 25, 1987

[4] Y. S. Shin and Z. Bien, "A Novel Method of Collision-Free Trajectory Planning for Two Robot Arms ," Proc. IEEE Int. Conf. Syst. Man, and Cybern., Beijing, China, Vol. 2, pp. 791 - 794, 1988

[5] S. Fortune, G. Wilfong, and C. Yap, "Coordinated Motion of Two Robot Arms," Proc. IEEE Int. Conf. Robotics Automat., San Francisco, pp. 1216 - 1223, 1986

[6] M. Erdmann and T. Lozano-Perez, "On Multiple Moving Objects," Proc. IEEE Int. Conf. Robotics Automat., San Francisco, pp. 1419 - 1424, 1986

[7] P. Tournassoud, "A Strategy for Obstacle Avoidance and Its Application to Multi-Robot Systems," Proc. IEEE Int. Conf. Robotics Automat., San Francisco, pp. 1224 - 1229, 1986

[8] E. G. Gilbert and D. W. Johnson, "Distance Functions and Their Application to Robot Path Planning in the Presence of Obstacles," IEEE J. Robotics Automat., Vol. RA-1, No. 1., Mar., pp. 21- 30, 1985

[9] Generozov V. L. , "Algorithms for Trajectory Planning in the Presence of Obstacles," Teknicheskaya Kibernetika, No. 1, 1984

[10] K. G. Shin and N. D. Mckay, "Selection of Near-Minimum Time Geometric Path for Robotic Manipulators," IEEE Trans. Automat. Contr., Vol. AC-31, No. 6, June, pp. 501 - 511, 1986

[11] ———, "A Dynamic Programming Approach to Trajectory Planning of Robotic Manipulators," IEEE Trans. Automat. Contr., Vol. AC-31, No. 6, June, pp. 491 - 500, 1986

[12] ———, "Minimum Time Control of Robotic Manipulators with Geometric Path Constraints," IEEE Trans. Automat. Contr., Vol. AC-30, No. 6, June, pp. 531 - 541, 1985

[13] J. E. Bobrow, S. Dubowsky, and J. S. Gibson, "Time-Optimal Control of Robotic Manipulators Along Specified Paths," Int. J. Robot. Res.,Vol. 4, No.3, Fall, pp. 3 - 17, 1985

[14] S. R. Oh, Z. Bien and I. H. Suh, "An Iterative Learning Control Method with Application for the Robot Manipulators," IEEE J. Robotics Automat., Vol. 4, No. 5, Oct., pp. 508 - 514, 1988

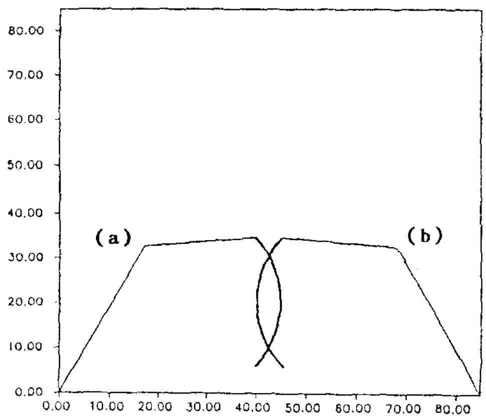

Fig. 1 Dual Robot Arms in a Common Workspace
(a) the master
(b) the slave

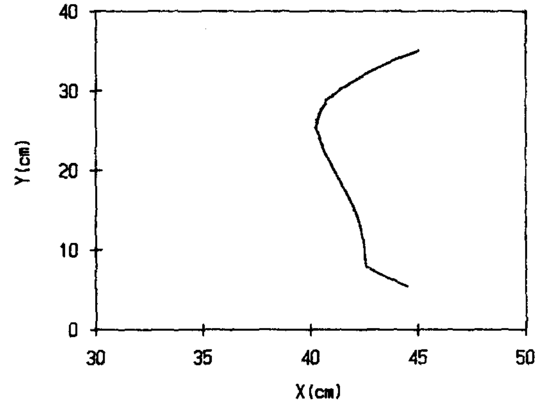

Fig. 2 The end-effector trajectory of the slave

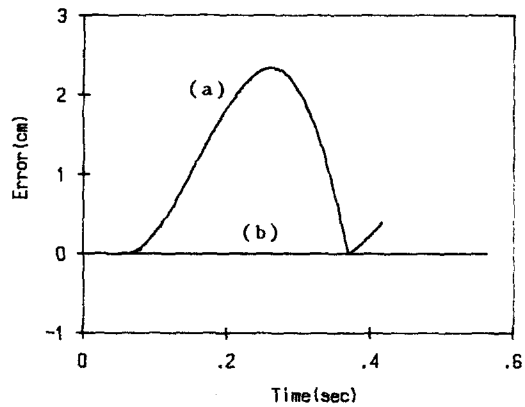

Fig. 3 The trajectory error of the slave
(a) after 1st iteration, T=0.417sec.
(b) after 2nd iteration, T=0.561sec.

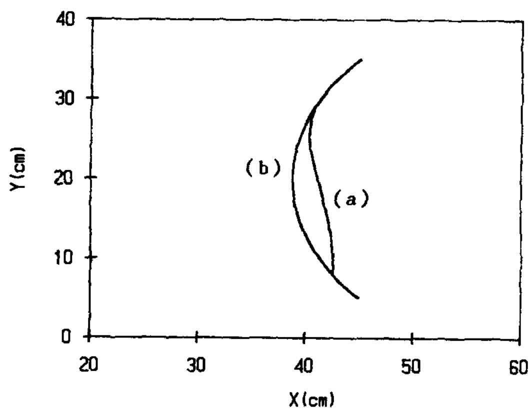

Fig. 4 The improvement of the trajectory through learning process
(a) after 1st iteration, T=0.417sec.
(b) after 2nd iteration, T=0.561sec.

TABLE 1 Link Lengths & Initial Configurations

	Master	Slave
link lengths [m]	$l_{m1} = 0.37$ $l_{m2} = 0.23$	$l_{s1} = 0.37$ $l_{s2} = 0.23$
initial configuration [degree]	$q_m^1 = 62$ $q_m^2 = -57$	$q_s^1 = 118$ $q_s^2 = 57$

TABLE 2 Joint Angle & Velocity Constraints of The Slave

	Lower Bounds	Upper Bounds
joint angle [degree]	$q_{1min} = -180$ $q_{2min} = 0$	$q_{1max} = 180$ $q_{2max} = 180$
joint velocity [rad/s]	$\dot{q}_{1min} = -2.0$ $\dot{q}_{2min} = -2.5$	$\dot{q}_{1max} = 2.0$ $\dot{q}_{2max} = 2.5$