'88 KACC 1988. 10. 21~22

COLLISION-FREE TRAJECTORY PLANNING FOR DUAL ROBOT ARMS

Nak Young Chong[*], [o] Dong Hoon Choi[**], Il Hong Suh[***]

[*] Graduate School, Hanyang Univ., Seoul, Korea

[**] Dept. of Machine Design and Production Engr., Hanyang Univ., Seoul, Korea

[***] Dept. of Electronics Engr., Hanyang Univ., Seoul, Korea

Abstract: A collision-free trajectory planning algorithm is proposed to optimally coordinate two robots working in a common 3-D workspace. Each link of the two robots is modeled as a line segment and by their motion priority, one of the two robots is chosen as the master and the other the slave. And the one-step-ahead minimum distance between the two robots is computed by moving the master to the next location on its specified trajectory. Then the nominal trajectory of the slave is modified such that the distance between the next locations of the master and the slave must be larger than a prespecified allowable minimum distance. Here the weighted sum of the trajectory error and the joint motions of the slave is minimized by using the linear programming technique under the constraints that joint angle and velocity limits are not violated. To show the validity of the proposed algorithm, a numerical example is illustrated by employing a two dof's and a three dof's planar robots.

I . INTRODUCTION

Recently, collision-free path or trajectory finding of several robots has received intensive attentions, since the multi-robot systems have been introduced in various single workstations to increase the productivity.

There exist only a few path planning schemes concerning the coordination of a multi-robot system in terms of the 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. B. H. Lee and C. S. G. Lee [2] developed the notion of collision map and the time scheduling and applied it for realizing a collision-free motion planning for two robots. But their scheme is restricted to avoiding potential wrist collision and does not include collision among inboard links which may be expected in many practical applications. In Yuh's work [3], the potential collision area is defined to be the operating space of a robot from its current position to its 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. It is not guaranteed, however, the resulting trajectory is the best, since his algorithm prohibits a robot from reaching even the area where the risk of collision may not actually exist. Y. S. Shin and Z. Bien [4] employed 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 demands too much calculations in mapping the virtual obstacle into the coordination chart.

In this paper, a collision avoidance algorithm is proposed to optimally coordinate the motions of two robots in a common workspace. Specifically, one of the two robots is assigned to be the master and the other the slave by the motion priority. The master is commanded to move along the specified trajectory and the increments of joint angles of the slave are evaluated at each time step to tract the assigned nominal trajectory as closely as possible with minimum joint movements without violating the mechanical constraints and collision avoidance condition requiring that the minimum distance between two robots be larger than prespecified allowable distance. Thus given starting and target points, the optimum trajectory of slave is found by minimizing the weighted sum of joint motions and trajectory error subject to the required constraints.

One class of the collision-free trajectory planning schemes so far is first to specify the collision-free end-effector path and later to stay away links of the robots from the obstacles and configuration limits. In the other class, the trajectory is developed in another mapped space with forbidden regions, which usually needs much calculation. This algorithm, however, takes into account the movement of all the links simultaneously in the Cartesian coordinate space where the optimum collision-free trajectory is described without preplanning the end-effector path or mapping into other spaces.

In the following section, the problem statement for collision-free trajectory planning for dual robot arms is presented. In Section III, the condition for collision avoidance is mathematically expressed. In Section IV, the problem is formulated in a form suitable for utilizing the efficient linear programming technique. Simulation results are summarized in Section V and the conclusion is drawn in the final section.

II . PROBLEM STATEMENT

The problem considered involves the evaluation of a sequence of the optimal admissible joint coordinate vectors of of the slave, $q(1)=q^0, q(2), \ldots, q(k), \ldots, q(n)$, which transfers the end-effector from the initial state x^0_e to the final state x^f_e, while the master must follow the pre-defined trajectory. The admissible configuration q of each state should lie within the physical and the velocity constraints and satisfy the condition of collision avoidance expressed in terms of the distance between two robots. The optimal joint configuration at each time step is determined in such a way that the weighted sum of the end-effector deviation from the nominal trajectory and joint motions is minimized.

Stationary and moving obstacles of arbitrary shapes can be easily included by modeling them as the closed connection of piecewise line segments. The number of conditions for collision avoidance will be greatly increased for the case of closely approximating the arbitrary-shaped curve with line segments, but the curse of dimensionality will be minimized since the linear programming technique employed.

Let n denote the number of joints of the slave and m be the number of dimensions necessary to describe the hand position and orientation. And let Δx_e represent the desired increments in Cartesian coordinates, $J \in R^{m \times n}$ be the Jacobian matrix relating the joint velocities to Cartesian velocities and β and γ_i be weighting factors. The objective can be state as follows :
Find Δq such that the performance index $I(\Delta q)$ given by

$$I(\Delta q)=\beta\{\max|(\Delta x_e)_j - J_j \Delta q|\}+\sum_{i=1}^{n}\gamma_i|\Delta q_i|$$

is minimized for each time step subject to the joint angle and velocity constraints as well as the collision avoidance. A mathematical presentation of the condition for collision avoidance is given in the next section.

III . COLLISION AVOIDANCE OF TWO ROBOTS

1 . DEFINITION OF DISTANCE FUNCTION

The various links of two robots are assumed to be represented by the line segments. Denote S and C as the subspaces at an instant in a common workspace occupied by the slave and the master respectively. Then it can be considered that two robots avoid the collision, if the following condition is satisfied.

$$d(S,C) \geq d_{min}, \quad t \in [0,T] \qquad (1)$$

where $$d(S,C) = \min_{x \in S} d(x,C) \qquad (2)$$

represents the minimum distance among the distances between all the combinations of the links of two robots and $d_{min}>0$ is a pre-specified allowable minimum distance given for some error margin. Here

$$d(x,C) = \min_{y \in C} \| x - y \| \qquad (3)$$

denotes the minimum distance between the point x on the link of the slave and all the points on the links of the master.

The assumption of treating each link as a line segment will not be impractical for most industrial robots if the midline of each link is taken and its width is included in the pre-specified allowable distance.

2. DETERMINATION OF CRITICAL POINTS AND MINIMUM DISTANCE

We now consider the determination of the critical points and the distance between the links of two robots such as 1 and 1' in Fig. 1. We introduce the scalar parameters $\lambda_1 \in [0,1]$ and $\lambda_2 \in [0,1]$ in order to indicate the various points on the links. Then the arbitrary points P and Q on the links can be represented by using λ_1 and λ_2 as follows :

$$\underline{P}=x_1\underline{i}+y_1\underline{j}+\lambda_1[(x_2-x_1)\underline{i}+(y_2-y_1)\underline{j}] \quad (4)$$

$$\underline{Q}=x_3\underline{i}+y_3\underline{j}+\lambda_2[(x_4-x_3)\underline{i}+(y_4-y_3)\underline{j}] \quad (5)$$

Thus any distance vecor between two links are defined as the vector difference of two parameterized points.

$$\underline{D} = \underline{PQ}$$
$$=[x_3-x_1+\lambda_2(x_4-x_3)-\lambda_1(x_2-x_1)]\underline{i}$$
$$+[y_3-y_1+\lambda_2(y_4-y_3)-\lambda_1(y_2-y_1)]\underline{j} \quad (6)$$

In order to find the minimum distance between two links, the following quadratic problem in λ_1 and λ_2 is formulated.

$$\text{Min.} \quad \| \underline{D} \|^2 \quad (7)$$
$$\text{s.t.} \quad 0 \le \lambda_1 \le 1 \quad (8)$$
$$0 \le \lambda_2 \le 1 \quad (9)$$

The solution indicates the critical points on two links which yield the minimum distance and the square root of the objective value gives the minimum distance between two links.

3 THE CONDITION FOR COLLISION AVOIDANCE

The property of distance function d(x,C) which is convex is used to impose the condition for collision avoidance on two robots. It satisfies the subgradient equation

$$d(z,C) \ge d(x,C) + \langle \nabla d, z-x \rangle \quad (10)$$

for any $z \in R^n$, $\langle x,y \rangle = \sum_{i=1}^{n} x_i y_i$. The vector ∇d is termed the subgradient of the convex function d(x,C) at the point x. Let us assume that at the kth step in trajectory planning, the distance between a link l of the slave and a link l' of the master is given by $d(x_l[k],y_{l'}[k+1])$, where $x_l[k]$ and $y_{l'}[k+1]$ denote the points on the lth l'th link respectively which is closest to each other. Here we define the distance between all the links with respect to the next location of the master, since we must consider the movement of master during the kth step. The evaluation of the increment of joint angles $\Delta q[k]$ at the kth step should provide that the displacement of the critical point satisfies the nonequality

$$d(x_l[k],y_{l'}[k+1])+\langle \nabla d^k_l, \Delta x_l[k] \rangle \ge d_{min} \quad (11)$$

where ∇d^k_l is the subgradient of distance function d(x,y) at the point $x_l[k]$ and is to be one-step-ahead minimum distance direction vector between two critical points. Assuming that the increments are small, the relationship $\Delta x_l[k] = J_l[k]\Delta q[k]$ may be established with $J_l[k]$ being the Jacobian matrix for the critical point $x_l[k]$. Thus the nonequality eq.(11) can now be written in the equivalent form

$$-\langle J^T_l[k]\nabla d^k_l, \Delta q[k] \rangle \le$$
$$d(x_l[k],y_{l'}[k+1])-d_{min}, \quad l=1,\dots,n \quad (12)$$

The constraints originating from the presence of the obstacle thus expressed as the linear constraints with respect to the increment of joint angle. At each optimization step k, the distance between all the links of two robots are to be determined , as well as the subgradients ∇d^k_l, and the Jacobian matrices $J_l[k]$ for the critical points on the links of the slave.

IV . TRAJECTORY PLANNIMNG BY LINEAR PROGRAMMING APPROACH

Now the problem can be formulated as follows :

$$\min_{\Delta q} \{\beta \max|(\Delta x_e)-J\Delta q|_j+\sum_{i=1}^{n}\gamma_i|\Delta q_i|\} \quad (13)$$

$$\text{s.t. } -(J^T{}_l \nabla d)\Delta q \le$$
$$d(x_i, y_l,) - d_{min} \quad l=1,\dots,n \quad (14)$$
$$q_{imin} \le q_i \le q_{imax} \quad i=1,\dots,n \quad (15)$$
$$\Delta q_{imin} \le \Delta q_i \le \Delta q_{imax} \quad i=1,\dots,n \quad (16)$$

Given the specified path segment in the external coordinate space, the optimum increments of joint angle $\Delta q[k]$ are evaluated with respect to the criterion (13) subject to the constraints (14), (15), and (16), where (14) is the constraint for the collision avoidance and (15) and (16) are the physical and the velocity constraints respectively. The optimality criterion (13) represents a weighted sum of the joint increments and the error in the Chebishev sense of the solution to the system $\Delta x_e[k] = J[k]\Delta q[k]$. In order to reduce the above formulation to the standard linear programming problem, the following nonnegative variable are introduced.

$$x_i - x_{n+1} = \Delta q_i \quad i=1,\dots,n \quad (17)$$
$$x_{n+2} = \max|\Delta x_e - J\Delta q|_j \quad j=1,\dots,m \quad (18)$$
$$x_{n+2+i} = |\Delta q_i| \quad i=1,\dots,n \quad (19)$$

where $x_k \ge 0$, $k=1,\dots,2n+2$. The optimization task (13) constrained by (14), (15), and (16) can be rewritten as

$$\min_{\Delta q} \{\beta x_{n+2} + \sum_{i=1}^{n} \gamma_i |\Delta q_i|\} \quad (20)$$

$$\text{s.t.} -(J^T{}_l \nabla d)\Delta q \le$$
$$d(x_l, y_l,) - d_{min}, \quad i=1,\dots,n \quad (21)$$
$$|(\Delta x_e) - J_j \Delta q| \le x_{n+2}, \quad j=1,\dots,m \quad (22)$$
$$|\Delta q_i| \le x_{n+2+i} \quad i=1,\dots,n \quad (23)$$
$$q_{imin} \le q_i \le q_{imax} \quad i=1,\dots,n \quad (24)$$
$$\Delta q_{imin} \le \Delta q_i \le \Delta q_{imax} \quad i=1,\dots,n \quad (25)$$

Equivalent form to (22) is

$$-x_{n+2} \le (\Delta x_e)_j - J\Delta q \le x_{n+2}$$

i.e.
$$(\Delta x_e)_j - J_j \Delta q \le x_{n+2} \quad (26)$$
$$-(\Delta x_e)_j + J_j \Delta q \le x_{n+2} \quad (27)$$

Equivalent form to (23) is

$$-x_{n+2+i} \le \Delta q_i \le x_{n+2+i}$$

i.e.
$$\Delta q_i \le x_{n+2+i} \quad (28)$$
$$-\Delta q_i \le x_{n+2+i} \quad (29)$$

It is remarked that this algorithm always gives an unique solution, by employing eq. (23) to eliminate the absolute values in objective function.

Substituting (17) into (20), (26), (27), (28), (29), (24), and (25) yields

$$\min_{\Delta q} \{\beta x_{n+2} + \sum_{i=1}^{n} \gamma_i x_{n+2+i}\} \quad (30)$$

$$\text{s.t.} -(J^T{}_l \nabla d_l)(x_i - x_{n+1}) \le$$
$$d(x_l, y_l,) - d_{min}, \quad l = 1,\dots,n \quad (31)$$
$$-J_j(x_i - x_{n+1}) - x_{n+2} \le -(\Delta x_e)_j, \quad j = 1,\dots,m \quad (32)$$
$$J_j(x_i - x_{n+1}) - x_{n+2} \le (\Delta x_e)_j, \quad j = 1,\dots,m \quad (33)$$
$$x_i - x_{n+1} \le x_{n+2+i}, \quad i = 1,\dots,n \quad (34)$$
$$x_{n+1} - x_i \le x_{n+2+i}, \quad i = 1,\dots,n \quad (35)$$
$$x_i - x_{n+1} \le q_{imax} - q_i, \quad i = 1,\dots,n \quad (36)$$
$$x_{n+1} - x_i \le q_i - q_{imin}, \quad i = 1,\dots,n \quad (37)$$
$$x_i - x_{n+1} \le \Delta q_{imax}, \quad i = 1,\dots,n \quad (38)$$
$$x_{n+1} - x_i \le -\Delta q_{imin}, \quad i = 1,\dots,n \quad (39)$$
$$x_k \ge 0, \quad k = 1,\dots,2n+2 \quad (40)$$

Now, the object function (30) and the inequality constraints (31), (32), (33), (34), (35), (36), (37), (38), (39) and the nonnegativity condition (40) constitute an LP problem.

V . SIMULATION RESULTS

A two dof's and a three dof's planar robots of the type of Fig. 2 are employed in this section to show the validity of proposed algorithm. The nominal path of the slave to be the straight line between x^o_e= [28.4249, 51.4909, 0.] [cm] and x^f_e= [-1.5761, 51.4909, 0.] [cm], while the master is commanded to move along a straight line between points x^o_e= [21.2132, 61.2132, 0.] [cm] and x^f_e= [21.2132, 21.2132, 0.]

[cm] with T = 6 seconds. The link lengths and the initial configurations of two robots are listed in Table 1. The physical constraints and the velocity constraints of the slave are given in Table 2. Fig. 3 shows the trajectory error of the slave with the variation of the weighting factors. It indicates that the error is diminished as the weighting factor is increased. End-effector trajectory of the slave is shown in Fig. 4 and simulation results for dual robot arm is also presented in Fig. 5.

VI . CONCLUSION

A collision-free trajectory planning algorithm was proposed for two robots in a common three-dimensional workspace to avoid the collision and the limits on joint angles and velocities. The suggested algorithm seems to be complete in the sense that the collisions between all the links of two robots as well as their end-effectors are simultaneously considered and also the algorithm does not require the mapping from one space to another. The proposed formulation allows the use of efficient linear programming technique which always guarantees to give the unique solution.

Extension to the three-dimensional multi-robot system is straightforward. The trajectory of each robot can be planned sequentially according to its motion priority.

REFERENCES

[1] E. Freund, and H. Hoyer, "Pathfinding in Multi-Robot Systems: Solution and Applications," Proc. IEEE Int. Conf. on Robotics and Automation, 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, January/February 1987

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

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

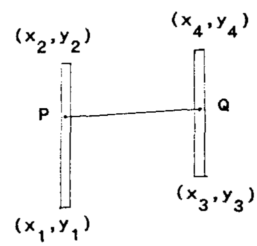

Fig.1 Distance between Two Links

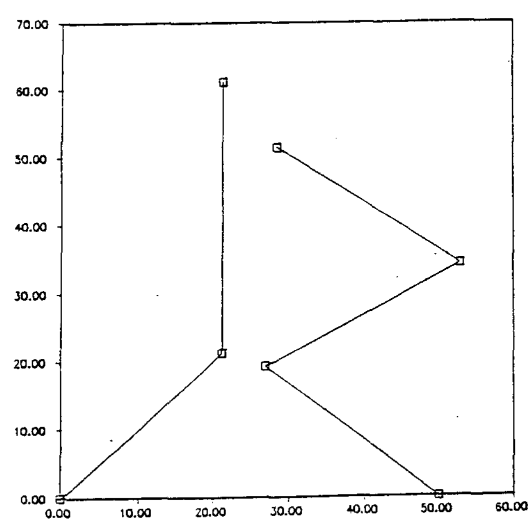

Fig. 2 Dual Robot Arms in a Common Workspace

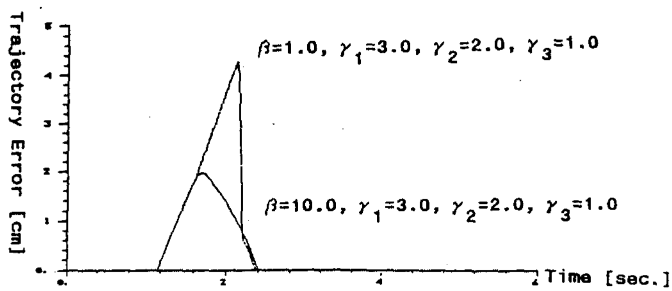

Fig. 3 Trajectory Error of the slave

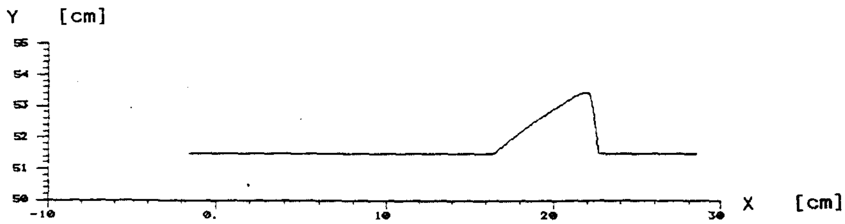

Fig. 4 End-Effector Trajectory of the Slave

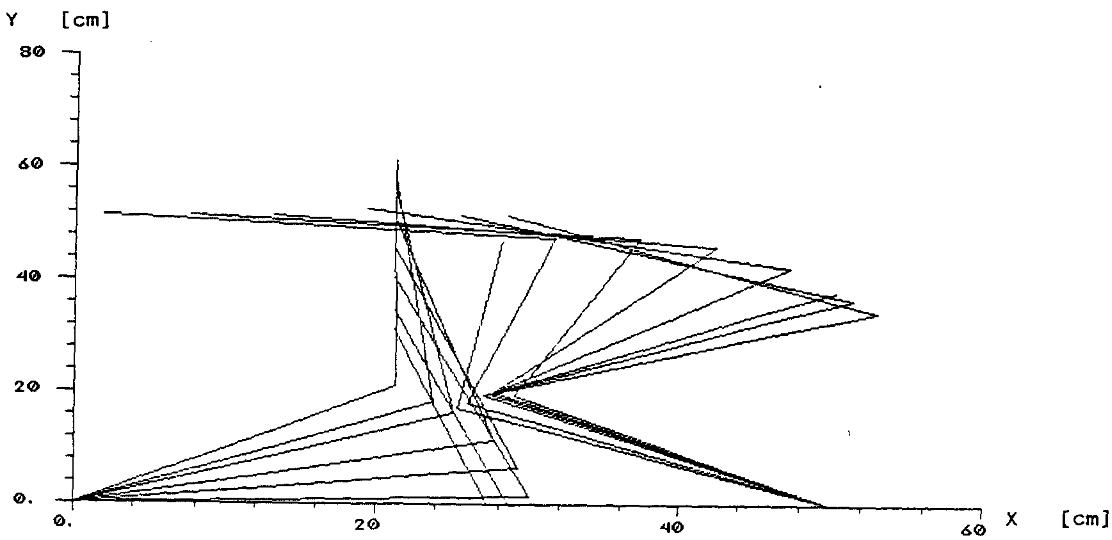

Fig. 5 Simulation Result for Dual Robot Arms

Table 1 Link Lengths & Initial Configurations

	Master	Slave
link lengths [cm]	Lm1 = 30 Lm2 = 40	Ls1 = 30 Ls2 = 30 Ls3 = 30
initial config-rations [rad]	theta1 = 0.7854 theta2 = 0.7854	q1 = 2.4435 q2 = 4.3633 q3 = 2.0071

Table 2 Physical & Velocity Constraints of The Slave

	Lower Bounds	Upper Bounds
physical con-straints [rad]	Q1min = -0.6109 Q2min = 0.1745 Q3min = 0.1745	Q1max = 3.7525 Q2max = 6.1087 Q3max = 2.9671
velocity con-straints [rad/s]	$\dot{Q}$1min = -3.2 $\dot{Q}$2min = -3.5 $\dot{Q}$3min = -3.0	$\dot{Q}$1max = 3.2 $\dot{Q}$2max = 3.5 $\dot{Q}$3max = 3.0