Delay Time Optimal Coordination Planning for Two Robot Systems

Ji-Hong Lee, Heon-Seong Nam, and Joon Lyou

Abstract

A practically applicable collision free trajectory planning technique for two robot systems is proposed. The robot trajectories considered in this work are composed of many segments, and at the intersection points between segments robots stop to assemble, weld, or do other jobs by the attached end-effectors. The proposed method is based on the Planning- Coordination Decompostion where planning is to find a trajectory of each robot independently according to their tasks and coordination is to find a velocity modification profile to avoid collision with each other. To fully utilize the independently planned trajectories and to ensure no geometrical path deviation after coordination, we develop a simple technique added the minimal delay time to avoid collision just before moving along path segments. We determine the least delay time by the graphical method in the Coordination Space where collisions and coordinations are easily visualized. We classify all possible cases into 3 groups and derive the optimal solution for each group.

I. Introduction

Multiple robots in a coordinated manner can increase the productivity and improve the versatility in complex tasks. However, when more than one robot are moving in a common work space, they become obstacles to each other. In multiple robot cases, a collision free motion planning for single robot can not be directly expanded, where stationary obstacles or moving obstacles with predefined velocity are under consideration.

One practical approach to the collision free motion planning for multiple robots is to decompose the problem into two subproblems: path planning and velocity planning[5]. Along this line, time optimal solutions for single robot had been studied in Bobrow et al.[7] and Shin and McKay[8], and for multiple robot in Bien and Lee[1], Lee[2], and Shin and Zheng[4] including additional considerations for collision avoidance between robots. A special technique of modifying the velocity for collision avoidance is to insert delay times at appropriate spatio-temporal positions. Because of the simplicity this method has been studied in many works[1, 3, 4, 6]. Among these methods, many works inserted a delay time before robots start to move along the next path segment. By doing so, the original velocity profile is only shifted along the time axis without deformation.

In practical application, trajectories are composed of many segments both geometrically and temporally. Actual contacts between end-effectors and objects occurs between these segments(see fig. 1).

The most widely used technique for collision avoidance for such cases is the semaphore method where one robot waits before running the next path segment till the other robot move out completely from the dangerous region. The main advantage of this technique lies in its simplicity and the fact that each resultant

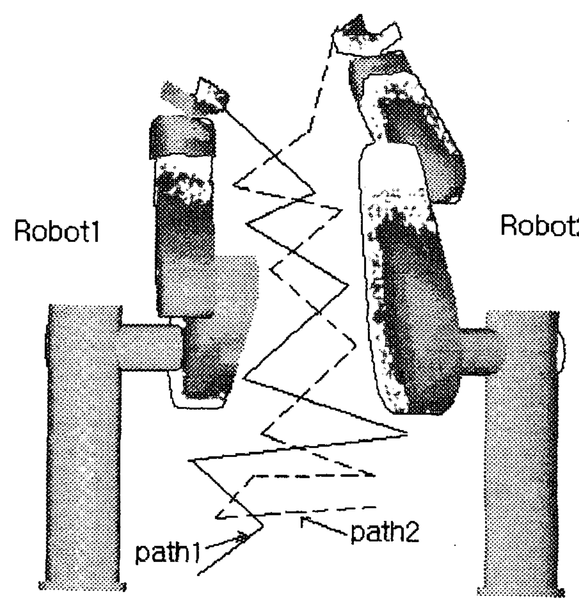

Fig. 1. Example configuration and paths for a dual-robot system.

Manuscript received July 15, 1996; accepted May 26, 1997.

J. H. Lee is with Dept. of Mechatronics Engineering and H. S. Nam and J. Lyou are with Dept. of Electronics Engineering, Chngnam National University.

velocity keeps certain original shape if described with respect to run-length along the paths. The main concern of this work is to minimize the sum of delay times inserted collision avoidance. We call the proposed method planning-coordination decomposition technique: in the planning stage individual trajectories(paths and velocities) are planned according to given tasks, and in the coordination stage velocities along the paths are modified to avoid possible collisions between the two moving robots. In this work, we assume that the overall paths are composed of several path segments(actually curves) and robots move to a task point, stop to execute some jobs by end-effectors and continue with this pattern. Even though zero job-execution times are assumed in the illustrative examples, the proposed method can be directly applied to general cases by simply overlapping the job-execution times to the calculated delay times.

For collision avoidance, we extend the result of Bien and Lee[1] by gathering all the coordination space constructed by given path segment pairs of two robots. The coordination space technique is summarized as follows. At first, paths and velocities(say, trajectories) of two robots are planned independently by taking only static obstacles into consideration. Next, a collected coordination space is constructed by collecting all the possible combinations of path segments (it was called coordination space cell here while called coordination space by Bien and Lee). And then, a collision map which shows collision regions in the collected coordination space is defined. After a coordination curve from independently planned trajectories is configured on the collected coordination space, we check whether the coordination curve passes through collision regions and insert minimal delay time between path segments for collision avoidance. By inserting a delay time, the corresponding robot waits for the delay time before moving its next path segment.

To calculate optimal delay time, all the possible coordination curves are classified into 3 groups according to the traveling times of two robots. And then we derive a optimal delay time coordination curve for the corresponding group. The proposed method is intended to add delay time mainly to faster one so that the overlapped total traveling times of the two robots may be minimized.

In section 2, we describe the concept and various properties of the collected coordination space, the collision region and the coordination curve for segmented paths. The relation delay times and collision avoidance is handled in section 3. With this concept, the method of classifying all the coordination curves into 3 groups and inserting minimal delay time at appropriate spatio temporal positions are introduced in section 4 which is accompanied with illustrative examples for each case.

II. Coordination Space for Segmented Paths

When a trajectory pair for two robots is given, geometrical descriptions of the paths as well as the velocities along the paths are specified. In the following subsections, brief descriptions for the concepts and properties of (collected) coordination space, collision regions, original coordination curve determined from independently planned trajectories and a coordination curve modified by inserting delay time between path segments are given.

1. Collected Coordination Space and Collision Map

The paths considered in this paper are composed of several segments, and the robots start to move along each segment with zero velocity and reach the end point with zero velocity. Actual operations which do not require any motions of robot bodies are executed by attached end-effectors between the segments. Let the distance from the initial position to the present position along the path of robot r be denoted by s^r. Then the i-th path segment of robot r is the interval described as

$$S_i^r \le s^r \le S_{i+1}^r,\ r=1,\ 2,\ i=0,\ \ldots,\ N^r-1. \quad (1)$$

where N^r denotes the number of path segments of robot r. Here, for completeness, we review the definition of the coordination space.

Definition 1. : Coordination Space is the collection of the ordered pairs $(s^1,\ s^2)$ with $0 \le s^1 \le S^1_{N^1}$ and $0 \le s^2 \le S^2_{N^2}$.

Note that we call the coordination space on which collision regions are defined the collision map.

Definition 2. : A subspace in a coordination space ordered pairs $(s^1,\ s^2)$ satisfying $S^1_i \le s^1 \le S^1_{i+1}$ and $S^2_j \le s^2 \le S^2_{j+1}$ is called (i, j)th Coordination Space Cell.

Note that there are 4×5=20 coordination space cells if one robot has 4-segmented path and the other robot has 5-segmented path. For the paths and configurations of Fig. 2(the details of data will be given later), the coordination space is constructed by collecting each coordination space cell which is specified by horizontal and vertical boundaries in the Fig. 2. Assuming nonredundant manipulators and nonsingular configurations, we can determine collision regions by the technique of Bien and Lee. The arm lengths of two identical robots are 0.4m and 0.3m, and the distance between two bases of the robots is 0.8m.

2. Coordination Curve

Assuming that the velocity of each robot along the path is given, which is the case under the consideration in this paper, we can construct a curve in the coordination space connecting initial point (lower left corner in the coordination space) to end point

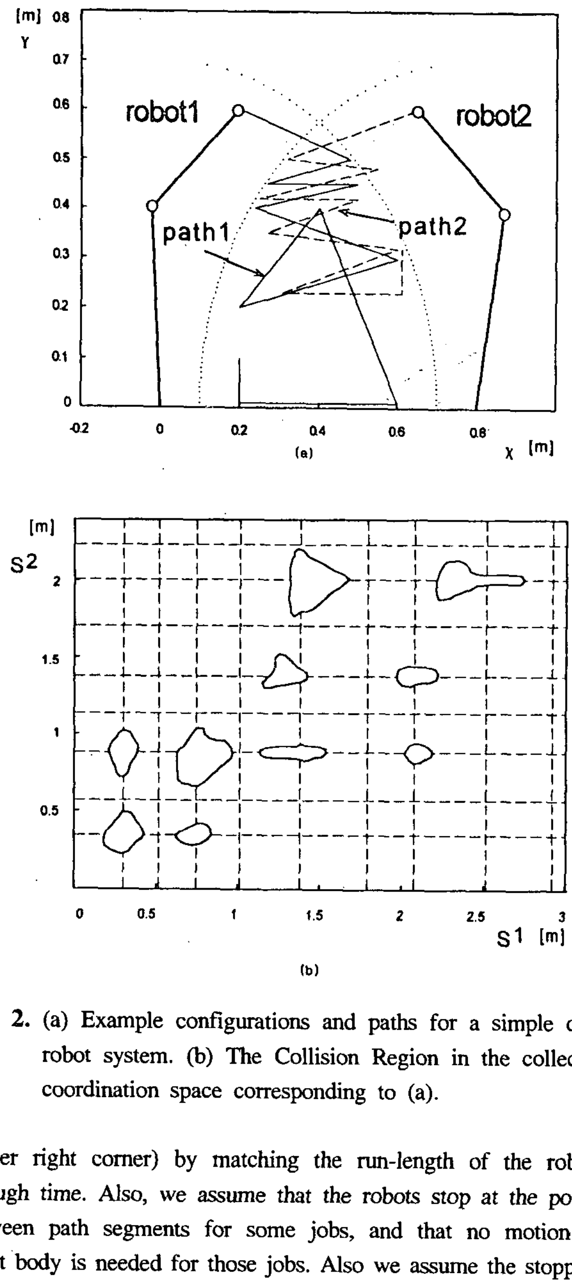

Fig. 2. (a) Example configurations and paths for a simple dual robot system. (b) The Collision Region in the collected coordination space corresponding to (a).

(upper right corner) by matching the run-length of the robots through time. Also, we assume that the robots stop at the points between path segments for some jobs, and that no motion of robot body is needed for those jobs. Also we assume the stopping time is zero, for conveniency. Then, for the example of Fig. 2 and the trapezoidal velocity profile (acceleration during 1/4 of total traveling time, 1/2 for constant speed of 1000 mm/sec, and 1/4 for deceleration), the corresponding coordination curve is given as Fig. 3. Total traveling times are 4.0067 sec and 3.1961 sec, respectively. For independent trajectory planning prior to the coordination, one may adopt path-velocity decomposition[5] technique or any other methods in which the paths and the velocities are obtained simultaneously. For any cases, the resultant trajectory is specified by the path and velocity along the path which are sufficient informations needed to apply the method proposed in this paper. In this work, however, we will adopt trapezoidal profiles for the simplicity of presentation.

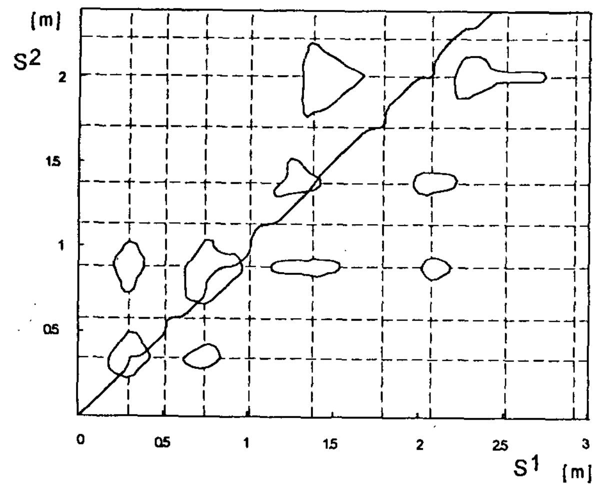

Fig. 3. Resultant Coordination Curve for the paths of Fig. 2 and trapezoidal velocity profiles.

3. Virtual Collision Region

In the works of Bien and Lee[1] and Lee[2], we assume that robots do not move backward (toward initial point) along the paths. This assumption means

$$\frac{ds^r}{dt} \geq 0, \quad r=1,\ 2. \tag{2}$$

and

$$\frac{ds^2}{ds^1} = \frac{\frac{ds^2}{dt}}{\frac{ds^1}{dt}} \geq 0. \tag{3}$$

With this assumption, a collision region can be transformed to a simpler quasi-equivalent collision region. For a given collision region, there may exist some areas around the collision region where any coordination curve satisfying the condition (3) can not pass through. When such regions are added up to the given collision region, the resultant region is called Virtual Collision Region(VCR). In Fig. 4, a collision region and corresponding virtual collision region are depicted.

4. Backward Motion

Any velocity profile including backward motion along the given path takes longer traveling time than the quasi-equivalent velocity profile of no backward motion. The quasi-equivalent velocity profile is a velocity profile that has same profile with the given one except in the intervals where more than one velocity values exist. In the overlapped intervals, we take the highest velocity curve to get a quasi-equivalent velocity profile. A velocity profile including backward motion interval and its quasi-equivalent velocity profile are shown in Fig. 5. As far as collision avoidance is concerned, we can replace the velocity profile including

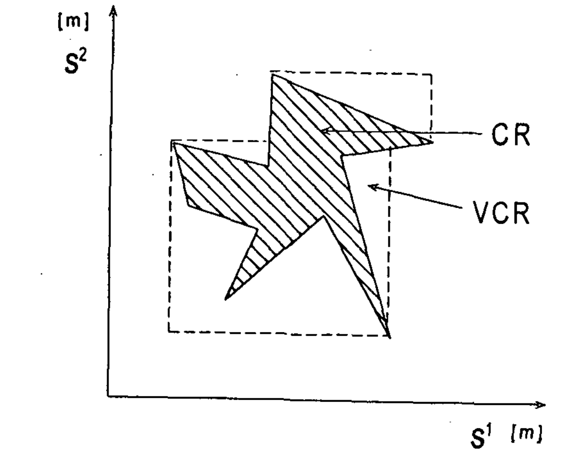

Fig. 4. A collision region and its virtual collision region.

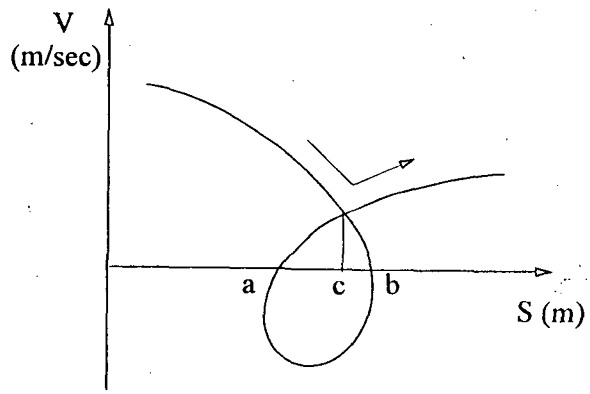

Fig. 5. A velocity profile including backward motion and its quasi-equivalent velocity profile.

backward motion with the quasi-equivalent velocity profile having only forward motion.

And related with the traveling times, we describe one more interesting property of a given velocity and its quasi-equivalent velocity profile in the followings. Let T_f be the total traveling time of any given trajectory, S be the total traveling length along the path, and $(a, b) \subset (0, S)$ be the interval of backward motion. Then,

$$\begin{aligned} T_f &= \int_0^{T_f} dt = \int_0^S \frac{dt}{ds} ds = \int_0^S \frac{1}{s} ds \\ &= \int_0^c \frac{1}{s} ds + \int_c^b \frac{1}{s} ds + \int_b^a \frac{1}{s} ds + \int_a^c \frac{1}{s} + \int_c^S \frac{1}{s} ds \\ &= \int_0^c \frac{1}{s} ds + \int_c^S \frac{1}{s} + \oint_c \frac{1}{s} ds \\ &= T_f' + \oint_c \frac{1}{s} ds \end{aligned} \tag{4}$$

As $T_f' = \int_0^c \frac{1}{s} ds + \int_c^S \frac{1}{s}$, T_f' is the total traveling time except the backward motion. Note that $T_f \geq T_f'$.

5. Slope at the Boundary

Another important property of coordination curve considered in this work is described here. Since robots stop between segments for a job, the slope of a coordination curve at the boundary of a coordination space cell are given as follows.

$$\frac{ds^2}{ds^1} = \begin{cases} \infty & \text{if } s^1 = S_i^1 \text{ and } s^2 \neq S_j^2 \\ 0 & \text{if } s^1 \neq S_i^1 \text{ and } s^2 = S_j^2 \\ \frac{v^2}{v^1} & otherwise \end{cases} \tag{5}$$

As a result, coordination curves intersect the horizontal boundaries and vertical boundaries of coordination space cells tangentially. An example of such property will be shown in the next section.

III. Collision Avoidance

From now on, we describe how to avoid collision by inserting delay time between path segments. In Fig. 6 are shown example velocity profiles and the resultant coordination curve for two robots.

Next, let's insert some delay time between the first path segment and the second path segment of robot 1 and between the second path segment and the third path segment of robot 2 as shown in Fig. 7-(a), (b). Then, we get the corresponding coordination curve as Fig. 7-(c). As shown in the figure the resultant coordination curve. has been transformed into the shape different with previous one. So, by determining where and how many to insert delay times can the original coordination curve be modified so as not to pass given collision regions, and therefore the robots do not collide with each other.

Note that, even if some delay times are inserted between path segments, the velocity profiles described along traveled distance are never changed. So, the velocity profiles described along traveled distance corresponding to Fig. 7 are exactly the same with the profiles in Fig. 6-(c), (d).

Let the original coordination curves and the corresponding traveling times of independently planned trajectory for each robot be c_o and T_o^r, $r=1, 2$, respectively, and the inserted delay times between path segments i and $i+1$ of robot r be Δt_i^r. Then, the total traveling times of robots after a coordination are given as

$$T_f^r = T_o^r + \sum_{i=0}^{N^r-1} \Delta t_i^r, \; r=1, 2. \tag{6}$$

From the description about coordination curves given so far, the collision free coordination planning for two robots is converted to a simpler problem of finding a continuous curve in two dimensional space while keeping the curve from passing through some regions (collision regions) in the space. The problem is formally described as follows.

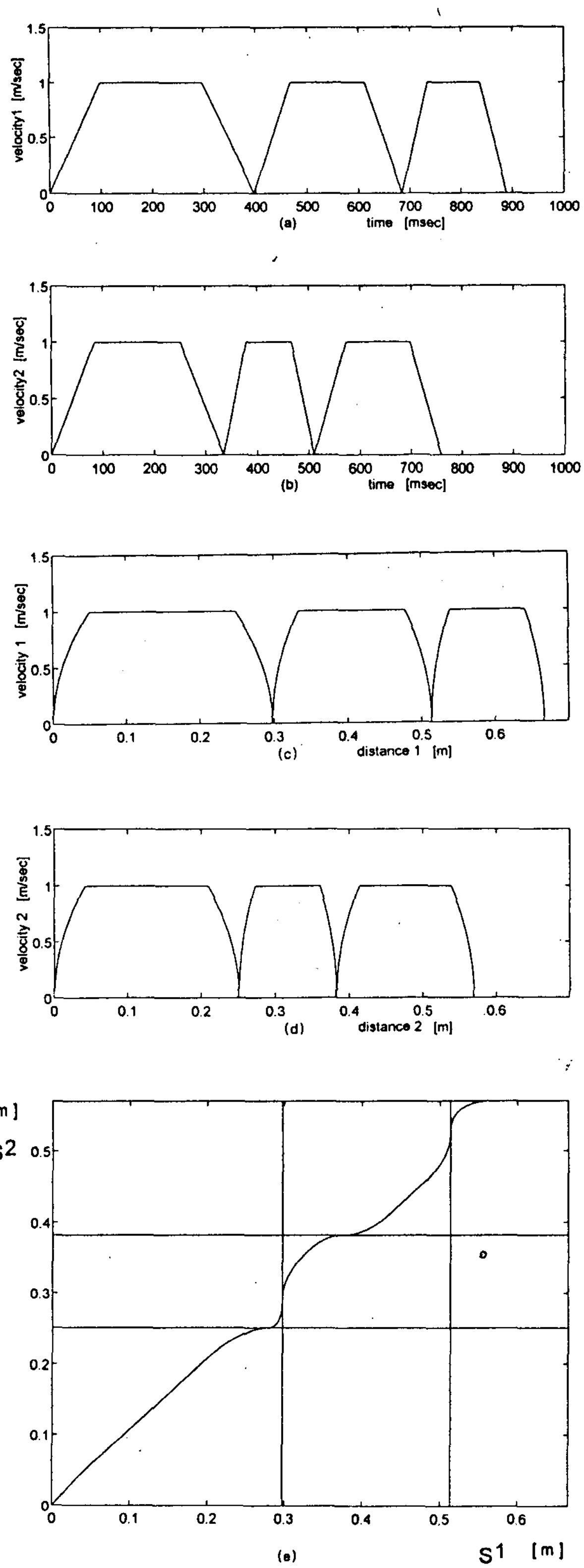

Fig. 6. Example velocity profiles along time for (a) robot 1 and (b) robot 2.
Redrawn velocity profiles along traveled distance for (c) robot 1 and (d) robot 2. (e) Resultant coordination curve.

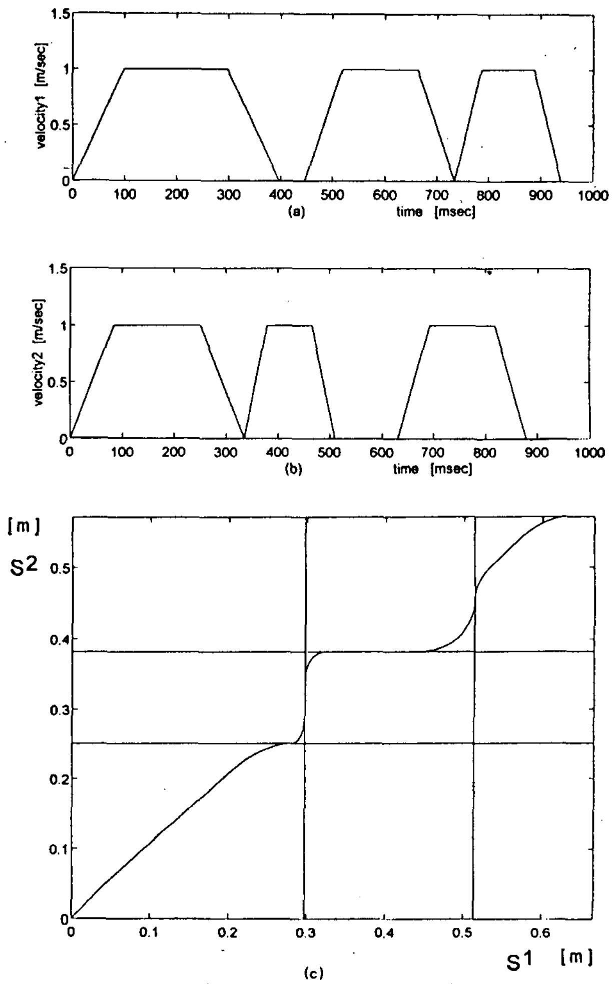

Fig. 7. Velocity profiles after some delay time are inserted for (a) robot 1 and (b) robot 2. (c) Resultant coordination curve after inserting some delay times.

Problem : For a given coordination curve, c_o, corresponding to independently planned trajectory pair, find the set of Δt_i^r so that the modified coordination curve may not pass through any collision regions as well as minimize the following quantity.

$$T = \max\{T_f^1,\ T_f^2\} \tag{7}$$

We call the solution of the problem as delay time optimal coordination curve. Before going into the problem directly, we would better define some notations. There may be infinitely many coordination curves which can be derived from c_o by inserting appropriate delay times between path segments so as not to pass through collision regions. We call a set of such coordination curves as $CFCC_{delay}$'s, where each character means collision, free, coordination, and curve, respectively. Among these curves, there are two special ones : $CFCC^-_{delay}$ and $CFCC^+_{delay}$. We describe the method how to get the $CFCC^-_{delay}$ and the $CFCC^+_{delay}$ in the followings. Note that both algorithms start with c_o from initial point $(s^1, s^2)=(0, 0)$.

[Algorithm A]

Step A1 : Follow the coordination curve under consideration from current initial point in forward direction till the curve enters a collision region or the curve hits final boundary (one of the two robots reaches its goal position). If it reaches final boundary, then stop. Else go to next step.

Step A2 : If the curve enters a collision region then follow the coordination curve in backward direction to find an intersecting point of the coordination curve with horizontal boundary of coordination space cell (an intermediate point of robot 2 between path-segments). Calculate minimal delay time for robot 2. After the delay time at the found intersecting point and updating the current initial point with this intersecting point, go to Step A1.

Definition 3 : A coordination curve derived through Step A1 to Step A2 is called $CFCC^-_{delay}$.

[Algorithm B]

Step B1 : Follow the coordination curve under consideration from current initial point in forward direction till the curve enters collision region or the curve hits final boundary (one of the two robots reaches its goal position). If it reaches final boundary, then stop. Else go to next step.

Step B2 : If the curve enters collision region then follow the coordination curve in backward direction to find an intersecting point of the coordination curve with vertical boundary of coordination space cell (an intermediate point of robot 1 between path segments). Calculate minimal delay time for robot 1. After the delay time at the found intersecting point and updating the current initial point with this intersecting point, go to Step B1.

Definition 4 : A coordination curve derived through Step Bl to Step B2 is called $CFCC^+_{delay}$. With $CFCC^-_{delay}$ and $CFCC^+_{delay}$, we can classify all the trajectory pairs into three groups.

Case 1 : Both $CFCC^-_{delay}$ and $CFCC^+_{delay}$ reach the boundary $s^1 = S^1_{N^1}$ before they reach the boundary $s^2 = S^2_{N^2}$.

Case 2 : Both $CFCC^-_{delay}$ and $CFCC^+_{delay}$ reach the boundary $s^2 = S^2_{N^2}$ before they reach the boundary $s^1 = S^1_{N^1}$.

Case 3 : $CFCC^-_{delay}$ reaches the boundary $s^1 = S^1_{N^1}$ before it reaches the boundary $s^2 = S^2_{N^2}$ and $CFCC^+_{delay}$ reaches the boundary $s^2 = S^2_{N^2}$ before it reaches the boundary $s^1 = S^1_{N^1}$.

If a coordination curve reaches $s^1 = S^1_{N^1}$ before it reaches the boundary $s^2 = S^2_{N^2}$, it means, in this coordination, robot 1 reaches its final goal position before robot 2 reaches its final goal position, say, the traveling time of robot 1 is smaller than that of robot 2.

Ⅳ. Delay Time Optimal Coordination

1. Delay Time Optimal Coordination Curve for Case 1.

Theorem 1 : For the Case 1, the delay time optimal coordination curve is $CFCC^+_{delay}$. The proof of the Theorem 1 is given in Appendix.

An example of case 1 is given in the Fig. 2. In Fig. 2, the paths are made up of straight lines, and the intermediate points of robot 1 in 2 dimensional work space are (0.2, 0.6), (0.48, 0.5), (0.27, 0.45), (0.5, 0.45), (0.24, 0.4), (0.6, 0.3), (0.2, 0.2), (0.4, 0.4), (0.6, 0.01), (0.2, 0.01), and (0.2, 0.1) in sequence from initial point and (0.65, 0.6), (0.32, 0.5), (0.55, 0.48), (0.25, 0.42), (0.5, 0.42), (0.27, 0.35), (0.6, 0.32), (0.31, 0.23), (0.61, 0.23), and (0.61, 0.32) for robot 2. The sum of inserted time for $CFCC^+_{delay}$ is 0.3 + 0.13=0.43 second and 0.16 second for $CFCC^-_{delay}$. So, the total traveling time of the time optimal solution are 4.0067 second and 3.3561 second for robot I and 2, respectively. Note that the traveling times of the original trajectories are 4.0067 second and 3.1961 second for robot I and 2, respectively. For the case of Fig. 2, we apply the above algorithms and get the result of Fig. 8.

2. Time Optimal Coordination Curve for Case 2.

Theorem 2 : For the case 2, the delay time optimal coordination curve is $CFCC^-_{delay}$.

Proof of the Theorem 2 is exactly the same with Theorem 1 if we interchange the robot 1 and robot 2 with each other, hence it is omitted here.

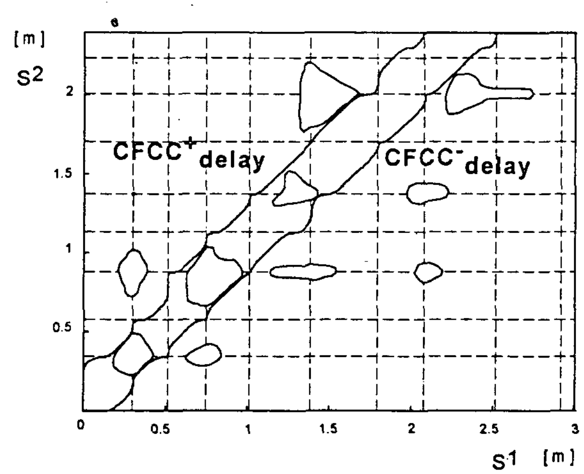

Fig. 8. The two boundary curves for the case of Fig. 2.

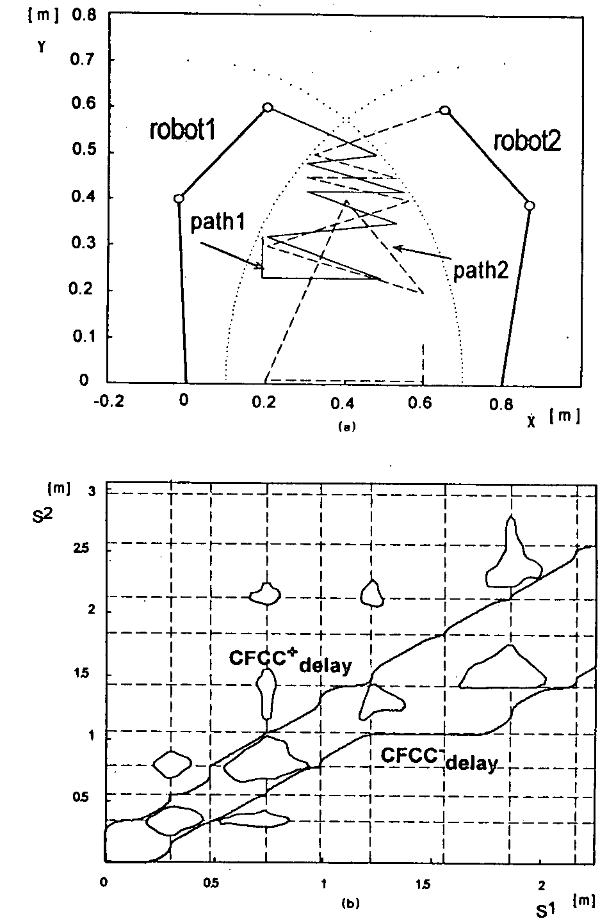

Fig. 9. (a) Example configurations and paths corresponding to case 2. (b) The resultant boundary coordination curves.

Example configurations of the case 2 is given in Fig. 9 where the intermediate points in 2 dimensional work space are (0.2, 0.6), (0.48, 0.5), (0.3, 0.48), (0.55, 0.42), (0.3, 0.42), (0.53, 0.35), (0.2, 0.32), (0.49, 0.23), (0.19, 0.23) and (0.19, 0.32) for robot 1 and (0.65, 0.6), (0.32, 0.5), (0.53, 0.45), (0.3, 0.45), (0.56, 0.4), (0.2, 0.3), (0.6, 0.2), (0.4, 0.4), (0.2, 0.01), (0.6, 0.01), and (0.6, 0.1) for robot 2. The total traveling lengths of the path and the total traveling times of the original trajectories are 2.2509 m, 3.0525 m, 3.0013 second and 4.0700 second for robot 1 and robot 2, respectively. With these paths and the same velocity patterns of case I example, we got the collision map, $CFCC^-_{delay}$, *and* $CFCC^+_{delay}$ as shown in Fig. 9. Total sums of delay times are 0.29 +0.12 = 0.41 second for $CFCC^+_{delay}$ and 0.2 +0.65 = 0.85 second for $CFCC^-_{delay}$. So, the optimal coordination curve gives 3.4113 second for robot 1 and 4.0700 second for robot 2.

3. Time Optimal Coordination Curve for Case 3.

Theorem 3 : Among $CFCC_{delay}$'s, the delay time optimal collision free coordination curve exists in the regions enclosed by $CFCC^+_{delay}$ and $CFCC^-_{delay}$ (including the boundaries).

Before proving the Theorem 3, we define the set of notations needed in the proof. Let the curve equation corresponding to $CFCC^-_{delay}$ be

$$s^2 = f^-(s^1) \tag{8}$$

and the curve equation corresponding to $CFCC^+_{delay}$ be

$$s^2 = f^+(s^1) \tag{9}$$

Then, some subspaces of the collision free coordination space are defined as follows.

Definition 4 : When an element in $CFCC_{delay}$ is described as $s^2 = f(s^1)$, C_{in}, C^-_{out}, C^+_{out} are defined as the collection of coordination curves which satisfy the following conditions in order.

$$f^-(s) \le f(s) \le f^+(s),\ 0 \le s \le s^1_{N^1} \tag{10}$$

$$f(s) \le f^-(s),\ 0 \le s \le s^1_{N^1} \tag{11}$$

$$f(s) \ge f^+(s),\ 0 \le s \le s^1_{N^1} \tag{12}$$

where $s^1_{N^1}$ is the entire traveling length of robot 1.

Definition 5 : When an element in $CFCC_{delay}$ is described as $s^2 = f(s^1)$, C^-_{cross} is defined as the collection of coordination curves which satisfy the following condition for some $s \in (0,\ S^1_{N^1})$.

$$f(s) = f^-(s) \tag{13}$$

C^+_{cross} is also defined by replacing $f^-(s)$ in the definition of C^-_{cross} with $f^+(s)$. One can easily know that

$$CFCC_{delay} = C_{in} \cup C^+_{out} \cup C^-_{out} \cup C^+_{cross} \cup C^-_{cross} \tag{14}$$

Lemma 1 : When the traveling times of an element c in $CFCC$ for robot 1 and 2 are given as $T^1(c)$ and $T^2(c)$, the following inequalities are satisfied for any $c \in C^-_{out}$.

$$T^1(c) \ge T^1(CFCC^-_{delay}) \tag{15}$$

$$T^2(c) \ge T^2(CFCC^-_{delay}) \tag{16}$$

Lemma 2 : The followings are satisfied for any $c \in C^+_{out}$.

$$T^1(c) \ge T^1(CFCC^+_{delay}) \tag{17}$$

$$T^2(c) \ge T^2(CFCC^+_{delay}) \tag{18}$$

The proof of Lemma 2 is omitted here because the concept is exactly the same with Lemma 1.

Lemma 3 : When two coordination curves intersect with each other, the slopes of the curves at the intersect points are the same with each other, say, the two curves intersect tangentially.

Lemma 4 : For any $c \in (C^-_{cross} \cup C^+_{cross})$, there exists $\hat{c} \in C_{in}$ satisfying

$$T(\hat{c}) \le T(c) \tag{19}$$

The proofs of Lemma 1, 3, and 4 are given in the appendix. With the Lemma 1 through 4, the Theorem 3 has been proven.

Even though the fact that the delay time optimal coordination curve exists in C_{in} is proven, any analytic method determining the curve has not been developed yet. So, in this work, we apply the method searching the solution efficiently in the space C_{in}. Whenever a coordination curve intersects a collision region, there are two choices to avoid the collision: one is to delay robot 1 and the other is to delay robot 2. Let the number of collision regions inside C_{in} be N. Then there are 2^N cases for search at most. So, after checking this finite number of cases one can determine the optimal solution. Actually, however, the search space is drastically reduced by introducing a checking step for the accumulated delay times at each step of inserting delay time for collision avoidance. After inserting a delay time for collision avoidance, accumulated delay times for robot 1 and 2 are constantly compared with the total inserted delay times of $CFCC^+_{delay}$ and $CFCC^-_{delay}$, respectively. If the sums of delay times just after inserting a delay time at a intermediate point between path segments is greater than those of $CFCC^+_{delay}$ or $CFCC^-_{delay}$, we stop there, exclude the coordination curve from solution candidates, and return to the initial point to start with next one. By doing so, many of the candidates are excluded from the solution in early stage of the checking. The optimal solution is selected from the remaining ones.

Example configurations corresponding to case 3 and the resultant solution with the velocity profiles of previous case is given in Fig. 10. Even though 5 collision regions (25 candidates for solution) are located inside the interesting region, only 3 or so candidates remain till the end and others are excluded from solution candidate in early stage of searching. In Fig. 10, the points between path segments are (0.2, 0.6), (0.48, 0.5), (0.27, 0.45), (0.5, 0.45), (0.24, 0.4), (0.45, 0.35), (0.3, 0.3), (0.35, 0.4), (0.35, 0.01), and (0.2, 0.01) for robot 1, and (0.65, 0.6), (0.32, 0.5), (0.55, 0.48), (0.25, 0.42), (0.5, 0.42), (0.27, 0.35), (0.6, 0.32), (0.3, 0.22), (0.3, 0.15), (0.35, 0.15), and (0.35, 0.2) for robot 2. The total lengths of the paths are 2.0337 m and 2.1896 m, and the total traveling times of original trajectories are 2.7117 second and 2.9195 second for robot 1 and 2 respectively. Also, the sum of inserted delay times for robot 1 in $CFCC^+_{delay}$ is 0.3 + 0.15 + 0.1 = 0.55 second, which gives 3.2617 second for robot 1 and 2.9195 second for robot 2. The sum of added delay times for robot 2 in $CFCC^-_{delay}$ is 0.17 + 0.23 = 0.40 second, which gives 2.7117 second traveling time for robot 1 and 3.3195 second traveling time for robot 2. Finally, the sum of inserted delay times of the optimal coordination curve are 0.15 second and 0.17 second for robot I and robot 2, respectively, which gives total traveling times of 2.8617 second for robot 1 and 3.0895 second for robot 2.

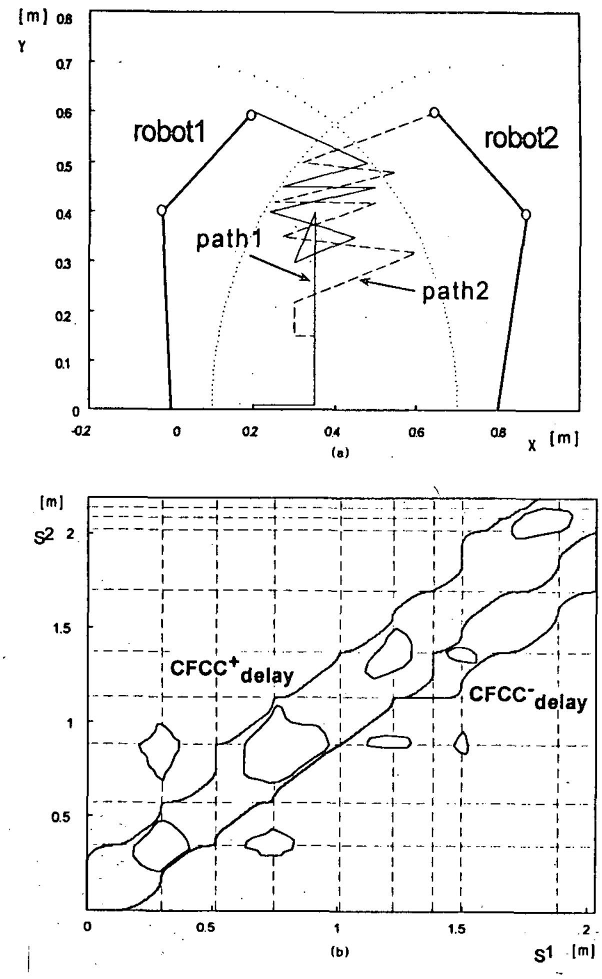

Fig. 10. (a) Example configurations of robots corresponding to case 3. (b) Solution for the example.

V. Concluding Remarks

In this paper, a coordination planning method for dual robot systems executing complex tasks is proposed. In the method, collision avoidance and time optimality are taken into consideration under the assumption that the trajectories be composed of several segments along which robots repeat moving and stopping. The proposed method may be called planning-coordination decomposition technique. Since the method intends to fully utilize the independently planned trajectory of both robots, it does not need any complex modifications to original trajectories such as speed changes or delaying the robots at the middle of the segments.

The proposed algorithm starts with given trajectories of two robots which are planned independently to each other. After constructing a coordination curve corresponding to the independently planned trajectories, we check whether collision occurs or not in the space called coordination space which shows collision map. If collision occurs, some delay times are inserted between path segments so that the resultant coordination curve may not pass through the predefined collision regions. The optimality criterion adopted in this paper is to minimize the traveling times of both robots.

When a coordination curve passes through a collision region, there are two ways for avoiding collision: one is to delay the start of robot 1 in the interested path segment, and the other is to delay the start of robot 2. The former causes the modified coordination curve to pass over the collision region and the latter causes the modified coordination curve to pass below the collision region. In calculating optimal collision free coordination curve, two special coordination curves are constructed to classify the whole solution space into three groups by examining the traveling times of these two special collision free coordination curves and the traveling time of original trajectories. The methods of finding solution for each case are developed and proved.

For the case of nonzero job execution times between path segments, the proposed method can be applied by overlapping the job execution times together to the delay times introduced for collision avoidance.

To apply the proposed technique in practical cases a safety consideration such as enlarging the collision regions should be included.

Acknowledgement

This work was supported by KOSEF Grant 94-0100-15-01-3.

Appendix

Proof of Theorem 1 : Let the independently planned traveling times of the robots be T_f^1 and T_f^2, respectively. And let the traveling time corresponding to the modified coordination curve of $CFCC^+_{delay}$ be T_f^{1+} and T_f^{2+}, respectively. Since we got $CFCC^+_{delay}$ after adding delay times,

$$T_f^1 \le T_f^{1+} \tag{20}$$

For any $CFCC^+_{delay}$ in case 1, robot 1 reaches its goal position before robot 2 reaches its final position. So,

$$T_f^{1+} \le T_f^{2+}, \tag{21}$$

and since no delay times are added to robot 2,

$$T_f^{2+} = T_f^2. \tag{22}$$

Finally, for the case 1, we get

$$T_f = \max\{T_f^1, T_f^2\} = \max\{T_f^{1+}, T_f^{2+}\} = T_f^+ \tag{23}$$

which means traveling time of slower robot for $CFCC^+_{delay}$ is same to the ones of slower robot of original trajectory, i. e, smallest traveling time.

Proof of Lemma 1 : From the definitions of C^+_{out} and C^-_{out}, c in C_{out} moves always below $CFCC^-_{delay}$ in coordination space. This is possible only when the motion of robot I of c is faster than that of $CFCC^-_{delay}$ or when the motion of robot 2 of c is slower than that of $CFCC^-_{delay}$. The first case is impossible because the motion of robot 1 of $CFCC^-_{delay}$ *is* original one (c_o), say, no delay times are added.

Proof of Lemma 3 : Even in the case where some delay times are inserted between path segments, the velocity profiles of robot I and 2 with respect to S^1 and S^2 respectively are never changed. Note that the two independent variables, S^1 and S^2, of a coordination space are the traveled distance of two robots, and the slope of a coordination curve at a point in the coordination space is determined from the velocities of two robots at that point. Since the velocities for a specified value of (S^1, S^1) are never changed by inserting any delay times and hence the slope of a coordination curve passing a point is never changed, the lemma is proven.

Proof of Lemma 4 : If we construct a coordination curve over $0 \le s^1 \le S^1_{N^1}$ from the following rules

$$\hat{f}(s^1) = \begin{cases} f^+(s^1) & f(s^1) > f^+(s^1) \\ f^-(s^1) & f(s^1) < f^-(s^1) \\ f(s^1) & f^-(s^1) \le f(s^1) \le f^+(s^1) \end{cases} \tag{24}$$

and denote the curve $s^2 = \hat{f}(s^1)$ as $\hat{c}$, then $\hat{c}$ is obviously a collision free coordination curve and is an element of C_{in}. Also, from Lemma 2 and 3, the equation (6) has shorter traveling time than that of c.

References

[1] Z. Bien and J. Lee, "A minimum-time trajectory planning method for two robots", *IEEE Trans. on Robotics and Automation*, Vol. 8, No. 3, pp. 414-418, June 1992.

[2] Jihong Lee, "A Dynamic Programming Approach to Near Minimum-Time Trajectory Planning for Two Robots," *IEEE Trans. on Robotics and Automation*, Vol. 11, No. 1, pp. 160-165, Feb. 1995.

[3] C. Chang, M. J. Chung, and B. H. Lee, "Collision avoidance of two general robot manipulators by minimum delay time", *IEEE Trans. on Syst., Man, and Cybern.*, Vol. 24, No. 3, pp. 517-522, March 1994.

[4] K. G. Shin and Q. Zheng, "Minimum-time collision-free trajectory planning for dual-robot systems", *IEEE Trans. on Robotics and Automation*, Vol. 8, No. 5, pp. 641-644, October 1992.

[5] K. Kant and S.W. Zucker, "Toward efficient trajectory planning: The Path-velocity decomposition," *Int. J. Robotics Res.*, Vol. 5, No. 3, pp. 72-89, Fall 1986.

[6] B. H. Lee and C. S. G. Lee, "Collision-free motion planning of two robots," *IEEE Trans. on Syst., Man, and Cybern.*, Vol. SMC-17, No. 1, pp. 21-32, 1987.

[7] J. E. Bobrow, S. Dubowski, and J. S. Gibson, "Time-optimal control of robotic manipulators along specified paths," *Int. J. Robotics Res.*, Vol. 4, No. 3, pp. 3-17, Fall 1985.

[8] K. G. Shin and N. D. McKay, "Minimum-time control of robotic manipulators with geometric path constraints," *IEEE Trans. Automatic Control*, Vol. AC-30, No. 6, pp. 531-541, June 1992.

Ji-Hong Lee received the B.S. degree in Electronics Engineering from Seoul National University, Seoul, Korea, in 1983 and the M.S. and Ph.D. degrees in Electrical Engineering from the Korea Advanced Institute of Science and Technology(KAIST), Taejon, Korea, in 1985 and 1991. From 1983 to 1988, he worked for the Hyundai Heavy Industry. From 1991 to 1993, he was an assistant professor in Electronics Control Engineering, Kum-Oh Technical College. He is now an associate professor in Mechatronics Engineering, Chungnam National University, Taejon, Korea. His current research interests include intelligence control, neural network, fuzzy control, and multiple robot path planning.

Heon-Seong Nam received the B.S. and M.S. degree in Elctronics Engineering from Chungnam National University, Taejon, Korea, in 1991 and 1993. Since 1993, he has been working toward Ph.D. degree at the same department. His current research interests include path planning, telerobot control, and process control.

Joon Lyou received the B.S. degree in Elctronic Engineering from Seoul National University in 1978 and the M.S. and Ph.D. degrees in Electrical Engineering from Korea Advanced Institute of Science and Technology in 1980 and 1984, respectively, Since 1984, he has been with Department of Electronics at Chungnam National University, where he is now professor. During the academic year of 1989-1990, he visited Michigan State University at East Lansing, MI, and for two months in early 1994, he visited Institute of Production Engineering and Automation(IPA) at Stuttgart, Germany. His research interests include control and automation, industrial applications of control theory, and mechatronics.