REAL-TIME CO-ORDINATION OF MULTIPLE ROBOTS WITH TEMPORAL UNCERTAINTY IN THE TASK EXECUTIONS[1]

Gustavo Raush and Raúl Suárez

Instituto de Organización y Control de Sistemas Industriales (UPC)
Av. Diagonal 647, Planta 11, 08028 Barcelona, Spain
Emails: {raush,suarez}@ioc.upc.es

Abstract: The paper proposes a method to deal with the co-ordination of multiple robots that perform independent tasks in the same workspace. It is assumed that the movements of each robot have been planned independently (geometric paths and velocity profiles) and that there exists temporal uncertainty at some points in each robot path, therefore a robot collision may occur. The proposed method was developed for two robots and it allows the co-ordination of the robots through the on-line modification of the robot velocity profiles according to the evolution of each robot task while the original geometric paths are maintained.

Keywords: robot arms, manipulators, task co-ordination, temporal uncertainty.

1. INTRODUCTION

The use of multiple robots in a common workspace can significantly improve the versatility and flexibility of the cell and, at the same time, save space if the robots have to do independent tasks. In this case, the robot movements must be coordinated in order to avoid collisions among themselves.

The coordination problem can be analyzed and solved off-line when the movements of the robots (geometric paths and temporal evolutions) can be fully determined before their execution. Nevertheless, if there is uncertainty in the temporal evolution of the task, from now on called *temporal uncertainty*, some decisions must be taken on-line to guaranty the coordination. This is the case when the geometric path is known but the time the task will need is not known, e.g. one robot may have to wait at some point for a feeder reply or an assembly with active compliance is being performed (reaction forces are controlled and the deviations in the path can be neglected).

The paper presents a method for the real-time robot coordination considering the existence of temporal uncertainty. The proposed approach has been initially developed for two robots. It was developed under the following assumptions: the geometric path and the initial velocity profile of each robot have been computed independently and are available, and the robots do not collide at their initial configurations (i.e. they have safe initial configurations). The target of the work is the modification of the robot velocity profiles so that collisions are avoided and minimum delays are introduced in the tasks.

2. PREVIOUS APPROACHES

The existent approaches to the coordination of multiple robots working in the same workspace can be classified into two main categories: coupled and uncoupled methods (Todt, *et al.*, 2000).

The coupled methods, also called centralized methods (Latombe, 1991) deal with the problem in a global way, such that the trajectories (i.e. the geometric path and the velocity at each point) of the robots are determined simultaneously as a unique problem of dimension equal to the sum of the degrees of freedom of each robot. The Configuration-Space

[1] This work has been partially supported by the projects CICYT TAP98-0471 and TAP99-0839

(Lozano-Pérez, 1983) and Configuration/Time-Space (Latombe, 1991) are typical tools for this purpose. Usually, the methods are complete but the weak points are that the complexity grows exponentially with the problem dimension and that temporal uncertainty cannot be handle in a simple way. The search for collision free paths can be done using potential functions (Khatib, 1986) with ad-hoc strategies to avoid local minimum, or harmonic potentials functions that do not have local minimum (Connolly and Burns, 1990; Connolly, 1992). Some numerical methods have been also presented to deal with potential functions in Confi-guration-Space (Barraquand and Latombe, 1991; Barraquand, *et al.*, 1992). In another approach, the free space is segmented into several parts, called *freeways*, that are represented in a graph (Brooks, 1993). The robots must then follow a set of connected freeways that is obtained by searching the graph for a path from the initial node to the goal one.

Other approach have been proposed to be applied in real-time requirement (Freund and Hoyer, 1986,1988). Each robot has a different hierarchical precedence level, and the robot with higher precedence level is allowed to continue its planned path while the others have to modify their path to avoid any predicted collisions.

The uncoupled methods split the global problem into two sub-problems of smaller complexity (Kant and Zucker, 1986). First, the trajectory of each robot is independently computed (using any of the techniques mentioned above for coupled methods), so the complexity of the problem is strongly reduced. Then, if it is necessary, in a second step the robot velocities are adjusted in order to avoid collisions. This can be done modifying the velocity profiles obtained in the original planned trajectories (Lee and Lee, 1987), or introducing a specific delay in the starting time of some robots (Bien and Lee, 1992; Chang, *et al.*, 1994), i.e. shifting the original velocity profile along the axis of time without any deformation. These works deal with tasks where the robots have null velocity only at the beginning and end of the path. In general, real tasks have at least one intermediate point of null velocity in the cycle (besides the initial and final configurations that quite frequently are coincident if the robot is performing a repetitive task). This situation was solved by applying pure delays at the beginning of the non null-velocity stage (Lee, *et al.*, 1995). Uncoupled methods are not complete, but the complexity is, in general, considerably smaller. Nevertheless, most of these approaches deal with the coordination problem of only two robots. Moreover, none of the previous approaches considers temporal uncertainty during the task execution.

In general, coupled methods are not useful for real time application due to their extremely high complexity (caused by the complete representation of the free space in a dynamic environment). On the

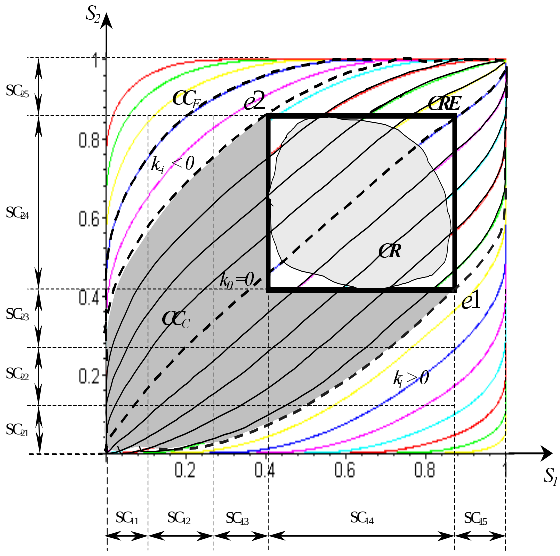

Figure 1: Collision Region *CR*, Collision Region Envelope *CRE*, escape points *e1* and *e2*, Basic Coordination Curves Set *CCS*, Shadow Cone *SC* corresponding to *CRE* and the five different entry zones for each robot.

contrary, the lower complexity of uncoupled methods makes them more suitable for on-line operation, since only robot velocities have to be adjusted on-line.

3. BASIC CONCEPTS FOR THE PROPOSED APPROACH

3.1 Coordination Space and Related Elements

Let us consider two robots, R_i $i\in\{1,2\}$, with a common workspace and independently planned trajectories $\boldsymbol{q}_i(t)$, $\boldsymbol{q}_i$ being the configuration of R_i. The geometric path of each robot can be parameterized as $\boldsymbol{q}_i(s_i)$ where s_i indicates the fraction of the total path covered in task and is normalized to satisfy $0 \le s_i \le 1$. Then, $s_i(t)$ represents the temporal evolution of the task performed by the robot R_i ($s_i(t) = 0$ for $t \le 0$).

Definition 1: *Coordination Space*, *CS*, is the space defined by the points $s = (s_1, s_2)$. ▪

Definition 2: *Collision Region*, *CR*, is the region in *CS* composed of the points that represent a collision between the robots in physical space. ▪

Depending on the problem, *CR* can be composed of non-connected subsets of points. The computation of *CR* is one of the weak points of the uncoupled approaches. In practice, *CS* is sampled and a collision test is run for each sample point, then, *CR* is determined from the finite set of collision sampled points (using, for instance, convex hull approximations).

Definition 3: *Collision Region Envelope*, *CRE*, is a rectangle in *CS* defined by the minimum and maximum values of s_1 and s_2 of any connected set of points of *CR*. ▪

The lower right vertex, $e1=(e1_1, e1_2)$, and the upper left one, $e2=(e2_1, e2_2)$, of a *CRE* will be called *escape points* (Figure 1).

Definition 4: *Coordination Curve*, *CC*, is any continuous curve in *CS* describing the relative motion of the robots from the initial to the goal configuration.■

A *CC* will be called *Collision-Free Coordination Curve*, CC_F, if it does not cross through any *CRE,* and *Collision Coordination Curve*, CC_C, otherwise. In this work it is assumed that the robots are not allowed to move backward, therefore $ds_2/ds_1 \geq 0 \ \forall s \in CC$ (Figure 1).

Definition 5: *Basic Coordination Curve Set*, *CCS*, is the set of curves *CC* determined by different initial delays τ_1 and τ_2 in the starting time of R_1 and R_2. ■

A $CC \in CCS$ is composed of points $s=(s_1(t-\tau_1), s_2(t-\tau_2))$ and can be characterized by a unique parameter $k=\tau_2-\tau_1$ such that $|k|$ indicates how much one robot is delayed with respect to the other and sign(k) indicates which is the delayed robot. CC_k indicates the Coordination Curve for a given k.

Definition 6: *Shadow Cone*, *SC*, is a region in *CS* covered by the portions of the curves $CC_C \in CCS$ that start with at least one coordinate $s_i = 0$ and end in a *CRE,* i.e. from an axis until *CRE* (Figure 1 illustrates a *SC*). ■

3.2 Decomposition of the Coordination Space into Cells

In general, the tasks performed by a robot can be divided into two parts, a zero-velocity track, ZVT, where the displacements and velocity of the robot are negligible (e.g. during a grasping or precise assembly operation), and a non-zero-velocity track, NZVT, otherwise (e.g. during the transportation of an object). In a ZVT, the robot is performing some operation or just waiting for an external signal, and therefore ZVT could not have a fixed predetermined execution time. In this work it is assumed that the temporal uncertainty appears in a ZVT.

Property 1: Let s_{iZ} represents a configuration of R_i where $\dot{q}_i(s_{iZ}) = 0$, $i \in \{1,2\}$, then, the curves $CC \in CCS$ that do not pass through a point (s_{1Z}, s_{2Z}) are tangent to the straight line defined by $s_i = s_{iZ}$■

The straight lines defined by $s_i = s_{iZ}$ (including $s_{iZ}=0$ and $s_{iZ}=1$) determine rectangular cells in *CS* such that the nominal velocities of both robots are non-null inside the cells, one of the robots has null velocity on the cell sides, and both robots have null velocities at the corners of the cells.

Considering the temporal uncertainty, the point s describing the relative evolution of the robots in *CS* during a real execution of the tasks satisfies:

1) s follows a curve CC_k inside a cell
2) if $s \in CC_k$ in the previous cell then s can:
   a) Enter into the new cell following the same CC_k
   b) Leave the curve CCk following the boundary of the cells, and enter into the new cell following a curve CCk' with:
      $k' > k$ if R_2 is delayed with respect to R_1
      $k' < k$ if R_1 is delayed with respect to R_2

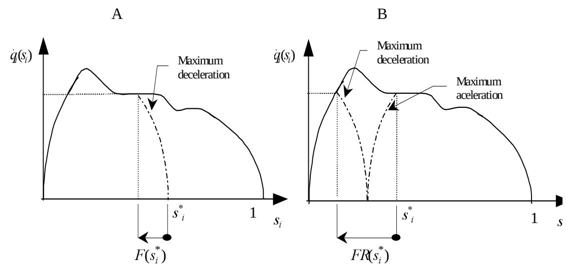

Figure 2: Examples of: (A) *Breaking Zone* and (B) *Breaking-Recovering Zone*.

Then, the coordination of the robots can be modeled as the problem of looking for curves *CC* that avoid any *CRE* inside a cell. From now on, in order to simplify and without lost of generality we will deal with the coordination problem considering the *CS* as a unique cell.

The robot accelerations, a_i, are constrained to a limited range, $a_{i,min} \leq a_i \leq a_{i,max}$, where $a_{i,max}>0$ is the maximum speed up acceleration and $a_{i,min}<0$ is the maximum breaking deceleration.

Definition 7: *Breaking Zone* of a point s_i^*, $F(s_i^*)$, is the minimum Δs_i needed to completely stop R_i at s_i^*, i.e. change from the actual velocity to null velocity at s_i^* (Figure 2-A).■

Definition 8: *Breaking-Recovering Zone* of a point s_i^*, $FR(s_i^*)$, is the minimum Δs_i before s_i^* that R_i needs to slow down from the nominal velocity at $(s_i^* - \Delta s_i)$ to null velocity and recover the nominal velocity at s_i^* (Figure 2-B).■

In order to keep the co-ordination between the robots and avoid collisions, different actions must be taken depending on the relative position of the robots when they enter into a new cell, i.e. depending on the point of entrance to the cell. For each robot R_i, five different regions are defined for each *CRE* (Figure 1). These regions are determined by the following partition of axis s_i :

$SC_{i1} = [0, ej_i - FR(ej_i)]$
$SC_{i2} = [ej_i - FR(ej_i), ej_i - F(ej_i)]$
$SC_{i3} = [ej_i - F(ej_i), ej_i]$
$SC_{i4} = [ej_i, ei_i]$
$SC_{i5} = [ei_i, 1]$

Note that $FR(ej_i) = SC_{i2} + SC_{i3}$.

4. PROPOSED METHODOLOGY

If the robots enter into a *CS* cell following a CC_C (i.e. the entrance point belongs to a SC) a collision will take place at some point in the cell. The goal is to minimally slow down one or both of the robots in order to commute to a CC_F. In this way, if R_i is delayed the system will move towards the *CC* over e_j (Figure 1). This approach should not be confused with the *time scheduling* proposed by Lee and Lee (1987), where the modification of the profile is based on the application of successive cutback of velocity formed by deceleration followed by acceleration until the collision is avoided.

The robot R_i to be delayed (in correspondence to the escape point *ej*) is selected trying to minimise:

- The delay $\Delta\tau_i$ of R_i, $i \in \{1,2\}$, if the robot R_i has higher priority, or
- The relative delay between the robots, i.e. minimise $\Delta\tau = |\Delta\tau_1 - \Delta\tau_2|$.

The necessary delay to be applied to a robot R_i can be *limited, LD,* if R_i has not sufficient path length to be completely stopped before arriving to *CRE*, or *non-limited, Non-LD*, otherwise. If the required delay is greater than the maximum available value then a solution does not exist through the selected escape point (*non-solution, NS*).

The co-ordination is called *complete* when the velocity of each robot at the escape point is the originally planned velocity, and it is called *incomplete* otherwise (in this case the velocity of at least one of the robots is smaller than the necessary to remain on the *CC* through the escape point). This paper deals with the *complete* co-ordinations.

Table I shows for robot R_i and escape point *ej* when the co-ordination is limited and complete as a function of the entrance region. The different possibilities to avoid the collision through *e2* are showed in Figure 3.

In order to avoid the collision, one of the robots must be slowed down, which is equivalent to reduce the area below the velocity profile between two particular points. The amount of removed area is equivalent to the path length executed into the CRE_i if the robot would continue with the initial velocity profile.

The modified portion of the velocity profile is called a *patch* profile. Several patch profiles can be used to introduce the minimum needed delay and allow the robot to recover the planned velocity avoiding the collision. Nevertheless, the constraints in the robot accelerations introduce constraints in the patch profiles that determine *extreme* patch profiles for the maximum accelerations and decelerations.

In this work linear piecewise functions are used for the velocity patch profiles, i.e.: the corresponding acceleration profiles are composed of constant segments. Two or three segments are used: one with negative slope (i.e. negative acceleration to slowdown the robot), one with positive slope (i.e. positive acceleration to speedup the robot), and between them it may exist one with null slope (a waiting stage while the other robot advances in the task).

Table 1: Co-ordination possibilities for robot R_i and escape point e_j.

SC_{i1}	SC_{i2}	SC_{i3}	SC_{i4}	SC_{i5}
NLD ⁞ C	NLD ⁞ C / I	LD ⁞ C / I; NS	Not Allowed	The initial plan is always possible.

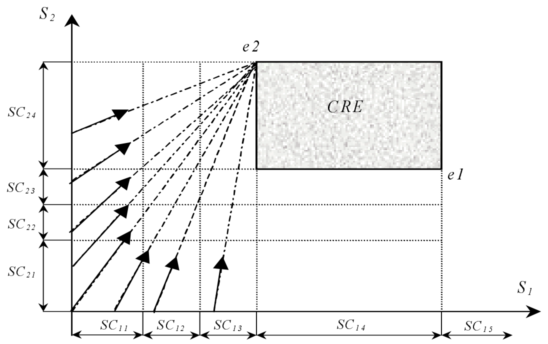

Figure 3: The schematic *CS* shows the different possibilities of avoiding collitions through *e2*.

Two cases are possible:

- Case 1: Slowdown the robot that is just starting the movement.
- Case 2: Slowdown the robot that is already in movement. Two sub-cases can take place:
  - Sub-case 2A: the robot is completely stopped.
  - Sub-case 2B: the robot is not completely stopped.

In order to compute the patch profiles, the required data are obtained from *CS* (Figure 1) under the following assumptions:

- Robot accelerations are constant, $da_i/dt = 0$, with limited modules ($a_{i,min} \le a_i \le a_{i,max}$).
- The nominal path evolutions $s_i(t)$ and $s_j(t)$ are known, as well as their inverse functions $t_i = s_i^{-1}(s_i)$, and $t_j = s_j^{-1}(s_j)$.
- The planned velocity profile $v_i = Q(s_i)$ is known.

From now on, sub-indexes *d* and *e* refer to departure and escape points respectively. The delay $\Delta\tau$ that must be applied is:

$$\Delta\tau = |\, k_e - k_d \,| \quad (1)$$

where k_e is the parameter of the *CC* through the chosen escape point, and k_d is the parameter of the *CC* at the entry point to the cell.

Let's assume that R_i is the robot to be delayed and *ej* is the escape point; s_{jd} is the entry point to the cell of the non-delayed robot R_j and s_{je} its position for the escape point. The path length that must be covered by R_i during the time that R_j needs for going from s_{jd} to s_{je} is Δs_i,

$$\Delta s_i = s_{ie} - s_{id} \quad (2)$$

where s_{id} is the position of R_i at the entry point to the cell and s_{ie} is its escape point.

The initial co-ordination curve CC_{kd} starts at s_{id} . According to the nominal plan, when R_j arrives to s_{je} R_i will arrive to a position s^*_{ie}.

The time needed by R_i to arrive to s^*_{ie} from the beginning of the motion is:

$$t^*_{ie} = s_i^{-1}(s^*_{ie}) \quad (3)$$

R_i has to be in s_{ie} at time t_{ie} obtained from (1) and (3) as,

$$t_{ie} = t^*_{ie} + \Delta\tau \quad (4)$$

The path length that has to be removed from the initial plan of R_i is,

$$\Delta s^*_i = s^*_{ie} - s_{ie} \quad (5)$$

Expression (5) gives the area to be removed from the initial velocity profile of R_i between t_{id} and t_{ie}. Since the nominal path Δs_{n_i} that R_i covers between t_{id} and t_{ie} is known from the original plan, the path length Δs_i that R_i has to cover during that period of time in order to avoid *CRE* can be obtained straight forward as,

$$\Delta s_i = \Delta s_{n_i} - s^*_{ie} \quad (6)$$

The next subsections describe how to obtain the patch profiles for the two possible cases. For the description of the procedures it is assumed that robot R_i is going to be delayed.

4.1 Solution to Case 1

Known data:

Positions: $s_{id} = 0$, s_{ie} , s_{jd} , s_{je}

Velocities: $v_{id}=0$, $v_{ie}=Q_i(s_{ie})$, $v_{jd}=Q(s_{jd})$, $v_{je}=Q_j(s_{je})$, where Q_i y Q_j are the nominal velocity profiles of R_i and R_j.

Times: $t_{je} = s^{-1}(s_{je})$; $t_{jd} = s^{-1}(s_{jd})$

Problem: Obtain $v_{Pi}(t)$ such that $v_{Pi}(0)=0$, $v_{Pi}(t_{ie})=v_{ie}$ and $\int_0^{t_{ie}} v_{pi} dt = s_{ie}$, where $t_{ie}=(t_{je}-t_{jd})$.

Solution: The patch profile is composed of two straight segments, one from the initial point to an auxiliar point C_P, and one from C_P to (t_{ie}, v_{ie}) (Figure 4). C_P lies on a vertical straight segment L_{CP} through the point C_{PExt} that is obtained as the intersection between the axis of time and a straight line with the slope corresponding to a_{imax} (i.e. considering the maximum acceleration) through (t_{ie}, v_{ie}).

The position of C_P on L_{CP} is determined so that the area under the patch profile (that is easy to be computed) equals the needed s_{ie} to avoid the collision.

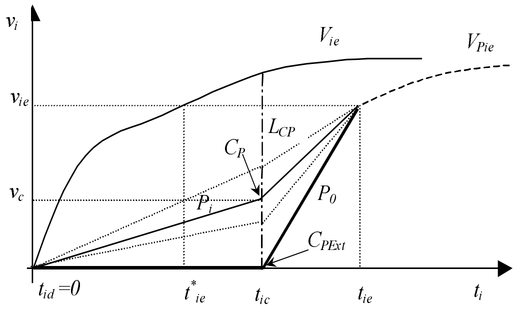

Figure 4: Patch profile for Case 1, (P_0 indicates the extreme patch profile and P_i an example of one).

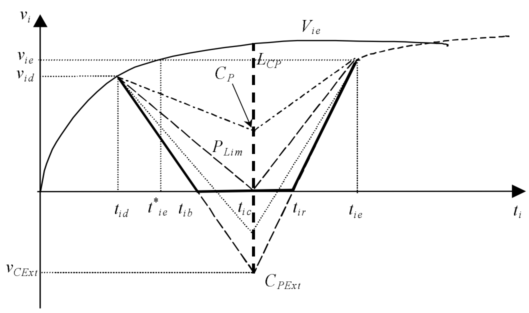

Figure 5: Patch profile for Case 2 (P_{Lim} is the patch profile between Sub-cases 2A and 2B).

The solution is complete because by construction R_i arrives to s_{ie} with the desired velocity v_{ie}; an incomplete solution will be the only option if using the slope a_{imax} the area under the second segment is smaller than s_{ie}.

4.2 Solution To Case 2

Known data:

Positions: s_{id}, s_{ie} , $s_{jd} = 0$, s_{je} , $\Delta s_i = s_{ie} - s_{id}$.

Velocities: $v_{id}=Q_i(s_{id})$, $v_{je}=Q_j(s_{je})$, $v_{jd}=0$, $v_{ie}=Q_i(s_{ie})$, where Q_i y Q_j are the nominal velocity profiles of both R_i and R_j.

Times: $t_{id} = s^{-1}(s_{id})$; $t_{je} = s^{-1}(s_{je})$; $t_{jd} = s^{-1}(s_{jd})$

Problem: Obtain $v_{Pi}(t)$ such that $v_{Pi}(t_{id})=v_{id}$, $v_{Pi}(t_{ie})=v_{ie}$ and $\int_{t_{id}}^{t_{ie}} v_{Pi} dt = s_{ie} - s_{id} = \Delta s_i$, where $t_{ie} = [(t_{je} - t_{jd}) + t_{id}]$.

Solution: The patch profile is composed of three (sub-case 2A) or two straight segments (sub-case 2B). Analogously to Case 1, two straight lines are determined using the minimum and maximum accelerations (i.e. maximum negative and maximum positive slopes) through (t_{id}, v_{id}) and (t_{ie}, v_{ie}) respectively (Figure 5). Let point C_{PExt} be the intersection of these two lines. If C_{PExt} is below the axis of time then the robot is able completely stop (allowing Sub-case 2A), wait still, and recover the velocity that it was initially supposed to have at time t^*_{ie}; otherwise, in order to arrive with the desired velocity at time t^*_{ie} the robot cannot completely stop (allowing only Sub-case 2B).

The desired co-ordination is obtained by selecting a point C_P (above or below the axis of time) such that the area under the patch profile (that is easy to be computed) equals the needed Δs_i to avoid the collision. If Δs_i is smaller than the minimum path, $SC_{i2}+SC_{i3}$, needed to stop and to recover the planned velocity at t_{ie} the co-ordination does not admit a complete solution (i.e the delayed robot will arrive to the escape point with a smaller velocity than the expected one). Then, the solutions for Sub-case 2A and 2B are (Figure 5):

Sub-case 2A. Three straight segments:

- Stage 1: breaking, t_{ib} - t_{id} , with constant deceleration.
- Stage 2: waiting, t_{ir} - t_{ib}
- Stage 3: recovering, t_{ie} - t_{ir} , with constant acceleration.

Sub-case 2B. Two straight segments:

- Stage 1: breaking, tic - tid , with constant deceleration.
- Stage 2: recovering, tie - tic , with constant acceleration.

5. CONCLUSIONS

A methodology for robot coordination has been presented and developed for the case of two robots. It is assumed that the nominal robot trajectories are independently planned and optimized (i.e. use the maximum possible velocity). These trajectories are analyzed in a Coordination Space and, if a collision can take place, the velocity profile of one of the robots is modified slowing it down. The task state may constrain which robot velocity profile should be modified; if a solution exists for each robot the best one can be chosen using different criteria. The main features of the proposed approach are:

- It deals with temporal uncertainty in some robot stages.
- It can be applied on-line according to the temporal uncertainty in the task executions.
- Limited accelerations and decelerations are considered.

The procedure can be extended to more than two robots, but the solutions to the different possible cases have to be determined in a general way. Nevertheless, the procedure developed for two robots can be applied in a iterative way to co-ordinate more than two robots taking two of them at each iteration. The result will not be an optimum co-ordination, but it can be obtained without major difficulties.

REFERENCES

Barraquand, J. and J.-C. Latombe (1991). Robot Motion Planning: A Distributed Representation Approach. *The Int. Journal of Robotics Research*, Vol. 10 (6), 628-649.

Barraquand, J., B. Langlois. and J.-C. Latombe (1992). Numerical Potential Field Techniques for Robot Path Planning. *IEEE Trans. on Sys., Man and Cybernetics*, Vol. 22 (2), 224-241.

Bien, Z. and J. Lee (1992). A Minimum-Time Trajectory Planning Method for Two Robots. *IEEE Transaction on Robotics and Automotion*, Vol. 8 (3), 414-418.

Brooks, R. A. (1983). Solving the Find-Path Problem by Good representation of Free Space. *IEEE Trans on Syst., Man and Cybernetics,* Vol. SMC-13(3), 190-197.

Chang, C., M. J. Chung and B. H. Lee (1994). Collision Avoidance of Two General Robot Manipulators by Minimum Delay Time. *IEEE Trans. on Sys. Man and Cybernetics*, Vol. 24 (3), 517-522.

Connolly, C.I. and J.B. Burns (1990). Path Planning Using Laplace's Equation. *Proc. IEEE Conf. on Robotics and Automotion,* 2102-2106.

Connolly, C. I. (1992). Applications of Harmonic Functions to Robotics. *Proc. 1992 The Int. Sym. on Intell. Control,* 498-502.

Freund, E. and H. Hoyer (1986). Pathfinding in Multirobot Systems: Solution and Applications. *Proc IEEE Int. Conf. on Rob. and Autom.,* Vol. 1(1), 103-111.

Freund, E. and H. Hoyer (1988). Real-Time Pathfinding in Multirobot Systems Including Obstacle Avoidance. *Intern. Journal of Robotics Research*, Vol. 7 (1), 42-70.

Kant, K. and S. W. Zucker (Fall 1986). Toward Efficient Trajectory Planning: The Path-Velocity Decomposition. *The Intern. Journal of Robotics Research,* Vol. 5 (3), 72-89.

Khatib, O. (1986). Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *The Inter. Journal of Robotics Research*, Vol. 5 (1), 90-98.

Latombe, J.-C. (1991). *Robot Motion Planning*. Kluwer Academic Publishers.

Lee, B. H. and C. S. G. Lee (1987). Collision-Free Motion Planning of Two Robots. *IEEE Trans. on Systems, Man and Cybernetics*, Vol. SMC-17 (1), 21-32.

Lee, J. H. S. Nam and J. Lyou (1995). A Practical Collision-Free Trajectory Planning for Two Robot Systems. *Proc. IEEE Int. Conf. on Rob. and Autom*., 2439-2444.

Lozano-Pérez, T. (1983). Spatial Planning: A Configuration Space Approach. *IEEE Transaction on Computers*, Vol. C-32 (2) 108-120.

Todt, E., G. Raush and R. Suárez (2000). Analysis and Classification of Multiple Robot Coordination Methods. *Proc. IEEE Int. Conf. on Rob. and Autom.*, 3158-3163.