# COLLISION FREE MOTION PLANNING FOR TWO ROBOTS OPERATING IN A COMMON WORKSPACE

C A Czarnecki

De Montfort University, UK

## ABSTRACT

This paper presents a collision-free motion planning method for two robots operating in a common three dimensional workspace. Initial time-optimal trajectories are constructed without attention to potential collision. These trajectories are then used to construct a collision map highlighting areas of collision. The collision map uses joint velocity to determine the optimum sampling frequency which makes the collision detection process computationally efficient and suitable for real-time implementation. Collision free trajectories are obtained either by time scheduling or alternatively, utilising a simple search technique to derive a new collision-free path and then deriving a minimum time trajectory for this path. An example is shown which shows the significance of the approach and implementation issues are discussed.

## INTRODUCTION

In many robotic applications ranging from simple material handling to complex assembly, greater flexibility and potential reduction in overall cycle time can be achieved by using additional robots, Grossman et al (1). To enable more than one robot to operate in a common workspace a robot motion planning method that can have the robots move safely and without collision is required. Motion planning involves determining the collision-free path to be traversed, followed by the planning of the trajectory along this path. The path planning problem for a single robot with stationary obstacles has been addressed by many researchers and numerous solutions proposed, Lozano-Perez (2), Brooks (3), However, collision-free motion planning for multi-robot systems has received relatively little attention in the literature. As a result most industrial implementations of multi-robot systems use a heuristic approach to planning the robot motions, for example only allowing one robot into a potential collision region at a time, Chimes (4). Whilst such approaches work adequately, they do not utilise the full potential of these systems.

This paper presents a novel solution to collision free motion planning, which in the context of this paper is defined as follows : Given a task that requires two robots to move from their initial positions to their final positions along predetermined paths, is there any way of accomplishing this task without the robots colliding with each other, and if so, how can we plan the collision free trajectories for both robots considering dynamic constraints on the torques and velocities of the actuators of the robots.

The approach adopted is essentially a three stage process. In a multi-robot environment, each robot presents a time-varying obstacle to all other robots operating in the common workspace. Any motion planning method must therefore provide a mechanism to detect potential collisions. In order to be able to detect collisions we may preplan the paths each robot will follow together with the associated trajectories along these paths. Straight line paths have well reported advantages for multi-robot systems, Lee and Lee (5), Chang et al (6). However, whilst simplifying the collision detection process such an approach may significantly increase the time it takes for the robots to perform a given task. To improve this situation time-optimal trajectories are first derived. These are then utilised to construct a collision map which allows potential collisions to be detected. Finally, if a collision is detected then either the trajectory or the path-trajectory pair are required to be modified to avoid the collision situation.

The format of this paper is as follows. A brief review of previous work related to the new approach is presented followed by a description of the technique used to obtain time-optimal trajectories. The procedure for collision detection is then introduced followed by a discussion of strategies for determining collision free trajectories. The method is illustrated by means of an example.

## PREVIOUS WORK

Much of the early work undertaken in multi-robot systems was reported by Freund and Hoyer (7) (8) (9). They developed a systematic design method, which uses a hierarchical structure for overall system control. Robot dynamics are included in their formulation and useful couplings between axes are utilised. The collision avoidance method uses a fictitious robot to define a collision free trajectory. Lee and Lee (5), provided a solution to the two robot case by deriving a collision map which incorporates both path and

CONTROL '94, 21–24 March 1994. Conference Publication No. 389, © IEE 1994.

trajectory information of the robots moving simultaneously and highlighting collision regions in the workspace. Collisions are then avoided by time scheduling one of the robots around the collision regions. This method relied on straight line motions and collision detection was restricted to the robot wrists. Shin and Bien (10), present an approach where the concept of a virtual obstacle is used to describe potential collision between the links of two robots along designated paths. A notion of virtual co-ordination space is then used to visualise all the collision free co-ordination's of two trajectories. From this the minimum-time collision-free trajectory pair for the two robots is sought. Chang et al (6), use minimum distance functions to describe the constraints that guarantee no collision between two robot arms. The collision-free motion planning problem is then formulated as a pointwise constrained non-linear minimisation problem, and solved by a conjugate gradient method with barrier functions. Lee and Bien (11), propose a method based on a neural optimisation network. The positions or configurations of robots are taken as the variables of the neural circuit and energy of the network is determined by combining various functions, in which one is to make each robot approach its goal and another helps each robot from colliding with other robots or obstacles. Park (12), presents a state space approach where an obstacle map is generated in N dimensional state space, where N is the total number of degrees of freedom of all robots. A local pathfinding method based on repelling pseudo forces is then used to automatically prevent the arms from colliding with fixed obstacles or with each other. Yuh (13), proposed a collision-free path finding algorithm for robots with prismatic joints and then used an adaptive control algorithm to follow the desired collision-free path. Each of these methods relies on the set of moves to be performed by the robots being known before the operation is performed. Whilst this is the case for many automated systems, an increasing number of applications are being reported where the path to be travelled is not well defined, for example in tele-operated robots, Beaumont and Crowder (14), Shaffer and Herb (15), who proposed real-time approaches to detecting collisions. Beaumont and Crowder (14) modelled each robot via a combination of spheres, polyhedrons and cylinders to define an exclusion volume, whilst Shaffer and Herb(15) used a hierarchical data structure to track occupied zones in the three dimensional workspace. A similar approach was reported by Fujimara and Samet (16), who considered two dimensional objects with time representing the third dimension. Dodds (17), and Zalzala et al (18) reported collision free on-line motion planning using a look ahead concept. Parallel processing is deemed necessary due to the vast computational complexities required to be completed in real-time. Example applications are presented for a two robot system.

## TIME-OPTIMAL TRAJECTORIES

Using the Lagrangian formulation, the dynamics of a general n-degree-of-freedom robot can be described by

$$H(q)\ddot{q} + h(q,\dot{q}) = T \qquad (1)$$

where q is the generalised co-ordinate, H(q) is the manipulator inertia matrix, $h(q,\dot{q})$ represents the centrifugal, coriolis and gravitational forces and T is the actuator forces/torques.

The problem is to determine the torques required to be applied such that the system is transferred from one physical configuration to another in the least possible time, subject to the constraints that the torques magnitude is bounded and the angles subtended by the links are physically realisable. For the case when the path to be followed by the robot is specified several solutions to obtaining the time optimal trajectory have been developed, Bobrow (19), Bobrow et al (20). The case when the path for a point to point robot motion can be taken as free has received little attention. Geering et al (21), utilised Pontryagins minimum principle to develop a solution but did not consider physical constraints of the robots in the analysis. Dissanayake et al (22), reformulated the problem incorporating physical constraints and provided a solution using the technique of control parameterisation, Goh and Teo (23). This is the approach utilised here to provide the time-optimal trajectories, for details of the procedure the reader is referred to Dissanayake et al (23). Having determined the time-optimal trajectories for all robots in the system, it is necessary to determine if a collision is going to occur when the robots follow these trajectories. The algorithm for collision detection is presented in the next section.

## COLLISION DETECTION ALGORITHM

The fundamental structure of the method for collision detection is extremely simple. The multi-robot workspace is divided into cuboids with edges parallel to the axes of the space. The limbs of any robots operating in this workspace can for the purpose of collision detection, also be considered as comprising as a series of cuboid primitives with equal dimensions to those used to represent the workspace. This representation has two advantages. Firstly, any robot configuration can be catered for regardless of the number of degrees of freedom and joint type, and secondly, static obstacles in the workspace can also be included in the collision detection process. Now, for any time instant t, a robot in the system can be represented as a series of cuboid primitives occupying cuboid elements of the operational space of the system. If any space cuboid is occupied by more than one object cuboid for the time instant t, a collision situation is deemed to have been detected. Thus, the collision detection process involves checking that no workspace cuboid is occupied by more than one robot or static object cuboid at any one time, for the duration of the complete motion. The practical problems that must be addressed to implement this algorithm are as follows.

1) How should the workspace be divided into cuboids ?

2) How may we efficiently search the operational space for collision conditions ?

The first step in implementing the algorithm involves discretizing the trajectories of each robot in the system. This provides us with a sequence of specific points in space-time. For each sequence of trajectory points at a time instant t, a list of occupied cuboids is constructed. The process is repeated for each robot in the system. For static obstacles the process is undertaken once only. Once a list has been constructed for each device at each sampling interval, a collision map for each sampling instant is constructed which illustrates if any collisions will occur at that time. The collision map is a three dimensional object comprising of cuboids of dimension equal to those used to represent the initial workspace and any robots/objects in that workspace. The map highlights those areas where collisions are predicted for the motions proposed. A collision map cuboid is empty if and only if one or less robots/objects occupy it for that time instant. Collision free motion is

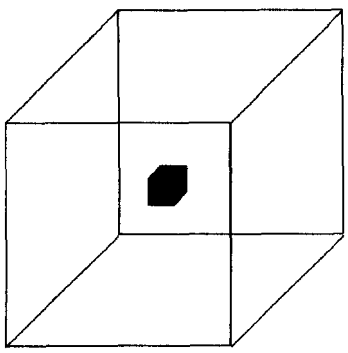


Figure 1. Example collision map showing collision region.

predicted for the time interval t if and only if each cuboid in the collision map is empty. A collision map highlighting collision regions (shaded areas) is shown in Figure 1.

The dimensions of the cuboids used to represent operational space are determined from the selected sampling interval of the time-optimal trajectory. We must ensure that no potential collision situations occur in the time intervals between two successive samples, and hence collision detection evaluations. To cater for this situation we consider the maximum velocity parameter from the proposed trajectories. For a given maximum velocity, there is a maximum distance a robot can move in a sampling period. This maximum distance defines the minimum dimension of the cuboid to be used in the collision detection process. A cuboid size smaller than this may not be used if collision free motion is to be guaranteed. If greater resolution in the collision detection process is required ,for example if very close operation of robots/objects is to be performed, then the sampling frequency must be appropriately increased.

## COLLISION FREE TRAJECTORIES

In the event of a collision situation being detected using the method described above, an avoidance strategy needs to be determined. This can be achieved in a variety of ways. For example, for a two robot system, the path of one robot could be altered whilst the second robot follows its original trajectory to avoid a collision situation. Another approach, as reported by Lee and Lee (5), is to allow the first robot to follow the original trajectory, whilst time scheduling the motion of the second robot along its original path, either via a time delay or a reduction in velocity, to avoid the collision situation.

Both methods have their merits and consequently the approach adopted here is to provide a combination of the two techniques. Consider firstly time scheduling. We aim to avoid the collision situation by either delaying the motion of one of the robots or varying the velocity and hence the trajectory to avoid the collision situation. To determine if time scheduling is to be utilised the following procedure is applied. The time at which a collision is detected is calculated as a percentage of the overall duration of each robot motion. If this percentage is close to the beginning or end of a move then it may be more appropriate to delay the motion of one of the robots. The reasoning behind this approach is that the trajectories followed by each robot have been determined to provide a time-optimal solution. If the collision has been detected and an alternative sub-optimal path is found, then the time to traverse the sub-optimal path is greater than the time delay plus the time to traverse the time- optimal path.

To calculate the time delay an iterative procedure is used. The time delay is always a multiple of the sampling interval used to perform the collision detection. The first collision map that indicates a collision situation requires us to calculate a suitable time delay. The robot to which time scheduling is applied is that which has traversed the smallest percentage of its path at collision time. This robot is then delayed for a period of one sample interval and the collision detection process repeated with the new time delayed trajectory. This process continues until either a suitable time delay is calculated, or no collision free trajectory can be found using a time delay.

If the time scheduling procedure is deemed unsuitable, either due to the percentage path travelled being too large or the time scheduling process failing to determine collision free trajectories, an alternative collision free path is required to be determined for one of the robots. It is assumed the other robot will follow its original trajectory. The robot which has travelled the smaller percentage of its path at the time of collision is the one that replans its path. The problem now becomes one of a collision free findpath in the presence of stationary obstacles as defined by the collision map. Many solutions to this problem exist, Brookes (3), Lozano-Perez, (2)(24), Sahar and Hollerbach (25). Any of these approaches may be utilised to determine the form of the collision free path. The trajectory of this path then needs to be determined. An algorithm for determining a time-optimal trajectory along a specified path is presented by Bobrow et al (20) and is ideally suited to this application.

## APPLICATION

To show the effectiveness of the proposed collision-free planning method, a case study of two robot arms working in close proximity is presented. The layout of

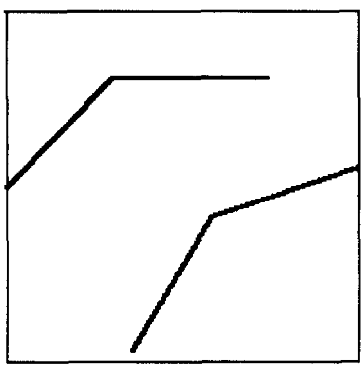
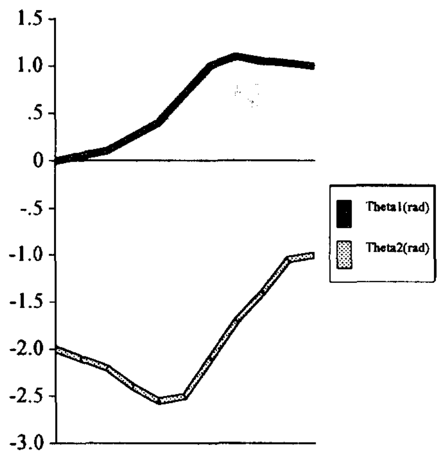


Figure 2. Two robot workcell

the two robot workcell is shown in Figure 2. Each robot is a two-degree-of-freedom planar device with initial and desired final conditions shown in Table 1.

| Inital position | | Final Position | |
|---|---|---|---|
| Theta 1 | Theta 2 (rad) | Theta 1 | Theta 2 (rad) |
| 0.0 | -2.0 | 1.0 | -1.0 |

Table 1. Robot configurations




Time 0.0 - 0.6711s

Figure 3. Time optimal joint trajectories

Using the method of Dissanayake et al (23), the time-optimal path and trajectories are determined. Figure 3 shows the trajectories derived. The total motion time was determined to be 0.6711s. It can be seen from Figure 3 that link 2 of the robot exhibits a clear inward motion during the initial part of the path, which is in the opposite direction to the required movement. Upon link 1 completing its move, link 2 rapidly moves outwards to the required final position. The reason for this is that the time-optimal trajectory method minimises the moment of inertia seen by the joint actuators at any time, so that they can achieve high acceleration and consequently reach their destination faster.

The next stage is to construct the collision map to determine if both robots following their time-optimal trajectories will collide. For this application a two-dimensional collision map suffices. A sampling frequency of 14.9 Hz was required. A collision was detected after 0.47 seconds of motion. The corresponding collision map is shown in Figure 4. The robot positions at this time are shown in Figure 5. At this point we need to determine if it is appropriate to apply time scheduling in order to prevent the collision occurring. With link 1 of both robots being within 10% of the final destination it is indeed appropriate to apply time scheduling.

The initial approach is to determine if a time delay will provide a satisfactory solution. Since both robots have travelled the same percentage of their paths at the time of collision, robot 1 was chosen as the one to be

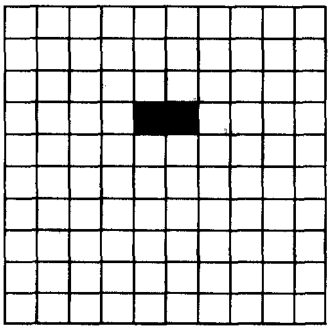


Figure 4. Collision map highlighting collision at t = 0.47s

delayed. The collision map was reconstructed with robot 1's motion being delayed by 0.06711 s. This time a collision was predicted after 0.53 s. The time delay for robot 1 was thus increased by a further 0.06711s, and the collision map reconstructed. Collision free

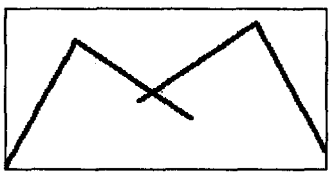


Figure 5. Robot configuration at t=0.47s

motion was now predicted for the robots following their original paths with time-optimal trajectories, with robot 1 delaying its motion by 0.13422s. Thus motion of both robots would be complete after 0.805s.

To put this performance into context a conventional trajectory planning algorithm was applied to the same two link robots for the same initial-final conditions, assuming no collisions. All the joint angles were monotonically increased or decreased according to a trajectory with a triangular velocity profile. The time for performing the motion, before the collision detection process has been applied, was determined as 1.092s.

## CONCLUSIONS

An approach has been proposed to solve the collision free motion planning problem for two robots operating in a common workspace. The approach is based upon predicting collisions via a collision map. The collision map presented is a new approach, which is versatile in that it caters for any robot configurations as well as stationary obstacles in the workspace. Further, collision detection is easily implemented and computationally efficient, utilising maximum joint velocity to determine an optimum sampling frequency. The feasibility of the proposed method is illustrated through an example involving two two-degree-of-freedom robots operating in close proximity.

## REFERENCES


1. Grossman, D.D, Evans, R.C., Summers, P.D., 1985, Rob.&Comp.Int. Manuf., 2, No.2, 135-142

2. Lozano-Perez, T., 1983, IEEE Trans. Comput. C-32, 108-120

3. Brooks, R.A.., 1983, IEEE Trans. Syst. Man. Cybern., SMC-13, 190-197

4. Chimes, P.R., 1985, Robotics Age, Oct 85, 5-10

5. Lee, B.H., Lee, C.S.G., 1987, IEEE Trans. SMC 17 No. 1, 21-32

6. Chang, C., Chung, M.J., Bien, Z., 1990, Robotica,8, 137-144

7. Freund, E., Hoyer, H., 1986, IEEE Int. Conf. RA, 103-111

8. Freund, E., Hoyer, H., 1985, Proc. 3rd Int. Symp. Rob. Res.,253-262

9. Freund, E., 1984, Proc. IEEE Int. Conf. R.A. ,477-490

10. Shin, Y., Bien, Z., 1989, Robotica, 7, 205-212

11. Lee, J., Bien, Z., 1990, Robotica, 8, 185-194

12. Park, W.T., 1984, Proc. 1th Int. Symp. Ind. Robots, 397-405

13. Yuh, J., Jnl. Manuf. Sys. ,9,No.1, 35-43

14. Beaumont, R.G., Crowder, R.M., 1989, IEE Colloquium, 1989/127, 1/1-1/4

15. Shaffer, C.A., Herb, G.M., 1992, IEEE Trans. RA, 8 No. 2, 149-160

16. Fujimura, K., Samet, H., 1989, IEEE Trans. RA, 5 No.1, 61-69

17. Dodds, G., 1989, IEE Colloquium 1989/127, 2/1-2/4

18. Zalzala, A.M.S, Dodds, G.I., Irwin, G.W., 1993, IEEE SERTA 93, 31-36

19. Bobrow, J.E., 1988, IEEE Jnl. RA, 4, No. 4, 443-453

20. Bobrow, J.E., Dubowsky, S., Gibson, J.S., 1985, Int. Jnl. Rob. Res., 4 No.3, 3-17

21. Geering, H.P., Guzzella, L., Hepner, S.A.R., Onder, C.H., 1986, IEEE Trans. AC-31 No.6, 512-518

22. Dissanayake, M.W.M.G., Goh, C.J., Phan-Thien, N., 1991, Robotica, 9, 131-138

23. Goh, C.J., Teo, K.L., 1988, Automatica, 24 No.1, 2-18

24. Lozano-Perez, T., 1982, IEEE Trans. Sys. Man. Cybern., SMC-11, 681 698

25. Sahar, G., Hollerbach, J.M., 1986, Int. Jnl. Rob. Res., 5, No.3, 90-100