Short Papers

Minimum-Time Collision-Free Trajectory Planning for Dual-Robot Systems

Kang G. Shin and Qin Zheng

Abstract—**Collision-free multirobot motion planning can be achieved in two steps: path planning and trajectory planning. Path planning finds for the robots geometric paths to avoid collision with *static* obstacles and trajectory planning determines how fast each robot must move along its geometric path to avoid collision with other *moving* robots. For a dual-robot system, a simple trajectory planning strategy is to let each robot move along its path as fast as possible and delay one robot at its initial position to avoid collision with the other robot. We derive in this paper a sufficient condition under which this simple (thus easy to implement) strategy for dual-robot systems can achieve time optimality as well as collision avoidance, i.e., the two robots reach their final positions without colliding with each other in the minimum amount of time. A demonstrative example is presented, showing how this strategy can be used for loading and unloading applications.**

Index Terms—**Collision avoidance, collision region, minimum-time trajectory planning, multirobot systems.**

I. Introduction

The use of multiple robots in a common workspace can enhance the utilization of robots, increase productivity, and improve the versatility in handling different applications. However, when more than one robot is used in a common workspace, they may become obstacles to each other. Therefore, in addition to avoiding collision with *static* objects, motion planning must also include collision avoidance between *moving* robots [1].

One approach to collision-free motion planning for multiple robots is to decompose the problem into two subproblems: path planning and trajectory planning [2]. Path planning finds for the robots geometric paths that do not intersect static obstacles, and trajectory planning determines how fast each robot must move along its path to avoid collision with others.

This paper deals with the problem of collision-free trajectory planning for multiple robots. For simplicity, we assume that the robots' paths are planned such that at most two robots might collide at a time. Therefore, only dual-robot systems need to be considered for trajectory planning.

The authors of [3] studied the dual-robot collision-avoidance problem. They modeled the robots as spheres and restricted each robot to move along a straight-line path. By approximating the collision region (to be defined later) with rectangles, they showed that delaying one robot at its initial position can achieve a smaller motion time than other collision-avoidance strategies (such as speed reduction or delaying the robot at the middle of its path) if the collision region contains only one rectangle. They also showed an example in which this conclusion may not be true if the collision region contains more than one rectangle.

We extend significantly the result of [3] by allowing arbitrary paths and more accurate geometric models of a robot. In other words, the robots do not have to move along straight lines, and more accurate geometric models than spheres can be used to describe the geometric shape of a robot. Under these relaxed conditions, we show that delaying one robot at its initial position can still achieve the minimum motion time among all collision-avoidance trajectory planning strategies if the collision region is strongly connected (to be defined later). A sufficient condition for verifying the strong connectivity of a collision region is that the two robots might collide at most once when they move along their paths. This condition is easy to check and can be satisfied in most multirobot applications.

The main intent of this paper is to provide an easy-to-use guideline for the collision-free trajectory planning of dual-robot systems. We prove that if the objective is to minimize the robots' motion time, then delaying one robot at its initial position for the minimum period required for collision avoidance is the best solution. So, we do not need more complex strategies like speed reduction or delaying the robot at the middle of the path, which are usually difficult or expensive to implement.

This paper is organized as follows. The main crux of the paper is given in Section II, presenting an algorithm for collision-free trajectory planning and stating its time optimality. A demonstrative example is given in Section III, and the paper concludes with Section IV.

II. Minimum-Time Collision-Free Trajectory Planning

Suppose there are two robots, R_1 and R_2, working in the same workspace. The path for each robot to follow is given in parametric form: $\boldsymbol{q}_i = \boldsymbol{g}_i(s_i)$, $0 \le s_i \le 1$, $i = 1, 2$, where $\boldsymbol{q}_i$ is R_i's joint position vector, and s_i is the parameter representing the robot's position. For example, s_i could be the normalized arc length along the path. Our trajectory planning problem is to determine a pair of feasible collision-free trajectories $\pi : s_i = f_i(t), i = 1, 2$ such that $f_i(0) = 0$, $f_i(t_{ie}) = 1$, $df_i(0)/dt = df_i(t_{ie})/dt = 0$, and the maximum motion completion time of the two robots, $J(\pi) = \max\{t_{1e}, t_{2e}\}$, is minimized, where t_{ie} is the (unconstrained) time for robot R_i to complete the motion.

A trajectory is said to be *feasible* if the above terminal conditions are satisfied and it does not require torques beyond the robots' capabilities. Clearly, a necessary condition for $s_i = f_i(t)$ to be a feasible trajectory is that $f_i(t)$ is C^1, i.e., continuously differentiable. A pair of trajectories $\pi : s_i = f_i(t), i = 1, 2$, is said to be *collision-free* if $V\Big(S_1\big(f_1(t)\big) \cap S_2\big(f_2(t)\big)\Big) = 0$, $0 \le t \le \max\{t_{1e}, t_{2e}\}$, where $S_i\big(f_i(t)\big)$ denotes the physical space occupied by R_i at position $s_i = f_i(t)$ and $V(S)$ denotes the volume of space S. Note that, according to this definition, R_1 does not collide with R_2 if R_1 slides on the surface of R_2. This can be easily met by enlarging the robots' physical dimensions in their mathematical models by the amounts determined based on the required clearance for safety.

There exist several algorithms that produce a minimum-time trajectory for a *single* robot [4]–[6]. One simple but elegant discrete approximation algorithm is the *perturbation trajectory improvement*

Manuscript received December 31, 1990; revised January 10, 1992. This work was supported in part by the National Science Foundation under Grant DMC-878721492. A portion of this paper was presented at the 28th IEEE Conference on Decision and Control, Tampa, FL, 1989.

The authors are with the Real-Time Computing Laboratory, Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor, MI 48109–2122.

IEEE Log Number 9201647

algorithm (PTIA) developed by Shin and Mckay [6]. Based on this algorithm, a pair of collision-free trajectories π^* for two robots can be obtained by delaying one of them at its initial position as follows:

Algorithm 1:

Step 1: Apply the PTIA in [6] to derive two minimum-time trajectories $s_1 = f_1^*(t)$ and $s_2 = f_2^*(t)$ individually for R_1 and R_2, respectively. Let $f_i^*(t) = 0$, $\forall t < 0$ and $f_i^*(t) = 1$, $\forall t > t_{ie}$, $i = 1, 2$.

Step 2: Let $T_{d1} = \{t_{d1} : t_{d1} \geq 0,\ s_1 = f_1^*(t - t_{d1})$ and $s_2 = f_2^*(t)$ are collision–free trajectories$\}$ and $T_{d2} = \{t_{d2} : t_{d2} \geq 0, s_1 = f_1^*(t)$ and $s_2 = f_2^*(t - t_{d2})$ are collision-free trajectories$\}$. Let t_{di}^* be the infimum of T_{di}, $i = 1, 2$.

Step 3: Let $\pi_1 : s_1 = f_{1d}^*(t) = f_1^*(t - t_{d1}^*), s_2 = f_2^*(t)$, and $\pi_2 : s_1 = f_1^*(t), s_2 = f_{2d}^*(t) = f_2^*(t - t_{d2}^*)$. Then $\pi^* = \pi_1$ if $J(\pi_1) \leq J(\pi_2)$, and $\pi^* = \pi_2$ otherwise.

To state the conditions for the time optimality of Algorithm 1, we need the following definitions.

Definition 1: The *collision region* on an $s_1 \times s_2$ plane is defined as $D_c = \{(s_1, s_2) : 0 \leq s_1 \leq 1,\ 0 \leq s_2 \leq 1,$ two robots collide with each other if they are at positions $\boldsymbol{q}_1 = \boldsymbol{g}_1(s_1)$ and $\boldsymbol{q}_2 = \boldsymbol{g}_2(s_2)$, respectively$\}$.

Definition 2: D_c is said to be *strongly connected* if $\forall\ 0 \leq s_1^1 \leq s_1^2 \leq 1,\ 0 \leq s_2^1 \leq s_2^2 \leq 1,\ D_c \cap \{(s_1, s_2) :\ s_1^1 \leq s_1 \leq s_1^2,\ s_2^1 \leq s_2 \leq s_2^2\}$ is either connected or an empty set. Clearly, the condition of strong connected D_c is stronger than connectedness but weaker than the convexity of D_c.

We have the following theorem stating the time optimality of Algorithm 1.

Theorem: Let π^* be the trajectories obtained from Algorithm 1. If A1) no collision occurs when at least one robot is at its starting position ($s = 0$) or ending position ($s = 1$), and A2) D_c is strongly connected, then $J(\pi^*) \leq J(\pi)$ for *any* feasible collision-free trajectories $\pi : s_i = f_i(t),\ i = 1, 2$.

The proof of this theorem is quite involved, so only a sketch of the proof is given here to illustrate the main ideas. A detailed, rigorous proof can be found in [7]. Specifically, we will prove the following results.

Result 1: If A1 holds, then Algorithm 1 can always produce a pair of feasible collision-free trajectories.

Result 2: If the collision region D_c is strongly connected, then we only need to consider strictly monotone trajectories in order to obtain minimum-time trajectories. By "strictly monotone trajectory" we mean that, once a robot starts moving, it keeps moving forward with a positive speed until it reaches its final position.

Result 3: Among all feasible collision-free strictly monotone trajectories, the one determined by Algorithm 1 has the minimum motion completion time.

It is easy to see Result 1 since Assumption A1 ensures that all possible collisions can be avoided by delaying one of the two robots at its starting position by at most the amount of time needed for the other robot to reach its final position.

Result 2 plays a key role in the proof of the theorem. Though Result 2 may seem obvious, it is difficult to prove rigorously. The difficulty lies in the dependence of Result 2 on the strong connectedness of D_c. If D_c is not connected, one can easily see a situation where Result 2 does not hold. The optimal trajectory on the $s_1 \times s_2$ plane could be the one going through between two disconnected collision regions. This means that R_2 might need to move to a middle point, wait there, and move again. However, the assumption on the connectedness of D_c still cannot lead to Result 2. The weakest assumption with which

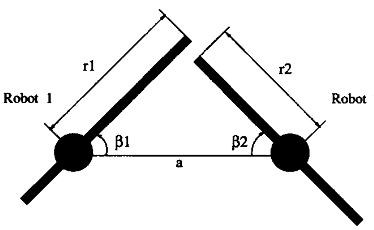

Fig. 1. The configuration of a dual-robot system.

we could prove Result 2 is the strong connectedness of D_c. See the proof of Lemma 2 in [7] for a detailed account of this.

Result 3, which leads to the conclusion of the theorem, can be easily obtained from Result 2 by using the fact that strictly monotone trajectories and the connected collision region, D_c, on the $s_1 \times s_2$ plane lead to connected collision regions on the $s_i \times t$ planes, $i = 1, 2$. We prove the theorem by replacing any monotone trajectories with those obtained from PTIA and shifting one of them upward by the minimum amount needed to avoid connection to the collision region on the $s_i \times t$ plane.

The above theorem is the main contribution of this paper. We have tried our best to impose the weakest assumption on the collision region D_c. Note that the strong connectedness of D_c is much weaker than the convexity of D_c, which could greatly simplify the proof of the theorem.

The practical significance of the above theorem depends on how easily Conditions A1 and A2 can be satisfied in real applications. In what follows we give some physical insights into these conditions.

Assumption A1 means that no collision will occur when at least one robot is at its initial position ($s = 1$) or final position ($s = 1$). This is usually true since these two positions are often the working positions of the robots, i.e., robots perform useful operations there. It is reasonable to plan one robot's path not to intersect the working positions of the other robot (at the path planning stage).

Assumption A2 requires that the collision region D_c be strongly connected, which depends on the geometric shapes of both robots and the given paths for the robots to follow. Geometrically, a sufficient condition for D_c to be strongly connected is that a robot may enter D_c at most once while traveling in the north–east direction on the $s_1 \times s_2$ plane. The physical meaning of this sufficient condition is that, if both robots always move forward (they may stop moving somewhere but never move back) along their paths, they may collide at most once. Under a proper path planning policy, this is not an unreasonable limitation for most multirobot applications. (See a typical loading and unloading example in the next section.)

The theorem also provides a tool for solving more complex multirobot motion planning problems. For example, each robot may be required to perform several sequential operations in different areas of its workspace. Using the theorem, one can search for an optimal combination of delays in these areas to obtain collision-free trajectories by minimizing the overall motion completion time.

Another potential application of the theorem is to combine path planning and trajectory planning. In this paper, we have separated path planning from trajectory planning in such a way that path planning is concerned only with collision avoidance between robots and *static* obstacles, whereas trajectory planning deals with collision avoidance between *moving* robots. However, there is no reason why path planning cannot assume some of the responsibilities of trajectory planning. At one extreme, the paths of two robots can be planned such that they do not intersect at all. Then there will be no need to delay a robot. However, this does not necessarily mean a smaller

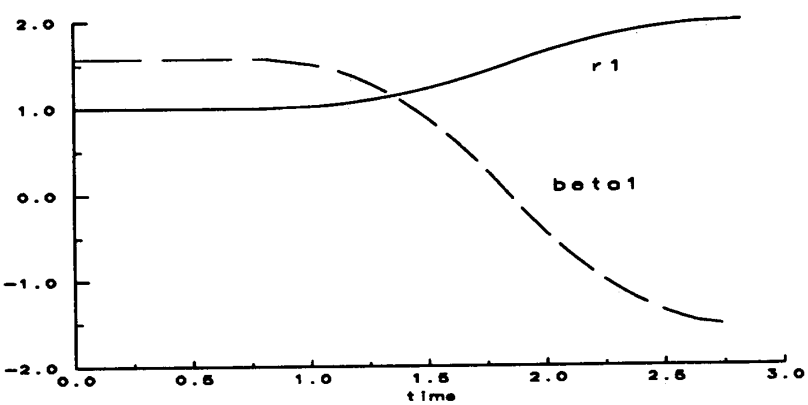

Fig. 2. Robot 1's trajectory.

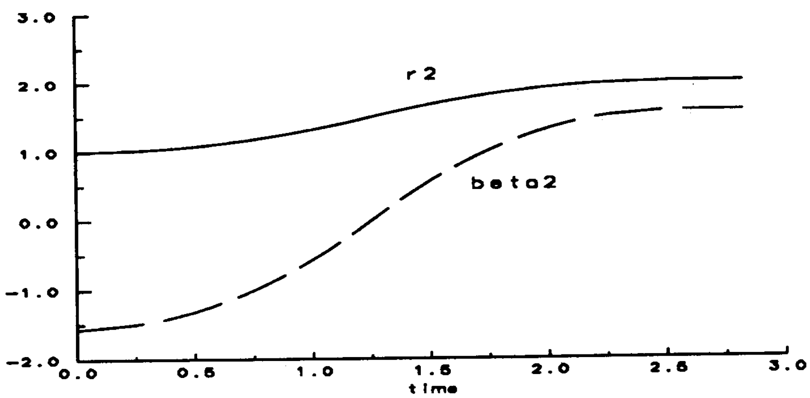

Fig. 3. Robot 2's trajectory.

motion completion time since nonintersecting paths may be take longer times than the intersecting ones. Thus, it is a better idea to construct several pairs of paths with different degrees of intersection and use the theorem to determine the completion time for each of them. The one with the minimum completion time is finally chosen.

III. An Example

In this section, we present an example to show the use of the above theorem for two robots in loading and unloading applications. As discussed in [8], each robot can be modeled with the cylindrical configuration of a rotational joint β and a translational joint r as shown in Fig. 1. The vertical movement is usually restricted and, therefore, not considered for collision avoidance.

The parameters to be used in the example are as follows. The distance between the rotational joints of two robots is a constant $a = 2$ m. The initial and final positions of $R1$ are $r_1 = 1$ m, $\beta_1 = \pi/2$ and $r_1 = 2$ m, $\beta_1 = -\pi/2$, and those of R_2 are $r_2 = 1$ m, $\beta_2 = -\pi/2$ and $r_2 = 2$ m, $\beta_2 = \pi/2$. $R1$ moves along the path $r_1 = 1 + s_1$, $\beta_1 = (1 - 2s_1)\pi/2$ and $R2$ along $r_2 = 1 + s_2$, $\beta_2 = (2s_2 - 1)\pi/2$, $0 \leq s_1 \leq 1$, $0 \leq s_2 \leq 1$. Clearly, two paths intersect and collisions may occur between the two robots moving along these two paths. For simplicity, the torque constraints are assumed to be $|d^2r_1/dt^2| \leq 1(m^2/s)$, $|d^2\beta_1/dt^2| \leq 3$, $|d^2r_2/dt^2| \leq 1(m^2/s)$, $|d^2\beta_2/dt^2| \leq 2$. If both robots are moving with the maximum velocities determined by the PTIA, a collision occurs.

Using a bisectional search for t^*_{di}, the collision-free trajectories π^* obtained from Algorithm 1 are shown in Figs. 2 and 3. Both robots move with their maximum speeds, and $R1$ is delayed at its initial position for 0.81 s to avoid collision with $R2$. The motion completion time is 2.86 s.

Clearly, no collision can occur when one robot is at its initial or final position. Also, the two robots can collide at most once when they move forward along their specified paths. So, both Assumptions A1 and A2 of the theorem are satisfied. We conclude that the collision-free trajectories obtained from Algorithm 1 are time optimal. In other words, no other collision-avoidance trajectory planning strategies can achieve smaller motion completion times than those obtained from this algorithm.

IV. Conclusion

We presented a sufficient condition under which the time optimality of dual-robot collision-free trajectory planning can be achieved by simply delaying one of the two robots. This result can relieve the users of the need to wade through more complex solutions to the collision-avoidance problem when this condition is satisfied. An example is also presented to show the applicability of the result.

References

[1] R. Mehrotra and M. R. Varanasi, *Multirobot Systems*. Los Alamitos, CA: IEEE Computer Soc. Press, 1990.

[2] 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.
[3] B. H. Lee and C. S. G. Lee, "Collision-free motion planning of two robots," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-17, no. 1, pp. 21–32, 1987.
[4] J. E. Bobrow, S. Dubowsky, and P. Gibson, "On the optimal control of robotic manipulators with actuator constraints," in *Proc. Automat. Contr. Conf.*, June 1983, pp. 782–787.
[5] K. G. Shin and N. D. McKay, "Minimum–time control of robotic manipulators with geometric path constraints," *IEEE Trans. Automat. Contr.*, vol. AC–30, no. 6, pp. 531–541, June 1985.
[6] ——, "Robust trajectory planning for robotic manipulators under payload uncertainties," *IEEE Trans. Automat. Contr.*, vol. AC-32, no. 12, pp. 1044–1054, Dec. 1987.
[7] K. G. Shin and Q. Zheng, "Minimum time trajectory planning for dual robot systems," in *Proc. Contr. Decision Conf.*, Dec. 1989, pp. 2506–2511.
[8] E. Freund and H. Hoyer, "Pathfinding in multi-robot systems: Solution and applications," in *Proc. IEEE Int. Conf. Robotics Automat.*, vol. 1, 1986, pp. 103–111.

Multiple-Goal Kinematic Optimization of a Parallel Spherical Mechanism with Actuator Redundancy

Ronald Kurtz and Vincent Hayward

Abstract—**A new kinematic design will be presented that is fully parallel and actuator redundant. Actuator redundancy refers to the use of more actuators than are strictly necessary to control the mechanism without increasing the mobility. The uses of this form of redundancy include the ability to partially control the internal forces, increase the workspace, remove singularities, and augment the dexterity. Optimization will take place based on several objective functions. The kinematic dexterity, the forces present at the actuators, and the uniformity of the dexterity over the workspace will all be investigated as potential objectives. Global measures will be derived from each of these quantities for optimization purposes. Examining only a single objective may not yield an acceptable design. Instead, optimization of several factors is done simultaneously by specifying a primary objective and minimum performance standards for the secondary measures.**

I. Introduction

Parallel kinematic structures are important means to improve the performance of robot manipulators. Justifications for this have been extensively discussed in the literature, for example [10], [13], [15], in terms of structural and actuator advantages. Much attention have been devoted to the analysis of these structures from the kinematic (position and velocity) and dynamic viewpoints. A survey of these techniques, even partial, would be quite impossible to fit here, so references will be made as needed.

Of the immense number of possibilities offered by parallel kinematic structures, it appears that two have been extensively used: the pantograph mechanism and its derivatives (vast numbers of industrial manipulators), and the iso-static (or Stewart) platform. Many others have been proposed and used, but once again, a survey would be outside the scope of this paper.

Manuscript received June 8, 1990; revised March 9, 1992. This work was supported by the Natural Sciences and Engineering Research Council of Canada; Les Fonds pour la Formation des Chercheurs et l'Aide à la Recherche, Québec; and the Institute for Robotics and Intelligent Systems Centers of Excellence of Canada.

The authors are with the McGill University Research Center for Intelligent Machines, Montreal, Qc, H3A 2A7, Canada.

IEEE Log Number 9202284.

The observation of biological manipulators was used to suggest alternate structures that could contribute to the design of manipulators. From a kinematic viewpoint, dualities between serial and parallel mechanisms have been pointed out [17]. One of them is not often discussed: the relative work volume for a given mechanical mobility. Serial chains have a large workspace (and poor structural properties); parallel ones have a reduced workspace (but good structural properties). It is not surprising that biological manipulators have hybrid structures: bones, tendons, and skeletal muscles form numerous chains closed regionally, yet the general architecture of biological manipulators is serial [8]. In addition, the work volume of biological manipulators can be large, for example, the human arm [12].

Part of the human shoulder can be approximated by a spherical ball and socket joint actuated by six muscle groups to control three degrees of freedom when four are strictly needed. This can be viewed as a case of actuator redundancy in which the redundant shoulder muscles are used to supply the internal forces needed to keep the humeral head (ball) firmly anchored to the glenoid (socket) throughout a large workspace. It also can be viewed as a means to increase workspace, among many other plausible interpretations such as the possibilities offered by antagonist actuation. This has inspired the development of a mechanical counterpart with similar properties: a three-degree-of-freedom mechanism fully parallel and actuator redundant. A detailed kinematic analysis has revealed that actuator redundancy can be used for more than just controlling internal forces [7]. Increasing the workspace, removal of singularities, decreasing joint forces, and improving dexterity are all possible with this technique.

This paper is concerned with the design optimization of the said mechanism, which consists of determining fixed geometric parameters in accordance with some set objectives. For any design problem, there will potentially be many objectives that cannot all be satisfied simultaneously. In addition, technological constraints must be considered before a practical design can be realized. Here, the focus is on kinematics and on the determination of a *range* of good designs from this perspective.

Purely numerical methods of optimization are avoided as they would give no insight into the workings of the mechanism. Instead, the approach is to form a hierarchy of objectives. Each objective will be examined in turn to reveal the best designs. The idea is to maximize the high-order objectives such that the low-order objectives satisfy some minimum criteria.

II. Description of the Mechanism

The general case of a spherical fully parallel platform mechanism with linear actuation consists of a movable body attached to n legs with one actuator per leg. Each leg has one prismatic joint interposed between two spherical joints. A point of the platform is constrained by a spherical joint permitting freedom of orientation (see Fig. 1). Let us denote the center of rotation as C, the point of attachment of each leg to the platform as P_i, and the point of attachment of each leg to a fixed frame A_i, $i = 1, ..., n$. For the nonredundant case, $n = 3$, with $n > 3$ for all the redundant cases. The parallel mechanism to be described will have four actuators, one being "redundant"—yet essential!