Coordinating Multiple Robots with Kinodynamic Constraints along Specified Paths

Jufeng Peng and Srinivas Akella

Department of Computer Science
Rensselaer Polytechnic Institute
Troy, New York 12180, USA

Abstract. This paper focuses on the collision-free coordination of multiple robots with kinodynamic constraints along specified paths. We present an approach to generate continuous velocity profiles for multiple robots that avoids collisions and minimizes the completion time. The approach, which combines techniques from optimal control and mathematical programming, consists of identifying collision segments along each robot's path, and then optimizing the robots velocities along the collision and collision-free segments. First, for each path segment for each robot, the minimum and maximum possible traversal times that satisfy the dynamics constraints are computed by solving the corresponding two-point boundary value problems. The collision avoidance constraints for pairs of robots can then be combined to formulate a mixed integer nonlinear programming (MINLP) problem. Since this nonconvex MINLP model is difficult to solve, we describe two related mixed integer linear programming (MILP) formulations that provide schedules that are lower and upper bounds on the optimum; the upper bound schedule is designed to be a continuous velocity schedule. The approach is illustrated with coordination of multiple robots, modeled as double integrators subject to velocity and acceleration constraints. Implementation results for coordination of 12 robots are described.

1 Introduction

Coordinating multiple robots with kinodynamic constraints, i.e. simultaneous kinematic and dynamics constraints ([11]), in a shared workspace without collisions has applications in manufacturing cells ([33]), AGV coordination in harbors and airports ([2]), and air traffic control ([5]). The general problem requires finding the trajectory (path and velocity profile) of each robot such that the specified objective, such as the task completion time, total time, or energy consumption, of the system is minimized.

This paper deals with the optimal coordination of multiple robots moving with kinodynamic constraints along specified paths. While prior work mostly addressed either the collision-free path coordination problem of several robots without considering dynamics constraints ([29],[26],[41]), or the search for time-optimal motions for a single robot ([7],[39]), the contribution of this paper is an approach to generate continuous velocity profiles that avoid collisions between multiple robots and that also minimize the task completion time under the kinodynamic constraints. An example application is the coordination of AGVs along fixed paths in harbors and airports. We must satisfy kinematic constraints, such as avoiding collisions between robots and with moving obstacles, and dynamics constraints, such as velocity and acceleration bounds, on the robot motions. By identifying the collision segments along a robot's path, we can combine the collision avoidance constraints for pairs of robots to formulate a mixed integer nonlinear programming (MINLP) problem. Since the resulting nonconvex MINLP formulation is difficult to solve, we use two related mixed integer linear programming (MILP) formulations, the *improved instantaneous model*, which provides a lower bound on the optimal solution, and the *setpoint model*, which provides a continuous velocity schedule that is an upper bound on the optimal solution. In this paper, we illustrate the approach using robots modeled as double integrators subject to velocity and acceleration constraints, and briefly discuss applications to more general robot systems.

2 Related Work

Multiple Robot Coordination: Motion planning for multiple robots requires moving each robot from its initial to its goal configuration, while avoiding collisions with static obstacles or with other robots ([24]). This problem is highly underconstrained, and Hopcroft, Schwartz, and Sharir [16] showed that even a simplified two-dimensional case of the problem is PSPACE-hard. Recent efforts have focused on reducing the dimension of the configuration space by grouping robots (Aronov et al. [3]) or using probabilistic approaches. A potential field randomized path planner was applied to multiple robot planning (Barraquand, Langlois, and Latombe [4]), and probabilistic roadmap planner have been developed for multiple car-like robots (Svestka and Overmars [43]) and multiple manipulators (Sanchez and Latombe [35]).

A slightly more constrained version of the problem is obtained when all but one of the robots have specified trajectories. This is the problem of planning a path and velocity for a single robot among moving obstacles (Reif and Sharir [32], Kant and Zucker [19]). Erdmann and Lozano-Perez [12] obtain a heuristic solution for planning the motions of multiple robots by assigning priorities to robots and sequentially searching for collision-free paths for the robots in the configuration-time space.

If the problem is further constrained so that the paths of the robots are specified, one obtains a path coordination problem. O'Donnell and Lozano-Perez [29] developed a method for path coordination of two robots. LaValle and Hutchinson [26] addressed a similar problem where each robot was constrained to a specified configuration space roadmap. Simeon, Leroy, and Laumond [41] perform path coordination for a very large number of car-like robots, where robots with intersecting paths can be partitioned into smaller sets. Trajectory coordination is a closely related problem where the trajectory (path and velocity) of each robot is specified. Previous work on trajectory coordination focused almost exclusively on dual robot systems (Shin and Zheng [40], Bien and Lee [6], Chang, Chung, and Lee [9]). Akella and Hutchinson [1] recently developed an MILP formulation for the trajectory coordination of large numbers of robots by changing robot start times. Our work extends these problem classes by additionally considering dynamics constraints and generating continuous velocity profiles.

Time-optimal Trajectory Planning: There is a large body of work on the time optimal control of a single manipulator. Bobrow, Dubowsky, and Gibson [7], and Shin and McKay [39] developed algorithms to generate the time-optimal velocity profile of a manipulator moving along a specified path. Trajectory planning directly in the $2n$-dimensional state space that considers both kinematic and dynamic constraints is called *kinodynamic planning*. Sahar and Hollerbach [34] and later Shiller and Dubowsky [38] developed algorithms for global near minimum-time trajectory generation (path and velocity) for a manipulator with dynamics and actuator constraints using grid-based search spaces. Donald et al. [11] developed a polynomial time approximation algorithm to generate near time-optimal trajectories for a single robot that satisfy kinematic and dynamic constraints. Fraichard [14] describes a trajectory planner for a car-like robot with dynamics constraints moving along a given path among moving obstacles. Recent work has focused on randomized kinodynamic planning, including the use of rapidly exploring random trees (Lavalle and Kuffner [27]) and probabilistic roadmaps (Hsu et al. [17]).

Air Traffic Control: Conflict resolution among multiple aircraft in a shared airspace is closely related to multiple robot coordination. Tomlin, Pappas, and Sastry [42] synthesized provably safe conflict resolution maneuvers for two aircraft using speed and heading changes. Kosecka et al. [21] used potential field planners to generate conflict resolution maneuvers. Bicchi and Pallottino [5] modeled aircraft with constant velocity and curvature bounds and generated minimum total time collision-free paths using given waypoints for three aircraft. Schouwenaars et al. [37] used a discretized system model to develop an MILP formulation for fuel-optimal path planning of multiple vehicles, including moving obstacles. Pallottino, Feron, and Bicchi [30] generate conflict-free paths to minimize the total flight time for the cases when either instantaneous velocity changes or heading angle changes are allowed.

3 Problem Overview

Given a set of n robots $\mathcal{A}_1, \ldots, \mathcal{A}_n$ with specified paths and dynamics constraints, the goal is to find the control inputs along the specified paths so that the completion time of the set of robots is minimized and their motions are collision free. We assume that the start and goal configurations of each robot are collision-free. This assumption guarantees the existence of a solution since a feasible schedule is for one robot at a time to move along its path, for some arbitrary sequencing of the robots. The only assumptions about the specified paths are that they are free of static obstacles, and can be traversed by the robots without violating kinematic constraints. We further assume that each robot moves only forward along its path.

3.1 Paths and Collision Zones

Each robot $\mathcal{A}_i$ is given a path γ_i, which is a continuous mapping $[0,1] \to \mathcal{C}_i^{free}$. Let $\mathcal{S}_i = [0,1]$ denote the set of parameter values s_i that place the robot along the path γ_i. The *coordination space* for n robots is defined as $\mathcal{S} = \mathcal{S}_1 \times \mathcal{S}_2 \times \cdots \times \mathcal{S}_n$. A feasible coordination is a schedule $\psi(t) : \mathcal{R}^+ \to \mathcal{S}$ in which $s_{init} = (0,0,\ldots,0)$ and $s_{goal} = (1,1,\ldots,1)$ and the robots do not collide. Note that there is a 1-to-1 mapping between s and the path length. Based on the topology of the coordination space, we define *collision zones*.

A *collision pair* $\mathcal{CP}_{ij}(s_i, s_j)$, where s_i and $s_j \in [0,1]$, is defined as a pair of configurations $(\gamma_i(s_i), \gamma_j(s_j))$ where robot $\mathcal{A}_i$ and robot $\mathcal{A}_j$ collide, i.e., $\mathcal{A}_i(\gamma_i(s_i)) \cap \mathcal{A}_j(\gamma_j(s_j)) \neq \emptyset$. A *collision segment* for robot $\mathcal{A}_i$ is a contiguous interval $[s_i^{start}, s_i^{end}]$ over which $\mathcal{A}_i$ collides with some $\mathcal{A}_j$. That is, $\forall s_i \in [s_i^{start}, s_i^{end}], \exists s_j$ such that $\mathcal{A}(\gamma_i(s_i)) \cap \mathcal{A}(\gamma_j(s_j)) \neq \emptyset$.

An ordered pair of maximal contiguous intervals $([s_i^{start}, s_i^{end}], [s_j^{start}, s_j^{end}])$ in the coordination space $\mathcal{S}$ constitute a *collision zone* $\mathcal{CZ}_{ij}$ if and only if any point in one interval results in a collision with at least one point in the other interval (Figure 1). That is, $\forall s_i \in [s_i^{start}, s_i^{end}], \exists s_j \in [s_j^{start}, s_j^{end}]$ such that $\mathcal{A}(\gamma_i(s_i)) \cap \mathcal{A}(\gamma_j(s_j)) \neq \emptyset$, and $\forall s_j \in [s_j^{start}, s_j^{end}], \exists s_i \in [s_i^{start}, s_i^{end}]$ such that $\mathcal{A}(\gamma_i(s_i)) \cap \mathcal{A}(\gamma_j(s_j)) \neq \emptyset$.

In Figure 1, the collision zones are $([a_1, a_2], [b_3, b_4])$, and $([a_3, a_4], [b_1, b_2])$. A maximal interval that is not within any collision zone is called a *collision-free segment* . Each robot's path is decomposed into one or more collision segments and collision-free segments.

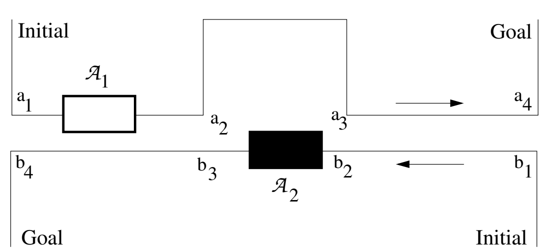


Fig. 1. Example with two translating robots with two collision zones.

3.2 Optimal Control Problem For A Single Robot

For a single robot $\mathcal{A}$ moving along a path segment, let $\mathbf{x}(t)$ represent the state, $\mathbf{u}(t)$ be the control, γ be the path of the robot, $J(\mathbf{x}, \mathbf{u})$ be the objective function, and $g(\mathbf{x})$ and $q(\mathbf{u})$ be the inequality constraints on the state variables and controls respectively. Then the optimal control problem, to compute the minimum and maximum time taken by the robot to traverse the segment subject to its dynamics and path constraints, can be written as:

$$
\begin{aligned}
&\text{Minimize} \quad J(\mathbf{x}, \mathbf{u}) \\
&\quad \text{subject to:} \\
&\qquad\qquad \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \\
&\qquad\qquad g(\mathbf{x}) \leq 0 \\
&\qquad\qquad q(\mathbf{u}) \leq 0 \\
&\qquad\qquad \mathbf{x}(0) = \mathbf{x}_{start} \\
&\qquad\qquad \mathbf{x}(\Delta T) = \mathbf{x}_{end} \\
&\qquad\qquad \mathbf{x} \in \gamma
\end{aligned}
$$

The minimum time control problem has $J(\mathbf{x}, \mathbf{u}) = \Delta T$, and the maximum time control problem has $J(\mathbf{x}, \mathbf{u}) = -\Delta T$ where $\Delta T = \int_0^{\Delta T} 1 dt$ is the time to traverse the segment. Feasible robot motions that give a minimum and a maximum of the objective over each segment are obtained by solving two TPBVPs (two-point boundary value problems) for each segment.

3.3 Coordination of Multiple Robots

Now consider the multiple robot system in which each robot has a specified path and dynamics constraints. The goal is to coordinate these robots to minimize a specified objective; in this paper it is the global completion time. The path of each robot is decomposed into collision segments and collision-free segments. The coordination of multiple robots can then be modeled as a mixed integer nonlinear programming (MINLP) problem, with each robot satisfying the traversal time and collision avoidance constraints over each of its segments. Since this MINLP problem with nonconvex constraints is difficult to solve, we obtain schedules that provide a lower bound and an upper bound on the optimal solution by solving two related mixed integer linear programming (MILP) problems. We illustrate this approach using the double integrator model from optimal control ([8]).

4 Instantaneous Model

We first consider a simplified model where each robot always moves using its highest speed v_{max}, and is permitted to instantaneously change its velocity.

That is, each robot has infinite acceleration, and can instantaneously accelerate to v_{max} or instantaneously decelerate to zero velocity from v_{max}. We will refer to this as the *instantaneous model*, since it provides a schedule with instantaneous starts and stops.

4.1 Mixed Integer Linear Programming (MILP) Formulation

We now present an MILP formulation for the instantaneous model. Let t_{ik} be the time when robot $\mathcal{A}_i$ begins moving along its kth segment and τ_{ik} be the traversal time for $\mathcal{A}_i$ to pass through segment k. Let ΔT_{ik}^{min} and ΔT_{ik}^{max} represent the minimum and maximum traversal time for $\mathcal{A}_i$ between the start point of segment k and the start point of segment $k+1$, giving the *traversal time constraints* $\Delta T_{ik}^{max} \geq \tau_{ik} \geq \Delta T_{ik}^{min}$. For the instantaneous model, $\Delta T_{ik}^{max} = \infty$. The minimum time for $\mathcal{A}_i$ to traverse a segment of length S_{ik} at its maximum velocity $v_{i,max}$ is $\Delta T_{ik}^{min} = S_{ik}/v_{i,max}$. The motion completion time C_{max} for the set of robots is greater than or equal to the completion time of each robot, leading to the *completion time constraint* $C_{max} \geq t_{i,last} + \tau_{i,last}$ for each robot $\mathcal{A}_i$. Consider robots $\mathcal{A}_i$ and $\mathcal{A}_j$ with a shared collision zone where k and h are their respective collision segments. A sufficient condition for collision avoidance is that $\mathcal{A}_i$ and $\mathcal{A}_j$ are not simultaneously in their shared collision zone. That is, $t_{jh} \geq t_{i(k+1)}$ (when $\mathcal{A}_i$ exits segment k before $\mathcal{A}_j$ enters segment h) or $t_{ik} \geq t_{j(h+1)}$ (when $\mathcal{A}_j$ exits segment h before $\mathcal{A}_i$ enters segment k). These disjunctive constraints are converted to standard conjunctive form ([28]) by introducing δ_{ijkh}, a binary variable that is 1 if robot $\mathcal{A}_i$ goes first along its kth segment and 0 if robot $\mathcal{A}_j$ goes first along its hth segment, and M, a sufficiently large positive number. The resulting *collision avoidance constraints* to ensure the two robots $\mathcal{A}_i$ and $\mathcal{A}_j$ are not simultaneously in their shared collision zone are:

$$
\begin{aligned}
&t_{jh} - t_{i(k+1)} + M(1 - \delta_{ijkh}) \geq 0 \\
&t_{ik} - t_{j(h+1)} + M\delta_{ijkh} \geq 0
\end{aligned}
$$

The constraints for all robots are combined to form the instantaneous MILP formulation:

$$
\begin{aligned}
&\text{Minimize } C_{max} \\
&\text{subject to:} \\
&C_{max} \geq t_{i,last} + \tau_{i,last} \qquad \text{for} \quad i = 1, \ldots, n \\
&t_{ik} \geq 0 \\
&t_{i(k+1)} = t_{ik} + \tau_{ik} \\
&\Delta T_{ik}^{max} \geq \tau_{ik} \geq \Delta T_{ik}^{min} \\
&t_{jh} - t_{i(k+1)} + M(1 - \delta_{ijkh}) \geq 0 \\
&t_{ik} - t_{j(h+1)} + M\delta_{ijkh} \geq 0 \\
&\delta_{ijkh} \in \{0, 1\}
\end{aligned}
$$

The above model is conservative in requiring that two robots should not simultaneously be in their shared collision zone, and can lead to solutions that are not truly optimal. When a robot has overlapping collision zones with more than one robot, we subdivide its overlapping collision segments into several subsegments. The relevant pairs of subdivided collision zones are used to generate collision avoidance constraints as before. This increases the number of binary variables in the formulation.

The coordination of multiple robots under the instantaneous model can be viewed as a *job shop scheduling problem* (JSP). Each *job*, composed of several operations, is a robot's motion along its path. Each *operation* is the motion along a segment. Each *machine* is a collision zone or a collision-free zone. The job shop scheduling problem is NP-hard ([31]), and by reduction, the instantaneous model for robot coordination is NP-hard.

5 Continuous Velocity Model

We now consider generating a schedule with continuous velocity profiles for the robots that is consistent with their maximum velocity and acceleration bounds. To find the minimum and maximum times taken by a robot to traverse a segment, we solve two TPBVPs (two-point boundary value problems) over the segment. We illustrate this procedure using the *double integrator* model from classical optimal control ([8]).

5.1 Single Robot on a Segment

A single robot moving along a path segment can be modeled as a double integrator ([8]) with inequality constraints on the control input (acceleration) and the velocity state variable. Let $x(t)$, $v(t) = \frac{dx(t)}{dt}$, and $a(t) = \frac{dv(t)}{dt}$ be the position, velocity, and acceleration of the robot at time t, S be the length of the segment, and ΔT be the time taken to traverse the segment. Computing the minimum and maximum time taken by the robot to traverse the segment, subject to constraints on its velocities v_{start} and v_{end} at the segment endpoints and inequality constraints on its velocity and acceleration, can be solved as a TPBVP. The double integrator system can be written as:

$$\text{Minimize or Maximize} \quad \Delta T = \int_0^{\Delta T} 1 dt$$

$$\begin{aligned} &\text{subject to:} \\ &\begin{pmatrix} \dot{x} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ v \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} a(t) \\ &x(0) = 0 \quad x(\Delta T) = S \\ &v(0) = v_{start} \quad v(\Delta T) = v_{end} \\ &0 \leq v \leq v_{max} \\ -a_{max} &\leq a \leq a_{max} \end{aligned}$$

5.2 Minimum and Maximum Time Control for a Single Robot over a Segment

The minimum time control of the double integrator model is well known ([8]). We have extended this to obtain the maximum time control using the restricted maximum principle ([20]). Basically these are TPBVPs, and the solutions have a bang-bang or bang-off-bang control structure. The minimum ΔT and maximum ΔT each have two different cases, depending on whether S is sufficiently long for the robot to reach v_{max} (zero) for the minimum (maximum) time case. Note that if $\frac{|v_{end}^2 - v_{start}^2|}{2a_{max}} > S$, there is no feasible velocity profile since the distance is too short.

1. Minimum ΔT (Figure 2):
   (a) If $S \geq \frac{1}{2}(\frac{(v_{max}^2 - v_{start}^2)}{a_{max}} + \frac{(v_{max}^2 - v_{end}^2)}{a_{max}})$,
   $\Delta T^{min} = \frac{S}{v_{max}} - \frac{((v_{max}^2 - v_{start}^2)+(v_{max}^2 - v_{end}^2))}{2a_{max} \cdot v_{max}} + \frac{v_{max} - v_{start}}{a_{max}} + \frac{v_{max} - v_{end}}{a_{max}}$
   (b) If $\frac{1}{2}(\frac{(v_{max}^2 - v_{start}^2)}{a_{max}} + \frac{(v_{max}^2 - v_{end}^2)}{a_{max}}) > S \geq \frac{1}{2}\frac{|v_{end}^2 - v_{start}^2|}{a_{max}}$,
   $\Delta T^{min} = \frac{(v_{middle} - v_{start})}{a_{max}} + \frac{(v_{middle} - v_{end})}{a_{max}}$
   where $v_{middle} = \frac{1}{2}(2v_{start}^2 + 2v_{end}^2 + 4Sa_{max})^{\frac{1}{2}}$
2. Maximum ΔT (Figure 3):
   (a) If $S \geq \frac{1}{2}\frac{(v_{start}^2 + v_{end}^2)}{a_{max}}$, $\Delta T^{max} = \infty$.
   (b) If $\frac{1}{2}\frac{(v_{start}^2 + v_{end}^2)}{a_{max}} > S \geq \frac{1}{2}\frac{|(v_{end}^2 - v_{start}^2)|}{a_{max}}$,
   $\Delta T^{max} = \frac{(v_{start} - v_{middle})}{a_{max}} + \frac{(v_{end} - v_{middle})}{a_{max}}$
   where $v_{middle} = \frac{1}{2}(2v_{start}^2 + 2v_{end}^2 - 4Sa_{max})^{\frac{1}{2}}$

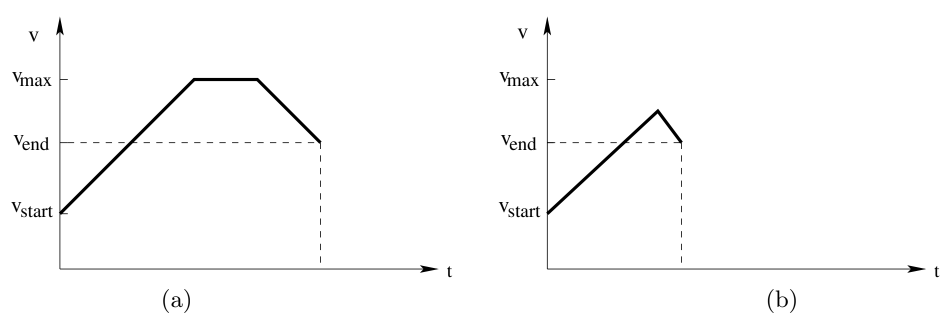


Fig. 2. Minimum ΔT. Case (a): Velocity reaches v_{max}. Case (b): Velocity cannot reach v_{max}.

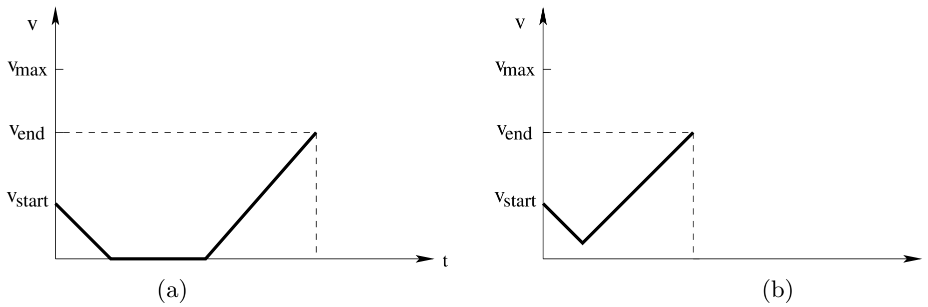


Fig. 3. Maximum ΔT. Case (a): Velocity can decrease to zero. Case (b): Velocity cannot decrease to zero.

5.3 Continuous Velocity MINLP Formulation

Since the robot velocities are variables in the minimum and maximum time control for a robot over a segment, they introduce nonlinear constraints. We therefore formulate a mixed integer nonlinear programming problem (MINLP) for the multiple robot coordination problem. We have the usual completion time constraints and collision avoidance constraints. The traversal time constraints are more complicated since the segment endpoint velocities are variables. Let $a_{i,max}$ be the maximum acceleration of robot $\mathcal{A}_i$ and let v_{ik} represent the velocity of $\mathcal{A}_i$ at the start of segment k. Let ΔT_{ik}^{min} and ΔT_{ik}^{max} be the minimum and maximum traversal times for $\mathcal{A}_i$ along the segment k. Let $\Delta T_{ik,1}^{min}(\Delta T_{ik,1}^{max})$ and $\Delta T_{ik,2}^{min}(\Delta T_{ik,2}^{max})$ represent the two possible minimum (maximum) traversal time values (as described in Section 5.2). The binary variables $y_{ik,1}$ and $y_{ik,2}$ ($z_{ik,1}$ and $z_{ik,2}$) depend on whether or not the values of v_{ik} and $v_{i(k+1)}$ permit the robot to reach v_{max} (zero) in S_{ik}, and are used to select the feasible value of $\Delta T_{ik}^{min}(\Delta T_{ik}^{max})$. The MINLP formulation for the optimal continuous velocity schedule is:

$$
\begin{aligned}
&\text{Minimize } C_{max}\\
&\text{subject to:}\\
&C_{max} \geq t_{i,last} + \tau_{i,last} \qquad \text{for } i = 1, \ldots, n\\
&t_{i(k+1)} - t_{ik} = \tau_{ik}\\
&t_{jh} - t_{i(k+1)} + M(1 - \delta_{ijkh}) \geq 0\\
&t_{ik} - t_{j(h+1)} + M\delta_{ijkh} \geq 0\\
&S_{ik} \geq \frac{(v_{i(k+1)}^2 - v_{ik}^2)}{2a_{max}} \geq -S_{ik}\\
&(S_{ik} - \frac{(v_{i,max}^2 - v_{ik}^2) + (v_{i,max}^2 - v_{i(k+1)}^2)}{2a_{i,max}}) - My_{ik,1} \leq 0\\
&(S_{ik} - \frac{(v_{i,max}^2 - v_{ik}^2) + (v_{i,max}^2 - v_{i(k+1)}^2)}{2a_{i,max}}) + My_{ik,2} \geq 0
\end{aligned}
$$

$$
\begin{aligned}
&(S_{ik} - \frac{v_{ik}^2 + v_{i(k+1)}^2}{2a_{i,max}}) - Mz_{ik,1} \le 0 \\
&(S_{ik} - \frac{v_{ik}^2 + v_{i(k+1)}^2}{2a_{i,max}}) + Mz_{ik,2} \ge 0 \\
&\Delta T_{ik,1}^{min} = \frac{S_{ik}}{v_{i,max}} - \frac{(v_{i,max}^2 - v_{ik}^2 + v_{i,max}^2 - v_{i(k+1)}^2)}{2a_{i,max}v_{i,max}} \\
&\qquad\qquad + \frac{v_{i,max} - v_{ik}}{a_{i,max}} + \frac{v_{i,max} - v_{i(k+1)}}{a_{i,max}} \\
&\Delta T_{ik,2}^{min} = \frac{(v_{middle,ik}^{min} - v_{ik})}{a_{i,max}} + \frac{(v_{middle,ik}^{min} - v_{i(k+1)})}{a_{i,max}} \\
&(v_{middle,ik}^{min})^2 = \frac{1}{4}(2v_{ik}^2 + 2v_{i(k+1)}^2 + 4S_{ik}a_{i,max}) \\
&\Delta T_{ik,1}^{max} = \infty \\
&\Delta T_{ik,2}^{max} = \frac{(v_{ik} - v_{middle,ik}^{max})}{a_{i,max}} + \frac{(v_{i(k+1)} - v_{middle,ik}^{max})}{a_{i,max}} \\
&(v_{middle,ik}^{max})^2 = \frac{1}{4}(2v_{ik}^2 + 2v_{i(k+1)}^2 - 4S_{ik}a_{i,max}) \\
&\Delta T_{ik}^{min} = y_{ik,1} \cdot \Delta T_{ik,1}^{min} + y_{ik,2} \cdot \Delta T_{ik,2}^{min} \\
&\Delta T_{ik}^{max} = z_{ik,1} \cdot \Delta T_{ik,1}^{max} + z_{ik,2} \cdot \Delta T_{ik,2}^{max} \\
&y_{ik,1} + y_{ik,2} = 1 \qquad y_{ik,1}, y_{ik,2} \in \{0,1\} \\
&z_{ik,1} + z_{ik,2} = 1 \qquad z_{ik,1}, z_{ik,2} \in \{0,1\} \\
&\Delta T_{ik}^{max} \ge \tau_{ik} \ge \Delta T_{ik}^{min} \\
&v_{i,max} \ge v_{ik} \ge 0 \\
&v_{i,initial} = v_{i,goal} = 0 \\
&t_{ik} \ge 0 \\
&\delta_{ijkh} \in \{0,1\}
\end{aligned}
$$

This MINLP problem describing the optimal solution has difficult nonconvex constraints. Existing techniques to solve MINLPs either require convexity or are not guaranteed to find the optimal solution for large problem sizes.

We therefore solve two MILPs, which differ only in their ΔT^{max} values, to obtain good lower and upper bounds on the optimal solution. For simplicity, the velocities at the initial and goal configurations, $v_{initial}$ and v_{goal}, are assumed zero for each robot. Initially also assume that the first segment is sufficiently long for the robot to reach $v_{i,max}$ by its end, and that the last segment is sufficiently long for the robot to decelerate from $v_{i,max}$ to zero.

1. Lower bound problem: A lower bound for the MINLP problem can clearly be obtained by solving the MILP for the instantaneous model, assuming infinite acceleration. We obtain a tighter lower bound by formulating an *improved instantaneous model* that considers the time to accelerate and decelerate over the first and last segments for each robot. Since the segments are sufficiently long for the robot to go from zero to v_{max} (and

vice versa), the minimum traversal times for the first and last segments are $\Delta T^{min} = S/v_{max} + v_{max}/2a_{max}$. Solving the MILP for this improved instantaneous model gives a lower bound for the MINLP problem.

2. Upper bound problem: Here the original MINLP model is transformed into an MILP problem by setting the velocities at the endpoints of each segment (except the initial and goal configurations) to v_{max}, the highest feasible velocity given the segment lengths. Solving the MILP problem for this *setpoint model* (described in the next section) gives a feasible continuous velocity schedule that is an upper bound for the MINLP problem.

Note that under the assumption that the initial and goal configurations are collision-free, a valid solution always exists for both problems since a feasible schedule is for one robot at a time to move along its path, for some arbitrary sequencing of the robots.

6 Setpoint Model

We now describe the *setpoint model* to generate a continuous velocity schedule for a set of robots with maximum velocity $v_{i,max}$ and maximum acceleration $a_{i,max}$. We generate a continuous velocity profile for each robot by making the following additional assumption: *Each robot travels at its maximum velocity at the endpoints of each of its collision segments.* The intuition is that by setting the velocity v_{ik} at the endpoints of the collision zones to be the maximum velocity $v_{i,max}$, the robots are biased to move through their collision zones in the least time. Since any continuous velocity schedule is an upper bound on the optimal continuous velocity schedule, this solution is guaranteed to be an upper bound on the optimal solution. Further, this reduces the MINLP formulation to an MILP formulation.

6.1 MILP Formulation

Since the segment endpoint velocities are $v_{i,max}$ by the setpoint assumption, the minimum and maximum times to traverse a segment of length S_{ik} are:

$$\Delta T_{ik}^{min} = \begin{cases} S_{ik}/v_{i,max} & \text{if } k \text{ an interior segment} \\ S_{ik}/v_{i,max} + v_{i,max}/2a_{i,max} & \text{if the first or last segment} \end{cases}$$

$$\Delta T_{ik}^{max} = \begin{cases} \infty & \text{if } S_{ik} \geq v_{i,max}^2/a_{i,max} \\ \frac{2v_{i,max} - 2(v_{i,max}^2 - a_{i,max}S_{ik})^{\frac{1}{2}}}{a_{i,max}} & \text{if } S_{ik} < v_{i,max}^2/a_{i,max} \end{cases}$$

The MILP formulation for the setpoint model is identical to the MILP formulation for the improved instantaneous model, except for the difference in the ΔT^{max} values. The solution to the setpoint MILP is a continuous velocity schedule that is guaranteed to be an upper bound on the optimal continuous velocity schedule. When the segment traversal time τ_{ik} generated by the MILP does not correspond to either minimum time or maximum time trajectories over the segment, we generate a velocity profile, based on a case analysis for the double integrator, so the robot can traverse the segment in the given amount of time subject to its velocity and acceleration constraints. Note that the setpoint model is also NP-hard.

6.2 Relaxing Segment Length Constraints

We now relax the requirement that the first and last segments are sufficiently long for the robot velocity to reach v_{max} and zero respectively, by assuming that each path is long enough for the robot to reach its highest speed v_{max} at some point along the path. We then set the velocity at each segment endpoint to be the highest velocity consistent with the initial and goal velocities.

The velocity v_{ik} at the start of the kth segment, assuming robot $\mathcal{A}_i$ begins with an initial velocity of zero and ends with a goal velocity of zero, is:

$$v_{ik} = \begin{cases} \sqrt{2a_{i,max}\sum_{j=1}^{k-1} S_{ij}} & \text{if } \sum_{j=1}^{k-1} S_{ij} \leq \frac{v_{i,max}^2}{2a_{i,max}} \\ \sqrt{2a_{i,max}\sum_{j\geq k} S_{ij}} & \text{if } \sum_{j\geq k} S_{ij} \leq \frac{v_{i,max}^2}{2a_{i,max}} \\ v_{i,max} & \text{otherwise.} \end{cases}$$

Note that the velocity v_{ik} at the start of each segment is uniquely determined since the path is sufficiently long for the robot to reach its highest speed $v_{i,max}$ at some point along the path.

The corresponding bounds on the segment traversal times ΔT_{ik}^{min} and ΔT_{ik}^{max} for the setpoint model are computed from $\Delta T^{min}(v_{ik}, v_{i(k+1)})$ and $\Delta T^{max}(v_{ik}, v_{i(k+1)})$ as described in Section 5.2.

We correspondingly update the improved instantaneous model so that the velocity v_{ik} at each segment endpoint is the highest velocity physically possible, which is the corresponding setpoint velocity. For the improved instantaneous model, ΔT_{ik}^{min} takes the same values as in the setpoint model, while $\Delta T_{ik}^{max} = \infty$ since the robot can pause its motion instantaneously.

7 General Robot Models

We can apply the approach for coordinating multiple robots in this paper to general robots with linear or nonlinear dynamics if we have analytical or numerical solutions for the extremal TPBVPs for each robot over a segment. Additionally, we should be able to compute the collision zones for the robots and express the collision avoidance constraints in the MILP formulation. We briefly discuss some illustrative cases of interest.

Moving obstacles: We have applied both the instantaneous and setpoint formulations to include multiple moving obstacles with known trajectories.

Each moving obstacle is treated like a robot with a specified trajectory. The collision constraints for each obstacle are computed from its known velocity profile, and are easily added to the MILP formulations.

Car-like robots: Our approach can be applied to car-like nonholonomic robots with continuous curvature paths. Paths that satisfy the nonholonomic constraints ([25]) typically require the robot to stop whenever there is a discontinuity in curvature (to change the steering direction) or there is a cusp point (to reverse the robot direction of motion). Therefore we use continuous curvature paths for a forward moving robot (Scheuer and Fraichard [36]) so that the robot does not have to stop at curvature discontinuities. When the class of continuous curvature paths includes cusp points ([15], [22]), we add the constraint that the robot velocity must be zero at every such point.

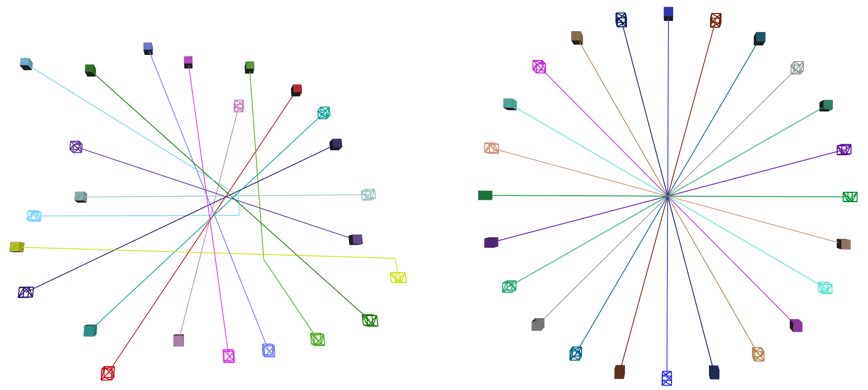
Aircraft: To apply this approach to aircraft, we can use a simplified planar aircraft model ([5]). Since the path is specified, the linear velocity is a variable with magnitude bounds and the acceleration is the control input. As with car-like robots, we require the paths to have continuous curvature.

Articulated robots: To apply this approach to articulated robots, we must compute appropriate collision zones and solve the TPBVPs for the dynamic model of the robots. Due to the difficulty of solving the TPBVPs analytically, numerical methods can be used to approximate the optimal controls. Algorithms for minimum time control along a given path exist ([7],[39],[10]), and we expect they can be modified for maximum time control.

8 Implementation

We have implemented software in C++ to coordinate the motions of polyhedral robots with specified paths (Figure 4). We compute the collision zones using the PQP collision detection package (Larsen et al. [23]) by sampling uniformly along each robot's path. We generate the MILP formulations from the computed collision zones and solve them using the AMPL [13] and CPLEX [18] optimization packages. Since the setpoint formulation with its tighter constraints is usually solved much more quickly than the improved instantaneous formulation, we use the setpoint solution as an upper bound constraint for the improved instantaneous formulation. See Table 1 for running times measured on a Sun Ultra 60. The problem complexity depends primarily on the number of collision zones, and to a lesser extent on the number of robots. For a particularly difficult problem (for example, the symmetric radial case with a bottleneck at the center) or for a sufficiently large number of collision zones, the MILP time will dominate the running time.

In these experiments, the gap between the objective function values returned by the instantaneous and setpoint formulations was zero in all the cases, indicating that an optimal solution had been found. Example animations may be seen at `www.cs.rpi.edu/~sakella/multikino/`.



(a) (b)

Fig. 4. Overhead view of example paths for 12 robots: (a) Typical paths (b) Radial paths with symmetry, with a bottleneck at the center. Goal configurations are indicated by solid cubes.

Num. of robots	Num. of collision zones	Collision time (secs)	Num. of binary variables	MILP-S time (secs)	Num. of binary variables	MILP-I time (secs)
5	13	18.67	20	0.04	14	0
8	42	55.67	64	0.13	62	0.08
10	71	88.26	102	0.53	100	0.17
12	82	115.81	124	0.61	123	0.25
8 (radial, unsymmetric)	32	54.76	54	0.167	54	0.095
12 (radial, unsymmetric)	94	170.15	128	2.2	128	0.49
8 (radial, symmetric)	29	30.53	54	3.87	54	0.095
12 (radial, symmetric)	86	70.53	128	160	128	60.67

Table 1. Sample run times for setpoint formulation (MILP-S) and improved instantaneous formulation (MILP-I). (The improved instantaneous formulation used the setpoint solutions as upper bounds.) Collision checks were performed at 200 points along each path.

9 Conclusion

We have developed an approach to generate continuous velocity profiles for (near) minimum time collision-free coordination of multiple robots with kinodynamic constraints along specified paths. The principal advantage of our MILP formulations is that they permit the collision-free coordination of a large number of robots. The MILP formulations for coordination of multiple robots are NP-hard, and their complexity increases directly with the number of potential collisions. However efficient collision detection software and integer programming solvers make this approach practical for reasonable problem sizes.

There are several directions for future work. Analytically characterizing the gap between the improved instantaneous model and setpoint model would indicate the optimality of the generated solutions. Developing polynomial time approximation or randomization algorithms to schedule robots can be useful in generating bounds for the MILP formulations. Developing computational procedures to solve the continuous velocity MINLP formulation is important. Demonstrating the approach on robots with more complex dynamics, such as car-like mobile robots, aircraft, and articulated robots, is an interesting next step since the complex constraints on the state and control variables make solving the TPBVPs for individual segments challenging. Automatic modification of robot paths to reduce completion time would be a useful extension. Another interesting direction is online coordination of multiple robots using sensor-based estimates of robot positions and velocities.

Acknowledgments

This work was supported in part by RPI and by NSF under CAREER Award No. IIS-0093233. Andrew Andkjar implemented animation software that interfaced with the PQP software from the University of North Carolina and helped generate examples. Thanks to Seth Hutchinson for early discussions, to Prasad Akella for suggesting the problem, and John Mitchell for advice. Thanks also to the reviewers for helpful suggestions.

References

1. S. Akella and S. Hutchinson. Coordinating the motions of multiple robots with specified trajectories. In *IEEE International Conference on Robotics and Automation*, pages 624–631, Washington, DC, May 2002.
2. R. Alami, S. Fleury, M. Herrb, F. Ingrand, and F. Robert. Multi-robot cooperation in the MARTHA project. *IEEE Robotics and Automation Magazine*, 5(1):36–47, Mar. 1998.
3. B. Aronov, M. de Berg, A. F. van der Stappen, P. Svestka, and J. Vleugels. Motion planning for multiple robots. *Discrete and Computational Geometry*, 22:505–525, 1999.
4. J. Barraquand, B. Langlois, and J.-C. Latombe. Numerical potential field techniques for robot path planning. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(2):224–241, 1992.
5. A. Bicchi and L. Pallottino. On optimal cooperative conflict resolution for air traffic management systems. *IEEE Transaction on Intelligent Transportation Systems*, 1(4):221–231, Dec. 2000.
6. Z. Bien and J. Lee. A minimum time trajectory planning method for two robots. *IEEE Transactions on Robotics and Automation*, 8(3):414–418, June 1992.

7. J. E. Bobrow, S. Dubowsky, and J. S. Gibson. Time-optimal control of robotic manipulators along specified paths. *International Journal of Robotics Research*, 4(3):3–17, Fall 1985.
8. A. E. Bryson and Y.-C. Ho. *Applied Optimal Control*. Hemisphere, Washington D.C., 1975.
9. C. Chang, M. J. Chung, and B. H. Lee. Collision avoidance of two robot manipulators by minimum delay time. *IEEE Transactions on Systems, Man, and Cybernetics*, 24(3):517–522, Mar. 1994.
10. Y. Chen and A. A. Desrochers. A proof of the structure of the minimum-time control law of robotic manipulators using a Hamiltonian formulation. *IEEE Transactions on Robotics and Automation*, 6(3):388–393, June 1990.
11. B. R. Donald, P. Xavier, J. Canny, and J. Reif. Kinodynamic motion planning. *Journal of the ACM*, 40(5):1048–1066, Nov. 1993.
12. M. Erdmann and T. Lozano-Perez. On multiple moving objects. *Algorithmica*, 2(4):477–521, 1987.
13. R. Fourer, D. M. Gay, and B. W. Kernighan. *AMPL: A Modeling Language for Mathematical Programming*. Duxbury Press, 1993.
14. T. Fraichard. Trajectory planning in dynamic workspace: a 'state-time space' approach. *Advanced Robotics*, 13(1):75–94, 1999.
15. T. Fraichard, A. Scheuer, and R. Desvigne. From Reeds and Shepp's to continuous-curvature paths. In *Proceedings of the International Conference on Advanced Robotics*, pages 585–590, Tokyo, Japan, Oct. 1999.
16. J. E. Hopcroft, J. T. Schwartz, and M. Sharir. On the complexity of motion planning for multiple independent objects: PSPACE-hardness of the "warehouseman's problem". *International Journal of Robotics Research*, 3(4):76–88, 1984.
17. D. Hsu, R. Kindel, J.-C. Latombe, and S. Rock. Randomized kinodynamic motion planning with moving obstacles. In B. R. Donald, K. M. Lynch, and D. Rus, editors, *Algorithmic and Computational Robotics: New Directions*, pages 247–264. A. K. Peters, Natick, Massachusetts, 2001.
18. ILOG, Inc., Incline Village, NV. *CPLEX 6.0 Documentation Supplement*, 1998.
19. K. Kant and S. W. Zucker. Toward efficient trajectory planning: The path-velocity decomposition. *International Journal of Robotics Research*, 5(3):72–89, Fall 1986.
20. G. Knowles. *An Introduction to Applied Optimal Control*. Mathematics in Science and Engineering, Volume 159. Academic Press, New York, 1981.
21. J. Kosecka, C. Tomlin, G. Pappas, and S. Sastry. Generation of conflict resolution maneuvers for air traffic management. In *IEEE/RSJ International Conference on Robots and Systems*, pages 1598–1603, Grenoble, France, 1997.
22. F. Lamiraux and J.-P. Laumond. Smooth motion planning for car-like vehicles. *IEEE Transactions on Robotics and Automation*, 17(4):498–502, Aug. 2001.
23. E. Larsen, S. Gottschalk, M. Lin, and D. Manocha. Fast distance queries using rectangular swept sphere volumes. In *IEEE International Conference on Robotics and Automation*, pages 3719–3726, San Francisco, CA, Apr. 2000.
24. J.-C. Latombe. *Robot Motion Planning.* Kluwer Academic Publishers, Norwell, MA, 1991.
25. J.-P. Laumond, editor. *Robot Motion Planning and Control*. Lecture Notes in Control and Information Sciences, No. 229. Springer, 1998.

26. S. M. LaValle and S. A. Hutchinson. Optimal motion planning for multiple robots having independent goals. *IEEE Transactions on Robotics and Automation*, 14(6):912–925, Dec. 1998.
27. S. M. LaValle and J. J. Kuffner. Randomized kinodynamic planning. *International Journal of Robotics Research*, 20(5):378–401, May 2001.
28. G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization.* John Wiley and Sons, New York, 1988.
29. P. A. O'Donnell and T. Lozano-Perez. Deadlock-free and collision-free coordination of two robot manipulators. In *IEEE International Conference on Robotics and Automation*, pages 484–489, Scottsdale, AZ, May 1989.
30. L. Pallottino, E. Feron, and A. Bicchi. Conflict resolution problems for air traffic management systems solved with mixed integer programming. *IEEE Transactions on Intelligent Transportation Systems*, 3(1):3–11, Mar. 2002.
31. M. Pinedo. *Scheduling: Theory, Algorithms, and Systems.* Prentice-Hall, Englewood Cliffs, NJ, 1995.
32. J. Reif and M. Sharir. Motion planning in the presence of moving obstacles. In *Proceedings of the 26th Annual Symposium on the Foundations of Computer Science*, pages 144–154, Portland, Oregon, Oct. 1985.
33. A. A. Rizzi, J. Gowdy, and R. L. Hollis. Distributed coordination in modular precision assembly systems. *International Journal of Robotics Research*, 20(10):819–838, Oct. 2001.
34. G. Sahar and J. M. Hollerbach. Planning of minimum-time trajectories for robot arms. *International Journal of Robotics Research*, 5(3):97–140, Fall 1986.
35. G. Sanchez and J. Latombe. On delaying collision checking in PRM planning — application to multi-robot coordination. *International Journal of Robotics Research*, 21(1):5–26, Jan. 2002.
36. A. Scheuer and T. Fraichard. Continuous-curvature path planning for car-like vehicles. In *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems, volume 2*, pages 997–1003, Grenoble, France, Sept. 1997.
37. T. Schouwenaars, B. D. Moor, E. Feron, and J. How. Mixed integer programming for multi-vehicle path planning. In *European Control Conference 2001*, Porto, Portugal, 2001.
38. Z. Shiller and S. Dubowsky. On computing the global time-optimal motion of robotic manipulators in the presence of obstacles. *IEEE Transactions on Robotics and Automation*, 7(6):785–797, Dec. 1991.
39. K. G. Shin and N. D. McKay. Minimum-time control of robotic manipulators with geometric path constraints. *IEEE Transactions on Automatic Control*, AC-30(6):531–541, June 1985.
40. K. G. Shin and Q. Zheng. Minimum-time collision-free trajectory planning for dual robot systems. *IEEE Transactions on Robotics and Automation*, 8(5):641–644, Oct. 1992.
41. T. Simeon, S. Leroy, and J.-P. Laumond. Path coordination for multiple mobile robots: A resolution-complete algorithm. *IEEE Transactions on Robotics and Automation*, 18(1):42–49, Feb. 2002.
42. C. Tomlin, G. J. Pappas, and S. Sastry. Conflict resolution for air traffic management: A study in multi-agent hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):509–521, Apr. 1998.
43. P. Švestka and M. H. Overmars. Coordinated motion planning for multiple car-like robots using probabilistic roadmaps. In *IEEE International Conference on Robotics and Automation*, pages 1631–1636, Nagoya, Japan, May 1995.