Jufeng Peng
Department of Mathematical Sciences
Rensselaer Polytechnic Institute
Troy, New York 12180, USA
pengj@rpi.edu

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

Coordinating Multiple Robots with Kinodynamic Constraints Along Specified Paths

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; these velocity profiles satisfy the dynamics constraints, avoid collisions, and minimize 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, which provide schedules that give lower and upper bounds on the optimum; the upper bound schedule is designed to provide continuous velocity trajectories that are feasible. The approach is illustrated with coordination of multiple robots, modeled as double integrators subject to velocity and acceleration constraints. An application to coordination of nonholonomic car-like robots is described, along with implementation results for 12 robots.

KEY WORDS—multiple robots, collision-free coordination, dynamics, mixed integer program

1. Introduction

Coordinating multiple robots with kinodynamic constraints (i.e., simultaneous kinematic and dynamics constraints) in a shared workspace without collisions has applications in manufacturing cells (Rizzi, Gowdy, and Hollis 2001), automated guided vehicle (AGV) coordination in harbors and airports (Alami et al. 1998), and air traffic control (Bicchi and Pallottino 2000). The general problem requires finding a trajectory (path and velocity profile) for each robot such that the specified objective, such as the task completion time, total time, or energy consumption, of the system is minimized. Optimization of the robot motions is especially important when the task is executed repeatedly or resources must be conserved.

This paper deals with the optimal coordination of multiple robots moving with kinodynamic constraints along specified paths. While previous work in robotics mostly addressed either the collision-free path coordination problem of several robots without considering dynamics constraints (O'Donnell and Lozano-Perez 1989; LaValle and Hutchinson 1998; Simeon, Leroy, and Laumond 2002), or the search for time-optimal motions for a single robot (Bobrow, Dubowsky, and Gibson 1985; Shin and McKay 1985), the contribution of this paper is an approach to generate continuous velocity profiles that satisfy the dynamics constraints, avoid collisions between robots, and minimize the task completion time. An example application is the coordination of AGVs along fixed paths in harbors and airports. The robot motions must satisfy kinematic constraints, such as avoiding collisions with other robots and with moving obstacles, and dynamics constraints, such as velocity and acceleration bounds. Our basic approach is to simultaneously tackle the problem of generating individual robot trajectories that satisfy the dynamics constraints, and the problem of generating optimal coordination schedules that satisfy the collision avoidance constraints. By identifying the collision segments along a robot's path, we combine the disjunctive 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

The International Journal of Robotics Research
Vol. 24, No. 4, April 2005, pp. 295-310,
DOI: 10.1177/0278364905051974

mixed integer linear programming (MILP) formulations. The "improved instantaneous model" provides a lower bound on the optimal solution, and the "setpoint model" provides a continuous velocity schedule that is both feasible and an upper bound on the optimal continuous velocity trajectories. In this paper, we illustrate the approach using robots modeled as double integrators subject to velocity and acceleration constraints, and we discuss an application to a system of car-like robots. Portions of this work were previously presented in Peng and Akella (2003a, 2003b).

The approach described here represents a step towards solving the challenging problem of coordinating multiple robots without specified paths. This approach can be combined with probabilistic techniques, which can generate paths (Švestka and Overmars 1998; Sanchez and Latombe 2002) or trajectories (LaValle and Kuffner 2001; Hsu et al. 2001) for the set of robots, to then optimize robot motions along those paths subject to dynamics constraints.

2. Related Work

There are two main bodies of related work in robotics, which partially overlap. One focuses on the coordination of multiple robots, typically without considering robot dynamics. The other focuses on trajectory optimization for a single robot while considering robot dynamics. Additionally, there has been recent work in coordinating air vehicles with simplified dynamics models.

2.1. 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 (Latombe 1991). This problem is highly underconstrained, and Hopcroft, Schwartz, and Sharir (1984) have shown 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. 1999) or using probabilistic approaches. A potential field randomized path planner was applied to multiple robot planning (Barraquand, Langlois, and Latombe 1992), and probabilistic roadmap planners have been developed for coordinating multiple car-like robots (Švestka and Overmars 1998) and multiple manipulators (Sanchez and Latombe 2002). However, these do not consider robot dynamics.

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 (Kant and Zucker 1986; Reif and Sharir 1994). Erdmann and Lozano-Perez (1987) 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, with previous robots treated as moving obstacles. Buckley (1989) presented a fast motion planner for multiple translating robots in the plane that prioritizes robots based on whether they can travel in a straight line to the goal. Parsons and Canny (1990) describe a cell decomposition based path planning algorithm for coordinating translating robots in the plane. Fujimura and Samet (1989) perform path planning for a robot in the presence of moving obstacles using the configuration space–time. They assume a translating robot with velocity and acceleration bounds, and use a hierarchical octree representation of the space. Fiorini and Shiller (1993, 1998) have developed an approach that computes velocity obstacles to perform trajectory planning of a single robot among multiple moving obstacles with known linear trajectories. They have also considered optimization of the generated trajectories (Fiorini and Shiller 1996). Shiller, Large, and Sekhavat (2001) have generalized the basic approach to deal with obstacles moving along arbitrary trajectories.

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 (1989) developed a coordination diagram representation for path coordination of two robots. LaValle and Hutchinson (1998) addressed a similar problem where each robot was constrained to a specified configuration space roadmap. Ghrist and Koditschek (2002) designed controllers for coordination of AGVs constrained to motion on graphs, based on an analysis of the configuration space of two robots on a Y-graph. Simeon, Leroy, and Laumond (2002) performed path coordination for a very large number of car-like robots, in part by exploiting the cylindrical structure of the coordination diagram and in part by partitioning robots with shared collision zones into smaller sets. Trajectory coordination is a closely related problem where the trajectory (path and velocity) of each robot is specified. Akella and Hutchinson (2002) recently developed an MILP formulation for the trajectory coordination of large numbers of robots by changing robot start times. Our work here extends these problem classes by additionally considering dynamics constraints and generating continuous velocity profiles.

2.2. Time-optimal Trajectory Planning

There is a large body of work on the time-optimal control of a single manipulator, going back to the early work of Kahn and Roth (1971). Bobrow, Dubowsky, and Gibson (1985) and Shin and McKay (1985) developed algorithms to generate the time-optimal velocity profile of a manipulator moving along a specified path. Subsequently, Pfeiffer and Johanni (1987), Slotine and Yang (1989), and Shiller and Lu (1992) refined these algorithms. Lamiraux and Laumond (1998) extended these methods to generate velocity profiles for a car-like robot with constraints on the robot velocity magnitude.

Trajectory planning directly in the $2n$-dimensional state space that considers both kinematic and dynamics constraints is called “kinodynamic planning”. Sahar and Hollerbach (1986), and later Shiller and Dubowsky (1991) 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. O’Dunlaing (1987) presented a polynomial-time algorithm for planning the motion of a particle moving in one dimension while subject to bounded acceleration constraints. Canny, Rege, and Reif (1991) developed an exact exponential-time algorithm for the time-optimal motion of a point robot, with velocity and acceleration bounds, in two dimensions. Donald et al. (1993) developed a polynomial-time approximation algorithm to generate near time-optimal trajectories that satisfy kinematic and dynamic constraints for a single point mass robot. Heinzinger et al. (1990) developed an approximation algorithm for time-optimal trajectory planning of an open-chain manipulator, using graph search in the discretized state space. Donald and Xavier (1995a) presented an improved algorithm for robots with decoupled dynamics bounds, and extended this work to robots with coupled dynamics bounds such as open-chain manipulators (Donald and Xavier 1995b). Reif and Wang (1997) developed approximation algorithms that use nonuniform grid decompositions for kinodynamic planning. Fraichard (1999) described 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 (RRTs) (Lavalle and Kuffner 2001) and probabilistic roadmaps (Hsu et al. 2001). These randomization approaches are capable of generating collision-free trajectories for multiple robots. For example, Frazzoli (2003) has recently applied randomization techniques to trajectory planning and coordination of a small number of spacecraft. However, these do not explicitly provide a method to optimize the coordination of the robots, a gap which the present work addresses.

2.3. Multiple Robot Coordination with Dynamics

Previous work on coordination of robots with dynamics has focused almost exclusively on dual robot systems (Shin and Bien 1989; Chang, Chung, and Bien 1990; Bien and Lee 1992; Chang, Chung, and Lee 1994). Lee and Lee (1987) considered the effects of delays and velocity changes on motion time. Freund and Hoyer (1988) implemented a hierarchical control scheme to modify trajectories of multiple robots to avoid collisions, using a specified right-of-way prioritization for the robots. Shin and Zheng (1992) showed that, for a two-robot system, generating time-optimal trajectories for each robot independently and then delaying the start time of one of the robots leads to a minimal finish time under certain assumptions.

The RRT approach (LaValle and Kuffner 2001) is capable of generating collision-free trajectories for multiple robots. However, it does not explicitly provide a method to optimize the coordination of the robots. Zefran, Desai, and Kumar (1997) consider the planning and control of multiple cooperating manipulators. In recent work, Pledgie et al. (2002) perform trajectory planning and control of groups of unmanned vehicles that are differentially flat systems. Hao et al. (2003) present a framework for planning and control of formations of three unmanned ground vehicles.

2.4. Air Traffic Control

Conflict resolution among multiple aircraft in a shared airspace is closely related to multiple robot coordination. Tomlin, Pappas, and Sastry (1998) synthesized provably safe conflict resolution maneuvers for two aircraft using speed and heading changes. Kosecka et al. (1997) used potential field planners to generate conflict resolution maneuvers. Bicchi and Pallottino (2000) modeled aircraft with constant velocity and curvature bounds and generated minimum total time collision-free paths using given waypoints for three aircraft. Frazzoli et al. (2001) formulated the planar multi-aircraft conflict resolution problem as a nonconvex quadratic program with quadratic constraints. They used semidefinite programming to find the lower bound on the optimum and randomization to find a feasible solution. Richards et al. (2001) used an MILP formulation to plan fuel-optimal paths for multiple spacecraft that avoid plume impingement from thrusters. They used a discretized model of the spacecraft dynamics. Schouwenaars et al. (2001) 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 (2002) generated conflict-free paths to minimize the total flight time for 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, the goal is to find the control inputs along the specified paths so that the dynamics constraints of the robots are satisfied, their motions are collision-free, and the completion time of the set of robots is minimized. 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 forward along its path without retracing its path.

3.1. Paths and Collision Zones

Each robot $\mathcal{A}_i$ is given a path γ_i, which is a continuous mapping $[0, 1] \rightarrow \mathcal{C}_i^{free}$. 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 $\psi(t) : [0, \infty) \rightarrow \mathcal{S}$ 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 one-to-one mapping between s and the path length.

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., $\text{Int}\mathcal{A}_i(\gamma_i(s_i)) \cap \text{Int}\mathcal{A}_j(\gamma_j(s_j)) \neq \emptyset$). 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 $\text{Int}\mathcal{A}_i(\gamma_i(s_i)) \cap \text{Int}\mathcal{A}_j(\gamma_j(s_j)) \neq \emptyset$.

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 $\text{Int}\mathcal{A}_i(\gamma_i(s_i)) \cap \text{Int}\mathcal{A}_j(\gamma_j(s_j)) \neq \emptyset$, and $\forall s_j \in [s_j^{start}, s_j^{end}], \exists s_i \in [s_i^{start}, s_i^{end}]$ such that $\text{Int}\mathcal{A}_i(\gamma_i(s_i)) \cap \text{Int}\mathcal{A}_j(\gamma_j(s_j)) \neq \emptyset$. 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.

We compute collision segments for pairs of robots, and then subdivide any overlapping collision segments so that each subdivided segment corresponds to a set of potentially colliding robots over its entire length. We also ensure that the order in which the robots traverse the subsegments does not change over the original (undivided) segments, since changing the order will result in collisions. For example, if robot $\mathcal{A}_2$ goes before $\mathcal{A}_3$ in the first subsegment, then $\mathcal{A}_2$ must go before $\mathcal{A}_3$ in all subsegments that make up their undivided collision zone.

3.2. Optimal Control Problem For A Single Robot

For a single robot $\mathcal{A}$ moving along a path segment, let $\mathbf{q}$ represent the configuration, let $\mathbf{x}(t)$ represent the state, let $\mathbf{u}(t)$ be the control, let γ be the path of the robot in configuration space, let $J(\mathbf{x}, \mathbf{u})$ be the objective function, and let $\mathbf{c}(\mathbf{x}, \mathbf{u})$ represent the inequality constraints on the state variables and controls. To determine when to speed up and slow down each robot as it moves along its specified path, we must first compute their fastest and slowest possible motions along each path segment. Thus, the optimal control problem, to compute the minimum and maximum times taken by the robot to traverse the segment subject to its dynamics and path constraints, can be written as

$$\begin{aligned} &\text{Minimize} \quad J(\mathbf{x}, \mathbf{u}) \\ &\text{subject to:} \\ &\qquad \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \\ &\qquad \mathbf{c}(\mathbf{x}, \mathbf{u}) \leq 0 \\ &\qquad \mathbf{x}(0) = \mathbf{x}_{start} \\ &\qquad \mathbf{x}(\Delta T) = \mathbf{x}_{end} \\ &\qquad \mathbf{q} \in \gamma. \end{aligned} \tag{1}$$

The minimum time control problem has $J(\mathbf{x}, \mathbf{u}) = \int_0^{\Delta T} 1\,\mathrm{d}t = \Delta T$, and the maximum time control problem has $J(\mathbf{x}, \mathbf{u}) = -\Delta T$ where ΔT 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 two-point boundary value problems (TPBVPs) for each segment.

3.3. Coordination of Multiple Robots

Now we 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 an MINLP problem, with each robot satisfying the traversal time and collision avoidance constraints over each of its segments. Since this MINLP problem with non-convex 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 MILP problems. We illustrate this approach using the double integrator model from optimal control (Bryson and Ho 1975).

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 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 let 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 times for $\mathcal{A}_i$ between the start point of segment k and the start point of segment $k + 1$,

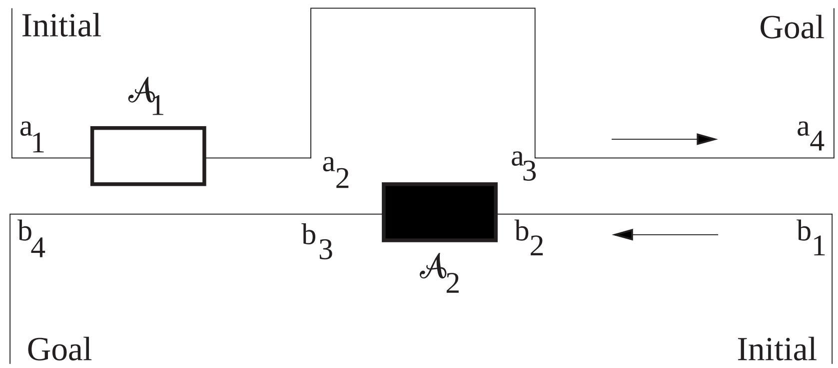

Fig. 1. Coordination example with two translating robots having two collision zones.

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 (Nemhauser and Wolsey 1988) 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 traversal time, completion time, and collision avoidance constraints for all robots are combined to form the instantaneous MILP formulation:

$$\begin{aligned} &\text{Minimize } C_{max} \\ &\text{subject to:} \\ &C_{max} \geq t_{i,last} + \tau_{i,last} \qquad \text{for } i = 1, \ldots, n \\ &t_{ik} \geq 0 \\ &t_{i(k+1)} = t_{ik} + \tau_{ik} \\ &\Delta T_{ik}^{max} \geq \tau_{ik} \geq \Delta T_{ik}^{min} \\ &t_{jh} - t_{i(k+1)} + M(1-\delta_{ijkh}) \geq 0 \\ &t_{ik} - t_{j(h+1)} + M\delta_{ijkh} \geq 0 \\ &\delta_{ijkh} \in \{0, 1\}. \end{aligned} \tag{2}$$

A solution to this MILP provides the completion time C_{max}, the start times t_{ik} and traversal times τ_{ik} for each robot along each segment, and the precedence order of the robots through the collision zones, specified by the δ_{ijkh} variables. The collision avoidance constraints in the above model are 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.

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 JSP is NP-hard (Pinedo 1995), 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, which is consistent with their dynamics constraints. Since we are trying to determine when to speed up and slow down each robot as it moves along its specified path, we first compute their fastest and slowest possible motions along each path segment. To find the minimum and maximum times taken by a robot to traverse a segment, we solve two TPBVPs over the segment. We illustrate this procedure using the "double integrator" model from classical optimal control (Bryson and Ho 1975) for robots with maximum velocity and acceleration bounds.

5.1. Single Robot on a Segment

A single robot moving along a path segment can be modeled as a double integrator with inequality constraints on the control input (acceleration) and the velocity state variable. Let $x(t)$, $v(t) = \mathrm{d}x(t)/\mathrm{d}t$, and $a(t) = \mathrm{d}v(t)/\mathrm{d}t$ be the position, velocity, and acceleration of the robot at time t, let S be the length of the segment, and let ΔT be the time taken to traverse the segment. Computing the minimum and maximum times 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

$$\text{Minimize or Maximize } \Delta T = \int_0^{\Delta T} 1 \mathrm{d}t$$

$$\begin{aligned} &\text{subject to:} \\ &\begin{pmatrix} \dot{x} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ v \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} a(t) \\ &x(0) = -S \quad x(\Delta T) = 0 \\ &v(0) = v_{start} \quad v(\Delta T) = v_{end} \\ &0 \le v \le v_{max} \\ &-a_{max} \le a \le a_{max}. \end{aligned} \tag{3}$$

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 (Bryson and Ho 1975). We have extended this to obtain the maximum time control using the restricted maximum principle (Knowles 1981). 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

$$S < \frac{|v_{end}^2 - v_{start}^2|}{2a_{max}},$$

there is no feasible velocity profile since the distance is too short.

1. Minimum ΔT (Figure 2)

   (a) If
   $$S \ge \frac{1}{2}\left(\frac{(v_{max}^2 - v_{start}^2)}{a_{max}} + \frac{(v_{max}^2 - v_{end}^2)}{a_{max}}\right),$$
   the robot velocity can reach v_{max} and
   $$\Delta T^{min} = \frac{S}{v_{max}} - \frac{((v_{max}^2 - v_{start}^2) + (v_{max}^2 - v_{end}^2))}{2a_{max} \cdot v_{max}} + \frac{v_{max} - v_{start}}{a_{max}} + \frac{v_{max} - v_{end}}{a_{max}}.$$

   (b) If
   $$\frac{1}{2}\left(\frac{(v_{max}^2 - v_{start}^2)}{a_{max}} + \frac{(v_{max}^2 - v_{end}^2)}{a_{max}}\right) > S \ge \frac{1}{2}\frac{|v_{end}^2 - v_{start}^2|}{a_{max}},$$
   the robot velocity cannot reach v_{max} and
   $$\Delta T^{min} = \frac{(v_{middle} - v_{start})}{a_{max}} + \frac{(v_{middle} - v_{end})}{a_{max}}$$
   $$\text{where } v_{middle} = \frac{1}{2}(2v_{start}^2 + 2v_{end}^2 + 4Sa_{max})^{1/2}.$$

2. Maximum ΔT (Figure 3)

   (a) If
   $$S \ge \frac{1}{2}\frac{(v_{start}^2 + v_{end}^2)}{a_{max}},$$
   the robot can go to zero velocity and $\Delta T^{max} = \infty$.

   (b) If
   $$\frac{1}{2}\frac{(v_{start}^2 + v_{end}^2)}{a_{max}} > S \ge \frac{1}{2}\frac{|(v_{end}^2 - v_{start}^2)|}{a_{max}},$$
   the robot cannot go to zero velocity and
   $$\Delta T^{max} = \frac{(v_{start} - v_{middle})}{a_{max}} + \frac{(v_{end} - v_{middle})}{a_{max}}$$
   $$\text{where } v_{middle} = \frac{1}{2}(2v_{start}^2 + 2v_{end}^2 - 4Sa_{max})^{1/2}.$$

5.3. Continuous Velocity Mixed Integer Nonlinear Programming Problem 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 MINLP problem for the multiple robot coordination problem, with the robot velocities at the segment endpoints being additional variables to be computed. We have the usual completion time constraints and collision avoidance constraints. The traversal time constraints are more complicated since the minimum and maximum possible traversal times now depend on the segment endpoint velocities, which are variables. Let $a_{i,max}$ be the maximum acceleration, let $v_{i,max}$ be the maximum velocity 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 possible traversal times for $\mathcal{A}_i$ along the segment k. Let $\Delta T_{ik,a}^{min}$ and $\Delta T_{ik,b}^{min}$ represent the two minimum traversal time values (described in Section 5.2). Similarly, let $\Delta T_{ik,a}^{max}$ and $\Delta T_{ik,b}^{max}$ represent the two maximum traversal time values. The binary variable y_{ik} encodes the disjunctive constraint that either the values of v_{ik} and $v_{i(k+1)}$ permit the robot to reach $v_{i,max}$ in S_{ik} or they do not, and it is used to select the corresponding feasible value of ΔT_{ik}^{min}. Similarly, the binary variable z_{ik}

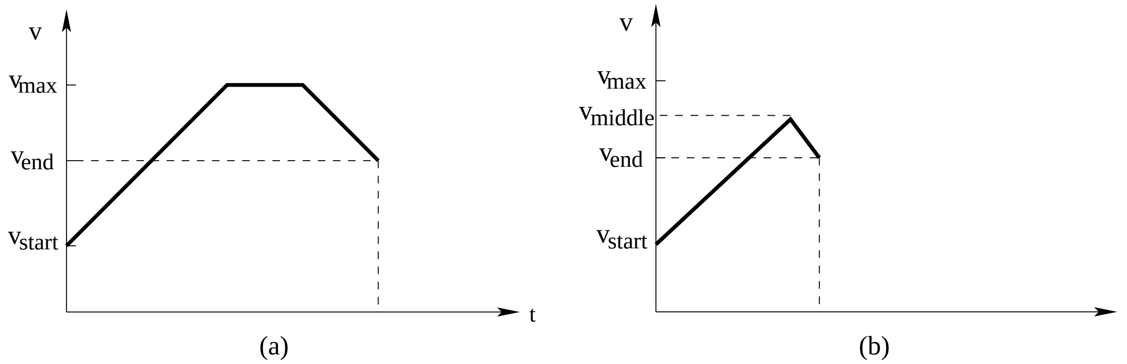

Fig. 2. Minimum ΔT: (a) velocity reaches v_{max}; (b) velocity cannot reach v_{max}. The maximum velocity attained is v_{middle}.

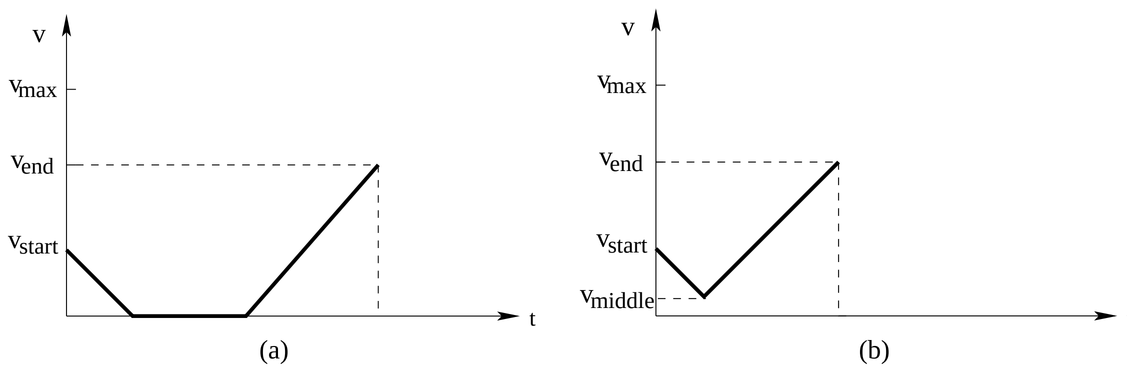

Fig. 3. Maximum ΔT: (a) velocity can decrease to zero; (b) velocity cannot decrease to zero. The minimum velocity attained is v_{middle}.

encodes the disjunctive constraint that either the values of v_{ik} and $v_{i(k+1)}$ permit the robot to reach zero velocity in S_{ik} or they do not, and it is used to select the corresponding feasible value of ΔT_{ik}^{max}. For simplicity, the velocities at the initial and goal configurations, $v_{i,initial}$ and $v_{i,goal}$, are assumed zero for each robot. The resulting MINLP formulation for the optimal continuous velocity schedule can be described by three sets of constraints. The first set of constraints are the standard completion time constraints, traversal time constraints, and collision avoidance constraints:

$$\begin{aligned}
&\text{Minimize } C_{max}\\
&\text{subject to:}\\
&C_{max} \geq t_{i,last} + \tau_{i,last} \qquad \text{for } i = 1, \ldots, n\\
&t_{ik} \geq 0\\
&t_{i(k+1)} = t_{ik} + \tau_{ik}\\
&\Delta T_{ik}^{max} \geq \tau_{ik} \geq \Delta T_{ik}^{min}\\
&t_{jh} - t_{i(k+1)} + M(1 - \delta_{ijkh}) \geq 0\\
&t_{ik} - t_{j(h+1)} + M\delta_{ijkh} \geq 0\\
&\delta_{ijkh} \in \{0, 1\}\\
&v_{i,max} \geq v_{ik} \geq 0\\
&v_{i,initial} = v_{i,goal} = 0.
\end{aligned} \tag{4}$$

The second set of constraints are the nonlinear minimum traversal time constraints:

$$\begin{aligned}
&S_{ik} \geq \frac{(v_{i(k+1)}^2 - v_{ik}^2)}{2a_{i,max}} \geq -S_{ik}\\
&\left(S_{ik} - \frac{(v_{i,max}^2 - v_{ik}^2) + (v_{i,max}^2 - v_{i(k+1)}^2)}{2a_{i,max}}\right) - My_{ik} \leq 0\\
&\left(S_{ik} - \frac{(v_{i,max}^2 - v_{ik}^2) + (v_{i,max}^2 - v_{i(k+1)}^2)}{2a_{i,max}}\right) + M(1 - y_{ik}) \geq 0\\
&\Delta T_{ik,a}^{min} = \frac{S_{ik}}{v_{i,max}} - \frac{(v_{i,max}^2 - v_{ik}^2 + v_{i,max}^2 - v_{i(k+1)}^2)}{2a_{i,max}v_{i,max}}\\
&\qquad + \frac{v_{i,max} - v_{ik}}{a_{i,max}} + \frac{v_{i,max} - v_{i(k+1)}}{a_{i,max}}\\
&\Delta T_{ik,b}^{min} = \frac{(v_{middle,ik}^{min} - v_{ik})}{a_{i,max}} + \frac{(v_{middle,ik}^{min} - v_{i(k+1)})}{a_{i,max}}\\
&(v_{middle,ik}^{min})^2 = \frac{1}{4}(2v_{ik}^2 + 2v_{i(k+1)}^2 + 4S_{ik}a_{i,max})\\
&\Delta T_{ik}^{min} = y_{ik} \cdot \Delta T_{ik,a}^{min} + (1 - y_{ik}) \cdot \Delta T_{ik,b}^{min}\\
&y_{ik} \in \{0, 1\}.
\end{aligned} \tag{5}$$

The third set of constraints are the nonlinear maximum traversal times constraints:

$$\begin{aligned}
&\left(S_{ik} - \frac{v_{ik}^2 + v_{i(k+1)}^2}{2a_{i,max}}\right) - Mz_{ik} \leq 0 \\
&\left(S_{ik} - \frac{v_{ik}^2 + v_{i(k+1)}^2}{2a_{i,max}}\right) + M(1 - z_{ik}) \geq 0 \\
&\Delta T_{ik,a}^{max} = \infty \\
&\Delta T_{ik,b}^{max} = \frac{(v_{ik} - v_{middle,ik}^{max})}{a_{i,max}} + \frac{(v_{i(k+1)} - v_{middle,ik}^{max})}{a_{i,max}} \\
&(v_{middle,ik}^{max})^2 = \frac{1}{4}(2v_{ik}^2 + 2v_{i(k+1)}^2 - 4S_{ik}a_{i,max}) \\
&\Delta T_{ik}^{max} = z_{ik} \cdot \Delta T_{ik,a}^{max} + (1 - z_{ik}) \cdot \Delta T_{ik,b}^{max} \\
&z_{ik} \in \{0, 1\}.
\end{aligned} \tag{6}$$

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. Initially, we 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 assumed sufficiently long for robot $\mathcal{A}_i$ to go from zero to $v_{i,max}$ (and vice versa), the minimum traversal times for the first and last segments are $\Delta T^{min} = S/v_{i,max} + v_{i,max}/2a_{i,max}$. 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 $v_{i,max}$, the highest feasible velocity given the segment lengths. Solving the MILP problem for this setpoint model gives a feasible continuous velocity schedule that is an upper bound for the MINLP problem, as described in the next section.

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 to pick some arbitrary sequencing of the robots, and to have only one robot at a time move along its path.

6. Setpoint Model

We now describe the setpoint model to generate a continuous velocity schedule for a set of robots with dynamics constraints; for double integrators, the dynamics constraints are the maximum velocity $v_{i,max}$ and maximum acceleration $a_{i,max}$ bounds. We first compute the time-optimal velocity profile for each robot over its entire path. We then make the following additional assumption: each robot travels at its maximum feasible velocity at the endpoints of each of its collision segments. The maximum feasible velocity at each segment endpoint is obtained from the corresponding point on the time-optimal velocity profile. The intuition is that by setting the velocity v_{ik} at the endpoints of the collision zones to be the corresponding maximum velocity, 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, setting the endpoint velocities reduces the MINLP formulation to an MILP formulation. The continuous velocity schedule generated by the setpoint model is the solution used to specify the controls for coordinating the robots.

6.1. Mixed Integer Linear Programming Formulation

For double integrators, the maximum feasible velocity is $v_{i,max}$. 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

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

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

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. Like the instantaneous model, the setpoint model is also NP-hard. 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, as described in Section 6.3, so the robot can traverse the segment in the given amount of time subject to its velocity and acceleration constraints.

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 $v_{i,max}$ and zero, respectively. We assume that each path is long enough for the robot to reach its highest speed $v_{i,max}$ at some point along the path. (If the path is not sufficiently long, we compute the highest speed attained by the time-optimal velocity profile along the path and treat that effective highest speed as $v_{i,max}$.) 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

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

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.

6.3. Generating Velocity Profiles

Solving the setpoint formulation gives a feasible coordination schedule, from which we obtain the traversal time τ_{ik} along each segment k for each robot $\mathcal{A}_i$. When this segment traversal time τ_{ik} does not correspond to an extremal trajectory (either a minimum time or maximum time velocity profile) over the segment, we must generate a feasible velocity profile with traversal time τ_{ik}. In general, this is not easy due to the velocity and control inequality constraints. (The robot's velocity profile cannot be generated by solving differential-algebraic equations, since the velocity constraints and accleration constraints that are active at each time instant are not known.)

For the double integrator, we exploit its dynamics to generate a velocity profile consistent with traversal time τ_{ik}. By observing that the nonextremal trajectories have v_{start} and v_{end} equal to v_{max}, we can solve analytically to show that a symmetric bang–off–bang velocity profile is a feasible trajectory (Figure 4). The velocity during the off-phase, $v_{ik,off}$, is obtained by solving the following quadratic equation:

$$\frac{v_{i,max}^2 - v_{ik,off}^2}{a_{i,max}} + v_{ik,off}\left(\tau_{ik} - 2\frac{v_{i,max} - v_{ik,off}}{a_{i,max}}\right) = S_{ik}.$$

Selecting the non-negative solution

$$v_{ik,off} = -\frac{1}{2}\tau_{ik}a_{i,max} + v_{i,max} + \frac{1}{2}\left(a_{i,max}(\tau_{ik}^2 a_{i,max} - 4\tau_{ik}v_{i,max} + 4S_{ik})\right)^{1/2}$$

yields the trajectory profile specified by

$$a_i(t) = \begin{cases} -a_{i,max}, & t_{ik} \le t < t_{ik} + \frac{v_{i,max}-v_{ik,off}}{a_{i,max}} \\ 0, & t_{ik} + \frac{v_{i,max}-v_{ik,off}}{a_{i,max}} \le t < t_{ik} + \tau_{ik} - \frac{v_{i,max}-v_{ik,off}}{a_{i,max}} \\ a_{i,max}, & t_{ik} + \tau_{ik} - \frac{v_{i,max}-v_{ik,off}}{a_{i,max}} \le t < t_{ik} + \tau_{ik}. \end{cases}$$

A few example velocity profiles generated by this algorithm are shown in Figure 5.

6.4. Optimality

While in the general case there can be a gap between the lower bound of the improved instantaneous formulation and the upper bound provided by the setpoint formulation, we have identified two cases for which the gap is zero. That is, here the setpoint formulation provides an optimal coordination schedule.

1. When each robot can collide with at most one other robot. Since each robot can collide with at most one other robot, the multiple robot problem can be reduced to a set of independent problems, each optimizing the motions of a pair of robots. It can be shown (Shin and Zheng 1992) that when the robots have a single shared collision zone, the optimal solution for a pair of robots is to independently generate the time-optimal velocity profile for each robot and then compute the start times for the two robots that minimize the total completion time. Clearly, the setpoint formulation will automatically generate such a solution.

2. When the length of each segment is sufficiently long for each robot to reach zero velocity in the interior of each segment. Here, the improved instantaneous and setpoint formulations are identical since their ΔT^{max} values are identical. When this occurs, the gap is guaranteed to be zero and the setpoint solution provides an optimal feasible velocity schedule.

Note that in the limit as robot maximum acceleration a_{max} approaches infinity, the setpoint model approaches the improved instantaneous model.

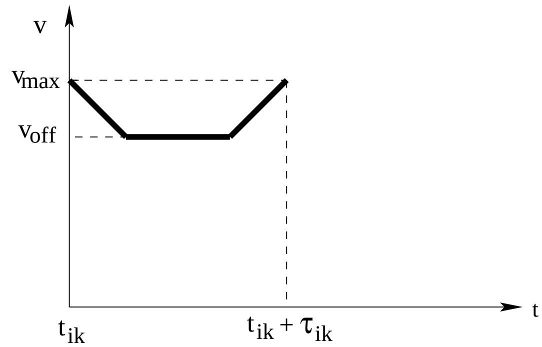

Fig. 4. A symmetric bang–off–bang velocity profile over a segment. Here, v_{start} and v_{end} are equal to v_{max}.

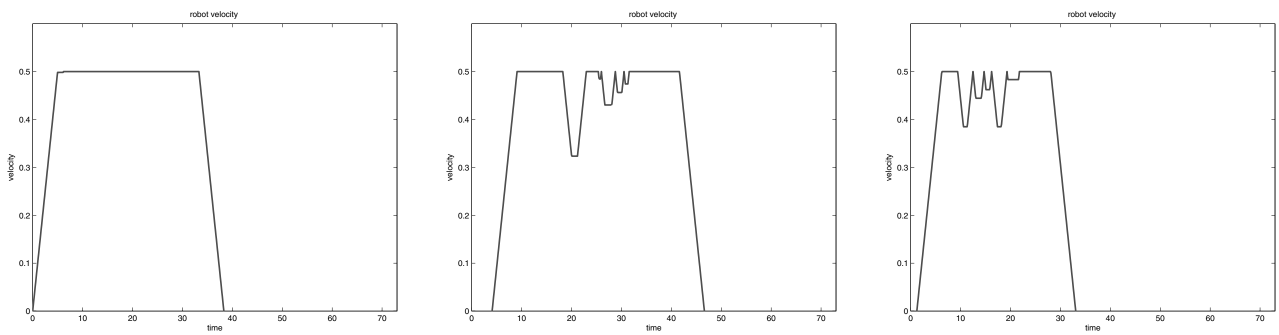

Fig. 5. Example velocity profiles for three robots, indicating their velocities over their entire paths.

7. Coordinating Multiple Car-like Mobile Robots

We now illustrate our coordination approach on nonholonomic car-like robots with dynamics constraints. Paths that satisfy the nonholonomic constraints such as Dubins' paths or Reeds and Shepp's paths (Laumond 1998) typically require the robot to stop when there is a discontinuity in curvature (to change the steering direction) or when there is a cusp point (to reverse the robot motion direction). Therefore, we use simple continuous curvature paths for a forward moving robot (Scheuer and Fraichard 1997).

7.1. Car-like Robot Model

The configuration of a robot is given by (x, y, θ, κ) where (x, y) represents the robot reference point at the mid-point of the rear axle, θ is the robot orientation, and κ is the signed path curvature. Let v be the robot velocity at its reference point.

We model a car-like robot of mass m moving on a horizontal plane with a friction coefficient μ as subject to the following dynamics constraints (Shiller and Chen 1990; Fraichard 1999).

1. Acceleration constraints.

   (a) Acceleration constraints due to the maximum engine force F_{max} and the maximum braking force F_{min} are

   $$\frac{F_{min}}{m} \leq a \leq \frac{F_{max}}{m}.$$

   (b) Sliding friction constraints to prevent the robot slipping off the path are

   $$-\sqrt{\mu^2 g^2 - \kappa^2 v^4} \leq a \leq \sqrt{\mu^2 g^2 - \kappa^2 v^4}.$$

   Thus, the (state-dependent) acceleration constraints are

   $$a \geq \max\left(\frac{F_{min}}{m}, -\sqrt{\mu^2 g^2 - \kappa^2 v^4}\right)$$

   and

   $$a \leq \min\left(\frac{F_{max}}{m}, \sqrt{\mu^2 g^2 - \kappa^2 v^4}\right).$$

2. Velocity constraints.

   (a) Maximum velocity constraints are $0 \leq v \leq v_{max}$. These can be used to enforce maximum speed constraints and to prevent tip-over at high speeds.

(b) Velocity magnitude constraints. To ensure that $\mu^2 g^2 - \kappa^2 v^4 \geq 0$ in the sliding friction constraints, we have the constraint

$$-\sqrt{\frac{\mu g}{|\kappa|}} \leq v \leq \sqrt{\frac{\mu g}{|\kappa|}}.$$

Thus, the (state-dependent) velocity constraints are

$$0 \leq v \leq \min\left(v_{max}, \sqrt{\frac{\mu g}{|\kappa|}}\right).$$

7.2. Paths

The specified paths are chosen to be simple continuous curvature (SCC) paths (Scheuer and Fraichard 1997). SCC paths are composed of straight lines, circular arcs, and clothoid curves. The curvature of a clothoid curve varies linearly with its path length and so the clothoidal segments bridge the straight-line segments and circular arcs. Each path is C^2 continuous, so the path has continuous curvature and no cusps. Since the robot can follow the path without having to stop or reverse direction, we assume the robot moves forward monotonically along its path. The curvature κ of an SCC path is upper bounded by κ_{max}. That is, the steering radius $\rho \geq \rho_{min} = 1/\kappa_{max}$. Additionally, there is an upper bound on the curvature derivative, $\dot{\kappa}$, since a robot must reorient its front wheels with a finite steering velocity.

7.3. Optimal Control Problem For A Single Car-like Robot

Consider a single car-like robot moving along an SCC path segment with x and v representing its position and velocity respectively. The optimal control problem is

$$\text{Min or Max} \quad \Delta T = \int_0^{\Delta T} 1 \mathrm{d}t$$

$$\begin{aligned}
&\text{subject to:}\\
&\begin{pmatrix} \dot{x} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ v \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} a(t)\\
&x(0) = -S \quad x(\Delta T) = 0\\
&v(0) = v_{start} \quad v(\Delta T) = v_{end}\\
&v \geq 0 \qquad v \leq v_{max} \qquad v \leq \sqrt{\frac{\mu g}{|\kappa|}}\\
&a(t) \geq \frac{F_{min}}{m} \qquad a(t) \leq \frac{F_{max}}{m}\\
&a^2(t) \leq \mu^2 g^2 - \kappa^2 v^4.
\end{aligned} \tag{7}$$

This TPBVP is difficult to solve because of the complex constraints on the state and control variables. We assume that $F_{min} = -F_{max}$ and additionally assume $\mu^2 g^2 - \kappa^2 v^4 \geq (F_{max}/m)^2$, which is reasonable for typical values of the variables. This constraint can be expressed as a minimum steering radius constraint $\rho_{min} \geq v_{max}^2/\sqrt{\mu^2 g^2 - (F_{max}/m)^2}$ during path generation. This implies that the maximum robot velocity is v_{max}. This minimum steering radius constraint also makes the upper and lower bounds on the acceleration state independent constants with magnitude F_{max}/m. Therefore, the double integrator formulation of Section 5.1 applies directly to these car-like robots.

7.4. Coordinating Multiple Car-like Robots

Given a set of n car-like robots $\mathcal{A}_1, \ldots, \mathcal{A}_n$ with specified SCC paths that satisfy the above minimum steering radius constraints, we generate collision-free continuous velocity profiles along the specified paths that minimize the completion time using the setpoint MILP formulation described earlier.

Our approach can be extended to car-like nonholonomic robots with piecewise continuous curvature paths. When the class of continuous curvature paths includes cusp points (Fraichard, Scheuer, and Desvigne 1999; Lamiraux and Laumond 2001), we must add the constraint that the robot velocity must be zero at every such point, and compute the velocity while taking into account the direction reversals at cusp points.

8. Implementation

We have implemented software in C++ to coordinate the motions of polyhedral robots with specified paths (Figure 6 and Multimedia Extensions 1–6). We compute the collision zones using the PQP collision detection package (Larsen et al. 2000) by sampling uniformly along each robot's path. The sampling resolution along the robots' paths must be sufficiently fine to avoid missing any part of the collision zones. Since the optimal coordination may cause a pair of robots to just touch at the boundary of their collision zone, we compute collision zones using a specified tolerance.

We generate the MILP formulations from the computed collision zones and solve them using the AMPL (Fourer, Gay, and Kernighan 1993) and CPLEX (ILOG Inc. 1999) 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 MILP 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 our experiments, an optimal solution, indicated by a zero gap between the minimum completion time values computed by the improved instantaneous and setpoint formulations, was found in almost all cases; the maximum gap observed was 8.84%. We have observed that a zero gap may

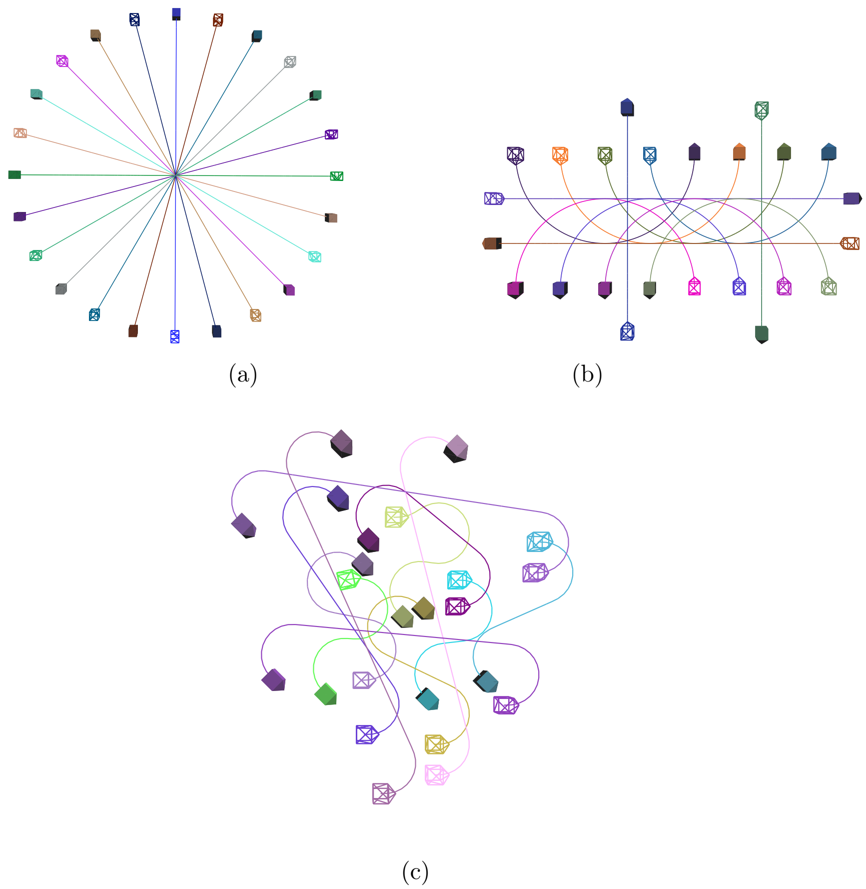

Fig. 6. Overhead view of example paths for 12 robots: (a) radial paths with symmetry, with a bottleneck at the center; (b) constant curvature straight-line and circular paths for car-like robots; (c) SCC paths for car-like robots. Goal configurations are indicated by solid polyhedra. Animations may be seen in Multimedia Extensions 1–6.

occur even with small values of a_{max}. Example animations can be seen at http://www.cs.rpi.edu/∼sakella/multikino/ and in the Appendix.

8.1. 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.

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. We use two MILP formulations, of which the setpoint model provides a continuous velocity schedule that is feasible and is an upper bound on the optimal continuous velocity schedule, and the improved instantaneous model provides a lower bound on the optimal solution. 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 collision zones. However, efficient collision detection software and integer programming solvers make this approach practical for reasonable problem sizes. Furthermore, the input set of robots may be partitioned into smaller disjoint sets of robots with shared collision zones, which may then be coordinated independently.

There are several directions for future work. While we have focused on reducing the completion time, other linear objective functions, such as the average completion time or the total execution time, may be optimized. Additionally, objective functions that may be described as piecewise linear functions or convex quadratic functions may also be solved using CPLEX (ILOG Inc. 1999). It is important to analytically characterize the gap between the improved instantaneous

Table 1. Sample Run Times for Setpoint Formulation (MILP-S) and Improved Instantaneous Formulation (MILP-I)

Number of Robots	Number of Collision Zones	Collision Time (s)	Number of Binary Variables	MILP-S Time (s)	Number of Binary Variables	MILP-I Time (s)	% Gap Between MILP-S and MILP-I
5	13	17.64	20	0.04	14	0	0.0
8	42	52.63	64	0.13	62	0.08	0.0
10	71	84.87	102	0.53	100	0.17	0.0
12	82	109.96	124	0.61	123	0.25	0.0
8 (radial, unsymm.)	32	62.89	54	0.167	54	0.095	0.0
12 (radial, unsymm.)	94	166.19	128	2.2	128	0.49	0.0
8 (radial, symm.)	29	28.4	54	3.87	54	0.095	0.0
12 (radial, symm.)	86	20.556	124	216.7	124	65.7	0.0
12 (const. curvature)	154	11.62	158	2.167	158	0.967	7.4
12 (SCC)	64	118.94	85	0.296	85	0.16	0.0

Note. (The MILP-I formulation used the MILP-S solutions as upper bounds.) Collision checks were performed at 65–200 points along each path for most paths, and at 605 points for the SCC paths. AMPL presolve times are not included.

model and setpoint model solutions, and to develop heuristic algorithms for closing the gap. Developing computational procedures to directly solve the continuous velocity MINLP formulation is important. We are currently characterizing the convexity of the MINLP constraints to develop an approach to directly solve for the global optimum of the MINLP.

Extending the approach to general car-like robots, aircraft, and manipulator robots is an interesting next step, since the complex constraints on the state and control variables make solving the TPBVPs for individual segments challenging. To apply this approach to aircraft, we can use a simplified planar aircraft model (Bicchi and Pallottino 2000). For specified continuous curvature paths, the velocity is a variable with magnitude bounds and the acceleration is the control input. To apply this approach to manipulator robots, we must compute appropriate collision zones and solve the TPBVPs for the dynamic model of the robots. One approach is to generate optimal velocity profiles by extending previous methods for optimizing motion of a manipulator along a path (Bobrow, Dubowsky, and Gibson 1985; Shin and McKay 1985; Slotine and Yang 1989; Shiller and Lu 1992). However, we must additionally address computing the maximum traversal times, and generating velocity profiles consistent with the computed traversal times. An alternative approach we have recently demonstrated is to use the time-scaling law for robot manipulators (Hollerbach 1984) to perform time-scaled coordination of multiple manipulators (Akella and Peng 2004).

To apply the approach for coordinating multiple robots in this paper to more general systems with complex linear or nonlinear dynamics, we must turn to numerical methods, as exact analytical solutions do not exist. A promising approach is to use specialized nonlinear programming techniques and to solve the discretized version of the optimal control problem as a large-scale sparse nonlinear programming problem (von Stryk 2000; Betts 2001). The advantages of this approach are that it does not require writing down the first-order necessary conditions, which may be tedious or even impossible to do for complex dynamics, and it leverages advances in nonlinear solvers, such as SNOPT (Gill, Murray, and Saunders 2002), which can solve nonlinear programming problems with many thousands of constraints and variables.

The approach described here represents a step towards solving the challenging problem of coordinating multiple robots without specified paths. To relax the fixed path constraint, we are developing formulations for coordination of multiple robots on a roadmap where each robot can select a path from a set of candidate paths. By introducing a binary variable for each candidate path to denote whether that path is chosen, the roadmap coordination can also be modeled as a mixed integer programming problem with additional roadmap constraints. The approach in this paper can also be combined with probabilistic techniques, which can generate paths (Švestka and Overmars 1998; Sanchez and Latombe 2002) or trajectories (LaValle and Kuffner 2001; Hsu et al. 2001) for the set of robots, to then optimize robot motions along those paths subject to dynamics constraints. Automatic modification of robot paths to reduce completion time would be a useful extension. Another interesting direction is on-line coordination of multiple robots using sensor-based estimates of robot positions and velocities.

Appendix: Index to Multimedia Extensions

The multimedia extension page is found at http://www.ijrr.org. Each example animation depicts a set of robots before and after coordination. Goal configurations are indicated by solid polyhedra. Collisions are indicated by changing the color of the colliding robots to red.

Table of Multimedia Extensions

Extension	Type	Description
1	Animation	12 robots moving along radial paths with symmetry, with a bottleneck at the center (before coordination)
2	Animation	12 robots moving along radial paths with symmetry, with a bottleneck at the center (after coordination)
3	Animation	12 car-like robots moving on constant curvature straight-line and circular paths (before coordination)
4	Animation	12 car-like robots moving on constant curvature straight-line and circular paths (after coordination)
5	Animation	12 car-like robots moving on simple continuous curvature paths (before coordination)
6	Animation	12 car-like robots moving on simple continuous curvature paths (after coordination)

Acknowledgments

This work was supported in part by RPI and by NSF under CAREER Award No. IIS-0093233. Thanks to Prasad Akella for getting us started, Seth Hutchinson for early discussions, and John Mitchell, John Wen, and Charles Wampler for advice along the way. Andrew Andkjar implemented animation software that interfaced with the PQP software from the University of North Carolina and helped generate examples. Thierry Fraichard generously provided code to generate continuous curvature paths, and Mayur Patel created a convenient interface.

References

Akella, S. and Hutchinson, S. 2002. Coordinating the motions of multiple robots with specified trajectories. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Washington, DC, May, pp. 624–631.

Akella, S. and Peng, J. 2004. Time-scaled coordination of multiple manipulators. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, New Orleans, LA, April, pp. 3337–3344.

Alami, R., Fleury, S., Herrb, M., Ingrand, F., and Robert, F. 1998. Multi-robot cooperation in the MARTHA project. *IEEE Robotics and Automation Magazine* 5(1):36–47.

Aronov, B., de Berg, M., van der Stappen, A. F., Švestka, P., and Vleugels, J. 1999. Motion planning for multiple robots. *Discrete and Computational Geometry* 22:505–525.

Barraquand, J., Langlois, B., and Latombe, J.-C. 1992. Numerical potential field techniques for robot path planning. *IEEE Transactions on Systems, Man, and Cybernetics* 22(2):224–241.

Betts, J. T. 2001. *Practical Methods for Optimal Control Using Nonlinear Programming*. Society for Industrial and Applied Mathematics, New York.

Bicchi, A. and Pallottino, L. 2000. On optimal cooperative conflict resolution for air traffic management systems. *IEEE Transactions on Intelligent Transportation Systems* 1(4):221–231.

Bien, Z. and Lee, J. 1992. A minimum time trajectory planning method for two robots. *IEEE Transactions on Robotics and Automation* 8(3):414–418.

Bobrow, J. E., Dubowsky, S., and Gibson, J. S. 1985. Time-optimal control of robotic manipulators along specified paths. *International Journal of Robotics Research* 4(3):3–17.

Bryson, A. E. and Ho, Y-C. 1975. *Applied Optimal Control*. Hemisphere, Washington DC.

Buckley, S. J. 1989. Fast motion planning for multiple moving robots. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Scottsdale, AZ, May, pp. 322–326.

Canny, J., Rege, A., and Reif, J. 1991. An exact algorithm for kinodynamic planning in the plane. *Discrete and Computational Geometry* 6:461–484.

Chang, C., Chung, M. J., and Bien, Z. 1990. Collision-free motion planning for two articulated robot arms using minimum distance functions. *Robotica* 8:137–144.

Chang, C., Chung, M. J., and Lee, B. H. 1994. Collision avoidance of two robot manipulators by minimum delay time. *IEEE Transactions on Systems, Man, and Cybernetics* 24(3):517–522.

Donald, B. R. and Xavier, P. 1995a. Provably good approximation algorithms for optimal kinodynamic planning: Robots with decoupled dynamics bounds. *Algorithmica* 14(6):443–479.

Donald, B. R. and Xavier, P. 1995b. Provably good approximation algorithms for optimal kinodynamic planning for Cartesian robots and open chain manipulators. *Algorithmica* 14(6):480–530.

Donald, B. R., Xavier, P., Canny, J., and Reif, J. 1993. Kinodynamic motion planning. *Journal of the ACM* 40(5):1048–1066.

Erdmann, M. and Lozano-Perez, T. 1987. On multiple moving objects. *Algorithmica* 2(4):477–521.

Fiorini, P. and Shiller, Z. 1993. Motion planning in dynamic environments using the relative velocity paradigm. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Atlanta, GA, May, pp. 560–565.

Fiorini, P. and Shiller, Z. 1996. Time-optimal trajectory planning in dynamic environments. *Proceedings of the IEEE International Conference on Robotics and Automation*

(ICRA), Minneapolis, MN, April, pp. 1553–1558.

Fiorini, P. and Shiller, Z. 1998. Motion planning in dynamic environments using velocity obstacles. *International Journal of Robotics Research* 17(7):760–772.

Fourer, R., Gay, D. M., and Kernighan, B. W. 1993. *AMPL: A Modeling Language for Mathematical Programming*. Duxbury Press.

Fraichard, T. 1999. Trajectory planning in dynamic workspace: a “state-time space” approach. *Advanced Robotics* 13(1):75–94.

Fraichard, T., Scheuer, A., and Desvigne, R. 1999. From Reeds and Shepp’s to continuous-curvature paths. *Proceedings of the International Conference on Advanced Robotics*, Tokyo, Japan, October, pp. 585–590.

Frazzoli, E. 2003. Quasi-random algorithms for real-time spacecraft motion planning and coordination. *Acta Astronautica* 53(4–10):485–495.

Frazzoli, E., Mao, Z., Oh, J., and Feron, E. 2001. Resolution of conflicts involving many aircraft via semidefinite programming. *AIAA Journal of Guidance, Control and Dynamics* 24(1):79–86.

Freund, E. and Hoyer, H. 1988. Real-time path finding in multirobot systems including obstacle avoidance. *International Journal of Robotics Research* 7(1):42–70.

Fujimura, K. and Samet, H. 1989. A hierarchical strategy for path planning among moving obstacles. *IEEE Transactions on Robotics and Automation* 5(1):61–69.

Ghrist, R. and Koditschek, D. 2002. Safe cooperative robot dynamics on graphs. *SIAM Journal on Control and Optimization* 40(5):1556–1575.

Gill, P. E., Murray, W., and Saunders, M. A. 2002. SNOPT: an SQP algorithm for large-scale constrained optimization. *SIAM Journal on Optimization* 12(4):979–1006.

Hao, Y., Laxton, B., Agrawal, S. K., Lee, E., and Benson, E. 2003. Planning and control of UGV formations in a dynamic environment: A practical framework with experiments. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Taipei, Taiwan, September, pp. 1209–1214.

Heinzinger, G., Jacobs, P., Canny, J., and Paden, B. 1990. Time-optimal trajectories for a robot manipulator: a provably good approximation algorithm. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Cincinnati, OH, May, pp. 150–156.

Hollerbach, J. M. 1984. Dynamic scaling of manipulator trajectories. *ASME Journal of Dynamic Systems, Measurement, and Control* 106(1):102–106.

Hopcroft, J. E., Schwartz, J. T., and Sharir, M. 1984. 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.

Hsu, D., Kindel, R., Latombe, J.-C., and Rock, S. 2001. Randomized kinodynamic motion planning with moving obstacles. *Algorithmic and Computational Robotics: New Directions*, B. R. Donald, K. M. Lynch, and D. Rus, editors, A. K. Peters, Natick, MA, pp. 247–264.

ILOG Inc. 1999. *CPLEX 6.6 User’s Manual*. Incline Village, NV.

Kahn, M. E. and Roth, B. 1971. The near-minimum-time control of open-loop articulated kinematic chains. *ASME Journal of Dynamic Systems, Measurement, and Control* 93(3):164–172.

Kant, K. and Zucker, S. W. 1986. Toward efficient trajectory planning: The path-velocity decomposition. *International Journal of Robotics Research* 5(3):72–89.

Knowles, G. 1981. *An Introduction to Applied Optimal Control*. Mathematics in Science and Engineering, Vol. 159, Academic Press, New York.

Kosecka, J., Tomlin, C., Pappas, G., and Sastry, S. 1997. Generation of conflict resolution maneuvers for air traffic management. *Proceedings of the IEEE/RSJ International Conference on Robots and Systems (IROS)*, Grenoble, France, pp. 1598–1603.

Lamiraux, F. and Laumond, J.-P. 1998. From paths to trajectories for multi-body mobile robots. *International Symposium on Experimental Robotics V*, Lecture Notes in Control and Information Sciences, A. Casals and A. T. De Almeida, editors, Springer-Verlag, Berlin.

Lamiraux, F. and Laumond, J.-P. 2001. Smooth motion planning for car-like vehicles. *IEEE Transactions on Robotics and Automation* 17(4):498–502.

Larsen, E., Gottschalk, S., Lin, M., and Manocha, D. 2000. Fast distance queries using rectangular swept sphere volumes. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, San Francisco, CA, April, pp. 3719–3726.

Latombe, J.-C. 1991. *Robot Motion Planning*, Kluwer Academic, Dordrecht.

Laumond, J.-P., editor. 1998. *Robot Motion Planning and Control*, Lecture Notes in Control and Information Sciences Vol. 229, Springer-Verlag, Berlin.

LaValle, S. M. and Hutchinson, S. A. 1998. Optimal motion planning for multiple robots having independent goals. *IEEE Transactions on Robotics and Automation* 14(6):912–925.

LaValle, S. M. and Kuffner, J. J. 2001. Randomized kinodynamic planning. *International Journal of Robotics Research* 20(5):378–401.

Lee, B. H. and Lee, C. S. G. 1987. Collision-free motion planning of two robots. *IEEE Transactions on Systems, Man, and Cybernetics* 17(1):21–32.

Nemhauser, G. L. and Wolsey, L. A. 1988. *Integer and Combinatorial Optimization*, Wiley, New York.

O’Donnell, P. A. and Lozano-Perez, T. 1989. Deadlock-free and collision-free coordination of two robot manipulators. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Scottsdale, AZ, May, pp. 484–489.

O'Dunlaing, C. 1987. Motion planning with inertial constraints. *Algorithmica* 2(4):431–475.

Pallottino, L., Feron, E., and Bicchi, A. 2002. Conflict resolution problems for air traffic management systems solved with mixed integer programming. *IEEE Transactions on Intelligent Transportation Systems* 3(1):3–11.

Parsons, D. and Canny, J. 1990. Motion planning for multiple mobile robots. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Cincinnati, OH, pp. 8–13.

Peng, J. and Akella, S. 2003a. Coordinating multiple robots with kinodynamic constraints along specified paths. *Algorithmic Foundations of Robotics V*, J.-D. Boissonnat, J. Burdick, K. Goldberg, and S. Hutchinson, editors, Springer-Verlag, Berlin, pp. 221–237.

Peng, J. and Akella, S. 2003b. Coordinating the motions of multiple robots with kinodynamic constraints. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Taipei, Taiwan, September, pp. 4066–4073.

Pfeiffer, F. and Johanni, R. 1987. A concept for manipulator trajectory planning. *IEEE Journal of Robotics and Automation* 3(2):115–123.

Pinedo, M. 1995. *Scheduling: Theory, Algorithms, and Systems*, Prentice-Hall, Englewood Cliffs, NJ.

Pledgie, S., Hao, Y., Ferreira, A., Agrawal, S. K., and Murphey, R. 2002. Groups of unmanned vehicles: differential flatness, trajectory planning, and control. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Washington, DC, May, pp. 3461–3466.

Reif, J. and Sharir, M. 1994. Motion planning in the presence of moving obstacles. *Journal of the ACM* 41(4):764–790.

Reif, J. and Wang, H. 1997. Non-uniform discretization approximations for kinodynamic motion planning and its applications. *Algorithms for Robotic Motion and Manipulation*, J.-P. Laumond and M. Overmars, editors, A. K. Peters, Wellesley, MA, pp. 97–112.

Richards, A., How, J., Schouwenaars, T., and Feron, E. 2001. Plume avoidance maneuver planning using mixed integer linear programming. *Proceedings of the AIAA Guidance, Navigation and Control Conference*, Montreal, Canada, August.

Rizzi, A. A., Gowdy, J., and Hollis, R. L. 2001. Distributed coordination in modular precision assembly systems. *International Journal of Robotics Research* 20(10):819–838.

Sahar, G. and Hollerbach, J. M. 1986. Planning of minimum-time trajectories for robot arms. *International Journal of Robotics Research* 5(3):97–140.

Sanchez, G. and Latombe, J. 2002. On delaying collision checking in PRM planning—application to multi-robot coordination. *International Journal of Robotics Research* 21(1):5–26.

Scheuer, A. and Fraichard, T. 1997. Continuous-curvature path planning for car-like vehicles. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Grenoble, France, September, Vol. 2, pp. 997–1003.

Schouwenaars, T., De Moor, B., Feron, E., and How, J. 2001. Mixed integer programming for multi-vehicle path planning. *European Control Conference 2001*, Porto, Portugal.

Shiller, Z. and Chen, J. C. 1990. Optimal motion planning of autonomous vehicles in three-dimensional terrains. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Cincinnati, OH, May, pp. 198–203.

Shiller, Z. and Dubowsky, S. 1991. 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.

Shiller, Z. and Lu, H. H. 1992. Computation of path constrained time-optimal motions with dynamic singularities. *ASME Journal of Dynamic Systems, Measurement, and Control* 114:34–40.

Shiller, Z., Large, F., and Sekhavat, S. 2001. Motion planning in dynamic environments: Obstacles moving along arbitrary trajectories. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Seoul, Korea, May, pp. 3716–3721.

Shin, Y. and Bien, Z. 1989. Collision-free trajectory planning for two robot arms. *Robotica* 7:205–212.

Shin, K. G. and McKay, N. D. 1985. Minimum-time control of robotic manipulators with geometric path constraints. *IEEE Transactions on Automatic Control* 30(6):531–541.

Shin, K. G. and Zheng, Q. 1992. Minimum-time collision-free trajectory planning for dual robot systems. *IEEE Transactions on Robotics and Automation* 8(5):641–644.

Simeon, T., Leroy, S., and Laumond, J.-P. 2002. Path coordination for multiple mobile robots: a resolution-complete algorithm. *IEEE Transactions on Robotics and Automation* 18(1):42–49.

Slotine, J.-J. and Yang, H. S. 1989. Improving the efficiency of time-optimal path-following algorithms. *IEEE Transactions on Robotics and Automation* 5:118–124.

Švestka, P. and Overmars, M. 1998. Coordinated path planning for multiple robots. *Robotics and Autonomous Systems* 23(3):125–152.

Tomlin, C., Pappas, G. J., and Sastry, S. 1998. Conflict resolution for air traffic management: a study in multi-agent hybrid systems. *IEEE Transactions on Automatic Control* 43(4):509–521.

von Stryk, O. 2000. *User's Guide for DIRCOL (Version 2.1): a direct collocation method for the numerical solution of optimal control problems*. Technische Universitat Darmstadt.

Zefran, M., Desai, J. P., and Kumar, V. 1997. Continuous motion plans for robotic systems with changing dynamic behavior. *Algorithms for Robotic Motion and Manipulation*, J.-P. Laumond and M. Overmars, editors, A. K. Peters, Wellesley, MA.