Proceedings of the 2003 IEEE
International Conference on Robotics & Automation
Taipei, Taiwan, September 14-19, 2003

Coordinating the Motions of Multiple Robots with Kinodynamic Constraints

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

Srinivas Akella
Department of Computer Science
Rensselaer Polytechnic Institute
Troy, New York 12180
sakella@cs.rpi.edu

Abstract

This paper focuses on the coordination of multiple robots with kinodynamic constraints along specified paths. The presented approach generates continuous velocity profiles that avoid collisions and minimize the completion time for the robots. The approach identifies collision segments along each robot's path and then optimizes the motions of the robots along their collision and collision-free segments. 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. Then the collision avoidance constraints for pairs of robots can 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 a continuous velocity schedule. The approach is illustrated with robots modeled as double integrators subject to velocity and acceleration constraints. An implementation that coordinates 12 nonholonomic car-like robots is described.

1 Introduction

Coordinating multiple robots with kinodynamic constraints, i.e. simultaneous kinematic and dynamics constraints ([8]), in a shared workspace without collisions has applications in manufacturing cells ([28]), AGV coordination in harbors and airports ([2]), and air traffic control ([24]). The general problem requires finding the trajectory (path and velocity profile) of each robot such that a specified objective, such as the task completion time, total time, or energy consumption, of the system is minimized.

We present here an approach to generate continuous velocity profiles for multiple robots with specified paths and dynamics constraints so their motions are collision-free and minimize the task completion time. This is in contrast to prior work that mostly addressed either the collision-free path or trajectory coordination of several robots without considering dynamics constraints ([23],[20],[36],[1]), or the search for time-optimal motions for a single robot ([5],[34]). An example application is the coordination of the motions of large numbers of AGVs along specified paths in harbors and airports ([2]). 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 and when it can enter and exit its collision segments, 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* and *setpoint* formulations, that provide schedules that are lower and upper bounds on the optimal solution. We illustrate the approach using robots modeled as double integrators, and demonstrate its application to nonholonomic car-like robots with dynamics constraints.

1.1 Related Work

Multiple Robot Coordination: The problem of motion planning for multiple robots is to have each robot move from its initial to its goal configuration, while avoiding collisions with obstacles or other robots ([18]). This problem is highly underconstrained, and Hopcroft, Schwartz, and Sharir [12] showed that even a simplified two-dimensional case of the problem is PSPACE-hard. Recent efforts have focused on probabilistic approaches. A potential field randomized path planner was applied to multiple robot planning ([3]), and probabilistic roadmap planners have been developed for multiple car-like robots ([38]) and manipulators ([30]).

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 ([27], [14]). To plan the motions of multiple robots, Erdmann and Lozano-Perez [9] assign priorities to robots and sequentially search for collision-free paths for the robots, in order of priority, 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 [23] developed a method for path coordination of two robots. LaValle and Hutchinson [20] addressed a similar problem where each robot was constrained to a specified configuration space roadmap. The work most closely related to ours is that of Simeon, Leroy, and Laumond [36]. They perform path coordination for a very large number of car-like robots in the plane, where robots with intersecting paths can be partitioned into smaller sets. A more constrained version of this problem is the trajectory coordination problem where the trajectory (path and velocity) of each robot is specified. Previous work on trajectory coordination has focused almost exclusively on dual robot systems ([4], [7], [35]). Akella and Hutchinson [1] recently developed an MILP formulation to coordinate large numbers of robots with specified trajectories by changing only robot start times.

Trajectory Planning: There is a large body of work on the time optimal control of a single manipulator. Bobrow, Dubowsky, and Gibson [5] and Shin and McKay [34] developed algorithms to generate the time-optimal velocity profile of a manipulator along a specified path. Algorithms for minimum-time trajectory generation for a manipulator with dynamics and actuator constraints have also been developed ([29], [33]). Trajectory planning directly in the $2n$-dimensional state space that considers both kinematic and dynamic constraints is called *kinodynamic planning*. Donald et al. [8] developed a polynomial time approximation algorithm for kinodynamic planning for a single robot to generate near time-optimal trajectories. Fraichard [11] describes a trajectory planner for a car-like robot with dynamics constraints moving along a given path. Recent work on randomized kinodynamic planning includes the use of rapidly exploring random trees ([21]) and probabilistic roadmaps ([15]).

Air Traffic Control: Conflict resolution among multiple aircraft in a shared airspace ([37], [32], [24]) is closely related to multiple robot coordination. Tomlin, Pappas, and Sastry [37] synthesized safe conflict resolution maneuvers for two aircraft using speed and heading changes. Kosecka et al. [16] use potential field planners to generate conflict resolution maneuvers. Schouwenaars et al. [32] developed an MILP formulation for fuel-optimal path planning of multiple vehicles by using a discretized system model. Pallottino, Feron, and Bicchi [24] generate optimal conflict-free paths to minimize the total flight time and solve cases when either instantaneous velocity changes or heading angle changes are allowed.

2 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

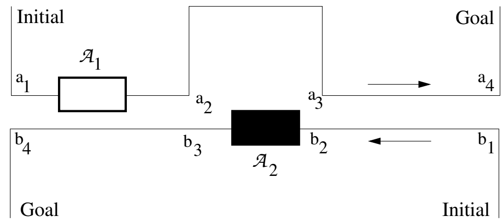

Figure 1: Example with two translating robots with two collision zones.

so that the completion time of the set of robots is minimized and their motions are collision free and satisfy their dynamics constraints. We assume that the start and goal configurations of each robot are collision-free, and that the specified paths for the robots are free of static obstacles. We further assume that each robot moves forward along its path without retracing its path.

2.1 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 $\psi(t) : \mathcal{R}^+ \rightarrow \mathcal{S}$ 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 1-to-1 mapping between s and the path length.

A *collision pair* $\mathcal{CP}_{ij}(s_i, s_j)$, where $s_i, s_j \in [0, 1]$ 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.

2.2 Optimal Control Problem For A Single Robot

Consider a robot $\mathcal{A}$ moving along a path segment. Let $\mathbf{x}(t)$ represent its state, $\mathbf{u}(t)$ be the control, γ be the path of $\mathcal{A}$, $J(\mathbf{x}, \mathbf{u})$ 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 segment traversal time for the robot subject to its dynamics and path constraints, can be written as:

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

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.

2.3 Coordinating 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 constraints 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 ([6]).

This approach easily incorporates multiple moving obstacles with known trajectories. Each moving obstacle is treated like a robot with a known velocity profile whose collision constraints are included in the MILP formulations.

3 Instantaneous Model

We first consider a simplified model, the *instantaneous model*, where each robot moves only at its highest speed v_{max}, and can instantaneously start or stop with infinite acceleration. The discontinuous velocity schedule provided by the instantaneous model is a lower bound to the optimal continuous velocity schedule.

3.1 MILP Formulation

We now present a mixed integer linear programming (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 $k+1$. 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 completion time C_{max} for the set of robots is greater than or equal to the completion time of each robot. 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 form ([22]) 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 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:

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

The collision avoidance constraints are conservative in not allowing two robots to simultaneously be in their collision zone, and in some cases 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.

The instantaneous model for multiple robot coordination can be viewed as a *job shop scheduling problem*, which is NP-hard ([26]). By reduction, the instantaneous model for robot coordination is NP-hard.

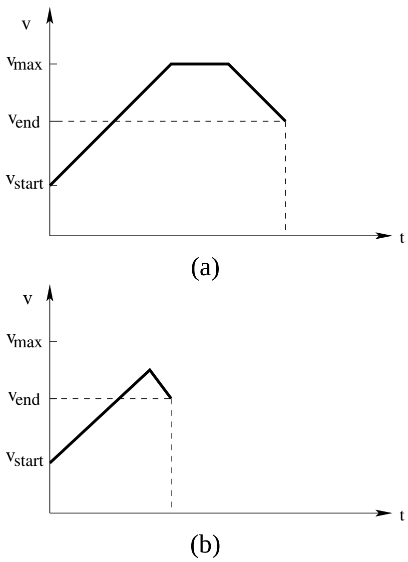

Figure 2: Minimum ΔT. Case (a): Velocity reaches v_{max}. Case (b): Velocity cannot reach v_{max}.

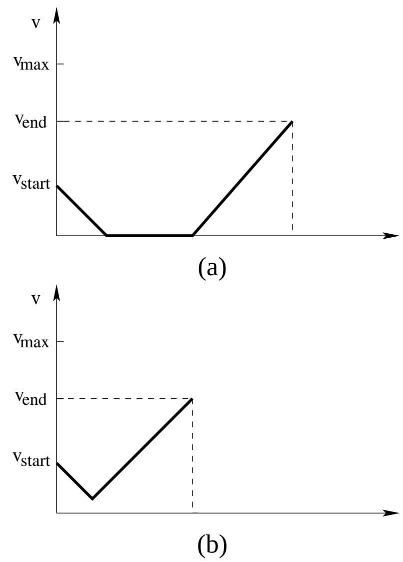

Figure 3: Maximum ΔT. Case (a): Velocity can decrease to zero. Case (b): Velocity cannot decrease to zero.

4 Continuous Velocity Model

We now consider generating a schedule with continuous velocity profiles for the robots 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 over the segment. We illustrate this procedure using the *double integrator* model from classical optimal control ([6]).

4.1 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. The minimum time control of the double integrator model is well known ([6]) and we have extended this to obtain the maximum time control. Basically the solutions to these TPBVPs have a bang-bang or bang-off-bang control structure. Let S be the length of the segment, ΔT be the time taken to traverse the segment, and v_{start} and v_{end} be the velocities at the segment endpoints. 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 solution since the distance is too short for a feasible velocity profile.

1. Minimum ΔT (Figure 2):

   (a) If $S \geq \frac{v_{max}^2 - v_{start}^2 + v_{max}^2 - v_{end}^2}{2a_{max}}$,

   $$\Delta T^{min} = -\frac{(v_{max}^2 - v_{start}^2 + v_{max}^2 - v_{end}^2)}{2a_{max} \cdot v_{max}} + \frac{S}{v_{max}} + \frac{v_{max} - v_{start}}{a_{max}} + \frac{v_{max} - v_{end}}{a_{max}}$$

   (b) If $\frac{2v_{max}^2 - v_{start}^2 - v_{end}^2}{2a_{max}} > S \geq \frac{|v_{end}^2 - v_{start}^2|}{2a_{max}}$,

   $$\Delta T^{min} = \frac{v_{middle} - v_{start}}{a_{max}} + \frac{v_{middle} - v_{end}}{a_{max}}$$

   where $v_{middle} = \frac{1}{2}(2v_{start}^2 + 2v_{end}^2 + 4Sa_{max})^{\frac{1}{2}}$

2. Maximum ΔT (Figure 3):

   (a) If $S \geq \frac{(v_{start}^2 + v_{end}^2)}{2a_{max}}$, $\Delta T^{max} = \infty$.

   (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 $v_{middle} = \frac{1}{2}(2v_{start}^2 + 2v_{end}^2 - 4Sa_{max})^{\frac{1}{2}}$

4.2 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 (MINLP) model for generating a minimum time continuous velocity schedule. We have the usual completion time and collision avoidance constraints. The traversal time constraints are more complicated. Let $a_{i,max}$ and $v_{i,max}$ be the maximum acceleration and velocity of robot $\mathcal{A}_i$. Let v_{ik} represent the velocity of robot $\mathcal{A}_i$ at the start of segment k. Let ΔT_{ik}^{min} and ΔT_{ik}^{max} be the minimum and maximum traversal times for robot $\mathcal{A}_i$ along 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) values (Section 4.1). 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 $v_{i,max}$ (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:

Minimize C_{max}
subject to:

$$C_{max} \geq t_{i,last} + \tau_{i,last} \qquad \text{for } i = 1, \ldots, n$$
$$t_{i(k+1)} = t_{ik} + \tau_{ik}$$
$$\Delta T_{ik}^{max} \geq \tau_{ik} \geq \Delta T_{ik}^{min}$$
$$t_{jh} - t_{i(k+1)} + M(1 - \delta_{ijkh}) \geq 0$$
$$t_{ik} - t_{j(h+1)} + M\delta_{ijkh} \geq 0$$
$$t_{ik} \geq 0$$
$$\delta_{ijkh} \in \{0, 1\}$$
$$S_{ik} \geq \frac{(v_{i(k+1)}^2 - v_{ik}^2)}{2a_{i,max}} \geq -S_{ik}$$
$$S_{ik} - \frac{(v_{i,max}^2 - v_{ik}^2 + v_{i,max}^2 - v_{i(k+1)}^2)}{2a_{i,max}} - My_{ik,1} \leq 0$$
$$S_{ik} - \frac{(v_{i,max}^2 - v_{ik}^2 + v_{i,max}^2 - v_{i(k+1)}^2)}{2a_{i,max}} + My_{ik,2} \geq 0$$
$$\Delta T_{ik,1}^{min} = \frac{S_{ik}}{v_{i,max}} - \frac{(v_{i,max}^2 - v_{ik}^2 + v_{i,max}^2 - v_{i(k+1)}^2)}{2a_{i,max}v_{i,max}} + \frac{v_{i,max} - v_{ik}}{a_{i,max}} + \frac{v_{i,max} - v_{i(k+1)}}{a_{i,max}}$$
$$\Delta T_{ik,2}^{min} = \frac{(v_{middle,ik}^{min} - v_{ik})}{a_{i,max}} + \frac{(v_{middle,ik}^{min} - v_{i(k+1)})}{a_{i,max}}$$
$$(v_{middle,ik}^{min})^2 = \frac{1}{4}(2v_{ik}^2 + 2v_{i(k+1)}^2 + 4S_{ik}a_{i,max})$$
$$\Delta T_{ik}^{min} = y_{ik,1} \cdot \Delta T_{ik,1}^{min} + y_{ik,2} \cdot \Delta T_{ik,2}^{min}$$
$$y_{ik,1} + y_{ik,2} = 1 \qquad y_{ik,1}, y_{ik,2} \in \{0, 1\}$$
$$(S_{ik} - \frac{v_{ik}^2 + v_{i(k+1)}^2}{2a_{i,max}}) - Mz_{ik,1} \leq 0$$
$$(S_{ik} - \frac{v_{ik}^2 + v_{i(k+1)}^2}{2a_{i,max}}) + Mz_{ik,2} \geq 0$$
$$\Delta T_{ik,1}^{max} = \infty$$
$$\Delta T_{ik,2}^{max} = \frac{(v_{ik} - v_{middle,ik}^{max})}{a_{i,max}} + \frac{(v_{i(k+1)} - v_{middle,ik}^{max})}{a_{i,max}}$$
$$(v_{middle,ik}^{max})^2 = \frac{1}{4}(2v_{ik}^2 + 2v_{i(k+1)}^2 - 4S_{ik}a_{i,max})$$
$$\Delta T_{ik}^{max} = z_{ik,1} \cdot \Delta T_{ik,1}^{max} + z_{ik,2} \cdot \Delta T_{ik,2}^{max}$$
$$z_{ik,1} + z_{ik,2} = 1 \qquad z_{ik,1}, z_{ik,2} \in \{0, 1\}$$
$$v_{i,max} \geq v_{ik} \geq 0$$
$$v_{i,initial} = v_{i,goal} = 0$$

This MINLP problem has very difficult nonconvex constraints. Existing optimization techniques to solve MINLPs either require convexity or are not guaranteed to find the optimal solution for large problem sizes. Hence we solve two MILPs that differ only in their ΔT^{max} values to obtain good lower and upper bounds on the optimal solution; the bounds have been very close in our experiments. Assume for simplicity that the first and last segments are sufficiently long for each robot to go from zero to v_{max} and vice versa. (This assumption can be relaxed as discussed in [25].)

1. Lower bound MILP: A lower bound for the MINLP problem can clearly be obtained by solving the MILP for the instantaneous model with infinite acceleration. We obtain a tighter lower bound by formulating an *improved instantaneous model* that considers the acceleration and deceleration time over the first and last segments for each robot. The minimum traversal times for the first and last segments are then $\Delta T^{min} = S/v_{max} + v_{max}/2a_{max}$. Solving the resulting MILP gives a lower bound for the MINLP problem.

2. Upper bound MILP: The MINLP is transformed into an MILP problem by setting the velocities at the endpoints of each segment (except the initial and goal velocities) to the maximum feasible velocity. Solving this setpoint MILP problem (see next section) gives a feasible continuous velocity schedule, which is therefore an upper bound for the MINLP problem.

5 Setpoint Model

The *setpoint model* is used to generate a continuous velocity schedule. Since any continuous velocity schedule is an upper bound on the optimal continuous velocity schedule, the setpoint model is guaranteed to provide an upper bound on the MINLP problem. Here each robot's velocity is set to its maximum feasible velocity at its collision zones endpoints, thereby biasing the robots to move through their collision zones in the shortest possible time. Setting the velocity v_{ik} of each robot at the endpoints of its segments to $v_{i,max}$ transforms the MINLP formulation to an MILP formulation with ΔT^{min} and ΔT^{max} as follows:

$$\Delta T^{min} = \begin{cases} \frac{S}{v_{max}} & \text{if interior segment} \\ \frac{v_{max}}{2a_{max}} + \frac{S}{v_{max}} & \text{if first or last segment} \end{cases}$$

$$\Delta T^{max} = \begin{cases} \infty & \text{if } S \geq \frac{v_{max}^2}{a_{max}} \\ \frac{2v_{max} - 2(v_{max}^2 - a_{max}S)^{\frac{1}{2}}}{a_{max}} & \text{if } S < \frac{v_{max}^2}{a_{max}} \end{cases}$$

5.1 MILP Formulation

The MILP formulation for the setpoint model is identical to the formulation for the improved instantaneous model, and differs only in the ΔT^{max} parameter values. When the segment traversal time τ_{ik} generated by the MILP does not correspond to either a minimum time or maximum time trajectory over the segment, we have a simple algorithm to generate a feasible velocity profile for the double integrator.

6 Car-like Mobile Robots

We now illustrate our coordination approach on nonholonomic car-like robots with dynamics constraints. Paths that satisfy the nonholonomic constraints (Laumond [19]) 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 [31]).

6.1 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 midpoint of the rear axle, θ is the robot orientation, and κ is the signed path curvature. v is the robot velocity at its reference point.

We model a car-like robot of mass m moving on a plane with a friction coefficient μ as subject to the following dynamics constraints (Fraichard [11]):

1. Tangential acceleration constraints:
   (a) Acceleration constraints due to the engine force F are: $\frac{F_{min}}{m} \leq a \leq \frac{F_{max}}{m}$.
   (b) Sliding constraints to prevent slipping 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: $a \geq \max(\frac{F_{min}}{m}, -\sqrt{\mu^2 g^2 - \kappa^2 v^4})$ and $a \leq \min(\frac{F_{max}}{m}, \sqrt{\mu^2 g^2 - \kappa^2 v^4})$.

2. Velocity constraints: In addition to the magnitude constraints $0 \leq v \leq v_{max}$, to ensure that $\mu^2 g^2 - \kappa^2 v^4 \geq 0$ we have the constraint $-\sqrt{\frac{\mu g}{|\kappa|}} \leq v \leq \sqrt{\frac{\mu g}{|\kappa|}}$.

   Thus the (state dependent) velocity constraints are: $0 \leq v \leq \min\left(v_{max}, \sqrt{\frac{\mu g}{|\kappa|}}\right)$.

6.2 Paths

The specified paths are chosen to be *simple continuous curvature paths* (SCC paths) (Scheuer and Fraichard [31]). 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 a path is upper bounded by κ_{max}, that is, the steering radius $\rho \geq \rho_{min} = 1/\kappa_{max}$. There is an upper bound on the time derivative of curvature, $\dot{\kappa}$.

We additionally assume $\mu^2 g^2 - \kappa^2 v^4 \geq (\frac{F_{max}}{m})^2$, which is true for typical values of the variables. This constraint can be expressed as a minimum steering radius constraint $\rho_{min} \geq v_{max}^2 / \sqrt{\mu^2 g^2 - (\frac{F_{max}}{m})^2}$ during path generation. This also implies that the maximum robot velocity is v_{max}.

6.3 Coordinating Multiple Car-like Robots

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

$$\text{Min or Max} \quad \Delta T = \int_0^{\Delta T} 1 dt$$

subject to:

$$\begin{aligned}
\begin{pmatrix} \dot{x} \\ \dot{v} \end{pmatrix} &= \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ v \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} a(t) \\
x(0) &= -S \quad x(\Delta T) = 0 \\
v(0) &= v_{start} \quad v(\Delta T) = v_{end} \\
0 \leq &v \leq v_{Max}(x) \\
-a_{Min}(x, v) \leq &a(t) \leq a_{Max}(x, v)
\end{aligned}$$

where $v_{Max}(x) = \min\left(v_{max}, \sqrt{\frac{\mu g}{|\kappa|}}\right)$, $-a_{Min}(x, v) = \max(\frac{F_{min}}{m}, -\sqrt{\mu^2 g^2 - \kappa^2 v^4})$, and $a_{Max}(x, v) = \min(\frac{F_{max}}{m}, \sqrt{\mu^2 g^2 - \kappa^2 v^4})$. This TPBVP is difficult to solve because of the complex constraints on the state and control variables.

The minimum steering radius constraint $\rho_{min} \geq v_{max}^2 / \sqrt{\mu^2 g^2 - (\frac{F_{max}}{m})^2}$ makes $v_{Max}(x)$, $a_{Min}(x, v)$, and $a_{Max}(x, v)$ state independent constants. Therefore the double integrator formulation of Section 4.1 applies to the car-like robots above. 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 can generate collision-free continuous velocity profiles along the specified paths that minimize the completion time using the MILP formulations described earlier.

7 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. [17]) by sampling uniformly along each robot's path. We generate the MILP formulations from the collision zones and solve them using the AMPL [10] and CPLEX [13] optimization packages. Since the setpoint formulation with its tighter constraints is solved much faster than the improved instantaneous formulation, we use the objective function value from the setpoint solution as an upper bound constraint in 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 radial case with a bottleneck at the center) or for a sufficiently large number of collision zones, the MILP time dominates the running time.

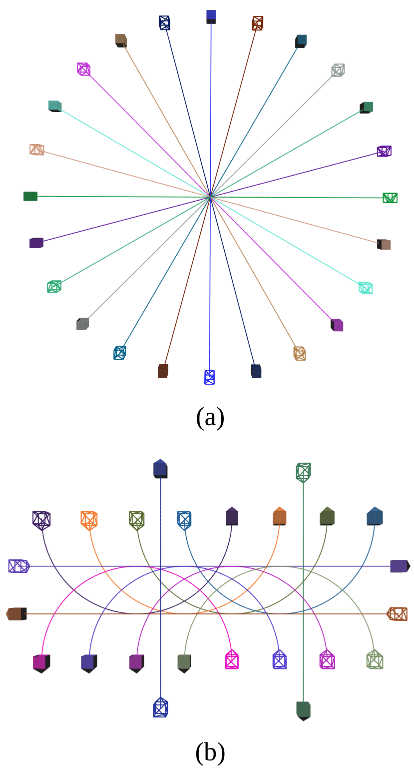

Figure 4: Overhead view of example paths for 12 robots: (a) Radial paths, with a bottleneck at the center (b) Simple continuous curvature paths. Goal configurations are indicated by solid polyhedra.

In our experiments, an optimal solution, indicated by a zero gap between the objective function values computed by the improved instantaneous and setpoint formulations, was found in almost all cases; the maximum gap observed was 8.84%. Example animations can be seen at `www.cs.rpi.edu/~sakella/multikino/`.

8 Conclusion

By combining techniques from optimal control and mathematical programming, we developed an MINLP formulation for minimum time collision-free coordination of multiple robots with kinodynamic constraints along specified paths. We then developed two related MILP formulations that give upper and lower bounds on the optimal solution. Although the MILP formulations for coordination of multiple robots are NP-hard, the availability of efficient collision detection software and integer programming solvers makes this approach practical for reasonable problem sizes.

There are several directions for future work. We have recently developed formulations for coordination of robots where each robot can move along a set of possible paths, and are investigating their computational feasibility. Analytically characterizing the gap between the improved instantaneous model and setpoint model solutions, and developing heuristic algorithms for closing the gap is important. Extending the approach to systems with more complex dynamics, including aircraft and articulated robots, appears to be an attainable next step. Another interesting direction is online coordination of multiple robots using sensor estimates of robot positions and velocities.

Num. of robots	Num. of collision zones	Collision detection time (secs)	MILP-S time (secs)	MILP-I time (secs)
5	13	18.67	0.04	0
8	42	55.67	0.13	0.08
10	71	88.26	0.53	0.17
12	82	115.81	0.61	0.25
8radial	29	30.53	3.87	0.095
12radial	86	70.53	160	60.67
12scc	154	65.62	12	1.167

Table 1: Sample run times for setpoint formulation (MILP-S) and improved instantaneous formulation (MILP-I). (The MILP-I formulation used the MILP-S solutions as upper bounds.) Collision checks were performed at 200 points along each path. AMPL presolve times are not included.

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 Prasad Akella for suggesting the problem, Seth Hutchinson for early discussions, and John Mitchell for advice.

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] 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.

[4] 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.

[5] 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.

[6] A. E. Bryson and Y.-C. Ho. *Applied Optimal Control*. Hemisphere, Washington D.C., 1975.

[7] 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.

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

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

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

[11] T. Fraichard. Trajectory planning in dynamic workspace: a 'state-time space' approach. *Advanced Robotics*, 13(1):75–94, 1999.

[12] 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.

[13] ILOG, Inc., Incline Village, NV. *CPLEX 6.6 User's Manual*, 1999.

[14] 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.

[15] R. Kindel, D. Hsu, J. Latombe, and S. Rock. Kinodynamic motion planning amidst moving obstacles. In *IEEE International Conference on Robotics and Automation*, pages 537–543, San Francisco, Apr. 2000.

[16] 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.

[17] 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.

[18] J.-C. Latombe. *Robot Motion Planning.* Kluwer Academic Publishers, Norwell, MA, 1991.

[19] J.-P. Laumond, editor. *Robot Motion Planning and Control*. Lecture Notes in Control and Information Sciences, No. 229. Springer, 1998.

[20] 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.

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

[22] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization.* John Wiley and Sons, New York, 1988.

[23] 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.

[24] 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.

[25] J. Peng and S. Akella. Coordinating multiple robots with kinodynamic constraints along specified paths. In *Fifth Workshop on the Algorithmic Foundations of Robotics*, Nice, France, Dec. 2002.

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

[27] 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.

[28] 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.

[29] 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.

[30] G. Sanchez and J.-C. Latombe. Using a PRM planner to compare centralized and decoupled multi-robot systems. In *IEEE International Conference on Robotics and Automation*, pages 2112–2119, Washington, D.C., May 2002.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.