Proceedings of the 2005 IEEE
International Conference on Robotics and Automation
Barcelona, Spain, April 2005

Coordinating Multiple Double Integrator Robots on a Roadmap: Convexity and Global Optimality

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
sakella@cs.rpi.edu

***Abstract*— This paper focuses on finding the global minimum time control for the collision-free coordination of multiple robots with double integrator dynamics and with additional robot state constraints and control constraints. We initially assume each robot's path is specified and decompose it into collision segments and collision-free segments. The collision avoidance constraints for pairs of robots and the dynamics constraints can then be combined to formulate the coordination problem as a mixed integer nonlinear program (MINLP). In this paper, we first show convexity of the constraints for an individual robot path segment under certain assumptions. We next establish that we are guaranteed to find the global optimum of the MINLP because each subproblem of the MINLP is a convex program, based on the convexity result on individual robot segments. To the best of our knowledge, this is the first result on directly obtaining the global optimum coordination of multiple (more than two) robots with dynamics constraints.**

Finally, we extend these results to the task of coordinating robots on a given roadmap, where the roadmap has multiple candidate paths for each robot. We present an approach to simultaneously select each robot's traversal path and generate its continuous velocity profile. These robot velocity profiles satisfy the dynamics constraints, avoid collisions, and globally minimize the completion time.

We use the MINLP Solver [25], which combines a branch-and-bound algorithm with a filterSQP algorithm, to solve the MINLP coordination problems. We illustrate the approach with multiple robot coordination examples with up to 156 collision zones.

I. INTRODUCTION

This paper focuses on finding the global minimum time control for the collision-free coordination of multiple robots with double integrator dynamics and additional robot state constraints and control constraints. Systems with double integrator dynamics are of great interest due to their simple and descriptive features, and have been used to describe the dynamics of agents [42], mobile robots [27], and spacecraft [18]. Coordinating multiple robots with simultaneous kinematic and dynamics constraints along specified paths or roadmaps without collisions has applications in manufacturing cells ([33]) and the coordination of AGVs ([2]). The robots must satisfy kinematic constraints, such as avoiding collisions with other robots and with moving obstacles, and dynamics constraints, such as velocity and acceleration bounds.

We present here a direct approach to find the globally optimal continuous velocity profiles that satisfy the dynamics constraints, avoid collisions between robots, and achieve the global minimum task completion time. We generalize this to select the robots' traversal paths on a roadmap. These results extend our previous results (Peng and Akella [30], [31]) on generating continuous velocity coordination schedules that upper bound the optimal schedule for robots on specified paths. In contrast, previous work in robotics mostly addressed either the collision-free path coordination problem of several robots without considering dynamics constraints ([26],[23],[40]), or the search for time-optimal motions for a single robot ([6],[38]).

We initially assume each robot's path is specified and decompose it into collision segments and collision-free segments. The disjunctive collision avoidance constraints for pairs of robots and the dynamics constraints are combined to formulate the coordination problem as a mixed integer nonlinear program (MINLP). Our goal is to determine the control inputs (accelerations) that minimize the completion time. It is hard, in general, to solve such MINLPs because integer variables are coupled with nonlinear constraints. In our previous work we developed MILP formulations to bound the optimal MINLP schedule [31]. Here we develop a direct approach to solve the MINLP that builds on recent advances in efficiently solving convex NLP relaxations of MINLPs [10][25]. We use solver MINLP [25], which implements a branch-and-bound algorithm searching a tree whose nodes correspond to continuous nonlinear optimization problems. The continuous problems are solved using filterSQP, a Sequential Quadratic Programming solver for solving large nonlinearly constrained problems.

The paper is organized as follows. After a discussion of related work in Section II, we review our previous work in Sections III and IV. In Section V, we show convexity of the constraints for an individual robot path segment under certain assumptions. In Section VI, we establish that we are guaranteed to find the global optimum of the MINLPs because the relaxation of the MINLPs are convex programs, based on the convexity result for individual robot segments. Finally, in Section VII we address the task of coordinating robots on a given roadmap, where the roadmap has multiple candidate paths for each robot.

II. Related Work

Multiple Robot Coordination: The task 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 ([22]). This problem is highly underconstrained, and Hopcroft, Schwartz, and Sharir [16] showed that even a simplified two-dimensional case of the problem is PSPACE-hard. Recent efforts have focused on 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 ([41]) and manipulators ([35]).

A slightly more constrained version of the problem is obtained when all but one of the robots have specified trajectories. This is the problem of planning a path and velocity for a single robot among moving obstacles ([32], [17]). To plan the motions of multiple robots, Erdmann and Lozano-Perez [11] 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. These problems typically do not consider robot dynamics. O'Donnell and Lozano-Perez [26] developed a method for path coordination of two robots. LaValle and Hutchinson [23] addressed a path coordination problem, and a generalization where each robot is constrained to a specified configuration space roadmap. Simeon, Leroy, and Laumond [40] 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. Ghrist, O'Kane, and LaValle [15] compute Pareto optimal coordinations for multiple robots moving on individual roadmaps, and give a finiteness bound for the number of Pareto optimal paths. A more constrained version of this problem is the trajectory coordination problem where the trajectory (path and velocity) of each robot is specified. Early work on trajectory coordination focused almost exclusively on dual robot systems ([5], [8], [39]). Akella and Hutchinson [1] 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 [6] and Shin and McKay [38] 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 ([34], [37]). Donald et al. [9] developed a polynomial time approximation algorithm for kinodynamic planning for a single robot to generate near time-optimal trajectories. Fraichard [14] 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 ([24]) and probabilistic roadmaps ([19]).

Air Traffic Control: Conflict resolution among multiple aircraft in a shared airspace ([43], [36], [28]) is closely related to multiple robot coordination. Tomlin, Pappas, and Sastry [43] synthesized safe conflict resolution maneuvers for two aircraft using speed and heading changes. Kosecka et al. [20] use potential field planners to generate conflict resolution maneuvers. Schouwenaars et al. [36] developed an MILP formulation for fuel-optimal path planning of multiple vehicles by using a discretized system model. Pallottino, Feron, and Bicchi [28] 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.

III. Coordination Problem Review

We first consider the coordination problem defined in [31]: *Given a set of n robots $\mathcal{A}_1, \ldots, \mathcal{A}_n$, each with a specified path, find the control inputs along the paths so that the robots' dynamics constraints are satisfied, their motions are collision free, and the completion time of the set of robots is minimized*. 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. We assume that the start and goal configurations of each robot are collision-free. 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.

A *collision segment* for robot $\mathcal{A}_i$ is a contiguous interval $[s_i^{start}, s_i^{end}]$ over which $\mathcal{A}_i$ can collide with another robot $\mathcal{A}_j$. 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 $([s_i^{start}, s_i^{end}], [s_j^{start}, s_j^{end}])$ constitute a *collision zone* if and only if any point in one interval results in a collision with at least one point in the other interval. 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. (For details, see [31].)

A. Optimal Control Problem For A 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 $x(t)$, $v(t) = \frac{dx(t)}{dt}$, and $a(t) = \frac{dv(t)}{dt}$ be the position, velocity, and acceleration of the robot at time t, S be the length of the segment, and τ be the time taken to traverse the segment. Computing the minimum (or maximum) time taken by the robot to traverse the segment, subject to constraints on its velocities v_s and v_e at the segment endpoints ($v(0) = v_s,\ v(\tau) = v_e$) and inequality constraints on its velocity ($0 \leq v \leq v_{max}$) and acceleration ($-a_{max} \leq a \leq a_{max}$), can be solved as a two-point boundary value problem (Bryson and Ho [7]).

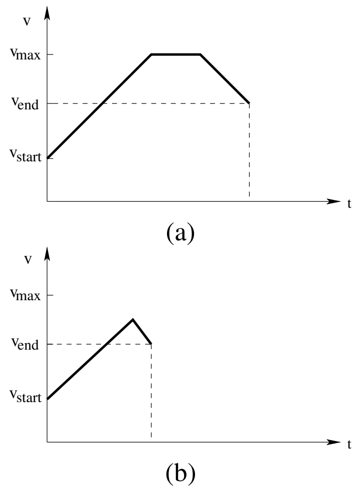


Fig. 1. Minimum τ. Case (a): Velocity reaches v_{max}. Case (b): Velocity cannot reach v_{max}.

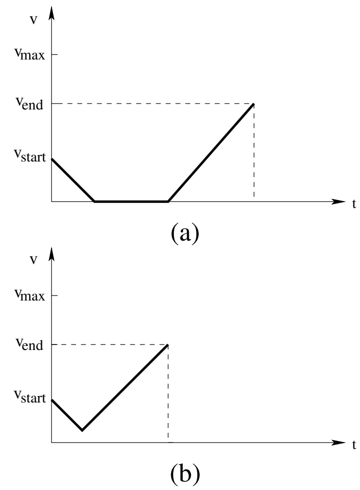


Fig. 2. Maximum τ. Case (a): Velocity can decrease to zero. Case (b): Velocity cannot decrease to zero.

We have previously described its parametric form with respect to the boundary values and also extended this to obtain the maximum time control [31].

B. Minimum traversal time τ^{min} (Figure 1):

Define

$$\mathcal{B} = \left\{(v_s, v_e) | S \leq \tfrac{1}{2}\left(\tfrac{(v_{max}^2 - v_s^2)}{a_{max}} + \tfrac{(v_{max}^2 - v_e^2)}{a_{max}}\right)\right\}. \quad (1)$$

Then,

$$\tau^{min} = \begin{cases} \tau_a^{min}, & \text{if} \quad (v_s, v_e) \notin \mathcal{B} \\ \tau_b^{min}, & \text{if} \quad (v_s, v_e) \in \mathcal{B} \end{cases}, \quad (2)$$

where

$$\begin{aligned} \tau_a^{min} &= \frac{Sa_{max} - \frac{1}{2}((v_{max}^2 - v_s^2) + (v_{max}^2 - v_e^2))}{a_{max} \cdot v_{max}} \\ &\quad + \frac{v_{max} - v_s}{a_{max}} + \frac{v_{max} - v_e}{a_{max}}, \\ \tau_b^{min} &= \frac{(\frac{1}{2}(2v_s^2 + 2v_e^2 + 4Sa_{max})^{\frac{1}{2}} - v_s)}{a_{max}} \\ &\quad + \frac{(\frac{1}{2}(2v_s^2 + 2v_e^2 + 4Sa_{max})^{\frac{1}{2}} - v_e)}{a_{max}} \end{aligned}$$

C. Maximum traversal time τ^{max} (Figure 2):

$$\tau^{max} = \begin{cases} \tau_a^{max} = \infty, \quad \text{if} \quad S \geq \frac{1}{2}\frac{(v_s^2 + v_e^2)}{a_{max}} \\ \\ \tau_b^{max} = \frac{(v_s - v_{middle})}{a_{max}} + \frac{(v_e - v_{middle})}{a_{max}}, \\ v_{middle} = \frac{1}{2}(2v_s^2 + 2v_e^2 - 4Sa_{max})^{\frac{1}{2}} \\ \text{if} \quad \frac{1}{2}\frac{(v_s^2 + v_e^2)}{a_{max}} > S \geq \frac{1}{2}\frac{|(v_e^2 - v_s^2)|}{a_{max}}, \end{cases} \quad (3)$$

Note that τ^{max} is discontinuous.

IV. Coordination of Multiple Robots: MINLP Formulation

We now describe generating a schedule for the robots that is consistent with their dynamics constraints. 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. 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 $t_{jh} \geq t_{i(k+1)}$ (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 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 robot velocities are variables in the minimum and maximum time control for a robot over a segment, and introduce nonlinear constraints. We therefore formulate a mixed integer nonlinear programming problem (MINLP) for the multiple robot coordination problem. The traversal time constraints are complicated since the minimum and maximum possible traversal times depend on the segment endpoint velocities, which are variables. Let $a_{i,max}$ be the maximum acceleration and $v_{i,max}$ be the maximum velocity of $\mathcal{A}_i$, and let v_{ik} represent the velocity of $\mathcal{A}_i$ at the start of segment k. Let τ_{ik}^{min} and τ_{ik}^{max} be the minimum and maximum possible traversal times for $\mathcal{A}_i$ along segment k. Let $\tau_{ik,a}^{min}$ and $\tau_{ik,b}^{min}$ represent the two minimum traversal time values (described in Section III-B). Similarly, let $\tau_{ik,a}^{max}$ and $\tau_{ik,b}^{max}$ represent the two maximum traversal time values. Without loss of generality, the velocities at the initial and goal configurations are assumed zero. The resulting MINLP formulation for the optimal continuous velocity schedule can be described by three sets of constraints.

1) The first set of constraints are the linear completion time constraints, traversal time constraints, and col-

lision avoidance constraints:

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

2) The second set of constraints are the nonlinear minimum and maximum traversal time constraints. Recall τ_{ik}^{min} defined in Section (III-B) and τ_{ik}^{max} defined in Section (III-C).
3) The third set of constraints are the feasible velocity constraints, whose convex representation is derived in Lemma 4.1.

$$\mathcal{V} = \left\{ (v_{ik}, v_{i(k+1)}) | \ S_{ik} \geq \tfrac{1}{2} \tfrac{|v_{ik}^2 - v_{i(k+1)}^2|}{a_{i,max}} \right\} \qquad (4)$$

To conclude this section, we prove the feasible velocity set is a convex set. For clarity, we drop index subscript ik here.

Lemma 4.1: The feasible velocity set (4) has a convex representation

$$\mathcal{V} = \left\{ (v_s, v_e) | \ \tau^{min} \geq \tfrac{-v_s + v_e}{a_{max}} \text{ and } \tau^{min} \geq \tfrac{v_s - v_e}{a_{max}} \right\} (5)$$

Proof: Recall τ^{min} defined in Equation (2). If $(v_s, v_e) \notin \mathcal{B}$, both (4) and (5) are satisfied trivially. If $(v_s, v_e) \in \mathcal{B}, \tau^{min} = \tau_b^{min} \geq \max\left(\frac{-v_s+v_e}{a_{max}}, \frac{v_s-v_e}{a_{max}} \right) \iff S \geq \frac{1}{2} \frac{|v_s^2 - v_e^2|}{a_{max}}$. ■

V. Differentiability and Convexity Properties of τ^{min}

Each robot's path is divided into collision segments and collision-free segments. All the constraints listed above, with the exception of τ_k^{min} and τ_k^{max}, are linear, and are clearly convex. To establish the global optimality of the solutions, we first show convexity of τ_k^{min}. Recall τ_a^{min} and τ_b^{min} defined in Equation (2).

Lemma 5.1: Given $\forall v_s, v_e$ feasible, $\tau_a^{min} \geq \tau_b^{min}$. Moreover, $\tau_a^{min} = \tau_b^{min}$ if and only if $S = \frac{1}{2}(\frac{(v_{max}^2 - v_s^2)}{a_{max}} + \frac{(v_{max}^2 - v_e^2)}{a_{max}})$.

Proof: We can show, after simplification, that

$$\tau_a^{min} - \tau_b^{min} = \frac{1}{2} \frac{(\sqrt{2} v_{max} - \sqrt{v_s^2 + v_e^2 + 2Sa_{max}})^2}{a_{max} v_{max}} \geq 0$$

Moreover, $\tau_a^{min} = \tau_b^{min}$ if and only if $S = \frac{1}{2}(\frac{(v_{max}^2 - v_s^2)}{a_{max}} + \frac{(v_{max}^2 - v_e^2)}{a_{max}})$ ■

Lemma 5.2: τ_a^{min} and τ_b^{min} are both differentiable and convex.

Proof: Obviously τ_a^{min} and τ_b^{min} are differentiable. We can prove the Lemma by showing that the Hessian matrices of τ_a^{min} and τ_b^{min} are positive definite, which can be verified easily.

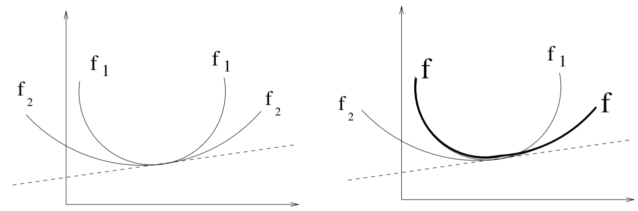


Fig. 3. If f_1 and f_2 are convex, then f is convex by Lemma (5.3).

■

Lemma 5.3: [29] Given convex functions $f_1(x), f_2(x) \in \mathcal{C}^1$, $\mathcal{R}^n \to \mathcal{R}$, and a set $\mathcal{S}$ such that $f_1(x) \geq f_2(x) \ \forall x \in \mathcal{R}^n$, $f_1(x) = f_2(x)$, $\forall x \in$ boundary of $\mathcal{S}$, then $f = \begin{cases} f_1, & x \notin \mathcal{S} \\ f_2, & x \in \mathcal{S} \end{cases}$ is convex. The Lemma is illustrated in Figure 3.

Now we are ready to give the cornerstone result of this paper.

Theorem 5.4: $\tau^{min} \in \mathcal{C}^1$ (first order differentiable) and is convex.

Proof:

1) Differentiability:
   At the boundary of $\mathcal{B}$ (Equation (1)), $\nabla\tau_a^{min} = \nabla\tau_b^{min}$, because $S = \frac{1}{2}(\frac{(v_{max}^2 - v_s^2)}{a_{max}} + \frac{(v_{max}^2 - v_e^2)}{a_{max}})$ on the boundary of $\mathcal{B}$.
2) Convexity:
   τ_a^{min} and τ_b^{min} are both convex functions. $\tau_a^{min} \geq \tau_b^{min}, \forall (v_s, v_e) \in \mathcal{R}^2$, and $\tau_a^{min} = \tau_b^{min}, \forall (v_s, v_e) \in$ boundary of $\mathcal{B}$. Thus $\tau^{min} = \begin{cases} \tau_a^{min}, x \notin \mathcal{B} \\ \tau_b^{min}, x \in \mathcal{B} \end{cases}$ is convex, according to Lemma (5.3).

■

Note that τ^{min} is not second order differentiable. However, most nonlinear programming algorithms, such as sequential quadratic programming (SQP) [12], do not require second order differentiability of the constraints.

VI. Finding a Global Optimal Schedule

It is convenient to view algorithms for solving MINLPs as searching a tree whose internal nodes depend on the partially instantiated integer values and leaf nodes correspond to continuous nonlinear optimization problems. We say an MINLP is convex if its NLP relaxation, obtained by ignoring the integer requirements of the integer variables, is convex.

There is a literature on the theory, algorithms, and solvers to find the global optimum solution of a convex MINLP ([10],[25]).

A. Assumptions

We now describe four assumptions, motivated by mobile robot and manufacturing automation applications respectively, where a robot must be able to come to rest to avoid a collision. Under one or more of these assumptions, we obtain various results on finding the global optimal coordination solution. These, to the best of our knowledge, are the first results on directly obtaining the global optimum solution of multiple (more than two) robot coordination with dynamics constraints. The key idea is to understand when the constraints $\tau^{max} \geq \tau$ can be made convex, or can be safely ignored.

Assumption 6.1: The robot must be able to stop in any segment. So for every segment, we enforce

$$\frac{1}{2}\frac{(v_s^2 + v_e^2)}{a_{max}} \leq S. \tag{6}$$

This implies $\tau^{max} = \infty$.

Assumption 6.2: The robot must be able to stop in any collision-free segment. So Equation (6) is true for all collision-free segments.

Assumption (6.2) is a relaxation of Assumption (6.1). Assumption (6.2) is important from a computational viewpoint since it greatly reduces the number of constraints related to the collision-free segments. It also provides nice convexity results, as we see later in this section.

Assumption 6.3: $\tau_k = \tau_k^{min}$ for every collision segment.

Assumption (6.3) implies that the robot passes through the collision zone as fast as possible, given the start and end segment velocities.

Assumption 6.4: Each path consists of an alternating sequence of collision segments and collision-free segments. Assumption (6.4) is true in general in uncrowded environments. If the environment is crowded, we can nevertheless concatenate adjacent collision segments into one large collision segment.

B. Existence and Sufficiency Results for Global Optimal Schedules

In the following, we refer to the MINLP model in Section III as the original MINLP.

Definition 6.5: MINLP-I is a MINLP model obtained from the original MINLP by dropping $\tau \leq \tau^{max}$ constraints and adding constraints of Assumption (6.1). *MINLP-II* is a MINLP model obtained from the original MINLP by dropping $\tau \leq \tau^{max}$ constraints and adding constraints of Assumption (6.2). *MINLP-III* is a MINLP model obtained from the original MINLP by dropping $\tau \leq \tau^{max}$ constraints and adding constraints of Assumptions (6.2) and (6.3).

Lemma 6.6: MINLP-I and MINLP-II are convex, and MINLP-III is not convex. Moreover, MINLP-II can be treated both as a relaxation of MINLP-I by dropping Equation (6) for collision segments and as a relaxation of MINLP-III by replacing $\tau^{min} = \tau$ with $\tau^{min} \leq \tau$ for collision segments.

Proof: $\left\{(v_s, v_e) | \frac{1}{2}\frac{(v_s^2+v_e^2)}{a_{max}} - S \leq 0\right\}$ is a convex set since $(\frac{1}{2}\frac{(v_s^2+v_e^2)}{a_{max}} - S)$ is a convex function, and $\left\{(v_s, v_e) | \tau^{min} \leq \tau\right\}$ is a convex set since τ^{min} is a convex function by Theorem (5.4). All other functions and constraints are linear for MINLP-I and MINLP-II. Therefore MINLP-I and MINLP-II are convex. $\left\{(v_s, v_e) | \tau = \tau^{min}\right\}$ is a nonconvex set, and so MINLP-III is nonconvex. ■

Note that any feasible solution of MINLP-I or MINLP-III is a feasible schedule. However a feasible solution of MINLP-II may not be a feasible schedule because possibly $\tau_k > \tau_k^{max}$ where k is a collision segment.

Theorem 6.7: (Global Optimality Result I): MINLP-I is convex. Therefore, a global minimum solution is guaranteed to be found for MINLP-I.

Proof: Follows from above convexity property. ■

Theorem 6.8: (Global Optimality Result II): MINLP-II gives a lower bound on the global optimum solution for the original MINLP. Furthermore, if the objective of MINLP-II equals the objective of MINLP-III, the solution of MINLP-III is a global optimum schedule in which each robot uses minimum traversal time along every collision segment.

Proof: Because MINLP-II is a convex relaxation of MINLP-III. ■

Theorem 6.9: (Global Optimality Result III): If Assumption (6.4) is true, solving MINLP-II will give the global optimum objective, even though solving MINLP-II may not give a feasible schedule. However a feasible global optimum schedule can be obtained through a post-processing step.

Proof: The schedule obtained by solving MINLP-II can have $\tau > \tau^{max}$ for collision segments. However, we can simply shift the additional traversal time $\tau - \tau^{min}$ to the neighboring collision-free segment based on Assumption (6.4) and obtain a feasible schedule, which is also a solution for MINLP-III. ■

VII. Coordination on Roadmaps

We now generalize the coordination problem to robots on a roadmap. The *coordination on a roadmap problem* is: Given a set of n robots $\mathcal{A}_1, \ldots, \mathcal{A}_n$ on a roadmap $\mathcal{R}$, which is composed of multiple candidate paths for each robot, select a path for each robot and find the control inputs along the selected paths so that the robots' dynamics constraints are satisfied, their motions are collision free, and the completion time of the set of robots is minimized. An illustrative example is shown in Figure 4 and Figure 5.

For n robots, the set of candidate path combinations is given by the Cartesian product of the sets of individual robot paths. The number of possible path combinations is $\prod_{i=1}^{n} p_i$, where p_i is the number of candidate paths for robot $\mathcal{A}_i$. For each path combination (i.e., a valid selection of paths for the set of robots), we have a corresponding MINLP problem (as in Section IV). A more efficient approach than solving the MINLP problems individually is to model the roadmap coordination problem as a single MINLP problem by using additional binary variables that denote whether each candidate path is selected or not. Integer programming techniques, such as branch-and-bound and branch-and-cut, can prune the search space effectively to more efficiently solve the roadmap coordination problem.

Definition 7.1: An *agent* A_I is a robot-path tuple $(\mathcal{A}_i, \mathcal{P}_I)$ that represents robot $\mathcal{A}_i$ and a valid path $\mathcal{P}_I$. Here index I depends on i and p_i.

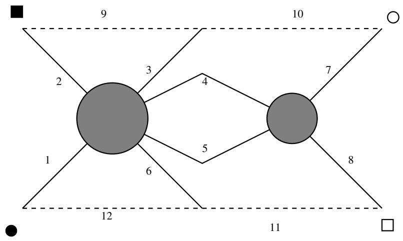


Fig. 4. Topology of the modified Bridges of Konigsberg problem: solid lines are bridges, dotted lines are roads on the land, and the two disks are islands. The problem has one rectangular robot and one circular robot, and each robot must traverse every bridge at least once. Each robot has a start position (black) and an end position (white). Each robot has multiple candidate paths, and the two robots cannot meet at the same time on any bridge.

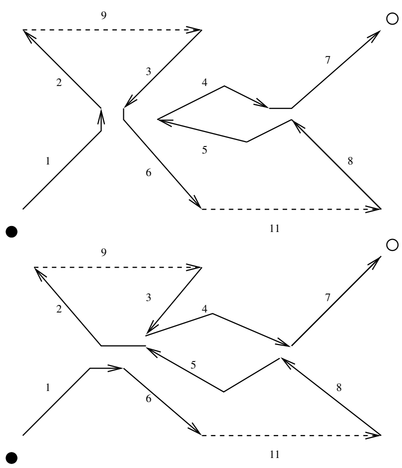


Fig. 5. Two candidate paths for the circular robot from Figure 4.

We introduce parameter ϕ_{IJ} for each pair of agents A_I and A_J.

$$\phi_{IJ} = \begin{cases} 1, & \text{if } \mathcal{A}_i = \mathcal{A}_j, \mathcal{P}_I \neq \mathcal{P}_J \\ 0, & \text{if } \mathcal{A}_i \neq \mathcal{A}_j \end{cases}$$

We also introduce binary variables χ_I to denote whether A_I is chosen. Let A be the set of agents, and the total number of agents is $N = |A| = \sum_{i=1}^{n} p_i$. The *path selection constraints* ensure that each robot choose only one of its candidate paths, which can be written as:

$$(\sum_{A_J \in A} (\phi_{IJ}\chi_J)) + \chi_I = 1 \text{ for each } A_I$$

The resulting MINLP formulation for the optimal continuous velocity schedule can be described by the previously described four sets of constraints (Section IV), of which only the first set must be modified as follows. The first set of constraints include the completion time constraints, traversal time constraints, collision avoidance constraints, and path selection constraints as shown below:

Minimize C_{max}
subject to:
Completion time constraints
$C_{max} \geq t_{I,last} + \tau_{I,last} - (1 - \chi_I)M \quad \text{for} \quad I = 1, \ldots, N$
Segment traversal time constraints
$t_{I(k+1)} = t_{Ik} + \tau_{Ik}$
$\tau_{Ik}^{max} \geq \tau_{Ik} \geq \tau_{Ik}^{min}$
Collision avoidance constraints
$t_{Jh} - t_{I(k+1)} + M(1 - \delta_{IJkh}) + (\phi_{IJ} + 2 - \chi_I - \chi_J)M \geq 0$
$t_{Ik} - t_{J(h+1)} + M\delta_{IJkh} + (\phi_{IJ} + 2 - \chi_I - \chi_J)M \geq 0$
$t_{Ik} \geq 0$
$\delta_{IJkh} \in \{0, 1\}$
Path selection constraints
$(\sum_{A_J \in A} (\phi_{IJ}\chi_J)) + \chi_I = 1$ for each agent A_I
$\chi_I \in \{0, 1\}$

The completion time constraint is active if and only if $\chi_I = 1$. The collision avoidance constraint is enforced if and only if $\chi_I = \chi_J = 1$, i.e., A_I and A_J are both selected.

VIII. Implementation and Examples

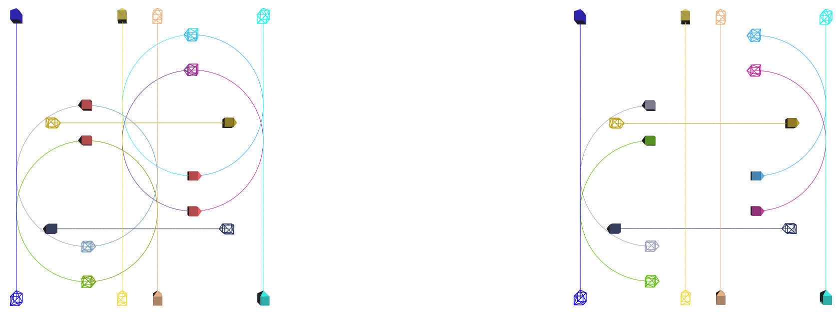
We conducted several experiments on multiple robot coordination problems. The MINLP models (MINLP-I, MINLP-II, and MINLP-III) are implemented in AMPL [13]. The resulting models were solved by using the solver MINLP [25], which is a nonlinear programming solver that integrates a sequential quadratic programming algorithm and a branch-and-bound strategy for solving mixed integer nonlinear programming problems. All MINLPs are solved on a Linux workstation with 2 GB memory and Intel XEON 2.00 GHz CPU.

The robots have specified initial positions, and each robot is required to move to a specified goal position. The collision zones are computed using the PQP collision detection package (Larsen et al. [21]).

We have two example test sets. For the first set, each robot has a fixed path. The second set illustrates coordination on a roadmap, where each robot may have a set of candidate paths and must select one path. (See Figure 4 and Figure 6.) Example animations can be seen at `www.cs.rpi.edu/~sakella/convex/`.

Example run times and objective values are in Table I. For all problems, MINLP-I, MINLP-II, and MINLP-III found an optimal solution, which implies that solving convex MINLP-I and MINLP-II found the global optimum solution, and solving nonconvex MINLP-III found at least a local optimum solution. Furthermore, MINLP-II and MINLP-III have the same objectives for every example, which shows that solving MINLP-III gives a global optimum solution by Theorem (6.8). Also we notice MINLP-II and MINLP-III have smaller objectives than those of MINLP-I for every example, which is consistent with Assumption (6.2) being less restrictive than Assumption (6.1).

Solving MINLP-I, MINLP-II, or MINLP-III gives a feasible coordination schedule, from which we obtain the traversal time τ along each robot segment. For the double integrator, we use its simple dynamics to generate a velocity profile consistent with traversal time τ. We can



(a) (b)

Fig. 6. Coordination on a roadmap with 10 car-like robots on constant-curvature paths. Case (a): Before coordination, the four robots with circular paths each have 2 candidate semi-circle paths. Case (b): After coordination, each robot moves along its selected path.

Num. of robots	Num. of collision zones	Collision time (secs)	Num. of binary variables	MINLP-I time (secs)	MINLP I Objective	MINLP-II time (secs)	MINLP II and III Objective	MINLP-III time (secs)
5	13	17.64	20	10.64	66.01	13.54	52.38	16.09
8	42	52.63	64	258.92	101.30	111.93	87.64	348.16
10	71	84.87	102	8106.44	106.11	9921.31	90.92	6849.31
8 (radial, unsymm.)	32	62.89	54	2232.33	82.54	7008.66	64.67	6206.54
8 (radial, symm.)	29	28.4	54	3770.68	95.93	2317.94	51.39	2795.99
12 (SCC)	64	118.94	85	3472.45	118.78	1226.98	108.47	1684.49
16 (rectangle)	64	102.82	128	11270.6	63.63	12141.9	57.41	9613.26
10 (4 with 2 paths)	46	11.62	76	142.42	60.70	760.84	52.24	788.23
2 (with 3 paths)	156	40.79	207	50.93	291.73	31.57	253.75	27.06

TABLE I

SAMPLE RUN TIMES AND OBJECTIVES FOR MINLP-I, MINLP-II, AND MINLP-III (AMPL PRESOLVE TIMES ARE NOT INCLUDED). NOTE MINLP II AND III HAVE THE SAME OBJECTIVES FOR EVERY EXAMPLE, WHICH IS CONSISTENT WITH THEOREM (6.8) AND THEOREM (6.9). MINLP-II AND MINLP-III HAVE SMALLER OBJECTIVES THAN THOSE OF MINLP-I, WHICH IS CONSISTENT WITH ASSUMPTION (6.2) BEING WEAKER THAN ASSUMPTION (6.1).

analytically show that one of four bang-off-bang velocity profiles is a feasible trajectory.

Our experiments show the MINLP solve times are large. With current solvers, directly solving for the global optimum may be best for use in off-line scheduling.

IX. CONCLUSION

We have developed MINLP models to find the global minimum time solution for collision-free coordination of multiple double integrator robots along specified paths or on a roadmap. The robots have state (velocity) constraints and control constraints. The MINLPs (actually, their relaxations) are proved convex under certain conditions. Therefore we are guaranteed to find the global optimal solutions for these coordination problems. The MINLP models built in this paper provide, to the best of our knowledge, one of the few, if not the only, results on global optimum coordination for multiple robot with dynamics constraints.

There are several directions for future work. While we have focused on reducing the completion time, other convex objective functions, such as the total execution time or average completion time, may be optimized. Additionally, piecewise linear or convex quadratic objective functions may also be solved. Future work on faster solution methods for the MINLP can explore interior point methods and cone programming. Furthermore, if only an approximate solution is needed, we can apply separable programming algorithms to obtain arbitrary ϵ-approximation solutions by solving mixed integer linear programs.

An immediate next step, due to the similarity of their dynamics to double integrators, is extension to Cartesian manipulators, and to car-like robots and simplified planar aircraft models ([4]) on continuous curvature paths. Aircraft have minimum velocity constraints and incorporating those will be useful for air traffic control applications. Automatic modification of robot paths to reduce completion time would be a useful extension.

ACKNOWLEDGMENTS

This work was supported in part by NSF under CAREER Award No. IIS-0093233. Thanks to Jong-Shi Pang, John

Mitchell, and John Wen for advice. Sven Leyffer provided the MINLP solver and Thierry Fraichard provided code to generate continuous curvature paths. Andrew Andkjar implemented animation software that interfaced with the PQP software from the University of North Carolina, and Mayur Patel created a convenient interface.

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

[5] 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.

[6] 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.

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

[8] 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.

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

[10] M. A. Duran and I. Grossmann. An outer approximation algorithm for a class of mixed integer nonlinear programs. *Mathematical Programming*, 36:307–339, 1986.

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

[12] R. Fletcher and S. Leyffer. Nonlinear programming without a penalty function. Technical report, Department of Mathematics and Computer Science, University of Dundee, Dundee DD1 4HN, UK, September 1997.

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

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

[15] R. Ghrist, J. O'Kane, and S. LaValle. Pareto-optimal coordination on roadmaps. In *Sixth Workshop on the Algorithmic Foundations of Robotics*, Utrecht/Zeist, The Netherlands, July 2004.

[16] J. E. Hopcroft, J. T. Schwartz, and M. Sharir. On the complexity of motion planning for multiple independent objects: PSPACE-hardness of the "warehouseman's problem". *International Journal of Robotics Research*, 3(4):76–88, 1984.

[17] 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.

[18] Y. Kim, M. Mesbahi, and F. Y. Hadaegh. Dual-spacecraft formation flying: Optimal collision-free reconfigurations. *Journal of Guidance, Control, and Dynamics*, 26(2):375–379, 2003.

[19] 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.

[20] 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.

[21] 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.

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

[23] 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.

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

[25] S. Leyffer. Integrating SQP and branch-and-bound for mixed integer nonlinear programming. *Computational Optimization and Applications*, 18:295–309, 2001.

[26] 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.

[27] P. Ogren. Split and join of vehicle formations doing obstacle avoidance. In *IEEE International Conference on Robotics and Automation*, pages 1951–1955, New Orleans, LA, Apr. 2004.

[28] 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.

[29] J. Peng. *Coordinating Multiple Robots: A Mathematical Programming Approach*. PhD thesis, Department of Mathematical Sciences, Rensselaer Polytechnic Institute, 2005. Forthcoming.

[30] J. Peng and S. Akella. Coordinating multiple robots with kinodynamic constraints along specified paths. In J.-D. Boissonnat, J. Burdick, K. Goldberg, and S. Hutchinson, editors, *Algorithmic Foundations of Robotics V*, pages 221–237. Springer-Verlag, Heidelberg, Germany, 2003.

[31] J. Peng and S. Akella. Coordinating the motions of multiple robots with kinodynamic constraints. In *IEEE International Conference on Robotics and Automation*, pages 4066–4073, Taipei, Taiwan, Sept. 2003.

[32] J. Reif and M. Sharir. Motion planning in the presence of moving obstacles. In *Proceedings of the 26th Annual Symposium on the Foundations of Computer Science*, pages 144–154, Portland, Oregon, Oct. 1985.

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

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

[35] G. Sanchez and J.-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.

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

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] 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.

[42] H. Tanner, A. Jadbabaie, and G. J. Pappas. Stable flocking of mobile agents, part I: Fixed topology. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, pages 2010–2015, Maui, HI, Dec. 2003.

[43] 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.