Decentralized Multi-Vehicle Path Coordination under Communication Constraints

Pramod Abichandani, Hande Y. Benson, and Moshe Kam

***Abstract*— We present a mathematical programming based decentralized framework to generate time optimal velocity profiles for a group of path constrained mobile vehicle robots subject to communication connectivity constraints. Each vehicle robot starts from a fixed start point and moves towards a goal point along a fixed path so as to avoid collisions with other robots, and remain in communication connectivity with other robots. The main contribution of this paper is the discrete time decentralized Receding Horizon Mixed Integer Nonlinear Programming (RH-MINLP) formulation of the multi-vehicle path coordination problem with constraints on kinematics, dynamics, collision avoidance, and communication connectivity, and the application of state-of-the-art MINLP solution techniques. We test scenarios involving up to ten (10) robots to demonstrate (i) the effect of communication connectivity requirements on robot velocity profiles; and (ii) the dependence of the solution computation time on communication connectivity requirements.**

I. INTRODUCTION

The coordination of the motion of a number of robots (say n) in a shared workspace so that they avoid collisions is known as the *multiple robot path coordination problem* [1], [2]. In previous work [3], [4], we addressed this problem under communication connectivity constraints by generating time optimal velocity profiles. The robots were confined to fixed paths and sought to arrive from a set of initial points to specified final destinations. The communication constraints in [3] ensured that each robot remains in immediate communication range with n_{conn} other robots, ($n_{conn} = 1, \ldots, n-1$). The Network Partition Elimination constraints in [4] eliminated partitions in the group by maintaining a fully connected communication network using intermediary robots as relays. The formulations in [3] and [4] were centralized in nature - all calculations were made by a single central coordinator. In [4], we studied the relationship between the number of robots in a given scenario and the solution computation times.

The high computational requirements of the centralized formulation motivate the study of a decentralized alternative. In addition, in many scenarios, new information about the environment becomes available to the robots as they progress, thereby requiring changes in the coordination scheme. In such cases, a decentralized receding horizon planning strategy is particularly attractive as it allows the robots to quickly resolve and communicate their plan with other robots in the group.

In this paper, we present a sequential, decentralized framework to solve the multi-vehicle path coordination problem under communication constraints. Each vehicle robot solves only its own coordination problem, formulated as a RH-MINLP problem subject to constraints on kinematics, dynamics, collision avoidance, and communication connectivity. The sequential nature of the formulation forces each robot to solve its own problem in a predetermined order and exchange trajectory information with other robots. In this way, each robot can take into account the latest trajectory of other robots while planning its own trajectory.

The fixed paths of the robots are represented by

Pramod Abichandani is a doctoral candidate at the Data Fusion Laboratory, Electrical and Computer Engineering Department, Drexel University, Philadelphia, PA 19104, USA pramod@drexel.edu

Hande Y. Benson, Ph.D., is an Associate Professor at the Decision Sciences Department, Drexel University, Philadelphia, PA 19104, USA hvb22@drexel.edu

Moshe Kam, Ph.D., is the Department Head and Robert Quinn Professor of Electrical and Computer Engineering, Drexel University, Philadelphia, PA 19104, USA kam@minerva.ece.drexel.edu

piecewise cubic spline curves. The feasibility criteria for trajectories require that the robots' kinematic and dynamic constraints be satisfied, along with the imperatives of avoiding collisions and obeying the communication connectivity constraints. The receding horizon formulation is implemented in MATLAB, which is interfaced with the MINLP solver MILANO [5]. MILANO is a MATLAB-based solver for mixed-integer linear and nonlinear programming problems. It uses a branch-and-bound method for handling integer variables, and an interior-point method for solving the nonlinear relaxations. Source code for MILANO has been made available online [6].

The following assumptions are made:

- Fixed path coordination and motion planning is fully decentralized.
- While robots are required to maintain a certain level of communication connectivity with their immediate neighbors, there is a central communication station that allows robots to communicate trajectory data to all the robots in the group and keep individual clocks synchronized.
- We assume that for each vehicle, there exists a perfect lower level control to achieve the planned position and heading angle without any time delay.
- There is no slipping in the motion of the robots. This is a valid assumption since we enforce kinematic (non-holonomic constraints, continuous first and second derivatives of the spline paths) and dynamic constraints (upper bounds on the velocities and accelerations) that disallow such slippage.
- Free space propagation between vehicle antennas is assumed. Other factors such as multi path propagation, fading, time delay, and crosstalk introduce uncertainty in the problem and are not considered here. Dealing with uncertainty will be subject of additional studies.

II. RELATED WORK

While a significant body of work was devoted to path planning for mobile robots (e.g., [1], [7]), we focus our work on the relatively untouched area of velocity planning along predetermined routes.

Several centralized approaches have been used to address the problem of path coordination of multiple robots [1], wherein multiple robots with fixed paths coordinate with each other so as to avoid collisions and reach destination points. These approaches include the use of coordination diagrams [8], constrained configuration space roadmap [9], grouping robots with shared collision zones into subgroups [2]. In [10], mixed integer linear programming (MILP) formulations were used to generate continuous velocity profiles for a group of robots that satisfy kinodynamics constraints, avoid collisions and minimize the task completion time. In [3] and [4] we extended this body of work by addressing communication requirements of the problem and using state of the art interior-point methods to solve the resulting nonlinear programming (NLP) problem.

Decentralized formulations for trajectory generation appear in [11] where the authors present a decentralized approach for multi-robot motion planning using potential field controllers based on navigation functions, [12] where the authors present a a decentralized nonlinear model predictive control (NMPC) formulation for trajectory generation of helicopters that includes a potential function that reflects state information of other moving vehicles into the cost function of the individual vehicle, and [13] where a decentralized algorithm for multiple aircraft coordination based on an iterative bargaining scheme is presented. Most closely related to our work is [14], where the authors present a decentralized receding horizon formulation for multiple aircraft path planning using mixed integer linear programming (MILP) to generate provably safe trajectories. Similar to [14], we use a sequential decision ordering mechanism as a part of our solution algorithm.

III. PROBLEM FORMULATION

A. Robot Motion and Path

Consider a two wheeled differential drive mobile robot shown in Fig. 1. The robot moves in a global (X, Y) Cartesian co-ordinate plane and is

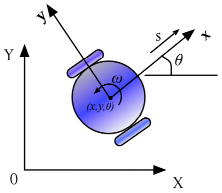

Fig. 1. Robot architecture and notations.

represented by the following kinematic model with associated non-holonomic constraint (that disallows the robot from sliding sideways).

$$\dot{x} = s\cos(\theta); \quad \dot{y} = s\sin(\theta); \quad \dot{\theta} = \omega \tag{1}$$

$$\dot{x}\sin(\theta) - \dot{y}\cos(\theta) = 0. \tag{2}$$

Here s and ω are the linear and angular speeds of the robot respectively; x, y and θ are the coordinates of the robot with respect to the global (X, Y) coordinate system.

Consider a group of n such mobile robots. Each robot $i = 1, 2, ..., n$ is represented by a common mathematical model (1) with associated non-holonomic constraint (2) and has a given start (origin) point o^i and a given end (goal) point e^i. O is the set of all start (origin) points. $o^i \in O, \forall i = 1, 2, ...n$. E is the set of all end points. $e^i \in E, \forall i = 1, 2, ...n$. The Euclidean distance between two robots i and j is denoted by d^{ij}. The robots are required to maintain a minimum safe distance d_{safe} from each other in order to avoid collisions. At any given discrete time step, the distance between the current location and the goal point for robot i is given by d^i_{goal}. s^i and ω^i denote the speed and angular velocity respectively of robot i along its path at a given time.

Each robot i follows a fixed path represented by a two dimensional piecewise cubic spline curve of length U^i, which is obtained by combining two one dimensional piecewise cubic splines $x(u)$, and $y(u)$, where the parameter u is arc length along the curve. Let $\kappa(u)$ be the curvature along the spline curve. For each robot i,

$$\omega^i(u) = s^i(u)\kappa^i(u) \tag{3}$$

These piecewise cubic splines have continuous first derivatives (slope) and second derivatives (curvature) along the curve. This property makes the path kinematically feasible. Furthermore, upper and lower bounds on the speed, acceleration and angular speed (turning rate) are enforced, thereby taking into account the robot dynamics. The paths represented by the two dimensional piecewise cubic splines along with the constraints on the speed, accelerations and turn rates result in a kinodynamically feasible trajectory. For a detailed discussion on spline curve design and analysis, see [15] and its references. Other path primitives that result in twice continuously differentiable functions in our constraints such as quintic curves, polar splines, cubic spirals can be easily accommodated using this framework.

B. Communication Model

Each robot is equipped with a wireless transceiver node. Consider two robots that try to communicate with each other at a given point in time. The Euclidean distance between them is denoted by d. The signal transmission power of the wireless node placed on the transmitter robot is denoted by P_{tr}. The received signal power of the wireless node placed on the receiver robot is denoted by P_r. The power experienced by the receiver robot node is calculated using Friis's equation [16]

$$P_r = P_{tr}G_tG_r\left(\frac{\lambda}{4\pi d}\right)^{\alpha} \tag{4}$$

where α is the path loss exponent. The noise σ is assumed to be thermal ($kTBF$). λ is the wavelength and is equal to c/f, where c = 3 x 10^8 m/s and f = 2.4 x 10^9 Hz. The values of G_t and G_r (antenna gains) are assumed here to be 1. The values of the path-loss exponent α range from 1.6 (indoor with line of sight) to 6 (outdoor obstructed) depending on the environment.

The Signal to Noise Ratio (SNR) experienced by the receiver robot is calculated using the relationship SNR $= P_r/\sigma$ to determine whether the robots are in communication range of each other. If the SNR experienced by a receiver node placed on a robot is above a predefined threshold η_c, the two robots are considered to be in communication range of each other. For a known P_{tr} and α, the condition SNR $\geq \eta_c$ can be modified appropriately as $d \leq \eta_d$, where η_d defines the maximum communication range beyond which path-loss results in loss of communication.

C. Receding Horizon

The parameter t represents steps in time. T_{hor} is the receding horizon time. At each time step t, each robot must calculate its plan for the next T_{hor} time steps, and communicate this plan with other robots in the network. While the robots compute their trajectory points and corresponding input commands for the next T_{hor} time steps, only the first of these solutions is implemented, and the process is repeated at each time step. T_{max} is the time taken by the last arriving robot to reach its end point. At $t = T_{max}$ the scenario is completed. If a robot reaches the goal point before T_{max}, it continues to stay there till the mission is over. However, if required, it can still communicate with other robots.

Each robot plans its own trajectory at each discrete time step, by taking into account the plans of all other robots. For a given time step t, each robot determines its speed for the next T_{hor} time steps starting time t i.e. $s^i(t),\ldots,s^i(t+T_{hor})$ and implements the first speed $s^i(t+1)$ out of all these speeds. In this way the plan starting at time step $t+1$ must be computed during time step t. Thus during each time step t, each robot communicates the following information about its plan $\mathscr{P}^i(t)$ to other robots: $\mathscr{P}^i(t)$ = $[\mathbf{p}^i(t)\ldots\mathbf{p}^i(t+T_{hor})]$, where $\mathbf{p}^i(t)$ = $(x^i(t), y^i(t))$ is the location of the robot i on its path at time t calculated based on the optimal speed $s^i(t)$.

D. Decision Ordering

The robots are assigned a pre-determined randomized decision order. The decentralized algorithm presented here sequentially cycles through each robot thereby allowing each robot to solve its planning problem in the order $ord(i)$, $i \in \{1,2,...,n\}$.

IV. MODEL FORMULATION

A. Optimization model

Each robot i solves the optimization problem $\mathscr{O}^i(t)$ indicated by (5)-(17) in the order $ord(i)$ that it has been assigned.

$$\text{minimize} \quad \sum_{k=t}^{t+T_{hor}} d^i_{goal}(k) \tag{5}$$

$$\text{subject to} \quad \forall j \in \{1,2,\ldots,n\}, j \neq i$$

$$\forall k \in \{t,\ldots,t+T_{hor}\}$$

$$(x^i(0), y^i(0)) = o^i \tag{6}$$

$$u^i(0) = 0 \tag{7}$$

$$u^i(k) \leq U^i \tag{8}$$

$$u^i(k) = u^i(k-1) + s^i(k)\Delta t \tag{9}$$

$$(x^i(k), y^i(k)) = ps^i(u^i(k)) \tag{10}$$

$$s_{min} \leq s^i(k) \leq s_{max} \tag{11}$$

$$a_{min} \leq a^i(k) \leq a_{max} \tag{12}$$

$$d^i_{goal}(k) = U^i - u^i(k) \tag{13}$$

$$d^{ij}(k) \geq d_{safe} \tag{14}$$

$$d^{ij}(k) \leq M(1 - C^{ij}(k)) + \eta_d \tag{15}$$

$$\sum_{j:j\neq i} C^{ij}(k) \geq n_{conn} \tag{16}$$

$$C^{ij}(k) \in \{0,1\} \tag{17}$$

B. Decision Variables

In (5)-(17), the main decision variables are the speeds, $s^i(t),\ldots,s^i(t+T_{hor})$, for robot i at time t. The values of the remaining variables are dependent on the speeds.

C. Objective Function

Equation (5) represents the objective function to be minimized. This formulation forces the robots to minimize the total distance between their current location and the goal position over the entire receding horizon. Constraint (13) defines the distance to goal $d^i_{goal}(k)$ for each robot $i = 1\ldots n$ at time-step k, $\forall k \in \{t,\ldots,t+T_{hor}\}$ as the difference between it's path length U^i and the total arc length travelled $u^i(k)$. The choice of this objective function results in the robots not stalling and moving to their goal position as fast as possible (minimum time solution).

D. Path (Kinematic) Constraints

Constraints (6)-(10) define the path of each robot. The constraints (6), (7), and (8) form the boundary conditions. Constraint (6) indicates that each robot i has to start at a designated start point o^i. Constraint (7) initializes the arc length travelled u to zero value. Constraint (8) provides the upper bound on the arc length travelled. Constraint (9) increments the arc length at each time step based on the speed of the robot ($\Delta t = 1$). Constraint (10) ensures that the robots follow their respective paths as defined by the cubic splines. The function $ps^i(u^i(k))$ denotes the location of robot i at time step k, $\forall k \in \{t,\ldots,t+T_{hor}\}$ after travelling an arc length of $u^i(k)$ along the piecewise cubic spline curves. It should be noted that the constraint (10) is a non-convex nonlinear equality constraint.

E. Speed and Acceleration (Dynamic) Constraint

Constraints (11)-(12) are dynamic constraints and ensure that the speed $s^i(k)$ (and hence, angular velocity) and the acceleration $a^i(k)$ for each robot $i = 1\ldots n$ at each time-step k, $\forall k \in \{t,\ldots,t+T_{hor}\}$ are bounded from above (by s_{max} and a_{max} respectively) and below (by s_{min} and a_{min} respectively). These constraints are determined by the capabilities of the robot and the curvature of the paths represented by the spline curve. Here we assume that the curvature of the paths is within the achievable bounds of the angular speed and radial acceleration of the robots. Hence the angular speed required by the robots corresponding to the optimal speed is always achievable, and can be determined by (3).

F. Collision Avoidance Constraint

The non-convex constraint (14) ensures that there is a sufficiently large distance d_{safe} between each pair of robots to avoid a collision at all times.

G. Communication Connectivity Constraint

Constraints (16) and (17) state that vehicle i should be in communication range of at least n_{conn} vehicles. This means that, for at least n_{conn} values of $j = 1,...,n,\ j \neq i$, the condition $d^{ij} \leq \eta_d$ should be satisfied. The remaining vehicles may or may not be in communication range of i. In order to express this requirement, we introduce a constant M and formulate constraint (15), which states that if $C^{ij}(k)$= 1 then vehicles i and j are within communication range. If $C^{ij}(k)$ = 0, then the constraint will be trivially satisfied for a sufficiently large M. Constraint (15) is an example of a big-M constraint [17].

Note: In our implementation, we used an equivalent form of constraint (16) as $\dfrac{(d^{ij}(k))^2}{(M(1-C^{ij}(k))+\eta_d)} \leq (M(1 - C^{ij}(k)) + \eta_d)$, in order to avoid the nondifferentiability of a Euclidean distance calculation within the nonlinear solver. The nondifferentiability is not going to occur at the optimal solution due to the collision avoidance constraint keeping d^{ij} sufficiently large, but during the initial iterations of the MILANO solver, it may cause numerical difficulties. The reformulation removes the potential of such an occurrence and provides numerical stability.

H. Algorithm

All robots are initially assumed to be in communication range of each other. The general outline of the algorithm is as follows:

For any time step t, let each robot i implement the following algorithm:

Start: Start at time t

- **Step 0** - An order is enforced in terms of which robot plans its trajectories first. The ordering can be randomly assigned or can be assigned a priori.
- **Step 1** - Based on its decision order $ord(i)$, each robot i solves the problem $\mathscr{O}^i(t+1)$ at time t by taking into account the following plans:
  1.1 Plans $\mathscr{P}^j(t+1)$ for robots j, $\forall j \in \{1,2,\ldots,n\}$, $j \neq i$ whose $ord(j) < ord(i)$ - these robots have already calculated their new plans, and
  1.2 Plans $\mathscr{P}^\zeta(t)$ for robots ζ, $\forall \zeta \in \{1,2,\ldots,n\}$, $\zeta \neq i$ whose $ord(i) < ord(\zeta)$ - these robots are yet to calculate their new plans.

- **Step 2** -
  - 2.1 If a feasible solution is found, the new plan is $\mathscr{P}^i(t+1)$.
  - 2.2 If $\mathscr{O}^i(t+1)$ is infeasible then use the previously available plan $\mathscr{P}^i(t)$ for the next $T_{hor}-1$ time steps i.e. the new plan

$$\mathscr{P}^i(t+1) = \mathscr{P}^i(t) \backslash \mathbf{p}^i(t) \tag{18}$$

    where $\mathbf{p}^i(t)$ = $(x^i(t), y^i(t))$
- **Step 3** - Broadcast this plan to the other robots.

End: End by $t+1$, and repeat

I. Remarks about the algorithm

In our numerical testing, we found that the decision order $ord(i)$ of the robots $i = 1, \ldots, n$ can qualitatively affect the solution of each robot depending on the geometry of the paths.

1) Due to the inherent decentralized decision making, certain robots' decisions may render the coordination problem difficult to solve for other robots. In some cases, reassigning a different decision order $ord(i)$ of the robots $i = 1, \ldots, n$ helped improve overall solutions.
2) Also in some cases, certain robots' decisions can render the coordination problem infeasible for other robots regardless of the decision ordering used. In such cases, the robots may use their plans from the previous time steps as indicated by Step 2.2 of the algorithm above.

V. SIMULATIONS AND RESULTS

We focus on studying the effect of communication constraint on the velocity profiles of the robots, and on the solution computation times.

A. Simulation Setup

The MATLAB function `spline()` was used to generate piecewise cubic splines passing through randomly generated waypoints. The spline paths are parametrized by arc length u. The optimization model, defined by (5)-(17) was implemented in MATLAB using the symbolic toolbox and the solver MILANO was used to perform numerical optimization. The MATLAB-MILANO combination was implemented on a 2.4 Ghz Intel Core 2 Duo with 4GB of main memory.

B. Simulations

We have tested our model for scenarios that include up to ten (10) mobile robots and a number of communications constraints. In the following discussion, we plot the spline curve paths of the robots with different colors indicating different robots. The parameters used in our simulations are listed in Table. I. For all the simulations the value of $\Delta t = 1$, and $T_{hor} = 5$.

1) Effect of varying n_{conn} on the velocity profiles: We vary n_{conn} between 0 to $n-1$ for $n = 2$ and 10.

- *2 robots*: For a scenario where $\eta_d = 2.5$ m, Fig. 2 shows the trajectories of the 2 robots for $n_{conn} = 0$ (no communication connectivity requirement) and $n_{conn} = 1$. It is observed, that the trajectories of both robots change as n_{conn} goes from 0 to 1. Fig. 3 shows the velocity profiles of both robots for these two scenarios. T_{max} increases from 7 to 10 as n_{conn} goes from 0 to 1.
- *10 robots*: For a scenario where $\eta_d = 4$ m, Fig. 4 shows the trajectories of the 10 robots for $n_{conn} = 0$ (no communication connectivity requirement) and $n_{conn} = 9$. The most visible trajectory changes are observed in Robot 1 and Robot 2 trajectories. T_{max} remains unchanged at 10 as n_{conn} goes from 0 to 9.

2) Effect of varying n_{conn} on solution computational time: For calculating the solution computation time T_{comp}, we measure the total time taken by all robots $i = 1 \ldots n$ to solve $\mathscr{O}^i$ for all discrete time steps till the scenario is completed. All other MATLAB-MILANO processing times are ignored.

TABLE I

PARAMETER VALUES USED FOR SIMULATIONS

d_{safe}	0.02 m	s_{min}	0	s_{max}	2 m/s
a_{min}	-1 m/s^2	a_{max}	0.5 m/s^2	M	10

TABLE II

T_{comp} IN SECONDS FOR VARIOUS SCENARIOS

	n_{conn}			
n	0	1	3	9
2 (η_d = 2.5 m)	21.3	23.2	-	-
6 (η_d = 3 m)	308.9	348.1	355.8	-
10 (η_d = 5 m)	983	937.2	2142.5	1093.5

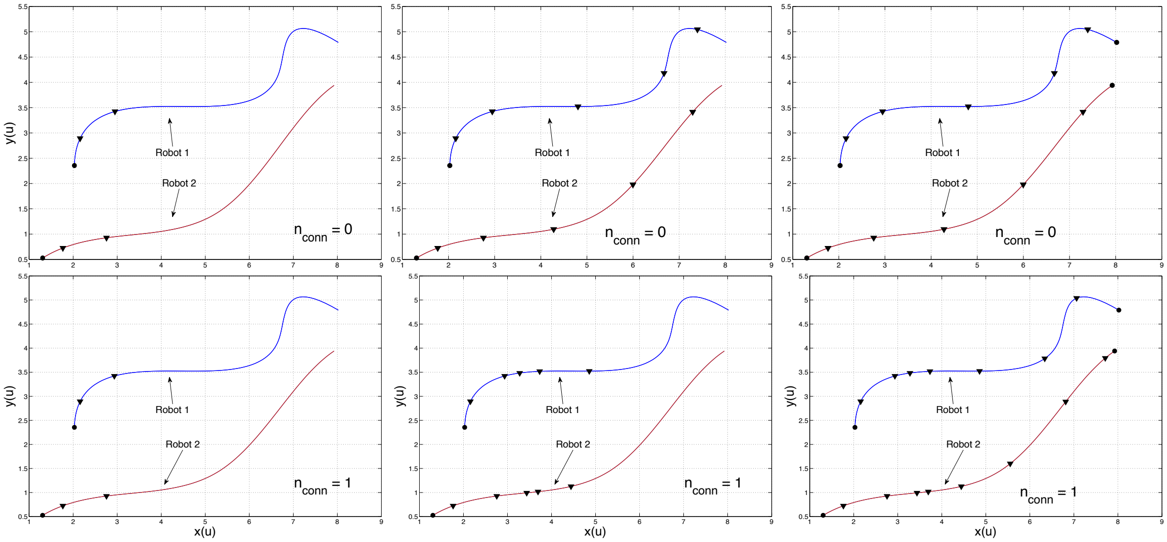

Fig. 2. The top three figures indicate robot trajectories for $n_{conn} = 0$ - both robots reach their destination by $T_{max} = 7$. The bottom three figures indicate robot trajectories for $n_{conn} = 1$. $T_{max} = 10$ for this case. Cartesian coordinate plane units are in meters. The origin and destination points of each robot are indicated by a dot marking. The triangular markings on the curves indicate the positions of the robot at each step in time while traveling at optimal speeds along the path.

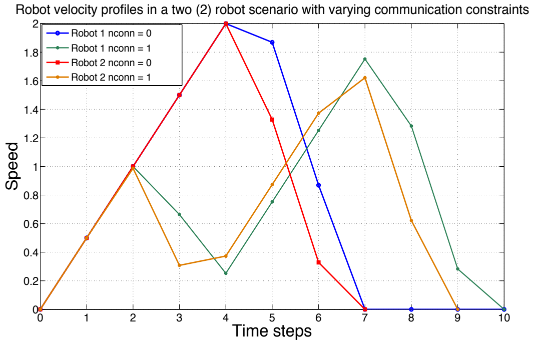

Fig. 3. Velocity profiles of robots in a 2 robot scenario with varying communication constraints

The results are listed in Table II. It is observed that for the scenarios where $n = 2$ and $n = 6$, T_{comp} steadily increases with an increase in n_{conn}. However, in the case when $n = 10$, T_{comp} varies unpredictably. As, mentioned earlier in subsection IV-I, this is due to the fact that certain robots' decisions may render the coordination problem difficult for other robots. In such cases, the value of T_{comp} can increase drastically as observed in the case when $n = 10$ and $n_{conn} = 3$ ($T_{comp} = 2142.5$ seconds).

MILANO issues infeasibility certificates if a robot's problem has no feasible solution. When such a situation is encountered, reassigning a different decision order $ord(i)$ of the robots $i = 1, \ldots, n$ can result in improved T_{comp} values. In the case where $n = 10$ and $n_{conn} = 9$, a decision order different than the case where $n = 10$ and $n_{conn} = 3$ was used. It is observed that the resulting value of T_{comp} improves considerably ($T_{comp} = 1093.5$ seconds).

In all plots, the triangular markings on different paths do not overlap with each other completely at any point in time. This observation indicates that the robots indeed do not collide with each other at any point in time (satisfying the collision avoidance constraint at all times).

VI. CONCLUDING REMARKS

A decentralized RH-MINLP formulation for solving multi-vehicle path coordination problem under communication constraints was presented

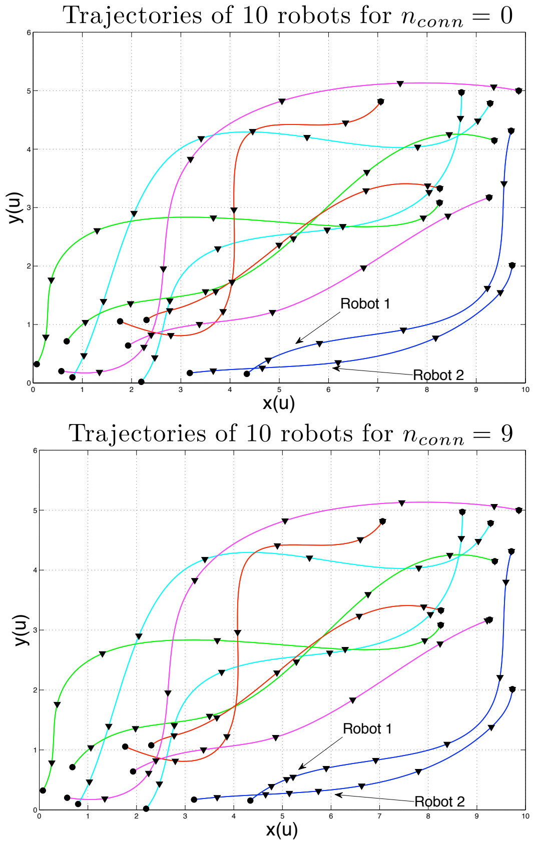

Fig. 4. A 10 robot scenario with varying communication constraints

and solved using the solver MILANO, which implements an interior-point method within a branch-and-bound framework. The effect of communication constraints on the robot velocity profiles and solution computation times were reported. As MINLP solution algorithms and associated numerical solvers continue to develop, we anticipate that MINLP solution techniques will be applied to an increasing number of multi-vehicle motion planning problems. Future work will focus on incorporating uncertainty due to communication requirements and using techniques from stochastic programming to solve the resulting formulations.

REFERENCES

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

[2] T. Simeon, S. Leroy, and J. Laumond, "Path coordination for multiple mobile robots: a resolution-complete algorithm," *IEEE Trans. on Robotics and Automation*, vol. 18, no. 1, pp. 42–49, 2002.

[3] P. Abichandani, H. Y. Benson, and M. Kam, "Multi-vehicle path coordination under communication constraints," in *Proc. American Control Conference*, WA, 2008.

[4] ——, "Multi-vehicle path coordination in support of communication," in *Proc. Int. Conf. on Robotics and Automation*, Japan, 2009.

[5] H. Y. Benson, "Using interior-point methods within an outer approximation framework for mixed-integer nonlinear programming," *IMA-MINLP Issue*, to appear; http://www.pages.drexel.edu/~hvb22/Benson-minlp.pdf.

[6] ——, "Milano: Mixed-integer linear and nonlinear optimizer." [Online]. Available: http://www.pages.drexel.edu/~hvb22/milano

[7] E. Todt, G. Raush, and R. Sukez, "Analysis and classification of multiple robot coordination methods," in *Proc. of the IEEE Int. Conf. on Rob.and Automation, CA*, 2000.

[8] P. O'Donnell and T. Lozano-Perez, "Deadlock-free and collision-free coordination of two robot manipulators," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 1989.

[9] S. LaValle and S. Hutchinson, "Optimal motion planning for multiple robots having independent goals," *IEEE Trans. on Robotics and Automation*, vol. 14, no. 6, pp. 912–925, 1998.

[10] J. Peng and S. Akella, "Coordinating multiple robots with kinodynamic constraints along specified paths," *The Int. J. of Robotics Research*, vol. 24, no. 4, pp. 295–310, 2005.

[11] G. A. S. Pereira, A. K. Das, V. Kumar, and M. F. M. Campos, "Decentralized motion planning for multiple robots subject to sensing and communication constraints," in *Proc. of the Second MultiRobot Systems Workshop*. Kluwer Academic Press, 2003, pp. 267–278.

[12] D. Shim, H. J. Kim., and S. Sastry, "Decentralized nonlinear model predictive control of multiple flying robots in dynamic environments," in *Proc. 44th IEEE Conference on Decision and Control (CDC03)*, Maui, HI, Dec 2003.

[13] G. Inalhan, D. Stipanovic, and C. Tomlin, "Decentralized optimization, with application to multiple aircraft coordination," in *Proc. IEEE Conf. on Decision and Control*, NV, 2002.

[14] T. Schouwenaars, J. How, and E. Feron, "Decentralized cooperative trajectory planning of multiple aircraft with hard safety guarantees," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, RI, 2004.

[15] M.Lepetic, G. Klancar, I. Skrjanc, D. Matko, and B. Potocnik, "Time optimal path planning considering acceleration limits," *Robotics and Autonomous Systems*, vol. 45, pp. 199–210, 2003.

[16] H. Friis, "A note on a simple transmission formula," *In Proceedings of the IRE*, vol. 34, no. 5, pp. 254–256, May 1946.

[17] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, pp. 407–427, 1999.