Multi-Vehicle Path Coordination under Communication Constraints

Pramod Abichandani, Hande Y. Benson, and Moshe Kam

Abstract— **We generate time-optimal velocity profiles for a group of path-constrained vehicles with fixed and known initial and goal locations. Each vehicle robot must follow a fixed path, arrive at its goal as quickly as possible (or at least not increase the time for the last robot to arrive at its goal) and stay in communication with other robots in the arena throughout its journey. We seek to solve this multi-objective optimization problem by generating optimal velocities along the paths. The problem is formulated as a nonlinear programming problem (NLP) with constraints on the kinematics, dynamics, collision avoidance and communication. Solutions demonstrate the trade off between the arrival time, the required transmission power and the communication connectivity requirements. Typically the optimization improved connectivity at no appreciable cost in journey time (as measured by the time of arrival of the last-arriving robot).**

I. INTRODUCTION

The coordination of the motion of n robots in a shared workspace so that they avoid collisions is known as the *multiple robot path coordination problem* [1, 2]. In this paper we study this problem under communication connectivity constraints. We plan the velocity of a group of mobile robots confined to fixed paths and seeking to arrive from a set of initial points to specified final destinations. A significant body of work was devoted to path planning for mobile robots [1], [3], but velocity planning along predetermined routes seems to be relatively untouched. However, more often than not, one does not get the liberty of planning an arbitrary path around sparse obstacles, and must rather follow a prescribed route. Here we are motivated by the additional need to maintain communication while in transit. A communication constraint forces the robots to stay within the communication range of each other. The specific problem in this paper is to generate time optimal speeds for robots in a group that moves along fixed paths in order to maintain communication with a specified number of co-travellers from the starting point till the goal point and avoid collisions. At the same time we seek to minimize the time it takes for the last-arriving robot to reach to its goal and avoid collisions. Examples of situations that may be represented by the problem of interest include military operations in an urban environment and search and rescue operation in a city.

Pramod Abichandani is a Ph.D. student at the Data Fusion Laboratory, Electrical and Computer Engineering Department, Drexel University, Philadelphia, PA 19104, USA pva23@drexel.edu

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

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

Several researchers have approached the problem of velocity planning for mobile robots along fixed paths. Kant and Zucker [4] introduced the concept of *path-velocity decomposition*, wherein the motion planning problem was broken up into two parts. The first part dealt with generating paths to avoid static obstacles, and the second part dealt with generating velocity profiles along these paths in order to avoid moving obstacles. Th. Fraichard and C. Laugier [5] extended the idea of path-velocity decomposition by introducing the concept of *adjacent paths*. Several 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 [6], constrained configuration space roadmap [7] and grouping robots with shared collision zones into subgroups [2]. In recent work, Peng and Akella [8] used mixed integer linear programming (MILP) formulations to generate continuous velocity profiles for a group of robots that satisfy kinodynamics constraints, avoid collisions and minimize the task completion time. We extend the body of work on such scenarios by addressing the problem of path coordination for multiple robots under communication constraints.

We formulate the problem as a nonlinear programming problem (NLP). The fixed paths of the robots are represented using piecewise cubic spline curves. The feasibility criterion for trajectories demands that the robots' kinematic and dynamic constraints be satisfied, along with avoiding collisions and obeying the communication constraint. The communication constraint demands that at all time, the robots be in communication range of at least k other robots, where k varies between 0 to $n-1$, n being the total number of robots. Calculations related to communication use distances and Signal to Noise ratios (SNR). Other factors such as multi path propagation, fading, time delay and crosstalk are ignored in the present exposition. The spline paths are generated using Matlab, which is interfaced with the modeling environment AMPL [9]. We use the software package LOQO [10] with AMPL to solve the NLP.

II. PROBLEM FORMULATION

A. Robot Motion and Path

Consider a two wheeled differential drive mobile robot as shown in Fig. 1. The robot moves in a global (X, Y) Cartesian co-ordinate plane and is represented by the following kinematic model with associated non-holonomic constraint (that disallows the robot from sliding sideways).

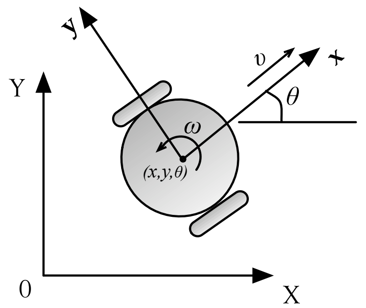

Fig. 1. Robot architecture and notations.

$$\dot{x} = vcos(\theta); \quad \dot{y} = vsin(\theta); \quad \dot{\theta} = \omega \tag{1}$$

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

Here v and ω are the linear and angular velocities 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 fixed path p^i to follow, with a given start (origin) point o^i and a given end (goal) point e^i. P is the set of all the fixed paths of each robot. $p^i(x(t), y(t)) \in P, \forall i = 1, 2, ...n$. 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} in order to avoid collisions with each other. At any given time step, the distance between the current location and the goal point for robot i is given by d^i_{goal}. s^i denotes the speed of the robot i along its fixed path at a given time. We formulate this scenario as a discrete time problem with the parameter t representing steps in time. $T_{\max}$ is the time taken by the last robot to reach its end point. At $t = T_{\max}$ the mission is complete. 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 follows a fixed path represented by a two dimensional piecewise cubic spline curve. The curve 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. These piecewise cubic splines have continuous first derivatives (slope) and second derivatives (curvature) along the curve. This property makes the path kinematically feasible for the robots. 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, readers are referred to [11] and its references. Fig. 2 shows how the spline curve paths are constructed.

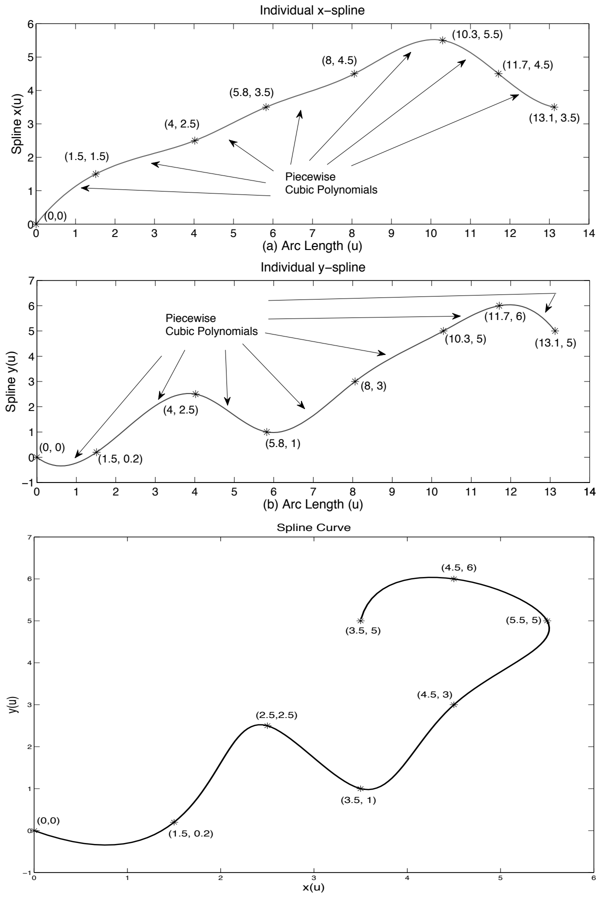

Fig. 2. Spline Curve Construction: $x(u)$ and $y(u)$ are individual cubic splines constructed where u is the arc length along the curve. The spline curve is obtained by combining $x(u)$ and $y(u)$.

B. Communication Model

Each robot is equipped with a wireless transceiver node. The communication constraint requires that at all times, every robot is in communication range of at least k other robots, where k varies between 0 to $n - 1$. The Signal to Noise Ratio (SNR) experienced by the receiver robot is calculated to determine whether or not the communication constraint is satisfied or not. If the SNR experienced by a receiver node placed on a robot is above a predefined threshold τ, the two robots are considered to be in communication range of each other.

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

$$P_r = P_{tr} G_t G_r \left(\frac{\lambda}{4\pi d} \right)^{\alpha} \tag{3}$$

where α is the path loss exponent. The noise 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) is assumed to be 1. The values of α range from 1.6 (indoor with line of sight) to 6 (outdoor obstructed) depending on the environment.

III. MODEL FORMULATION

We seek to minimize $T_{\max}$, the time of arrival of the last arriving robot, while each robot is in communication with at least k other robots at all time steps and satisfying the kinodynamic and collision avoidance constraints. The optimization is performed over the speeds of the robots along the specified paths. Since $T_{\max}$ is not known *a priori*, we pick a sufficiently large number of time steps T ($T_{\max} \leq T$) in our model so that it will yield a feasible solution. For a given value of k between 0 to $n-1$, the following problem is solved:

$$\text{minimize} \quad T_{\max} + \sigma \sum_{i,t} d^i_{goal}(t) \tag{4}$$

$$\forall i \in \{1,2,...n\}, \quad \forall t \in \{1,2,...T\}, \quad \forall j \in \{1,2,...n\}, j \neq i$$

$$\text{subject to} \quad (x^i(0), y^i(0)) = o^i \tag{5}$$

$$(x^i(T), y^i(T)) = e^i \tag{6}$$

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

$$u^i(t) = u^i(t-1) + s^i(t)\Delta t \tag{8}$$

$$(x^i(t), y^i(t)) = ps^i(u^i(t)) \tag{9}$$

$$s_{min} \leq s^i(t) \leq s_{max} \tag{10}$$

$$\dot{s}_{min} \leq \dot{s}^i(t) \leq \dot{s}_{max} \tag{11}$$

$$d^{ij}(t) \geq d_{safe} \tag{12}$$

$$0 \leq A^i(t) \leq 1 \tag{13}$$

$$A^i(t) d^i_{goal}(t) = 0 \tag{14}$$

$$\forall i, T_{\max} \geq \left(\sum_{t=0,\ldots,T} \left(1 - A^i(t)\right) \right) \tag{15}$$

$$0 \leq C^{ij}(t) \leq 1 \tag{16}$$

$$l^{ij}(t) = \text{SNR}^{ij}_r(t) - \tau \tag{17}$$

$$C^{ij}(t) l^{ij}(t) \geq 0 \tag{18}$$

$$\sum_{j:j\neq i} C^{ij}(t) \geq k \tag{19}$$

A. Decision Variables

In (4)-(19), the main decision variables are the speeds, $s^i(t)$, for vehicle i at time t. The values of the remaining variables are dependent on the speeds, as described in the following sections on the problem constraints.

B. Objective Function

Equation (4) represents the objective function to be minimized. The first term of the objective function is $T_{\max}$ which represents the time taken by the last robot to reach its goal point. By using a penalty parameter σ, the second term forces the robots to minimize the distance between their current location and the goal position. This term prevents the robots from stalling.

C. Path (Kinematic) Constraint

Constraints (5)-(9) define the path of each robot. Constraints (5) and (6) form the set of boundary requirements that each robot i has to start at a designated start point o^i and finish at a designated end point e^i at the end of the planning horizon. Constraint (7) initializes the arc length travelled u to zero value. Constraint (8) increments the arc length at each time step based on the speed of the robot ($\Delta t = 1$). Constraint (9) ensures that the robots follow their respective paths as defined by the cubic splines. The function $ps^i(u^i(t))$ denotes the location of robot i at time step t after travelling an arc length of u along the piecewise cubic spline curves.

D. Speed and Acceleration (Dynamic) Constraint

Constraints (10) and (11) are dynamic constraints and ensure that the speed and the acceleration respectively are bounded from above and below. These constraints are determined by the capabilities of the robot.

In general, solving an optimal path planning problem consists of finding a set of feasible pairs of linear and angular velocities that minimize a given cost function. Here, we assume that the maximum curvature of the path is within the achievable bounds of the angular velocity and radial acceleration of the robots and so the angular velocity corresponding to the optimal speed will always be achievable. Hence the overall solution will be feasible. In absence of such an assumption, by adding a constraint on the angular velocity and radial acceleration in the model, the feasible set of solutions for any given path with an associated curvature can be determined.

E. Collision Avoidance Constraint

Constraint (12) ensures that there is a sufficiently large distance between each pair of robots to avoid a collision.

F. Definition of T_{max}

As defined by constraints (13) and (14), $A^i(t)$ measures the number of time periods for which the robot is not at the destination. The equilibrium constraint (14) and the bounds on $A^i(t)$ (13) ensure that when $d^i_{goal}(t) > 0$, the value of $A^i(t) = 0$. Therefore, if $A^i(t) = 1$ for all (i,t) with $d^i_{goal}(t) = 0$, the total amount of time it takes a robot i to reach its destination can be calculated as

$$\sum_{t=0,\ldots,T} \left(1 - A^i(t)\right)$$

The equilibrium constraint (14) cannot guarantee by itself that this property will hold. However, constraint (15) specifies $T_{\max}$ as an upper bound for this sum, and equation (4) minimizes $T_{\max}$. Therefore at the optimal solution, for the last robot(s) to reach its destination, $A^i(t) = 1$ when robot i is at its destination at time t and that (15) will hold with equality.

Note that the solution obtained by including the constraints (13)-(15) is equivalent to the one obtained by using the following mixed-integer definition:

$$A^i(t) = 0 \text{ if } (d^i_{goal}(t) \neq 0) \qquad (20)$$
$$\quad = 1 \text{ if } (d^i_{goal}(t) = 0)$$

$$T\text{max} = \max_{i=1,\ldots,n} \left(\sum_{t=0,\ldots,T} \left(1 - A^i(t)\right) \right) \qquad (21)$$

It is, however, more advantageous for efficiency of the solution algorithm to solve an NLP instead of a mixed integer nonlinear programming problem (MINLP). With recent research in handling equilibrium constraints in NLPs, handling the resulting nonsmoothness is not a complicating factor in the solution process. For details on how the solver handles equilibrium constraints, see [12].

G. Communication Constraint

Constraints (16)-(19) define the requirement that each robot must be in communication with at least k other robots at all times. If there is a need for each robot to communicate with a greater number of robots (e.g., for contingency planning), the right-hand side of the constraint (19) can be increased.

Constraint (17) defines an intermediate variable, $l^{ij}(t)$, that aids in defining the communications constraint. The sign of $l^{ij}(t)$ at any given point in time indicates whether the robots i and j are within communication range of one other: $l^{ij}(t) \geq 0$ indicates that the two robots are in communication range whereas $l^{ij}(t) < 0$ indicates that the two robots are not in communication range of each other.

Constraint (18) then ensures that if there is no communication between robots i and j at time t, then the variable $C^{ij}(t)$ must necessarily equal 0. That is, if pairwise communication is lost, we have that $l^{ij}(t) < 0$ and since constraint (16) requires that the value of $C^{ij}(t) \geq 0$, the only way to satisfy constraint (18) is to have $C^{ij}(t) = 0$. If there is communication between the two robots at time t, then $C^{ij}(t)$ can take on any value between 0 and 1, inclusive, as allowed by constraint (16).

Finally, constraint (19) ensures that for each robot i at time t, at least k of the $C^{ij}(t)$, $j \in \{1, 2, ...n\}$, $j \neq i$ must be greater than zero. There are several issues to consider here:

- This formulation avoids the use of a binary variable to define the communications constraint. Each variable $C^{ij}(t)$ is continuous and bounded below by 0 and above by 1. Doing so greatly reduces the complexity of the problem.
- For any pair of robots that are in communication at time t, there may be an infinite number of optimal values for $C^{ij}(t)$. As an example, assume that in the optimal solution, robot i is within communication range of robots j and m. Then, as long as $C^{ij}(t) + C^{im}(t) \geq 1$, the communication constraint will be satisfied. Optimal values for $(C^{ij}(t), C^{im}(t))$ include $(0, 1)$, $(1, 1)$, $(1, 0)$, $(0.75, 0.75)$, among others. This does not constitute a difficulty for the solver, since the set of optimal solutions is bounded by constraint (16). The values of the variables can be reset to binary values after the optimal solution is found simply by observing the sign of $l^{ij}(t)$.
- The intermediate variables $l^{ij}(t)$ are provided here to simplify the exposition, but are not necessary to express the same requirements. In fact, constraints (17) and (18) can be replaced by a single set of constraints of the form

$$C^{ij}(t)(\text{SNR}^{ij}_r(t) - \tau) \geq 0, j \neq i, j = 1, \ldots, n. \qquad (22)$$

IV. SIMULATIONS AND RESULTS

A. Simulation Setup

Paths for each of the robots were generated randomly in Matlab, using 6 waypoints for each robot. The function `spline()` was used to generate piecewise cubic splines passing through the waypoints, parametrized by arc length u. Finally the individual splines were combined to generate the spline curve.

The optimization model, defined by (4)-(19) was implemented in the modeling environment AMPL and the solver LOQO was used. The AMPL-LOQO combination solves all problems discussed below in under 1 minute of real-time on a PC running RedHat Linux 2.4.20-8 with 512MB of main memory and a 2.4GHz clock speed. In our numerical testing, we have used LOQO Version 6.07 compiled with the AMPL solver interface Version 20021031.

B. Simulations

We focus on the effect of the communication constraint on the velocity profile design and on the transmission power requirements. We have tested our model using scenarios with 2, 6, and 10 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 starting (origin) point of each robot is indicated by a dot marking and the end point is indicated by a square marking. The triangular markings on the curves indicate the position of the robot in the (X,Y) Cartesian coordinate plane at each step in time while traveling at optimal speeds along the path. The parameters used in our

TABLE I

PARAMETER VALUES USED FOR SIMULATIONS

d_{safe}	0.01 m	s_{min}	0	s_{max}	2 m/s
$\dot{s}_{min}$	-1 m/s^2	$\dot{s}_{max}$	0.5 m/s^2	σ	100
α	2	f	2.4 x 10^9 Hz	τ	4.5 x 10^6

simulations are listed in Table. I. For all the simulations the value of $T = 10$ and P_{tr} is in milliwatts. 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 (thus satisfying the collision avoidance constraint at all times).

1) Effect of the communication constraint on the velocity profiles: We will start by demonstrating the effect of varying k between 0 to $n-1$ for $n = 2, 6$, and 10 on the velocity profiles.

- *2 robots*:
  For a scenario where $P_{tr} = 2.5$, Fig. 3 shows the trajectories of the 2 robots for $k = 0$ (no communication connectivity requirement) and $k = 1$. It is observed, that the trajectory of the Robot 1 changes as k goes from 0 to 1. Fig. 4 shows the velocity profiles of both the robots for scenarios when $k = 0$ and $k = 1$. For both the cases, $T_{\max} = 7$.
- *6 robots*:
  Fig. 5 shows the trajectories of the 6 robots for scenarios when $k = 1$ ($P_{tr} = 2.7$) and $k = 5$ ($P_{tr} = 10.1$). The trajectory of the Robot 1 changes with a change in the value of k. Fig. 6 shows the velocity profile of the Robot 1 for $k = 1$ and $k = 5$. For both the cases, $T_{\max} = 9$.
- *10 robots*:
  Fig. 7 shows the trajectories of the 10 robots for scenarios when $k = 0$ and $k = 9$ ($P_{tr} = 10.5$). The most visible changes in the trajectories that are observed correspond to the Robots 1 and 2. Fig. 8 shows the velocity profile of the Robots 1 and 2 for $k = 0$ and $k = 9$. For $k = 9$, Robot 1 slows down at times steps 2, 3 and 4 as compared to the case when $k = 0$ in order to maintain communication with the other robots, but speeds up during the latter part of its journey when its path is closer to the other robots. Similar behavior is observed in case of Robot 2. For both the cases, $T_{\max} = 9$.

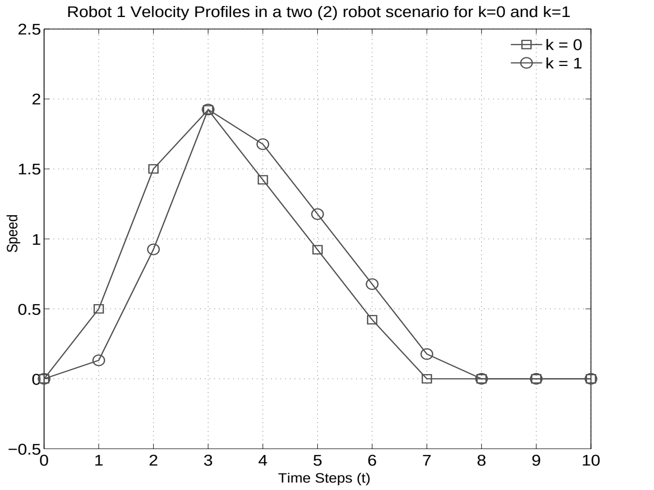

Fig. 4. Effect of communication constraint on the velocity profiles in a 2 robot scenario

From the above results, the following points are observed

- With an increase in the value of k, the velocity profile of the robot(s) change in order to satisfy the communication constraint.
- Even when the communication constraint becomes more stringent, the value of $T_{\max}$ in these examples remained the same, i.e. the cost incurred does not change. Typically, the robots whose times of arrival at their

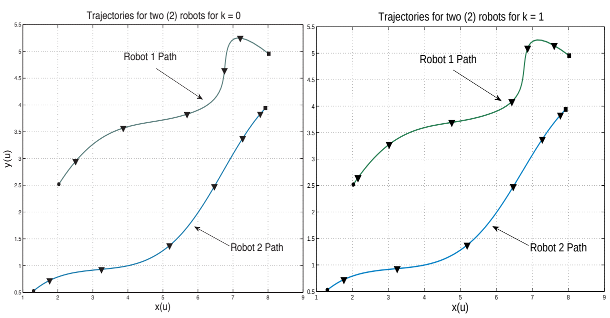

Fig. 3. A 2 robot scenario with varying communication constraint

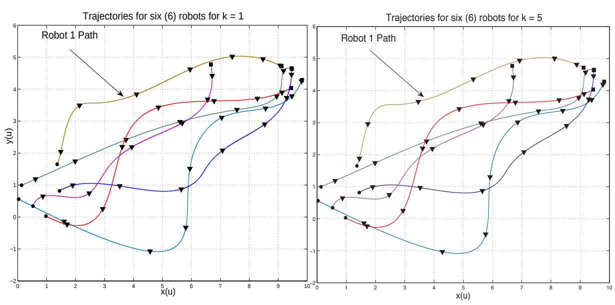

Fig. 5. A 6 robot scenario with varying communication constraint

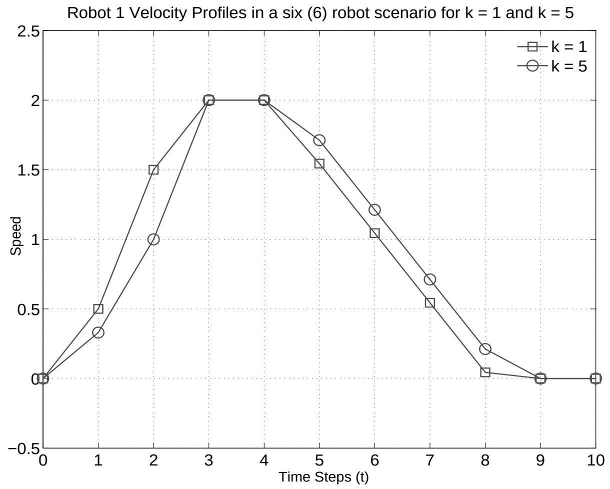

Fig. 6. Effect of communication constraint on the velocity profiles in a 6 robot scenario

respective destinations are less than $T_{\max}$ change their velocity profiles to comply with the new communication constraint, without affecting $T_{\max}$.

2) Effect of the communication constraint on transmit power: We determined the minimum transmit power required for the robots to satisfy the communication constraint for all the values of k.

Fig. 9 and Fig. 10 indicate the minimum transmit power required for the robots with paths as shown in Fig. 5 and 7 respectively, as the value of k is varied. Typically as the value of k is increased, the value of the minimum P_{tr} also needs to be increased.

3) Effect of the penalty parameter σ on the velocity profiles: We demonstrate the effect of the penalty parameter σ in the objective function for a 2 robot scenario. Fig. 11 indicates the trajectories of the two robots for $k = 1$ and $P_{tr} = 2.5$. Clearly for $\sigma = 100$, the robots are much more active as compared to the case where $\sigma = 0$. This results in them ending up closer to the destination at the penultimate time step before reaching the goal as indicated by the circles on both the plots. For both cases $T_{\max} = 7$.

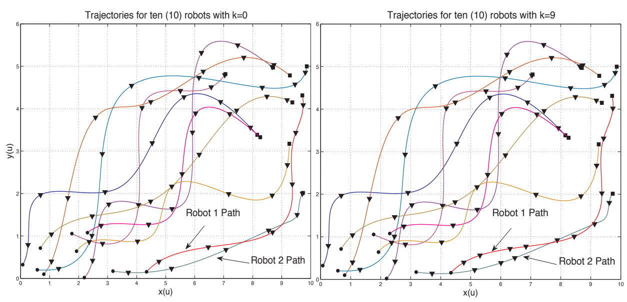

Fig. 7. A 10 robot scenario with varying communication constraint

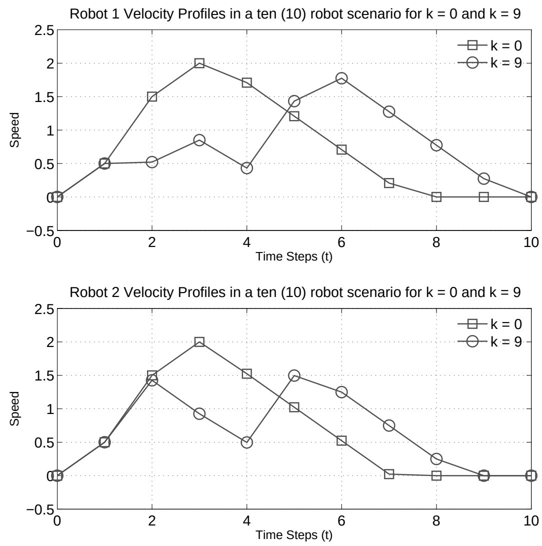

Fig. 8. Effect of communication constraint on the velocity profiles in a 10 robot scenario

Without penalty, there are infinitely many optimal solutions, each of which satisfies the constraints and has all robots reach their destination within $T_{\max}$. One such solution as depicted on the left side of Fig. 11. With the penalty, however, the robots must travel as close to the destination as possible at each time step, and in this example we show one optimal solution, which is indicated on the right hand side plot of Fig. 11.

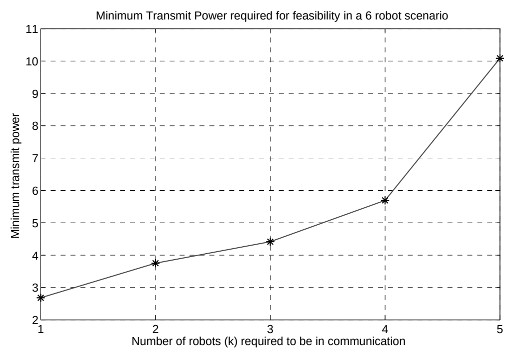

Fig. 9. Minimum power required to satisfy the communication constraint for six robots

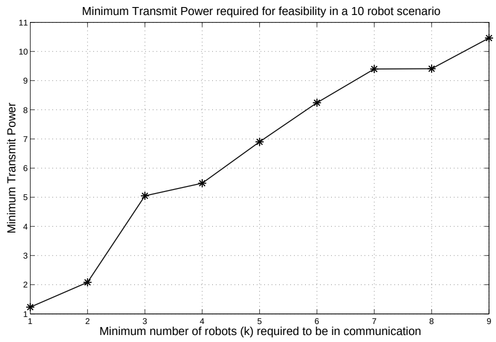

Fig. 10. Minimum power required to satisfy the communication constraint for 10 robots

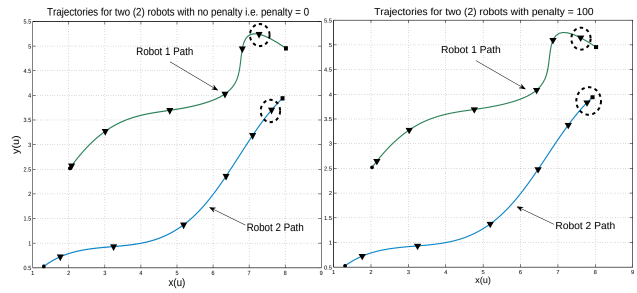

Fig. 11. A 2 robot scenario with and without the penalty term σ for $k = 1$

V. CONCLUSIONS

We generated time optimal velocity profiles for a group of mobile robots along fixed paths with kinodynamic, collision avoidance and communication constraints. We demonstrated the effect of the communication constraint on the velocity profiles and observed that in most scenarios studied by us, a change in the communication constraint requirements affected the velocity profile without deteriorating the cost incurred ($T_{\max}$). We further demonstrated that the transmit power required increases as the communication constraint becomes more demanding. Finally, we demonstrated the effect of penalizing the distances from the goal in the objective function on the velocity profiles. We observe that with a non-zero penalty term ($\sigma = 100$) in the objective function, the robots tend to move closer to the goal positions at each time step before reaching their destination as compared to cases when there is no penalty term ($\sigma = 0$).

VI. ACKNOWLEDGMENTS

Support for this work was provided by an internal summer research grant awarded through the Drexel University Provost's office. The authors thank Gabriel Ford for the useful discussions.

REFERENCES

[1] J.-C. 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 Transactions on Robotics and Automation*, 18(1):4249, 2002.

[3] E. Todt, G. Raush and R. Sukez, "Analysis and Classification of Multiple Robot Coordination Methods", *Proceedings of the International Conference on Robotics and Automation*, San Francisco, CA, 2000

[4] K. Kant and S. Zucker, "Towards efficient trajectory planning: The path-velocity decomposition.", *International Journal of Robotics Research*, 5(3):72-89, 1986.

[5] Th. Fraichard and C. Laugier, "Dynamic Trajectory Planning, Path-Velocity Decomposition and Adjacent Paths", *Proceedings of the International Joint Conference on Artificial Intelligence*, vol. 2, 1993, pp 1592-1597.

[6] P. A. O'Donnell and T. Lozano-Perez, "Deadlock-free and collision-free coordination of two robot manipulators.", *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA),* Scottsdale, AZ, 1989, pp. 484489.

[7] S. LaValle and S. Hutchinson, "Optimal motion planning for multiple robots having independent goals.", *IEEE Transactions on Robotics and Automation*, 14(6):912925,1998.

[8] J. Peng and S. Akella, "Coordinating Multiple Robots with Kinodynamic Constraints along Specified Paths", *The International Journal of Robotics Research*, vol. 24, No. 4, 2005, pp. 295-310.

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

[10] R. J. Vanderbei and D. F. Shanno, "An interior-point algorithm for nonconvex nonlinear programming." *Computational Optimization and Applications*, 13:231–252, 1999.

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

[12] H. Y. Benson, D. F. Shanno, A. Sen, and R. J. Vanderbei., "Interior-Point Methods, Complementarity Constraints, and Penalty Methods.", *Computational Optimization and Applications*, 34(2): 155-182; 2006.