A Mixed Integer Nonlinear Programming Framework for Fixed Path Coordination of Multiple Underwater Vehicles under Acoustic Communication Constraints

Solmaz Torabi, Shambadeb Basu, Hande Benson, and Pramod Abichandani

Abstract

Mixed Integer Nonlinear Programming (MINLP) techniques are increasingly used to address challenging problems in robotics, especially Multi-Vehicle Motion Planning (MVMP). The main contribution of this paper is a discrete time-distributed Receding Horizon Mixed Integer Nonlinear Programming (RH-MINLP) formulation of the underwater multi-vehicle path coordination problem with constraints on kinematics, dynamics, collision avoidance, and acoustic communication connectivity, and the application of state-of-the-art MINLP solution techniques.

Each vehicle robot starts from a fixed start point and moves toward a goal point along a fixed path, so as to avoid collisions and remain in communication connectivity with other robots. Acoustic communication connectivity constraints account for the attenuation due to signal propagation and delays arising from multi-path propagation in noisy communication environments, and specify inter-vehicle connectivity in terms of a signal-to-noise ratio (SNR) threshold. Scenarios including up to 4 robots are simulated to demonstrate (i) the effect of communication connectivity requirements on robot velocity profiles, and (ii) the dependence of the solution computation time on the communication connectivity requirement. Typically the optimization improved connectivity at no appreciable cost in journey time (as measured by the arrival time of the last-arriving robot). Results also demonstrate the responsive nature of robot trajectories to safety requirements with collision avoidance being achieved at all times despite overlapping and intersecting paths.

Index Terms

motion planning, autonomous underwater vehicles, acoustic communication mixed integer nonlinear programming, path coordination

I. Introduction

THE coordination of the motion of a number of robotic vehicles (n) in a shared workspace so that they avoid collisions is known as the *Multi-Vehicle Path Coordination (MVPC) problem* [1], [2]. In previous work, we reported on the results of simulations and experiments for real-time Receding Horizon Mixed Integer Nonlinear Programming (RH-MINLP) based decentralized MVPC under communication connectivity constraints [3], [4]. The experiments involved a group of 5 robotic ground vehicles moving along pre-determined and fixed paths while avoiding collisions and maintaining communication connectivity with the other robots. Only a handful of studies have documented practical implementations of mathematical optimization-based motion planning (e.g., [5], [6]), and to the best of our knowledge, our work in [4] presented the first experimental results for MVPC under communication connectivity constraints using MINLP.

In this paper, we extend this body of work by presenting a sequential, distributed framework to solve the MVPC problem under acoustic communication constraints for underwater vehicles. The MVPC problem becomes important when the robots are unable to follow arbitrary paths and must rather follow

S. Torabli, S. Basu, and P. Abichandani are with the Electrical and Computer Engineering Department, Drexel University, Philadelphia, PA, 19104 USA e-mail: pva23@drexel.edu.

H. Benson is an Assistant Professor in the Decision Sciences Department, LeBow College of Business, Drexel University

prescribed roadmaps or paths underwater. Example applications include autonomous ship hull monitoring and underwater distributed simultaneous localization and mapping (SLAM) [7]. It is conceivable to engage a platoon of networked underwater vehicles to autonomously image/map the hull of a ship in order to detect any unwanted objects or structural damage. Additionally, maintaining fixed path formations underwater can assist in performing distributed multi-vehicle SLAM.

For the problem at hand, each vehicle robot solves only its own coordination problem, formulated as a RH-MINLP problem subject to constraints on kinematics, dynamics, collision avoidance, and acoustic 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, and any new information, while planning its own trajectory.

Acoustic communication constraints account for the path loss experienced by a signal due to attenuation, noise in the channel, and delays arising from to multi-path propagation of signals [8] - [9]. The fixed paths of the robots are represented by 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 acoustic communication connectivity constraints. The receding horizon formulation is implemented in MATLAB, which is interfaced with the MINLP solver MILANO [10]. 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 [11]. 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 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.

II. RELATED WORK

The body of work that addresses motion planning of underwater vehicles continues to grow [12]-[13]. Approaches used to solve underwater motion planning problems include Sequential Quadratic Programming (SQP) using multi-beam forward-looking sonar images [14], fast marching algorithms [15], and incorporating ocean model predictions in trajectory planning [16]. In [17], the authors propose a nonlinear control strategy for an underwater vehicle that achieved global convergence to a reference path by explicitly taking its dynamics into account. To achieve this, the authors cast the vehicle dynamic equations into standard integrator form using nonlinear dynamic inversion and applied backstepping techniques. In [18], the authors propose decentralized coordinated path-following control laws by using Lyapunov-based techniques for a fleet of underwater vehicles. The control laws explicitly considered the topology of the inter-vehicle communication network by accounting for communication failures and delays. In [19], the authors utilize Mixed Integer Linear Programming (MILP) to generate optimal paths for underwater vehicles to perform adaptive sampling in oceans. In [20], the authors present a centralized extended Kalman filter designed for a novel navigation system that employs a Doppler sonar, depth sensors, synchronous clocks, and acoustic modems to achieve simultaneous acoustic communication and navigation. Recently, authors in [13] present theoretical and experimental results that demonstrate cooperative localization of underwater vehicles over a faulty and extremely bandwidth-limited underwater communication channel.

We extend the above body of work by focusing on the relatively untouched area of underwater fixed path coordination under acoustic communication connectivity constraints. Specifically, we use physical layer acoustic communication channel models, and state-of-the-art MINLP techniques to formulate and solve this problem by incorporating kinematics, dynamics, collision avoidance, and communication connectivity constraints.

III. PROBLEM FORMULATION

There are three basic elements of the problem formulation: robot architecture and motion, robot paths, and inter-robot communication.

A. Robot Architecture

Consider an underwater vehicle as shown in Fig. 1 [12], [21]. The vehicle has 6 degrees of freedom along its 3 body axes, and moves in a global (X, Y, Z) reference frame. The equations of motion for the vehicle are given by the following:

$$\dot{\mathbb{X}} = \mathbb{J}(\mathbb{X})\dot{q} \tag{1}$$

$$\mathbb{M}\ddot{q} + \mathbb{C}(\dot{q})\dot{q} + \mathbb{D}(\dot{q})\dot{q} + \mathbb{G}(\mathbb{X}) = \mathbb{B}(\dot{q})\mathbb{U}, \tag{2}$$

where $\mathbb{X}$ is the vector of global coordinates of the vehicle, and q is the vector of coordinates in the vehicle body frame. $\mathbb{J}(\mathbb{X})$ is the 6×6 transformation matrix. The inertia matrix $\mathbb{M}$ is represented by a sum of the rigid body matrix $\mathbb{M}_{\mathbb{R}}$ and the added mass matrix $\mathbb{M}_{\mathbb{A}}$. $\mathbb{C}(\dot{q})$ is the Coriolis matrix and $\mathbb{D}(\dot{q})$ the damping matrix. $\mathbb{G}(\mathbb{X})$ is the 6×1 vector containing restoring forces developed due to the vehicle's buoyancy and gravitational terms. $\mathbb{B}(\dot{q})$ is the controlling matrix of appropriate dimensions. $\mathbb{U}$ is the vector of control inputs.

Consider a group of n underwater vehicle robots described by (1) and (2). Each robot $i = 1, \ldots, n$ has a fixed path to follow, with a given start (origin) point o^i and a given end (goal) point e^i. O is the set of all start (origin) points. $o^i \in O, i = 1, \ldots, n$. E is the set of all end points. $e^i \in E, i = 1, \ldots, n$. 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. The distance between the current location and the goal point for robot i is given by d^i_{goal}. The speed of the robot i along its fixed path is denoted by s^i.

B. Robot Paths

Each robot follows a fixed path represented by a 3-dimensional piecewise cubic spline curve. The curve is obtained by combining three 1-dimensional piecewise cubic splines, $px(u)$, $py(u)$, and $pz(u)$, where the parameter u is the arc length along the curve. These piecewise cubic splines have continuous first derivatives (slope) and second derivatives (curvature) along the curve. The paths represented by the 3-dimensional piecewise cubic splines, along with the constraints on the speed, acceleration and turn rate, result in a kinodynamically feasible trajectory. For a detailed discussion on spline curve design and analysis, readers are referred to [22] and its references. Fig. 2 shows how the spline curve paths are constructed.

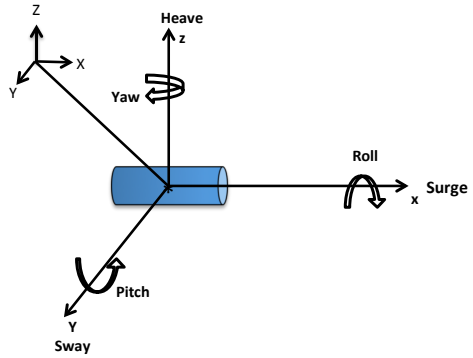

Fig. 1. Robot architecture and coordinate system.

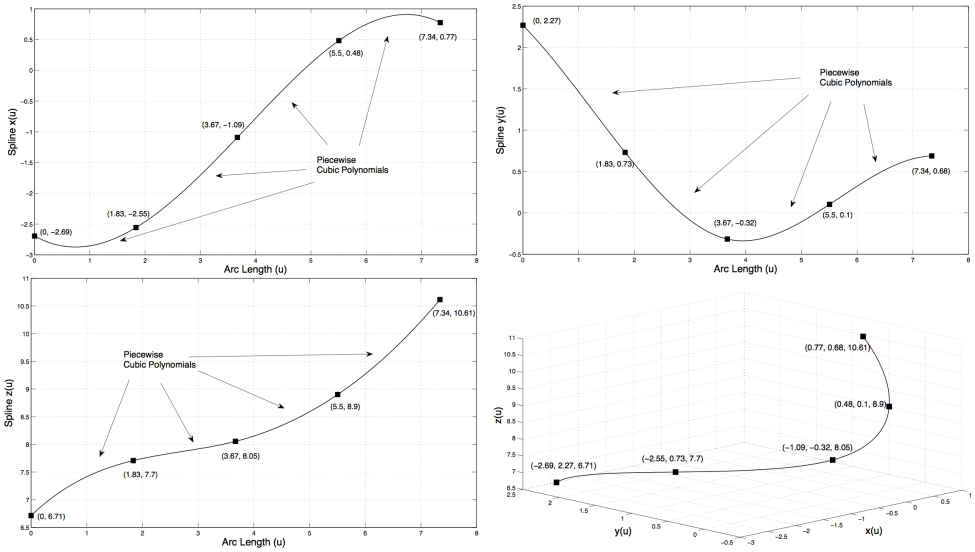

Fig. 2. Spline Curve construction: $x(u)$, $y(u)$, and $z(u)$ are individual cubic splines constructed where u is the length along the curve. The spline curve is obtained by combining $x(u)$, $y(u)$ and $z(u)$.

C. Acoustic Communication Model

Each robot is equipped with a wireless transceiver. The acoustic communication constraint requires that at all times, every robot is in communication range of at least n_{conn} other robots, where n_{conn} varies between 0 to $n-1$. These constraints account for the attenuation and multi-path propagation in noisy communication environments, and specify inter-vehicle connectivity in terms of a signal-to-noise ratio (SNR) threshold. If the SNR experienced by a receiver node placed on a robot is above a predefined threshold η_c, i.e., $SNR(l,f) \geq \eta_c$, the two robots are considered to be in one-hop communication range of each other. We refer to [8], [23], and [9] to develop our models.

1) Acoustic pathloss due to the direct line of sight (LOS) component: The acoustic path loss experienced by a signal frequency f traveling over distance l is given by

$$A(l,f) = A_0 l^{k_s} a(f)^l, \tag{3}$$

where A_0 is a constant scaling factor, and k_s is the spreading factor whose value is normally between 1 and 2. The absorption coefficient $a(f)$ is an increasing function of frequency. $a(f)$ can be expressed empirically in dB/km when f is in kHz using the following equation [8]:

$$10 \log a(f) = 0.11\frac{f^2}{1+f^2} + 44\frac{f^2}{4100+f^2} + 2.75.10^{-4}f^2 + 0.003.$$

Noise in an acoustic channel contains ambient and site-specific noise. The ambient noise, which is always present, may be modeled as Gaussian, but it is not white. Noise is approximated as $N(f) = 50 - 18\log(f)$. The power spectral density decays at approximately 18 dB/decade [8]. If we define a narrow band of frequencies of width Δf around some frequency f, SNR in this band can be expressed as

$$SNR(l,f) = \frac{S^{tr}(f)}{A(l,f)N(f)}, \tag{4}$$

where $S^{tr}(f)$ is the power spectral density of the transmitted signal. Typically, the transmission power may be adjusted based on the distance between the transmitter and receiver robots [23].

2) Multi-path propagation: In an acoustic channel, propagation occurs over multiple paths, as illustrated in Fig. 3. The impulse response of an acoustic channel is influenced by the geometry of the channel and its reflection properties, which determine the number of significant propagation paths, and their relative

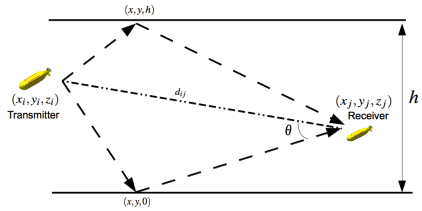

Fig. 3. Geometry of a shallow water channel.

strengths and delays [9]. Since the exact geometry of the channel is unknown, we derive the lower bounds on the lengths of the indirect paths.

Bottom reflection: As shown in Fig. 3, (x_i, y_i, z_i) and (x_j, y_j, z_j) are the coordinates of the transmitter and the receiver robots respectively. Consider an arbitrary point $(x, y, 0)$ at the bottom of this geometry. The length l_{b_p} of a path which is reflected at this arbitrary point is:

$$l_{b_p} = \sqrt{(x-x_i)^2 + (y-y_i)^2 + (z_i)^2} + \sqrt{(x-x_j)^2 + (y-y_j)^2 + (z_j)^2} \tag{5}$$

Since we do not know all the paths arising due to bottom reflection, we calculate the minimum length of these paths. l_b denotes the minimum length of a path arising due to bottom reflection. l_b is obtained by evaluating the derivatives $\frac{dl_{b_p}}{dx}$, equation (6), and $\frac{dl_{b_p}}{dy}$, equation (7), and is of the form described in (8). The path length l_{b_p} of any path arising due to bottom reflection will be equal to or greater than l_b:

$$\frac{dl_{b_p}}{dx} = \frac{(x-x_i)}{\sqrt{(x-x_i)^2 + (y-y_i)^2 + (z_i)^2}} + \frac{(x-x_j)}{\sqrt{(x-x_j)^2 + (y-y_j)^2 + (z_j)^2}} \tag{6}$$

$$\frac{dl_{b_p}}{dy} = \frac{(y-y_i)}{\sqrt{(x-x_i)^2 + (y-y_i)^2 + (z_i)^2}} + \frac{(y-y_j)}{\sqrt{(x-x_j)^2 + (y-y_j)^2 + (z_j)^2}} \tag{7}$$

$$l_{b_p} \geq l_b = \sqrt{\frac{(x_i-x_j)^2}{4} + \frac{(y_i-y_j)^2}{4} + z_i^2} + \sqrt{\frac{(x_i-x_j)^2}{4} + \frac{(y_i-y_j)^2}{4} + z_j^2} \quad . \tag{8}$$

Surface reflection: As shown in Fig. 3, the length l_{s_p} of any indirect path which is reflected at an arbitrary point (x, y, h) on the surface is:

$$l_{s_p} = \sqrt{(x-x_i)^2 + (y-y_i)^2 + (h-z_i)^2} + \sqrt{(x-x_j)^2 + (y-y_j)^2 + (h-z_j)^2}, \tag{9}$$

where h is the depth of the acoustic communication area. Similar to (8), the length l_{s_p} of any path arising due to surface reflection is equal to or greater than the minimum length l_s of the surface reflected paths:

$$l_{s_p} \geq l_s = \sqrt{\frac{(x_i-x_j)^2}{4} + \frac{(y_i-y_j)^2}{4} + (h-z_i)^2} + \sqrt{\frac{(x_i-x_j)^2}{4} + \frac{(y_i-y_j)^2}{4} + (h-z_j)^2} \quad . \tag{10}$$

It should be noted that l_b and l_s can be calculated using the position of the transmitter and receiver vehicles.

3) SNR due to multi-path propagation: Since the transmitted signal experiences multi-path propagation, the total power at the receiver is a superposition of the direct (LOS) path and multiple non-LOS paths. In the following discussions, the non-LOS paths that have experienced multiple reflections are ignored since they experience large attenuation.

Assuming that there are q significant non-LOS paths, the path gain H_p for the p-th non-LOS path is equal to $\Gamma_p/\sqrt{A_p}$, where $p = 1, \ldots, q$, Γ_p is the cumulative reflection coefficient along this path, and A_p is the propagation loss associated with this path. The p-th non-LOS path acts as a low-pass filter whose

transfer function can be modeled as:

$$H_p(l_p,f) = \frac{\Gamma_p}{\sqrt{A(l_p,f)}} e^{-j2\pi f \tau_p}, \tag{11}$$

where l_p is the length of this path, and $\tau_p = l_p/c$ is time delay associated with this path. c is the underwater speed of sound (1500m/s). $e^{-j2\pi f\tau_p}$ captures the effect of the phase delay associated with this path.

Incorporating the path gains of the p significant non-LOS paths, the total transfer function and the total SNR can be described by:

$$H(l,f) = \frac{1}{\sqrt{A(d^{ij},f)}} + \sum_{p=1}^{q} \frac{\Gamma_p}{\sqrt{A(l_p,f)}} e^{-j2\pi f\tau_p}, \tag{12}$$

$$SNR(l,f) = \frac{S^{tr}(f)}{N(f)} \left\| \frac{1}{\sqrt{A(d^{ij},f)}} + \sum_{p=1}^{q} \frac{\Gamma_p}{\sqrt{A(l_p,f)}} e^{-j2\pi f\tau_p} \right\|^2, \tag{13}$$

where d^{ij} is the Euclidean distance between the transmitter and receiver, and denotes the length of the LOS path.

Since all the non-LOS paths can be divided into two groups, bottom reflected and surface reflected paths, the total SNR of the receiver can be expressed as:

$$SNR(l,f) = \frac{S^{tr}(f)}{N(f)} \left\| \frac{1}{\sqrt{A(d^{ij},f)}} + \sum_{p=1}^{q_1} \frac{\Gamma_s}{\sqrt{A(l_{s_p},f)}} e^{-j2\pi f\tau_{s_p}} + \sum_{p=1}^{q_2} \frac{\Gamma_b}{\sqrt{A(l_{b_p},f)}} e^{-j2\pi f\tau_{b_p}} \right\|^2,$$

where q_1 is the number of non-LOS paths resulting due to surface reflection with reflection coefficient Γ_s. q_2 is the number of non-LOS paths resulting due to bottom reflection with reflection coefficient Γ_b. It should be noted that $q_1 + q_2 = q$.

Utilizing (14) to calculate *SNR* is non-trivial due to a lack of knowledge of the non-LOS paths. Instead, the worst case *SNR*, SNR_{worst}, is found using the (known) minimum non-LOS path lengths derived in (8) and (10). SNR_{worst} requires that all non-LOS signals resulting due to multi-path propagation follow the shortest paths with lengths l_s or l_b, and have a phase delay of π associated with them. SNR_{worst} is expressed as:

$$SNR_{worst}(d^{ij},f) = \frac{S^{tr}(f)}{N(f)} \| \frac{1}{\sqrt{A(d^{ij},f)}} - \frac{q_1\Gamma_s}{\sqrt{A(l_s,f)}} - \frac{q_2\Gamma_b}{\sqrt{A(l_b,f)}} \|^2 \tag{14}$$

SNR_{worst} provides the lower bound on *SNR*, and alleviates the need to calculate the actual value of *SNR*. Instead, providing a lower bound η_c on SNR_{worst} will guarantee the satisfaction of *SNR* requirements. Also, since SNR_{worst} is a function of d^{ij} for a given f, a lower bound on SNR_{worst} implies that there exists an upper bound (η_d) on the distance between the transmitter and receiver.

$$\begin{aligned} SNR(l,f) &\geq SNR_{worst}(d^{ij},f) \\ SNR_{worst} \geq \eta_c &\Rightarrow SNR(l,f) \geq \eta_c \\ SNR(l,f) \geq \eta_c &\Rightarrow d^{ij} \leq \eta_d \end{aligned} \tag{15}$$

4) One-hop communication connectivity: We introduce a binary variable $C^{ij}(t)$ that indicates whether the two robots are in one-hop communication range of each other or not. $C^{ij}(t)$=1 for the robots i and j if they are in one-hop communication range of each other at time t, and $C^{ij}(t)$=0 otherwise. The communication constraint requires that each robot be in one-hop communication range of n_{conn} other

robots, and can be expressed as:

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

$$\begin{aligned} C^{ij}(t) &= 1 \quad if \quad SNR_{worst} \geq \eta_c \\ &= 0 \quad if \quad SNR_{worst} < \eta_c \end{aligned} \tag{17}$$

Since one-hop communication constraints (16) and (17) are disjunctive in nature (either-or), we reformulate them as Big-M constraints given by (18)-(20). For this reformulation, we introduce a constant M and formulate constraint (19).

In case $SNR_{worst} > \eta_c$, $C^{ij}(t)$ can assume a value of either 0 or 1. The constraint (20) forces at least n_{conn} of them to be set to 1. This means that only n_{conn} of the C^{ij} variables, and not necessarily all, are guaranteed to have the correct value $C^{ij}(t) = 1$. It is easily seen that if $C^{ij}(t) = 0$, then the constraint will be trivially satisfied for a sufficiently large M.

$$C^{ij}(t) \in \{0,1\} \tag{18}$$

$$\eta_c \leq M(1 - C^{ij}(t)) + SNR_{worst} \tag{19}$$

$$\sum_{j:j\neq i} C^{ij}(t) \geq n_{conn}, \quad C^{ij}(t) \in \{0,1\} \tag{20}$$

D. 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 untill 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 accounting for 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 $\mathbf{p}^i(t) = (x^i(t), y^i(t), z^i(t))$ is the location of the robot i on its path at time t, calculated based on the optimal speed $s^i(t)$.

E. 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 (21)-(33) in the order $ord(i)$ that it has been assigned.

B. Decision Variables

In (21)-(33), 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 (21) 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 (29) 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 its 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).

$$\begin{aligned}
\text{minimize}\quad & \sum_{k=t}^{t+T_{hor}} d^i_{goal}(k) && (21)\\
\text{subject to}\quad & \forall j\in\{1,2,\ldots,n\}, j\neq i\\
& \forall k\in\{t,\ldots,t+T_{hor}\}\\
& (x^i(0),y^i(0),z^i(0)) = o^i && (22)\\
& u^i(0)=0 && (23)\\
& u^i(k)\leq U^i && (24)\\
& u^i(k)=u^i(k-1)+s^i(k)\Delta t && (25)\\
& (x^i(k),y^i(k),z^i(k)) = ps^i(u^i(k)) && (26)\\
& s_{min}\leq s^i(k)\leq s_{max} && (27)\\
& a_{min}\leq a^i(k)\leq a_{max} && (28)\\
& d^i_{goal}(k)=U^i-u^i(k) && (29)\\
& d^{ij}(k)\geq d_{safe} && (30)\\
& M(1-C^{ij}(k))+SNR_{worst}\geq \eta_c && (31)\\
& \sum_{j:j\neq i} C^{ij}(k)\geq n_{conn} && (32)\\
& C^{ij}(k)\in\{0,1\} && (33)
\end{aligned}$$

D. Path (Kinematic) Constraints

Constraints (22)-(26) define the path of each robot. The constraints (22), (23), and (24) form the boundary conditions. Constraint (22) indicates that each robot i has to start at a designated start point o^i. Constraint (23) initializes the arc length travelled u to zero. Constraint (24) provides the upper bound on the arc length travelled. Constraint (25) increments the arc length at each time step based on the speed of the robot ($\Delta t=1$). Constraint (26) 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 constraint (26) is a non-convex nonlinear equality constraint.

E. Speed and Acceleration (Dynamic) Constraint

Constraints (27)-(28) are dynamic constraints that 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.

F. Collision Avoidance Constraint

The non-convex constraint (30) 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 (32) and (33) 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 $SNR_{worst} \geq \eta_c$ 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 (31), 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 (31) is an example of a big-M constraint [24].

H. Decision Making Algorithm

All robots are initially assumed to be in communication range of each other. The general outline of the decision-making 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 trajectory 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

$$\mathscr{P}^i(t+1) = \mathscr{P}^i(t) \backslash \mathbf{p}^i(t), \tag{34}$$

  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.

TABLE I
MOTION AND COMMUNICATION PARAMETER VALUES USED FOR SIMULATIONS

d_{safe}	0.02 m	s_{min}	0	s_{max}	2.5 m/s
a_{min}	-1 m/s^2	a_{max}	0.5 m/s^2	M	1000
f	15 KHz	h	20 m	k_s	1.5

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 acoustic communication constraints on the velocity profiles of the underwater robots, and on the solution computation times. Additionally, we depict the effect of stricter safety requirements on the velocity profiles of the robots.

A. Simulation Setup

The MATLAB function `spline()` was used to generate piecewise cubic splines passing through 8 randomly generated waypoints. The spline paths are parametrized by arc length u. The optimization model, defined by (21)-(33) 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 2.2 GHz Intel Core i7 with 4GB of main memory.

B. Simulations

We have tested our model for 3 different scenarios that include up to 4 underwater robots and a number of communications connectivity constraints. In the following discussions, we plot the spline curve paths

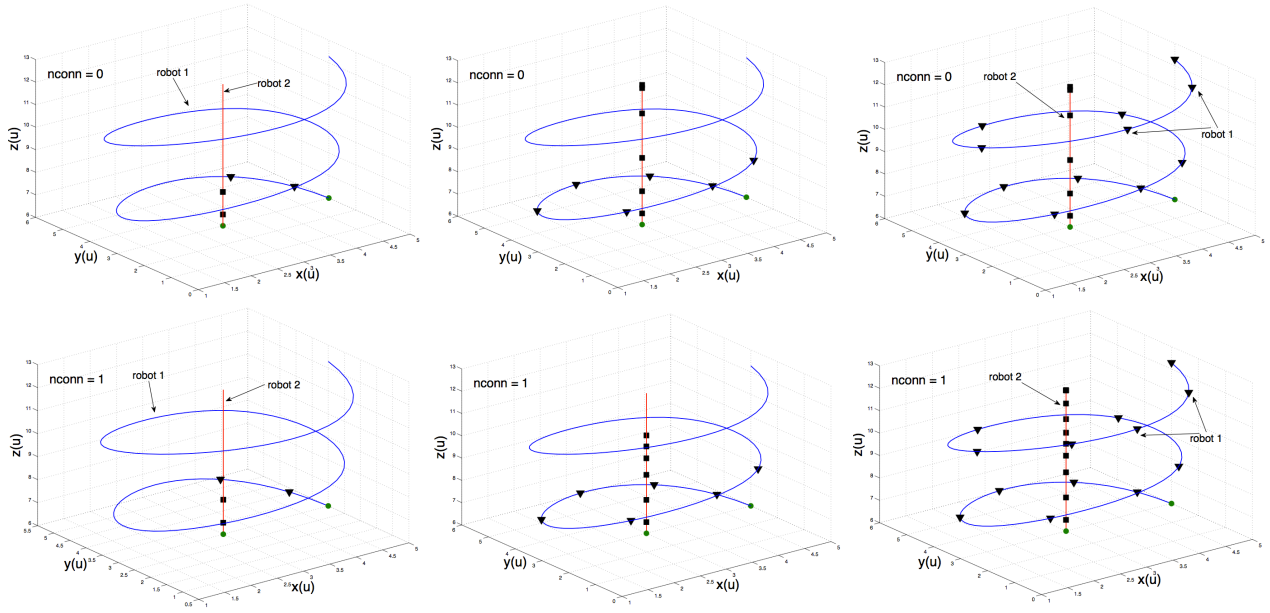

Fig. 4. The top 3 figures indicate robot trajectories for $n_{conn} = 0$ - both robots reach their destination at Tmax =14. The bottom 3 figures indicate robot trajectories for $n_{conn} = 1$. For this case, $T_{max} = 15$. Cartesian coordinate plane units are in meters. The origin and destination points of each robot are indicated by a dot marking. The triangular and square markings on the curves indicate the positions of the robots at each step in time while traveling at optimal speeds along the path.

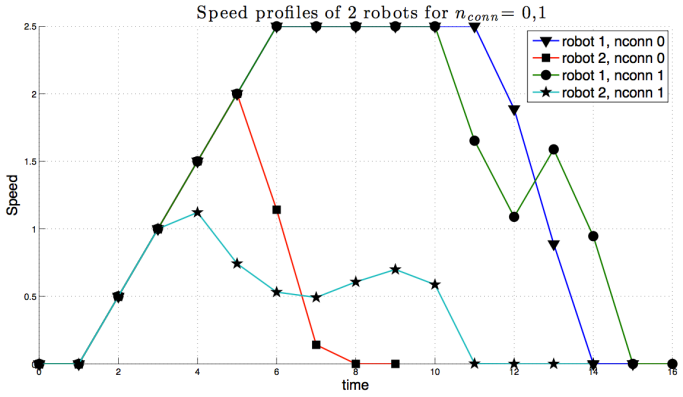

Fig. 5. Velocity profiles of robots in a 2-robot scenario with varying communication connectivity constraints.

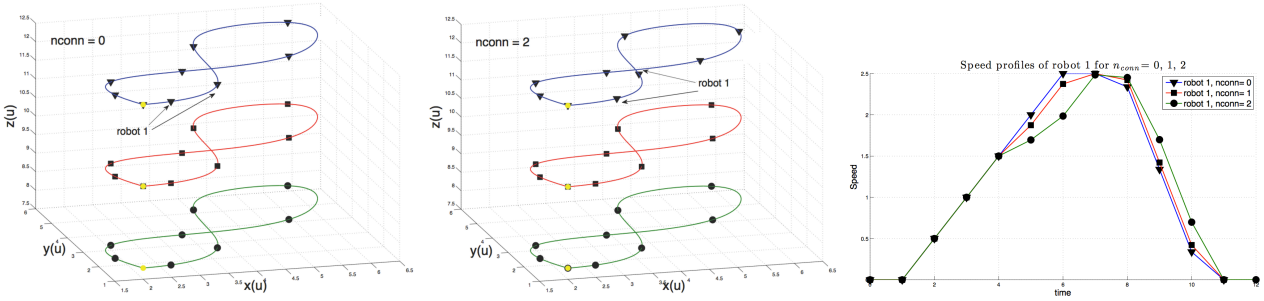

Fig. 6. Trajectories and velocity profiles for 3 robots with varying communication connectivity constraint.

of the robots with different colors indicating different robots. The motion and communications parameters used in our simulations are listed in Table. I. For all simulations, the value of $T_{hor} = 3$. $\Delta t = 1$, unless otherwise specified.

1) Effect of varying n_{conn} on the velocity profiles: We vary n_{conn} between 0 and $n-1$ for $n = 2$, 3, 4.

- *2 robots*: For a scenario where $\eta_c = 20$ dB and $S^{tr} = 20$W, Fig. 4 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. 5 shows the velocity profiles of both robots for this scenario. T_{max} changes from 14 to 15 as n_{conn} goes from 0 to 1.

- *3 robots*: For a scenario where 3 robots follow the parallel paths, $\eta_c = 15dB$ and $S^{tr} = 53$W, Fig. 6 shows the trajectories of the robots for $n_{conn} = 0$ (no communication connectivity requirement) and $n_{conn} = 2$. Fig. 6 also shows the speeds of robot 1 for $n_{conn} = 0, 1,$ and 2. It is observed that for this scenario, the speed of robot 1 changes as n_{conn} goes from 0 to 2.

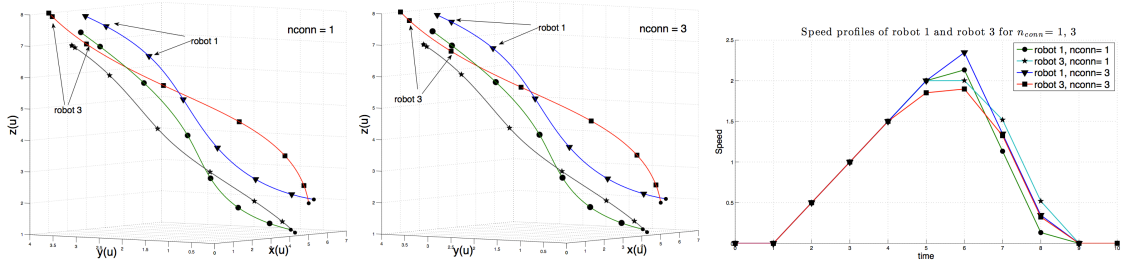

Fig. 7. Trajectories and velocity profiles of a 4-robot scenario with varying communication connectivity constraints.

TABLE II
EFFECT OF VARYING n_{conn} ON T_{comp} IN SECONDS

	n_{conn}			
n	0	1	2	3
2	303.75	31.2	-	-
3	940.94	520.74	66.15	-
4	3152.16	2313.6	921.24	157.77

TABLE III
EFFECT OF VARYING Δt ON T_{comp} (SECONDS) FOR A 2 ROBOT SCENARIO. n_{conn}= 1

Δt	T_{comp}
1	21.78
0.5	41
0.25	107.82

- *4 robots*: For a scenario where $\eta_c = 24$ dB and $S^{tr} = 60$W, Fig. 7 shows the trajectories and speed profiles of the 4 robots for $n_{conn} = 1$ and $n_{conn} = 3$. It is observed that robots 1 and 3 change their speeds in order to accommodate additional communication connectivity requirements n_{conn} goes from 1 to 3.

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 until the scenario is completed. All other MATLAB-MILANO processing times are ignored. T_{comp} values for the scenarios depicted in Figures 4, 6, and 7 are listed in Table II. Following observations are made:

- As expected, T_{comp} increases as the number of robots in the scenario increase. This is due to the increase in the number of variables and constraints in the problem.
- T_{comp} decreases with an increase in n_{conn} for a given number of robots. This is due to the fact that when n_{conn} is larger, a higher number of C^{ij} variables are set to 1 during the solution process. Simply stated, the are a fewer number of feasible binary solutions with increased n_{conn} as more of the variables are required to be 1. Hence the branch-and-bound algorithm of MILANO runs through fewer combinations of 0s and 1s. This results in reduced computation times.

3) Effect of varying Δt on velocity profiles and computation time: For all the numerical results provided thus far, Δt was kept equal to 1. However, Δt, being the discrete time step value, can have an important impact on the underwater vehicular control performance. In order to study the effect of Δt on the solution process, we vary Δt from 1 to 0.5 and 0.25 for a 2-robot scenario similar to the one shown in Fig. 4. Here, $n_{conn} = 1$ with $d_{safe} = 0.02$. It is observed that T_{comp} increases with a decrease in Δt for a given number of robots. This observation presents an important trade-off for implementing the underlying controllers that are responsible to drive the robots along the fixed path at the optimal speeds as determined by solving the RH-MINLP.

4) Collision avoidance and effect of varying d_{safe} on velocity profiles: In all plots, the markings on different paths do not completely overlap with each other at any point in time. This observation indicates that in fact, the robots do not collide with each other at any point in time (satisfying the collision avoidance constraint at all times). In order to study the responsiveness of the solutions to safety requirements, we study the effect of varying d_{safe} on velocity profiles and robot trajectories. For a 2-robot scenario we vary d_{safe} between 0.5m, 0.8m, and 0.9m. Fig. 8 show the trajectory and velocity profiles with different d_{safe} respectively. It is clearly observed that the robots change their trajectories to accommodate more stringent safety requirements while maintaining desired communication connectivity levels.

VI. CONCLUSION

A distributed RH-MINLP framework for solving a fixed path coordination problem for multiple underwater vehicles in the presence of acoustic communication constraints was presented. This problem was solved using the solver MILANO, which implements an interior-point method within a branch-and-bound framework. Several scenarios were studied to quantify the effect of communication connectivity

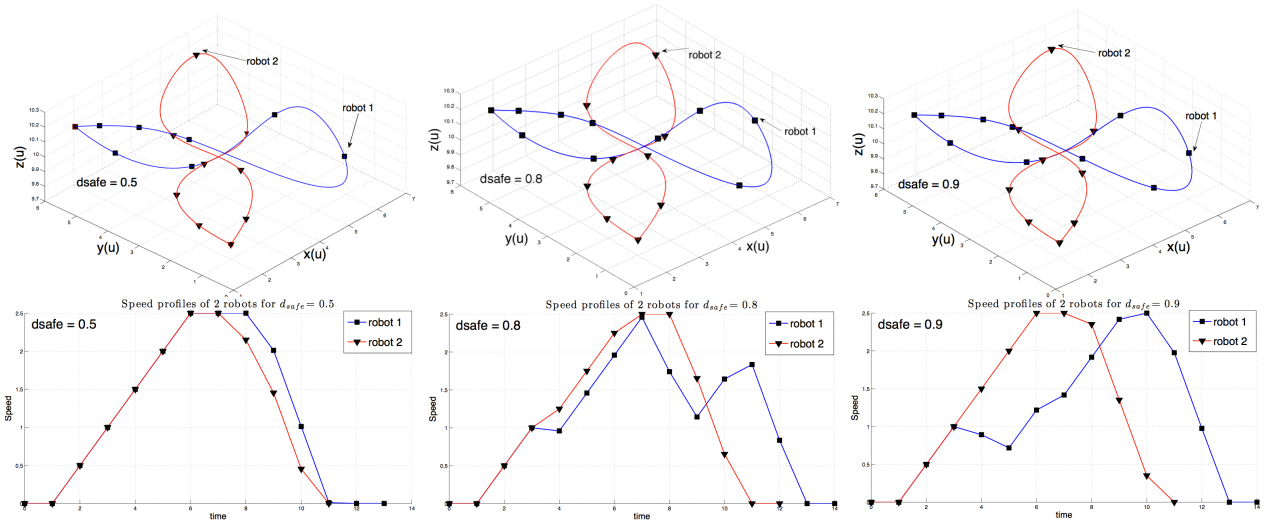

Fig. 8. Trajectories and velocity profiles of a 2-robot scenario ($n_{conn} = 1$) with varying collision avoidance constraint.

constraints on the robot velocity profiles. The solution computation times for these cases were reported. It is found that the communication connectivity requirements have a noticeable effect on the velocity profiles of the robots. It was also found that as the connectivity requirements n_{conn} are increased, the solution computation time decreases steadily. Lastly, we demonstrated that this framework provides collision free trajectories as displayed by the robots changing their trajectories to accommodate stricter satefy (d_{safe}) requirements despite intersecting or overlapping paths. Typically, the optimization improved connectivity at no appreciable cost in journey time. Future work will focus on experimentally validating this framework in an indoor shallow water environment.

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 Transactions on Robotics and Automation*, vol. 18, no. 1, pp. 42–49, 2002.

[3] P. Abichandani, H. Benson, and M. Kam, in *Decentralized Path Coordination in Support of Communication, Proceedings of the International Conference on Robotic Systems*, San Francisco, CA, September 2011.

[4] P. Abichandani, K. Mallory, and M. A. Hsieh, "Experimental multi-vehicle path coordination under communication connectivity constraints," in *Proceedings of the 13th International Symposium on Experimental Robotics (ISER 2012)*, Quebec City, Canada, June 2012.

[5] T. Schouwenaars, J. How, and E. Feron, in *Decentralized Cooperative Trajectory Planning of Multiple Aircraft with Hard Safety Guarantees, AIAA Guidance, Navigation, and Control Conference and Exhibit*, Providence, RI, August 2004.

[6] D. Mellinger, A. Kushleyev, and V. Kumar, "Mixed-integer quadratic program trajectory generation for heterogeneous quadrotor teams," in *Proceedings of the International Conference on Robotic and Automation*, St. Paul, MN, May 2012.

[7] J. Vaganay, M. Elkins, S. Willcox, F. Hover, R. Damus, S. Desset, J. Morash, and V. Polidoro, "Ship hull inspection by hull-relative navigation and control," in *Proc. IEEE/MTS OCEANS Conf. Exhib.*, 2005, pp. 761–766.

[8] M. Stojanovic, "On the relationship between capacity and distance in an underwater acoustic communication channel," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 11, no. 4, pp. 34–43, 2007.

[9] A. Radosevic, J. Proakis, and M. Stojanovic, "Statistical characterization and capacity of shallow water acoustic channels," in *OCEANS 2009 - EUROPE*, May, pp. 1–8.

[10] H. Y. Benson, "Using interior-point methods within an outer approximation framework for mixed integer nonlinear programming," in *Mixed Integer Nonlinear Programming*, ser. The IMA Volumes in Mathematics and its Applications, J. Lee and S. Leyffer, Eds. Springer New York, 2012, vol. 154, pp. 225–243.

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

[12] J. Yuh, "Design and control of autonomous underwater robots: A survey," *Auton. Robots*, vol. 8, no. 1, pp. 7–24, Jan. 2000. [Online]. Available: http://dx.doi.org/10.1023/A:1008984701078

[13] J. M. Walls and R. M. Eustice, "An exact decentralized cooperative navigation algorithm for acoustically networked underwater vehicles with robustness to faulty communication: Theory and experiment," in *Proceedings of the Robotics Science and Systems Conference*, June 2013.

[14] Y. Petillot, I. Tena Ruiz, and D. Lane, "Underwater vehicle obstacle avoidance and path planning using a multi-beam forward looking sonar," *Oceanic Engineering, IEEE Journal of*, vol. 26, no. 2, pp. 240–251, Apr 2001.

[15] C. Petres, Y. Pailhas, P. Patron, Y. Petillot, J. Evans, and D. Lane, "Path planning for autonomous underwater vehicles," *IEEE Transactions on Robotics*, vol. 23, no. 2, pp. 331–341, April 2007.
[16] R. Smith, A. Pereira, Y. Chao, P. Li, D. Caron, B. Jones, and G. Sukhatme, "Autonomous underwater vehicle trajectory design coupled with predictive ocean models: A case study," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, May, pp. 4770–4777.
[17] P. Encarnacao and A. Pascoal, "3d path following for autonomous underwater vehicle," in *Proceedings of the 39th IEEE Conference on Decision and Control*, vol. 3, 2000, pp. 2977–2982 vol.3.
[18] R. Ghabcheloo, A. Aguiar, A. Pascoal, and C. Silvestre, "Coordinated path-following control of multiple auvs in the presence of communication failures and time delays," in *Proceedings of the 7th Conference on Maneuvering and Control of Marine Craft (MCMC2006)*, Sept 2006.
[19] N. Yilmaz, C. Evangelinos, P. F. J. Lermusiaux, and N. Patrikalakis, "Path planning of autonomous underwater vehicles for adaptive sampling using mixed integer linear programming," *IEEE Journal of Oceanic Engineering,*, vol. 33, no. 4, pp. 522–537, Oct. 08.
[20] S. E. Webster, R. M. Eustice, H. Singh, and L. Whitcomb., "Advances in single-beacon one-way-travel-time acoustic navigation for underwater vehicles," *The International Journal of Robotics Research*, vol. 31, no. 8, pp. 935–950, July 2012.
[21] T. Fossen, *Guidance and control of ocean vehicles*. Wiley, 1994. [Online]. Available: http://books.google.com/books?id=cwJUAAAAMAAJ
[22] 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.
[23] M. Stojanovic, "Underwater acoustic communications: Design considerations on the physical layer," in *Fifth Annual Conference on Wireless on Demand Network Systems and Services (WONS 2008)*, Jan. 2008, pp. 1–10.
[24] A. Bemporad and M. Morari, *Control of Systems Integrating Logic, Dynamics, and Constraints, Automatica*, vol. 35, pp. 407–427, 1999.

Solmaz Torabi received her B.S. degree in Electrical Engineering from Sharif University of Technology, Tehran, Iran, in 2011. Her research interests are in the areas of acoustic communication and optimal decision-making. She currently works as a research assistant at Drexel University's Electrical and Computer Engineering department.

Shambadeb Basu received his B.Tech. degree from West Bengal University of Technology, Calcutta, India. Subsequent to this, he worked as a Research Fellow at Jadavpur University, Calcutta, India. His interests are in control and motion planning for autonomous underwater robots. He currently works as a research assistant at Drexel University's Electrical and Computer Engineering department.

Hande Benson is an Associate Professor at the Decision Sciences Department in the LeBow College of Business of Drexel University. Her research interests lie in the field of nonlinear programming, particularly in using interior-point methods to solve large-scale optimization problems. Research sponsors include the National Science Foundation (NSF) and the Office of Naval Research (ONR).

Pramod Abichandani is a Senior Researcher and an Assistant Teaching Professor at Drexel University. His research interests are centered around optimal, multi-dimensional, data-driven decision-making, through the use of techniques from mathematical programming, linear and nonlinear systems theory, statistics, and machine learning. Research sponsors include the Office of Naval Research (ONR), Mathworks, and Drexel University.