Optimal Control of an Underwater Sensor Network for Cooperative Target Tracking

Kelli A. C. Baumgartner, *Member, IEEE*, Silvia Ferrari, *Senior Member, IEEE*, and Anil V. Rao

***Abstract*—Optimal control (OC) is a general and effective approach for trajectory optimization in dynamical systems. So far, however, it has not been applied to mobile sensor networks due to the lack of suitable objective functions and system models. In this paper, an integral objective function representing the quality of service of a sensor network performing cooperative track detection over time is derived using a geometric transversals approach. A set of differential equations modeling the sensor network's dynamics is obtained by considering three dependent subsystems, i.e., underwater vehicles, onboard sensors, and oceanographic fields. Each sensor-equipped vehicle is modeled as a bounded subset of a Euclidian space, representing the sensor's field of view (FOV), which moves according to underwater vehicle dynamics. By this approach, the problem of generating optimal sensors' trajectories is formulated as an OC problem in computational geometry. The numerical results show that OC significantly improves the network's quality of service compared to area-coverage and path-planning methods. Also, it can be used to incorporate sensing and energy constraints on the sensors' state and control vectors, and to generate fronts of Pareto optimal trajectories.**

***Index Terms*—Cooperative, detection, optimal control (OC), optimization, planning, sensor networks, target tracking, track coverage, trajectory.**

I. Introduction

THE development of reliable sensor networks and autonomous-vehicle technologies are producing advanced surveillance systems that are characterized by a high degree of functionality and reconfigurability. Examples include mobile sensor networks for tracking and monitoring endangered species [1], [2], or for tracking and detecting possible intruders [3]–[6]. Many of these applications employ wireless sensors that have limited range and are characterized by frequent false alarms due to highly variable ocean environments [5]. Thus, their objective is to cooperatively detect target tracks, i.e., to obtain multiple elementary target detections over time [7], where a track is said to be detected when multiple independent sensor detections can be fused to form a feasible target track in an approach known as *track-before-detect* [5]. The quality of service of a sensor network performing cooperative track detection, referred to as *track coverage*, has received considerable attention [5], [7]–[12]. Several authors have addressed the placement of sensors to provide a desired level of track coverage in the region of interest (ROI), assuming that they remain stationary at all times [8], [13]–[16]. This paper addresses the case in which these sensors are expected to operate in a dynamic ocean environment, and are installed on autonomous underwater vehicles (AUVs) that can survey the ROI for several days.

Approaches for generating a sensor's path or trajectory include area coverage (AC) [17], [18], random [18], grid [19], and optimal search strategies [19], [20]. Optimal search strategies have been shown to outperform other approaches in applications where prior information is available, such as sensor models, environmental conditions, and prior measurements. However, they typically are not applicable to sensors with a limited range, or to cooperative sensing objectives, such as track coverage. Cooperative control methods have been developed to provide point or AC [17], [21], or to cooperatively manage the sensors' formation in response to the sensed environment [18], [22], [23]. Although they are very effective in some applications, these methods are not applicable to networks that perform other cooperative sensing tasks, such as target tracking, or that are characterized by nonlinear dynamics and time-varying environments. This paper presents a novel optimal control (OC) approach for generating the trajectories of a mobile sensor network composed of a fleet of sensor-equipped underwater vehicles that perform cooperative track detection. Its advantage over the aforementioned techniques is that it can be used to optimize any cooperative sensing task, such as area or track coverage, subject to nonlinear sensor dynamics and time-varying environmental conditions.

The problem formulation most closely related to that presented in this paper is an approach that generates the optimal trajectories of a cooperative sensor network for the purpose of ocean sampling [24]. In this sampling approach, a performance metric is obtained from the background covariance function to represent the ability of a network of underwater gliders to sample an ROI [24]. A coordinated control strategy is then obtained by modeling each glider as a point mass that moves at constant heading and speed, and by parameterizing families of collectives over closed curves, such as circles, that represent

Manuscript received October 02, 2008; accepted May 27, 2009. First published October 30, 2009; current version published November 25, 2009. This work was supported by the U.S. Office of Naval Research Young Investigator Program (Code 321).

Associate Editor: L. L. Whitcomb.

K. A. C. Baumgartner was with the Department of Mechanical Engineering and Materials Science, Duke University, Durham, NC 27708 USA. She is now with the System Development and Integration Division, Analex Corporation, Brook Park, OH 44142 USA (e-mail: kelli.baumgartner@qinetiq-na.com).

S. Ferrari is with the Department of Mechanical Engineering and Materials Science, Duke University, Durham, NC 27708 USA (e-mail: sferrari@duke.edu).

A. V. Rao is with the Department of Aerospace Engineering, University of Florida, Gainesville, FL 32611-6250 USA (e-mail: anilvrao@ufl.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JOE.2009.2025643

0364-9059/$26.00 © 2009 IEEE

the trajectories of subgroups of vehicles in the network. As pointed out in [24] and [25], these sensor-equipped AUVs are among the newest available sensing technology, and offer compelling opportunities because of their ability to operate cooperatively, and to scour the ocean in search for data and information. Although motivated by the same technology, ocean sampling approaches cannot be extended to the problem of cooperative track detection, because they employ different performance metrics and assume vehicle formations that are not necessarily optimal for track coverage. Classical trajectory-planning methods for AUVs (e.g., [26]–[29]) are also inapplicable because they are vehicle-centric and, as such, they do not account for the performance of the onboard sensors, or for the cooperative nature of the network.

Although OC is considered the most general and effective approach to trajectory optimization [28], [30]–[32], so far it has not been applied to mobile sensor networks due to the lack of suitable objective functions and system models. In this paper, the geometric transversals formulation of track coverage presented in [8] is extended to mobile sensors with time-varying positions and fields of view. By this approach, an integral track-coverage objective function is obtained that represents the network's ability to cooperatively detect targets over time. Bridging recent developments in sensor networks research [5], [8], [33] and AUVs [26], [27], a model of sensor network's dynamics is obtained by considering three dependent subsystems, i.e., underwater vehicles, onboard sensors, and oceanographic fields. The field-of-view (FOV) of many wireless sensors with limited range can be represented by a bounded subset of a Euclidian space (e.g., a disk or a sector) whose geometry and dimensions depend on the physical principles underlying the measurement process, and on the environmental conditions [5], [13], [33]–[35]. In this paper, each sensor's FOV is represented by a disk of time-varying radius that moves according to underwater vehicle dynamics. The influence of the ocean currents on vehicle dynamics is modeled by a neural network obtained from an ocean current forecast, using Bayesian regularization [36], oceanographic prediction models [37], and real data [38]. Thus, over time, the vehicles' energy consumption can be minimized while the network's track coverage is maximized.

Using this model of the sensor network's dynamics (presented in Section III), the sensors' trajectories can be generated via OC (reviewed in Section II). Since the dynamics are nonlinear, near-optimal trajectories are computed numerically using the direct shooting method (DSM), and are verified using the Gauss pseudospectral method (GPM), reviewed in Section IV. In Section VI, the OC approach is demonstrated by generating the trajectories of sensor networks deployed in an ROI near the New Jersey coast, using real ocean current data obtained from the Coastal Ocean Observation Laboratory (COOL), Rutgers—The State University of New Jersey, Piscataway [38], and the vehicle models in [28]. Since track coverage and energy consumption are competing objectives, a parametric study is conducted in Section VI-B to determine the Pareto front and analyze the influence of the objective function weighting on the sensors' trajectories. The numerical results show that OC significantly improves performance compared to methods that are obtained by modifying AC and path planning (PP) techniques (Section VI-C). These results also illustrate the influence of the ocean currents on the optimal sensors' trajectories, and demonstrate the importance of forecast models of the ocean dynamics in sensor deployment techniques. Finally, in Section VI-D, the generality of the OC approach is illustrated by computing sensors' trajectories that guarantee a minimum required track-coverage performance, with minimum energy consumption, through the use of inequality constraints.

II. Background on Optimal Control

The trajectories of a cooperative sensor network are generated using the OC framework reviewed in this section. Consider a system whose dynamics can be approximated by a nonlinear differential equation

$$\dot{\mathbf{x}}(t) = \mathbf{f}\left[\mathbf{x}(t), \mathbf{p}(t), \mathbf{u}(t), t\right], \qquad \mathbf{x}(t_0) = \mathbf{x}_0 \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the system state, $\mathbf{u} \in \mathbb{R}^m$ is the control, and $\mathbf{p} \in \mathbb{R}^l$ is a vector of time-varying parameters that represent the physical characteristics of the system, and scale the system's response to control inputs and to its own motions [30]. OC seeks to determine the state and control trajectories that optimize an integral cost function

$$J = \phi\left[\mathbf{x}(t_f)\right] + \int_{t_0}^{t_f} \mathcal{L}\left[\mathbf{x}(t), \mathbf{u}(t), t\right] dt \tag{2}$$

over a time interval $\Delta T = [t_0, t_f]$, subject to (1) and to the r-dimensional inequality constraint on the state and control

$$\mathbf{c}\left[\mathbf{x}(t), \mathbf{u}(t), t\right] \leq \mathbf{0}_{r\times 1}. \tag{3}$$

In this paper, it is assumed that t_f is fixed and finite. The OC equations (1)–(3) representing the sensor network's dynamics and performance are derived in Section III.

The necessary conditions for optimality are given by the well-known Euler–Lagrange equations, which can be derived using calculus of variations, as shown in [30]. For a nonlinear system, such as (1), these equations amount to a Hamiltonian boundary-value (HPBV) problem for which there is no closed-form solutions, and therefore, they typically are solved numerically [30], [39], [40]. As reviewed in [40], numerical methods for solving OC problems can be classified into direct and indirect methods. Indirect methods solve the HPBV problem numerically, to determine candidate optimal trajectories known as extremals. Direct methods determine near-optimal solutions by discretizing the continuous-time problem about collocation points and then transcribing it into a finite-dimensional nonlinear program (NLP). The NLP is then solved using an appropriate optimization method, such as sequential quadratic programming (SQP) [41], [42]. Direct methods are typically easier to implement than indirect methods and can be applied to a wider range of OC problems [43]–[45]. Therefore, in this paper, two popular direct methods, DSM and GPM, reviewed in Section IV are applied and compared to solve the problem of optimally controlling a cooperative sensor network, which is derived in the next section.

III. Mathematical Models

In this section, the OC framework is used to formulate the problem of optimizing the state and control trajectories of a network of n cooperative underwater sensors in a dynamic ocean environment. Every sensor in the network is installed on an AUV, and is assumed to be omnidirectional, where the sensor's FOV is approximated by a circle centered at the vehicle's location. The sensor network is deployed in an ROI for the purpose of tracking and detecting unauthorized targets that are assumed to maintain a constant heading over a time interval ΔT, without any prior knowledge of the target tracks. A model of the system dynamics (1) is obtained in Sections III-A and III-B, by considering the interactions of three subsystems composed of n AUVs, the onboard sensors, and the oceanographic fields in the ROI. The cost function (2) is defined as a tradeoff between the network's quality of service and energy consumption. As shown in Section III-C, the quality of service of a mobile network that performs cooperative track detection is formulated in closed form with respect to the sensors' positions and FOVs, using the geometric transversals approach presented in [8]. Finally, the inequality constraint (3) is used to specify physical bounds representing the ROI, the available control power, and, possibly, the minimum quality of service to be achieved by the network, as shown in Section III-D.

A. Vehicle Equations of Motion

For the purposes of trajectory generation, the ith underwater vehicle in the network can be assumed to be a freely swimming rigid body, propelled by onboard thrustors, that obeys the equation of motion

$$\mathbf{M}_i\dot{\boldsymbol{\nu}}_i + \mathbf{C}_i(\boldsymbol{\nu}_i)\boldsymbol{\nu}_i + \mathbf{D}_i(\boldsymbol{\nu}_i)\boldsymbol{\nu}_i + \mathbf{g}_i(\boldsymbol{\xi}_i) = \tilde{\mathbf{B}}_i\mathbf{T}_i(\boldsymbol{\nu}_i, \mathbf{u}_i), \qquad i = 1, \ldots, n \tag{4}$$

as shown in [46]–[48]. The positive-definite matrix $\mathbf{M}_i$ contains the vehicle inertia and the hydrodynamic added inertia. $\mathbf{g}_i$ is a vector of gravitational forces and moments, and the external forcing $\mathbf{T}_i$ generated by the control inputs $\mathbf{u}_i$ is premultiplied by a control-effect matrix $\tilde{\mathbf{B}}_i$, which is determined by the location of the actuators. The matrices $\mathbf{C}_i$ and $\mathbf{D}_i$ represent Coriolis and dissipative hydrodynamic effects, respectively, and depend on the vehicle's speed in the body coordinate frame (fixed to the vehicle) represented by $\boldsymbol{\nu}_i \in \mathbb{R}^6$. Let $\boldsymbol{\xi}_i = [x_i\ y_i\ z_i\ \phi_i\ \theta_i\ \psi_i]^T$ denote a vector of the ith vehicle's position coordinates x_i, y_i, and z_i, and orientation ϕ_i, θ_i, and ψ_i, with respect to an inertial coordinate frame. Then, if $\boldsymbol{v}_i \in \mathbb{R}^6$ represents the local current speed vector, the relative vehicle's speed can be written as $\boldsymbol{\nu}_i = \dot{\boldsymbol{\xi}}_i - \boldsymbol{v}_i$. As shown in [28], every element of the control vector $\mathbf{u}_i \in \mathbb{R}^m$ represents the thrust from one of the m propellers of the ith vehicle. By performing a coordinate transformation [46], the vehicle dynamics in the inertial frame can be obtained from (4), and can be written in state–space form

$$\dot{\mathbf{x}}_i(t) = \begin{bmatrix} \mathbf{J}(\phi_i, \theta_i, \psi_i)\boldsymbol{\nu}_i \\ \mathbf{M}_i^{-1}\left[-\mathbf{C}_i(\boldsymbol{\nu}_i)\boldsymbol{\nu}_i - \mathbf{D}_i(\boldsymbol{\nu}_i)\boldsymbol{\nu}_i - \mathbf{g}_i(\boldsymbol{\xi}_i)\right] \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{M}_i^{-1}\tilde{\mathbf{B}}_i \end{bmatrix}\mathbf{u}_i(t) \tag{5}$$

$$= \mathbf{f}_i\left[\mathbf{x}_i(t), \boldsymbol{v}_i\right] + \mathbf{B}_i\mathbf{u}_i(t), \qquad i = 1, \ldots, n \tag{6}$$

where $\mathbf{x}_i \equiv \left[\boldsymbol{\xi}_i^T\ \boldsymbol{\nu}_i^T\right]^T$. The Jacobian $\mathbf{J}$ is a block-diagonal matrix containing a 3×3 orthogonal rotation matrix, and a 3×3 transformation matrix that are functions of the roll, pitch, and yaw angles ϕ, θ, and ψ, respectively (see [28] for more details).

The model parameters $\mathbf{B}_i$, $\mathbf{C}_i$, $\mathbf{D}_i$, and $\mathbf{g}_i$ can be assumed constant, provided the inner-loop controller used for trajectory tracking is designed to compensate for parameter variations, e.g., through integral compensation or adaptive filtering techniques [30, pp. 283–284], [49]. Two additional assumptions that are commonly introduced to reduce the dimensionality of the trajectory optimization problem are that the AUV maintains a constant depth and a zero pitch angle, and that the roll angle is zero [26], [27]. The sensors' trajectories are generated over a time interval of several days, in an ROI for which the current speed varies significantly with respect to both time and position. Thus, assuming the vertical and rotational components of the current speed are negligible compared to the translational components [26], [27], the current speed vector at the ith sensor location can be defined in terms of its components in the inertial coordinate frame

$$\boldsymbol{v}_i = \left[v_x(x_i,\ y_i,\ z_i, t) \quad v_y(x_i,\ y_i,\ z_i, t)\right]^T = \boldsymbol{v}_i\left(\mathbf{x}_i(t), t\right). \tag{7}$$

Since z_i is assumed constant, the ith-sensor state to be optimized can be simplified to $\mathbf{x}_i = [x_i\ y_i]^T$, and the state of the sensor network is defined as $\mathbf{x} \equiv [\mathbf{x}_1\ \ldots\ \mathbf{x}_n]^T$. Similarly, the control and current speed vectors for the sensor network are defined as $\mathbf{u} \equiv [\mathbf{u}_1\ \ldots\ \mathbf{u}_n]^T$ and $\boldsymbol{v} \equiv [\boldsymbol{v}_1\ \ldots\ \boldsymbol{v}_n]^T$, respectively. Then, from (5), the dynamics of a network with n sensors can be approximated by the $(2n)$th-order differential equation

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} \mathbf{f}_1\left[\mathbf{x}_1(t), \boldsymbol{v}_1\left(\mathbf{x}_1(t), t\right)\right] \\ \vdots \\ \mathbf{f}_n\left[\mathbf{x}_n(t), \boldsymbol{v}_n\left(\mathbf{x}_n(t), t\right)\right] \end{bmatrix} + \begin{bmatrix} \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_n \end{bmatrix} \begin{bmatrix} \mathbf{u}_1(t) \\ \vdots \\ \mathbf{u}_n(t) \end{bmatrix}$$

$$\equiv \mathbf{f}\left[\mathbf{x}(t), \boldsymbol{v}\left(\mathbf{x}(t), t\right)\right] + \mathbf{B}\mathbf{u}(t). \tag{8}$$

The same approach can be applied to other underwater vehicles by modifying (8). Also, the above simplifying assumptions can be relaxed at the expense of computation time.

The influence of ocean currents on the sensors' motion is modeled in (8) in order to exploit the natural dynamics for sensors' transport to minimize energy consumption (as shown by the results in Section VI). The model of the ocean current field over the ROI is obtained from a forecast that is initialized with real-time measurements and estimates [50], and is computed from physical models and data via state-estimation techniques [37], [51]. A current speed forecast typically consists of a multidimensional array containing estimated values of the components in (7) at sample points in space and time. Although tabular data representations can potentially be treated in the OC framework, they typically lead to problems that are computationally intractable, and may compromise convergence [52]. To circumvent these problems, an approach based on open-boundary model analysis and B-splines basis functions was proposed in [26], [27], and [53]. In this paper, the neural network approach presented in [54], and reviewed in this section, is adopted to model the current speed forecast, and account for its influence on the sensors' dynamics (8). The advantages of this approach over B-splines are that it can be carried out in one step, without preprocessing the speed forecast, and it is very effective both at smoothing and generalizing the data [36], [55]–[58]. The approach in [54] obtains a neural network representation of the spatial and temporal characteristics of the current speed field over a 2-D ROI $\mathcal{A} \subset \mathbb{R}^2$, and a time interval ΔT. Let $\boldsymbol{\eta}(t) \equiv [\mathbf{x}_i^T(t)\ t]^T \in \mathcal{N}$, where $\mathcal{N} = \mathcal{A} \times \Delta T$, and $\mathcal{F} = \{\mathbf{x}_i^\kappa, \boldsymbol{v}_i^\kappa, t^\kappa\}_{\kappa=1,\ldots,p}$ be a training set containing the value $\boldsymbol{v}_i^\kappa$ of the current speed vector (7), at the location $\mathbf{x}_i^\kappa$, and instant t^κ, for p sample points in $\mathcal{N}$. Then, $\mathcal{F}$ is used to obtain a feedforward neural network

$$\boldsymbol{v}_i\left(\mathbf{x}_i(t), t\right) = \mathbf{W}_2 \boldsymbol{\Phi}\left[\mathbf{W}_1\, \boldsymbol{\eta}(t) + \mathbf{b}_1\right] + \mathbf{b}_2 \qquad (9)$$

with two linear output neurons, and one hidden layer of s sigmoidal functions represented by the operator

$$\boldsymbol{\Phi}(\mathbf{n}) \equiv \left[\sigma(n_1)\ \cdots\ \sigma(n_s)\right]^T \qquad (10)$$

where $\mathbf{n} = [n_1 \cdots n_s]^T$ is the input to the hidden layer, and $\sigma(n_i) \equiv 1/(1 + e^{-n_i})$. The neural network weights $\mathbf{W}_1 \in \mathbb{R}^{s \times 3}$, $\mathbf{W}_2 \in \mathbb{R}^{2 \times s}$, $\mathbf{b}_1 \in \mathbb{R}^s$, and $\mathbf{b}_2 \in \mathbb{R}^2$ are determined from $\mathcal{F}$ by a Bayesian regularization backpropagation algorithm (“trainbr’ [59]) that minimizes a linear combination of neural network squared errors and weights, as shown in [36], [54], and [60]. The effectiveness of this approach has been verified through numerical experiments using forecast data as well as real measurements, as shown in Section V.

The differential equations (8) combined with the neural network (9) are used to model the sensor network dynamics in the OC problem formulation. As shown in the next section, the measurement process is represented by a moving binary sensor model in the shape of a disk whose dimensions may be obtained from oceanographic field information and acoustic propagation models.

B. Sensor Model

Many applications in tracking and surveillance employ passive sensors, such as proximity sensors, that have limited processing abilities and must minimize wireless information transfer to a central processor [5]. Particularly, in highly variable and unknown ocean environments, which are characterized by frequent false alarms, tracking is performed by a central fusion processor based on multiple and independent detection events reported in the ROI. For example, each sensor may report only its location and one value of received signal level at the closest point of approach (CPA). In this paper, we assume that the signal received by the ith sensor is isotropic energy attenuated by the environment according to the following power law:

$$E_i(t) = cF\lambda_i^{-\alpha}(t) \qquad (11)$$

where λ_i is the distance between the ith sensor and the target at the time of the CPA detection. The values of the attenuation coefficient α and of the scaling constant c depend on the physical mechanisms of wave propagation and on the environmental conditions. F represents the target source level that is independent of both time and sensor location. The measurement model in (11) can be applied to acoustic, magnetic, and optical sensor measurements that are governed by linear wave propagation models [61], [62].

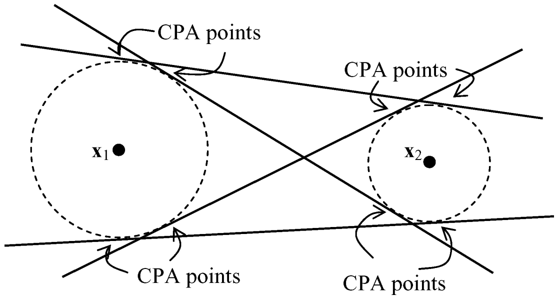


Fig. 1. Geometry of interior and exterior tracks formed from two CPA detections obtained by two omnidirectional sensors, placed at $\mathbf{x}_1$ and $\mathbf{x}_2$ (adapted from [62]).

A CPA detection occurs when E_i exceeds a threshold ϑ_i, which is typically tuned by an operator for each sensor [63], and the values of E_i and $\mathbf{x}_i$ at the CPA time are reported by the ith sensor to the central processor. These values are then fused with the information obtained from the other CPA detections, using (11), to form hypothetical target tracks without knowledge of c and F [62]. For example, if two sensors indexed by $i = 1, 2$ are located at $\mathbf{x}_1$ and $\mathbf{x}_2$ and have each reported a CPA detection, four hypothetical target tracks can be obtained, as illustrated in Fig. 1. The number of required target detections k depends on the false-alarm rate, on the measurement errors, and on the track accuracy required by the surveillance system [62]. In [62], $k = 3$ was found to provide accurate tracking by proximity sensors subject to few false alarms, and errors normally distributed with a standard deviation of 20%.

From (11), the maximum range at which the ith sensor can report a CPA detection, given a target source level F and a threshold ϑ_i, is $r_i^{\mathrm{CPA}} = (cF)^{1/\alpha}\vartheta_i$. The value of r_i^{CPA} can be estimated from the environmental conditions and, in this paper, it is assumed constant for simplicity [5], [8], [33]. This approach also can be extended to account for effects of a nonuniform and time-varying ocean environment on r_i^{CPA}, by using high-fidelity acoustic wave propagation models and oceanographic field estimates [35]. The noise generated by the propellers to control the vehicle increases the signal required to report a detection and, consequently, decreases the maximum sensor range. Thus, the effective sensor range is modeled as the sum of the CPA detection range and of a monotonically decreasing function of the controls

$$r_i(t) = (cF)^{1/\alpha}\vartheta_i - \frac{1}{ae^{b\|\mathbf{u}_i(t)\|}} \qquad (12)$$

where a and b are positive constants and, thus, $0 < r_i(t) \leq r_i^{\mathrm{CPA}}$. The constants a and b are chosen based on the sensor and propeller design characteristics [64]. A CPA detection is reported by the ith sensor only if the target comes within its effective range (12). Let the FOV of the ith sensor, located at $\mathbf{x}_i(t) \in \mathcal{A}$ at time $t \in [t_0, t_f]$, be defined as a closed and bounded subset of the ROI, $C_i(t) \subset \mathcal{A}$, such that the ith sensor has a nonzero probability of detecting a target located at a point $\mathbf{x}_T \in C_i(t)$, at t. For an acoustic omnidirectional sensor that obeys the isotropic

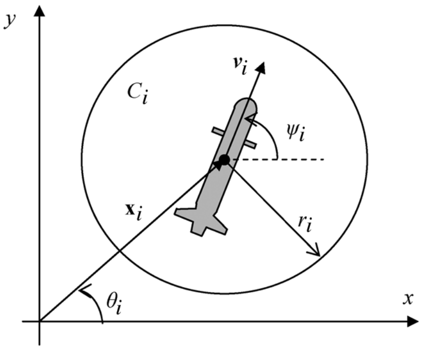


Fig. 2. Schematic of the ith mobile sensor (not to scale).

law (11), $C_i(t) = C_i[\mathbf{x}_i(t), r_i(t)]$ is a disk of time-varying radius $r_i(t)$, centered at $\mathbf{x}_i(t)$ [5], [13], [33]. Then, as schematized in Fig. 2, each sensor in the network can be viewed as a disk that moves in $\mathcal{A}$ according to the dynamic equation (5), and whose radius changes in time as shown in (12). For an omnidirectional network, the range vector $\mathbf{r}(t) \equiv [r_1(t) \cdots r_n(t)]^T$ and the state vector $\mathbf{x}(t)$ together specify the FOVs of all sensors that are represented by the set $S(t) = \{C_1(t), \ldots, C_n(t)\}$.

Recently, the performance of a network composed of n omnidirectional sensors deployed in an ROI to cooperatively track unauthorized targets was shown to be a function of the sensors' ranges and positions [8]. In the next section, the approach presented in [8] is extended to a network of sensors with time-varying positions and ranges, and the objective function of the sensor network is obtained in closed form.

C. Objective Function

The problem of cooperative track detection is motivated by surveillance systems in which little or no prior information is available about the target, and sensor measurements are infrequent and prone to false alarms [5], [7], [8], [62]. For example, in track-before-detect surveillance systems, the presence of an unauthorized target is established only after a hypothetical track can be formed from multiple independent detections obtained by the sensor network in $\mathcal{A}$. The dimensions of $\mathcal{A}$ and the time interval ΔT are chosen such that the target can be assumed to move at a constant speed and heading, and to maintain a constant source amplitude. After a minimum of k detections are obtained from k distinct sensors in the network, a tracking algorithm is used to form the target track [6], [62], [65]. When the sensors are deployed at a depth that enables target detections within their effective ranges, the ROI can be assumed to be 2-D and, in this paper, it is represented by a fixed rectangle $\mathcal{A} = [0, L_1] \times [0, L_2]$.

The quality of service of a sensor network performing cooperative track detection in a polygonal ROI is referred to as *track coverage*, and has received considerable attention in the literature [5], [7], [8], [11], [12]. In [7], [11], and [12], closed-form solutions for probability of multiple target detections were obtained using search theory and Poisson approximations. This approach assumed a uniform distribution of sensors with constant range, and modeled the moving target as a two-state Markov processes. More recent studies showed that the track coverage of a network with binary sensor models can be formulated as a function of the sensors' ranges and positions $\mathbf{r}$ and $\mathbf{x}$, allowing to compute the probability of multiple detections for sensors with variable range, located anywhere in the ROI [5], [8]. Consequently, track coverage can be optimized with respect to the free design parameters to position the sensors in $\mathcal{A}$ [8], or to design the sensors' ranges [66]. As a result, the quality of service of the network can be increased by several orders of magnitude compared to existing deployment schemes [13], [67].

Unlike these previous studies, which considered static sensor networks, in this paper, the track coverage of a mobile sensor network is considered and optimized subject to the dynamic equation of the vehicles (8). The geometric transversals approach presented in [8] is adopted here because it leads to a track coverage function that is formulated in terms of the individual sensors' ranges and positions. In this section, we show that track coverage can be formulated as a function of time, and integrated over the time interval ΔT to represent the quality of service of a mobile sensor network that is deployed for a period of time that can last up to several days. In order for the mission to be sustainable, the energy utilized by the vehicles must be minimized subject to the model of ocean currents in (9). Thus, in this section, a cost function that includes both objectives is obtained by defining the Lagrangian to be a weighted combination of track coverage and power required.

As a first step, we derive the time-dependent track coverage function by representing the target tracks detected by S as geometric transversals. A set of geometric objects in $\mathbb{R}^d$ is said to have a j-transversal when all the objects are simultaneously intersected by a j-dimensional flat or translate of a linear subspace [68]. A line transversal with $j = 1$, $d = 2$, and $k \geq 1$, also referred to as *stabber*, is a straight line that intersects at least k members of a family of objects. For example, line transversals of a family of five disks, with $k = 3$, are shown in Fig. 3. Since the targets can be assumed to move at a constant speed V and heading θ, every target track in $\mathcal{A}$ can be represented by a ray or half line $\mathcal{R}_\theta(b)$ with intercept $b \in \partial\mathcal{A}$, where $\partial\mathcal{A}$ is the perimeter of $\mathcal{A}$. Since there is no prior knowledge of the target tracks, the track parameters θ and b have a uniform probability distribution over their range. Thus, assuming all targets traverse $\mathcal{A}$ during the time interval ΔT, the track coverage of the network at any time $t \in \Delta T$ is directly proportional to its probability of track detection, as shown in [8]. The track coverage function at t can be derived by observing that the ith sensor in the network detects a track $\mathcal{R}_\theta(b)$ at time t if and only if the track intersects its FOV, i.e., $\mathcal{R}_\theta(b) \cap C_i(t) \neq \emptyset$, and $\mathcal{R}_\theta(b)$ is a stabber of $C_i(t)$ at t. Then, a target track that is cooperatively detected by k sensors in the network during the time interval ΔT is a stabber of $S(t)$. Since the target tracks are uniformly distributed and may intersect members of $S(t)$ at any time during ΔT, the set of all stabbers of $S(t)$ for all $t \in \Delta T$ represents the set of all tracks detected by the network.

Let the inertial xy-frame of reference be placed along two sides of $\mathcal{A}$, with the origin $(0,0)_{xy}$ at the lower left corner, such that all target tracks traverse $\mathcal{A}$ in the positive orthant $\mathbb{R}^2_+$ (Fig. 4). As proven in Appendix I, the set of stabbers of a disk

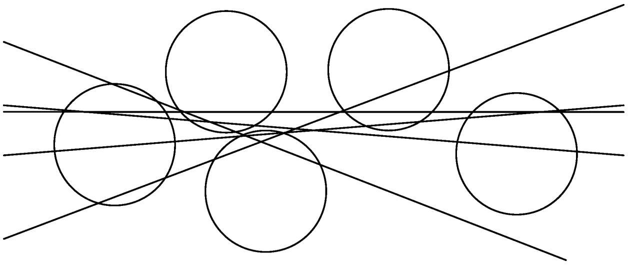


Fig. 3. Example of line transversals for a family of $n = 5$ disks and $k = 3$ (adapted from [68, p. 182]).

$C_i(t) = C_i[\mathbf{x}_i(t), r_i(t)]$ in $\mathbb{R}^2$, with y-intercept b_y, can be represented by the cone that is finitely generated by the following unit vectors:

$$\begin{aligned}\hat{\mathbf{h}}_i(b_y, t) &= \begin{bmatrix} \cos\alpha_i(t) & -\sin\alpha_i(t) \\ \sin\alpha_i(t) & \cos\alpha_i(t) \end{bmatrix} \frac{\mathbf{v}_i(t)}{\|\mathbf{v}_i(t)\|} \\ &= \mathbf{Q}_i^{+}\left[\alpha_i(t)\right]\hat{\mathbf{v}}_i(t) \end{aligned} \tag{13}$$

and

$$\begin{aligned}\hat{\mathbf{l}}_i(b_y, t) &= \begin{bmatrix} \cos\alpha_i(t) & \sin\alpha_i(t) \\ -\sin\alpha_i(t) & \cos\alpha_i(t) \end{bmatrix} \frac{\mathbf{v}_i(t)}{\|\mathbf{v}_i(t)\|} \\ &= \mathbf{Q}_i^{-}\left[\alpha_i(t)\right]\hat{\mathbf{v}}_i(t) \end{aligned} \tag{14}$$

where

$$\mathbf{v}_i(t) \equiv \mathbf{x}_i(t) - [0\ b_y]^T$$

and

$$\alpha_i(t) = \sin^{-1}(r_i(t)/\|\mathbf{v}_i(t)\|).$$

The *coverage cone* $K \subset \mathbb{R}^2$ is finitely generated by the above unit vectors when it contains all of their linear combinations [69], that is

$$\begin{aligned} K\left[C_i(t), b_y\right] &\equiv \operatorname{cone}(\hat{\mathbf{l}}_i, \hat{\mathbf{h}}_i) \\ &= \left\{\mathbf{q} | \mathbf{q} = c_1\hat{\mathbf{l}}_i(b_y, t) + c_2\hat{\mathbf{h}}_i(b_y, t), c_1, c_2 \geq 0\right\}. \end{aligned} \tag{15}$$

Since the unit vectors $\hat{\mathbf{l}}_i(b_y, t)$ and $\hat{\mathbf{h}}_i(b_y, t)$ are implicit functions of $\mathbf{x}_i(t)$ and $r_i(t)$, as shown in (13) and (14), the above coverage cone is a function of the ith sensor's position and range at t. $K[C_i(t), b_y]$ contains the set of all tracks that are detected by the ith sensor at time t, and its opening angle is a Lebesgue measure over this set.

The unit vectors (13) and (14) are also used to determine the stabbers of families of k disks in S, which represent tracks detected by at least k of the n sensors. Let all unit vectors in $\mathbb{R}^2$ be ordered based on the orientation of the xy-frame, where two vectors $\mathbf{q}_i, \mathbf{q}_j \in \mathbb{R}^2$ are ordered as $\mathbf{q}_i \prec \mathbf{q}_j$ if when these vectors are translated such that their origins coincide, and $\mathbf{q}_i$ is rotated through the smallest possible angle to meet $\mathbf{q}_j$, then this orientation is in the same direction as the orientation of the xy-frame [70]. Then, the family of stabbers with y-intercept b_y can be obtained for a family of k disks in $\mathbb{R}^2$, as shown by the following result.

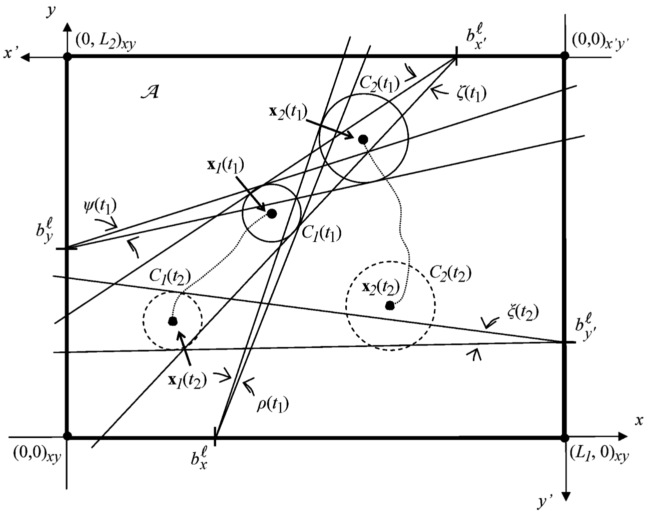


Fig. 4. Reference frames used to define the k-coverage cones illustrated at time t_1 for $S_2(t_1) = \{C_1(t_1), C_2(t_1)\}$, and at time t_2 for $S_2(t_2) = \{C_1(t_2), C_2(t_2)\}$, for a network with $n = k = 2$.

Proposition 3.1: The set of all stabbers of a family of disks $S_k(t) = \{C_1(t), \ldots, C_k(t)\}$, through $y = b_y$, is contained by the finitely generated cone

$$K_k\left[S_k(t), b_y\right] = \operatorname{cone}(\hat{\mathbf{l}}^*, \hat{\mathbf{h}}^*) \tag{16}$$

where

$$\begin{aligned}(\hat{\mathbf{l}}^*, \hat{\mathbf{h}}^*) = (\hat{\mathbf{l}}_\imath, \hat{\mathbf{h}}_\jmath)\ \imath, \jmath \in I_{S_k}, \qquad &\text{such that}\quad \hat{\mathbf{l}}_\imath \succeq \hat{\mathbf{l}}_i(b_y, t), \\ \hat{\mathbf{h}}_\jmath \preceq \hat{\mathbf{h}}_j(b_y, t),\ \hat{\mathbf{l}}_\imath \prec \hat{\mathbf{h}}_\jmath \qquad &\forall i, j \in I_{S_k} \end{aligned} \tag{17}$$

and I_{S_k} denotes the index set of $S_k(t)$. If $\hat{\mathbf{l}}_\imath \succeq \hat{\mathbf{h}}_\jmath$, then $K_k[S_k(t), b_y] = \emptyset$.

A proof is provided in Appendix II.

The cone $K_k[S_k(t), b_y]$ contains the set of tracks detected by a family of k sensors at time t, and is referred to as k-*coverage cone*. The opening angle ψ of the k-coverage cone is a Lebesgue measure over the set of line stabbers of $S_k(t)$ [8], and is obtained by the cross product

$$\psi\left[S_k(t), b_y\right] = \sin^{-1}\|\hat{\mathbf{l}}^* \times \hat{\mathbf{h}}^*\| = H(D_{\imath\jmath})\sin^{-1}(D_{\imath\jmath}) \tag{18}$$

with

$$\begin{aligned} D_{\imath\jmath} \equiv \frac{1}{w_\imath^2 w_\jmath^2} &\left\{\left[x_\imath p_\imath + (y_\imath - b_y)r_\imath\right]\left[x_\jmath r_\jmath + (y_\jmath - b_y)p_\jmath\right]\right. \\ &\left. - \left[x_\jmath p_\jmath - (y_\jmath - b_y)r_\jmath\right]\left[(y_\imath - b_y)p_\imath - x_\imath r_\imath\right]\right\} \end{aligned} \tag{19}$$

where $w_i(t) \equiv \|\mathbf{v}_i(t)\| = \sqrt{x_i^2(t) + [y_i(t) - b_y]^2}$, and $p_i(t) \equiv \sqrt{w_i^2(t) - r_i^2(t)}$ for $i = \imath, \jmath$. In the above equations, the time argument is omitted for brevity, and the indices $\imath$ and $\jmath$ are obtained from (17). The Heaviside function $H(\cdot)$ guarantees that if $\hat{\mathbf{l}}^* \succ \hat{\mathbf{h}}^*$, the opening angle of the coverage cone is equal to zero. By placing a second inertial frame of reference $x'y'$ along the remaining sides of $\mathcal{A}$, Proposition 3.1 can be applied to stabbers that intercept the x-, y'-, and x'-axis at b_x, $b_{y'}$, and $b_{x'}$, respectively, as shown in Fig. 4. The opening angles of the corresponding k-coverage cones are denoted by ρ, ξ, and ζ, respectively, and are illustrated in Fig. 4 for $n = k = 2$. The set of tracks traversing $\mathcal{A}$ and intersecting at least k disks in $S(t)$ is approximated by the union of the k-coverage cones over a finite set of intercept values indexed by ℓ. The intercept values are obtained by discretizing the perimeter of the ROI $\partial\mathcal{A}$, using a constant interval δb. Then, as shown in Appendix III, the following track coverage function is a Lebesgue measure on the set of tracks that intersect at least k disks in $S(t)$ at time t:

$$\begin{aligned}\mathcal{T}_{\mathcal{A}}^{k}[\mathbf{x}(t),\mathbf{r}(t)] &= \frac{1}{2}\sum_{\ell=1}^{L_2/\delta b}\sum_{j=1}^{q}(-1)^{j+1}\sum_{1\leq i_1<\ldots<i_j\leq q}\Big\{\psi\big[S_p^{i_{1,j}}(t),b_y^{\ell}\big] \\ &\qquad +\xi\big[S_p^{i_{1,j}}(t),b_{y'}^{\ell}\big]\Big\} \\ &+\frac{1}{2}\sum_{\ell=0}^{(L_1/\delta b-1)}\sum_{j=1}^{q}(-1)^{j+1}\sum_{1\leq i_1<\ldots<i_j\leq q}\Big\{\rho\big[S_p^{i_{1,j}}(t),b_x^{\ell}\big] \\ &\qquad +\zeta\big[S_p^{i_{1,j}}(t),b_{x'}^{\ell}\big]\Big\}\end{aligned} \tag{20}$$

where the summation $\sum_{1\leq i_1<\ldots<i_j\leq q}$ is the sum over all $[q!/(q-j)!j!]$ distinct integer j-tuples $(i_1,\ldots,i_j)$ satisfying $1 \leq i_1 < i_j \leq q$, and $S_p^{i_l}(t)$ denotes the i_lth p-subset of $S(t)$, at time t [71]. The proof is based on the principle of inclusion–exclusion [8], and is shown in Appendix III. Since the track parameters are uniformly distributed over their range, and all targets are assumed to traverse $\mathcal{A}$ in time ΔT, the network's quality of service can be obtained by integrating (20) with respect to time.

The power required by the n sensors to move and search for a target in $\mathcal{A}$ is minimized to obtain sustainable missions that can last up to several days [24], [27]. By assuming that the thrusters are driven by direct current (dc) motors, and that the power dissipation due to the armature resistance is much smaller than that due to the interaction between the propeller and the water, the mechanical power required by the jth propeller of the ith vehicle can be approximated by

$$p_{ij} = \frac{2\pi K_{Q_{ij}}}{\sqrt{\rho_w}d_{ij}\left|K_{T_{ij}}\right|^{1.5}}|u_{ij}|^{1.5} = \beta_{ij}|u_{ij}|^{1.5} \tag{21}$$

and is a function of the control input u_{ij} [28], where u_{ij} is the jth element in $\mathbf{u}_i$, and β_{ij} is assumed to be a known constant that can be computed from the water density ρ_w, the diameter d_{ij}, and the torque and thrust coefficients $K_{Q_{ij}}$ and $K_{T_{ij}}$ of the jth propeller of the ith vehicle [28]. Let $\mathbf{R} \in \mathbb{R}^{nm\times nm}$ denote a diagonal weighting matrix with elements that represent the relative importance of the nm control inputs given, for example, by the coefficients $\beta_{11},\ldots,\beta_{nm}$. Then, the quadratic integral energy function

$$\mathcal{E}\left[\mathbf{u}(t)\right] = \mathbf{u}^T(t)\mathbf{R}\mathbf{u}(t) = \sum_{i=1}^{n}\sum_{j=1}^{m}\beta_{ij}|u_{ij}|^2 \tag{22}$$

is used to minimize the power required by the sensor network, because it has a smooth gradient variation near the minimum, and it penalizes large power dissipations more heavily than small dissipations [30, p. 190].

The sensor network objective function

$$J = \phi\left[\mathbf{x}(t_f)\right] + \int_{t_0}^{t_f}\left\{W_{\mathcal{T}}\,\mathcal{T}_{\mathcal{A}}^{k}\left[\mathbf{x}(t),\mathbf{r}(t)\right] - W_{\mathcal{E}}\,\mathcal{E}\left[\mathbf{u}(t)\right]\right\}dt \tag{23}$$

combines the track coverage and energy functions to simultaneously optimize the quality of service and power required over the time interval ΔT. The weights $W_{\mathcal{T}}$ and $W_{\mathcal{E}}$ are used to specify the desired tradeoff between these two conflicting objectives. In Section VI-B, a parametric cost–benefit analysis is performed to illustrate how the choice of weights influences the optimal sensors' trajectories. The terminal cost

$$\phi\left[\mathbf{x}(t_f)\right] = W_{\phi}\left[\mathbf{x}_f - \mathbf{x}(t_f)\right]^T\left[\mathbf{x}_f - \mathbf{x}(t_f)\right] \tag{24}$$

can be used to specify desired final sensor positions $\mathbf{x}_f$, for example, in case that the vehicles need to be recovered, or must meet additional objectives. The optimal sensors' trajectories are computed by minimizing the cost function in (23) subject to the system dynamics in (8) and to the inequality constraints described in Section III-D.

D. Inequality Constraints on the State and the Control

The terminal cost function described in Section III-C can be used to express state constraints at the final time. Inequality and equality constraints that must be met at every instant of the trajectories can also be introduced via penalty functions or Lagrange multipliers [30, p. 231]. In the case of a mobile sensor network performing cooperative track detections in $\mathcal{A}$, this class of constraints is used to guarantee that the sensors obtain k-independent detections (Section III-B), remain in $\mathcal{A}$, and demand reasonable control usage at all times.

When two or more FOVs intersect in $\mathcal{A}$, multiple detections by different sensors may be due to the same false alarm, such as the same environmental conditions or false target. Therefore, the ability of a sensor network to provide independent detections in the ROI, also known as AC [72], [73], is maximized by deploying the sensors such that their FOVs are disjoint. When the sensor network is static, this can be accomplished by means of circle packing algorithms [74], [75], or mathematical programming [8], [16]. When the sensor network is mobile, trajectories that prevent intersections between the n FOVs can be computed by introducing an AC function in the Lagrangian, or by means of inequality constraints on the state. Adding an additional term to the Lagrangian is known as a soft constraint, and the result is that a tradeoff between AC and other sensors' objectives is optimized. Introducing hard inequality constraints on the state guarantees that, if a solution is found, its AC is maximum at all times. In this paper, disjoint FOVs are guaranteed by using the c_1-dimensional state inequality constraint $\mathbf{c}_f[\mathbf{x}(t)] \leq \mathbf{0}_{c_1\times 1}$, where each constraint has the form

$$\begin{aligned}&[r_i(t)+r_j(t)]^2 - [x_i(t)-x_j(t)]^2 - [y_i(t)-y_j(t)]^2 \leq 0, \\ &\qquad\qquad i,j = 1,\ldots,n, \qquad i \neq j\end{aligned} \tag{25}$$

and $c_1 = n!/2(n-2)!$.

Since both area and track coverage decrease if the sensors leave $\mathcal{A}$, an additional constraint is introduced to guarantee that the sensors remain in $\mathcal{A}$ during the entire time interval ΔT. Thus, the sensors' trajectories are optimized subject to the $4n$-dimensional state inequality constraint

$$\mathbf{c}_{\mathcal{A}}\left[\mathbf{x}(t)\right] \equiv \begin{bmatrix} \mathbf{x}(t) - \mathbf{L} \\ -\mathbf{x}(t) \end{bmatrix} \leq \mathbf{0}_{4n\times 1} \tag{26}$$

where $\mathbf{L} \equiv [L_1 \; L_2 \; \ldots \; L_1 \; L_2]^T \in \mathbb{R}^{2n}$. Finally, to obtain feasible trajectories, the physical limitations of the vehicles are taken into account by specifying the maximum available thrust of each vehicle denoted by the vector $\mathbf{u}_{\max} \in \mathbb{R}^{nm}$ through the following inequality constraint on the control:

$$\mathbf{c}_u\left[\mathbf{u}(t)\right] = |\mathbf{u}(t)| - \mathbf{u}_{\max} \leq \mathbf{0}_{nm\times 1} \tag{27}$$

where $|\cdot|$ denotes the absolute value of a vector, defined in [76, p. 343]. Then, the inequality constraint (3) for the sensor network is defined as $\mathbf{c} \equiv \left[\mathbf{c}_f^T \; \mathbf{c}_{\mathcal{A}}^T \; \mathbf{c}_u^T\right]^T \leq \mathbf{0}_{r\times 1}$, where $r = c_1 + 4n + nm$. Since all r constraints consist of inequalities, they are evoked only some of the time, and they are not overly restrictive, typically resulting in a successful optimization, as shown by numerical results in Section VI.

As shown in this section, the OC of a mobile sensor network involves nonlinear system dynamics and constraints, and an objective function that is not quadratic. Thus, the optimal sensors' trajectories are computed numerically, using the direct methods reviewed in the next section.

IV. Numerical Solution of the Optimal Control Problem

Nonlinear OC problems with no closed-form solution arise in many practical applications, ranging from spacecraft trajectory optimization [43] to OC of diseases [77]. As a consequence, many numerical methods have been devised in the literature to find approximate near-optimal solutions, based on a parametrization of the control, of the time interval, or both. An excellent survey of numerical methods for solving OC problems can be found in [40]. In recent years, direct methods have become increasingly popular thanks to their ability to solve a wide range of complex OC problems [43]–[45]. Of the methods investigated in this research, which include parametric control [30] and various direct methods [44], [45], it was found that only direct shooting [40], [78], [79] and GPMs [43] were capable of computing near-optimal trajectories for various real-size networks (shown in Section VI-A). In particular, the DSM, reviewed in Section IV-A, was found to be the fastest and most reliable method for solving the OC problem stated in (8) and (23)–(27). When both the system dynamics and the Lagrangian are time invariant, the minimum principle can be used as a global criterion for optimality to evaluate the trajectories obtained by the numerical optimization [30, p. 217]. However, since the sensor network's dynamics (8) are time varying, the trajectories computed by DSM are evaluated by comparing their performance (23) to those computed by GPM (reviewed in Section IV-B), and by performing random initializations to avoid local maxima.

A. Background on Direct Shooting Method

The DSM solves nonlinear OC problems numerically through parametric zero-order hold control of a uniformly discretized dynamical system [40], [78], [79]. The system dynamics, cost function, and inequality constraints are evaluated at discrete points in time, known as *collocation points*, over the interval $\Delta T = [t_0, t_f]$. Then, the nonlinear OC problem can be transformed into an NLP, for which several effective algorithms are available [80], [81]. Between collocation points, the control is assumed to be piecewise constant and the system dynamics (8) are integrated by Euler (rectangular) integration [30, p. 77]. Although this is the crudest type of numerical integration, it was found to be adequate for integrating (8) when compared to more accurate and computationally expensive methods, such as the Runge–Kutta integration.

For N equally spaced collocation points $t_k = t_0 + k\Delta t$, $k = 0, \ldots, (N-1)$, where $\Delta t = (t_f - t_0)/N$ is the discretization interval, let $\boldsymbol{\chi} \in \mathbb{R}^M$ denote an unknown parameter vector that uniquely defines the system state and control trajectories, and is defined as

$$\boldsymbol{\chi} \equiv \begin{bmatrix} \mathbf{u}_0^T & \ldots & \mathbf{u}_{N-1}^T & \mathbf{x}_0^T & \ldots & \mathbf{x}_N^T \end{bmatrix}^T \tag{28}$$

where $\mathbf{x}_k \equiv \mathbf{x}(t = t_k)$, $\mathbf{u}_k \equiv \mathbf{u}(t = t_k)$, and $M = 2(2N+1)n$. Then, the OC problem (Section II) is transformed into a finite-dimensional NLP by discretizing the cost function (2) with respect to time

$$J(\boldsymbol{\chi}) = \phi(\mathbf{x}_N) + \Delta t \sum_{k=0}^{N-1} \mathcal{L}(\mathbf{x}_k, \mathbf{u}_k, t_k) \tag{29}$$

and by performing an implicit integration of the system dynamics (1). Using Euler integration, the residuals

$$\boldsymbol{\rho}_k(\boldsymbol{\chi}) = \mathbf{x}_{k+1} - \mathbf{x}_k - \mathbf{f}(\mathbf{x}_k, \mathbf{p}_k, \mathbf{u}_k, t_k)\Delta t, \qquad k = 0, \ldots, N\text{–}1 \tag{30}$$

are computed at all collocation points using only one function evaluation per residual, and are driven to zero as part of the optimization process [79]. When the inequality constraint (3) is evaluated at every collocation point t_k, it is transformed into N inequality constraints $\mathbf{c}_k(\boldsymbol{\chi}) \equiv \mathbf{c}(\mathbf{x}_k, \mathbf{u}_k, t_k) \leq \mathbf{0}_{r\times 1}$, $k = 0, \ldots, N-1$, that are a function of the parameter vector $\boldsymbol{\chi}$. Thus, the solution $\boldsymbol{\chi}^*$ of the NLP

$$\begin{aligned} &\text{maximize} && J(\boldsymbol{\chi}) && (31)\\ &\text{subject to} && \boldsymbol{\rho}_k(\boldsymbol{\chi}) = \mathbf{0}_{n\times 1}, \; k = 0, \ldots, N-1 && (32)\\ & && \mathbf{c}_k(\boldsymbol{\chi}) \leq \mathbf{0}_{r\times 1}, \; k = 0, \ldots, N-1 && (33) \end{aligned}$$

is used to obtain near-optimal state and control trajectories for the OC problem (1)–(3). The NLP solution can be made arbitrarily close to the OC problem solution by using a higher order integration rule and by choosing the size of the parameter vector arbitrarily large, letting $\Delta t \to 0$ and $M \to \infty$, at the expense of the computation time [40].

In Section VI, the near-optimal trajectories of a cooperative sensor network are obtained by transforming the OC problem (8),(23)–(27) into an NLP, and by computing its solution using

SQP [41], [42]. In order to validate this approach, the performance of the near-optimal trajectories is compared to that of trajectories computed by GPM, as described in Section IV-B.

B. Background on Gauss Pseudospectral Method

Pseudospectral methods are a well-known class of direct methods that parameterize the state and control using global polynomials, and approximate the system dynamics via orthogonal collocation. They have been shown in [82] and [83] to exhibit faster convergence rates than other methods. Recently, the GPM has been successfully implemented for multivehicle trajectory optimization problems in the form (1)–(3) [43], [85]. In GPM, orthogonal collocation of the dynamics (1) is performed at the Legendre–Gauss (LG) points, where the state and control are approximated, but not at the boundary points. This collocation leads to a set of Karush–Kuhn–Tucker (KKT) conditions that are equivalent to the discretized first-order optimality conditions at the LG points, leading to accurate estimates of the state, costate, and control vectors (see [86] and [87] for more details).

In order to apply GPM, the time interval $\Delta T = [t_0, t_f]$ must be transformed to the time interval $\tau \in [-1, 1]$ using the mapping

$$\tau = \frac{2t - (t_f + t_0)}{t_f - t_0} \qquad \forall t \in \Delta T. \tag{34}$$

It follows that the cost function (2) can be written as

$$J = \phi\left[\mathbf{x}(+1)\right] + \frac{t_f - t_0}{2} \int_{-1}^{1} \mathcal{L}\left[\mathbf{x}(\tau), \mathbf{u}(\tau), \tau\right] d\tau \tag{35}$$

and is to be optimized subject to the constraints

$$\dot{\mathbf{x}}(\tau) = \frac{t_f - t_0}{2} \mathbf{f}\left[\mathbf{x}(\tau), \mathbf{p}(\tau), \mathbf{u}(\tau), \tau\right] \tag{36}$$

$$\mathbf{c}\left[\mathbf{x}(\tau), \mathbf{u}(\tau), \tau\right] \leq \mathbf{0}_{r \times 1}. \tag{37}$$

The state and control are parameterized using a basis of Lagrange interpolating polynomials on the interval $\tau \in [-1, 1]$

$$L_k(\tau) = \prod_{j=0, j \neq k}^{N} \frac{\tau - \tau_j}{\tau_k - \tau_j}, \quad L_k(\tau_j) = \delta_{kj} \tag{38}$$

where δ_{kj} is the Kronecker delta, such that

$$\mathbf{x}(\tau) \approx \mathbf{X}(\tau) = \sum_{k=0}^{N} \mathbf{X}(\tau_k) L_k(\tau) \tag{39}$$

and

$$\mathbf{u}(\tau) \approx \mathbf{U}(\tau) = \sum_{k=1}^{N} \mathbf{U}(\tau_k) L'_k(\tau) \tag{40}$$

where L'_k is given by (38) after shifting the index j from 0 to 1. The system dynamics are approximated at the collocation points by differentiating (39) with respect to time

$$\dot{\mathbf{x}}(\tau) \approx \dot{\mathbf{X}}(\tau) = \sum_{k=0}^{N} \mathbf{X}(\tau_k) \dot{L}_k(\tau) \tag{41}$$

and by representing the time derivative of each Lagrange polynomial at the LG points by a differential approximation matrix $\mathbf{N} \in \mathbb{R}^{N \times (N+1)}$, with elements given by

$$\mathbf{D}_{(i,k)} = \dot{L}_k(\tau_i) = \sum_{l=0}^{N} \frac{\prod_{j=0, j \neq k, l}^{N} (\tau_i - \tau_j)}{\prod_{j=0, j \neq k}^{N} (\tau_k - \tau_j)} \tag{42}$$

and computed offline for $i = 1, \ldots, N$ and $k = 0, \ldots, N$.

TABLE I
SENSOR NETWORKS' SIZE AND CPA DETECTION RANGES

n	$\{r_1^{CPA}, \ldots, r_n^{CPA}\}$ (km)
10	{ 3, 3, 5, 5, 6, 6, 8, 8, 10, 10 }
15	{ 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4 }
20	{ 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4 }

With the above transformations, the cost function (35) is approximated using a Gauss quadrature [88], the system dynamics (36) are approximated by algebraic constraints, and the path constraint (37) is evaluated at the LG points, leading to N r-dimensional inequality constraints. Thus, the OC problem (1)–(3) is transformed into the NLP

$$\text{maximize} \quad J = \phi(\mathbf{X}_f) + \frac{t_f - t_0}{2} \sum_{k=1}^{N} w_k \mathcal{L}(\mathbf{X}_k, \mathbf{U}_k, \tau_k) \tag{43}$$

$$\text{subject to} \quad \sum_{k=0}^{N} \mathbf{X}_k \mathbf{D}_{(i,k)} - \frac{t_f - t_0}{2} \mathbf{f}(\mathbf{X}_k, \mathbf{p}_k, \mathbf{U}_k, \tau_k) = \mathbf{0}_{n \times 1}, \quad k = 1, \ldots, N \tag{44}$$

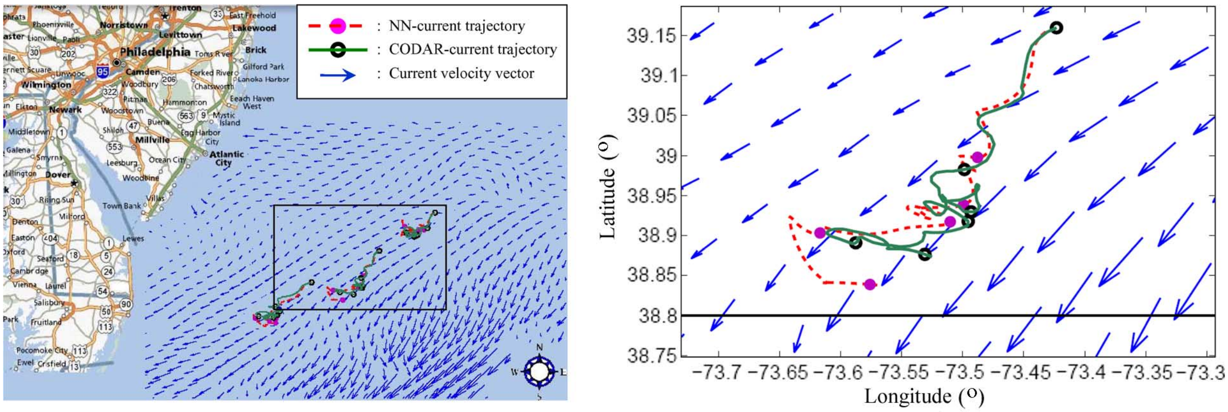
$$\mathbf{c}(\mathbf{X}_k, \mathbf{U}_k, \tau_k) \leq \mathbf{0}_{r \times 1}, \qquad k = 1, \ldots, N \tag{45}$$

with variables $\mathbf{X}_k \equiv \mathbf{X}(\tau_k) \in \mathbb{R}^n$ and $\mathbf{U}_k \equiv \mathbf{U}(\tau_k) \in \mathbb{R}^m$, $k = 1, \ldots, N$, where w_k are the Gauss weights, and $\mathbf{p}_k \in \mathbb{R}^l$ denotes the vector of system parameters evaluated at the collocation point τ_k.

The numerical solution of the above NLP is obtained using the software of [90] and [91] that implements the method developed in [87] and [89], and provides near-optimal state and control trajectories for the OC problem (1)–(3). In the next sections, this approach is implemented to compute the near-optimal trajectories of a cooperative sensor network in an ROI near the New Jersey coast. The results presented in Section VI show that the resulting sensors' trajectories outperform existing techniques, including trajectory planning by area-coverage and energy optimization.

V. IMPLEMENTATION

The OC approach presented in the previous sections is implemented on three networks with the sizes and CPA detection ranges listed in Table I, and different values of required detections k. Every sensor in the network is mobile, and is composed of a low-speed underwater vehicle equipped with an acoustic sensor. As an example, the network is deployed in an ROI $\mathcal{A}$ of dimensions $L_1 = 90$ km and $L_2 = 82.5$ km, shown by a black solid line in Fig. 5(a), for a time interval ΔT lasting up to five days. The CPA detection ranges (Table I) can be estimated



(a) (b)

Fig. 5. Point-mass trajectories obtained from the neural network model and from CODAR measurements in an ROI (black rectangle) with longitude of 72.7° W to 74.1° W, and latitude of 38.6° N to 39.5° N (taken from [54]).

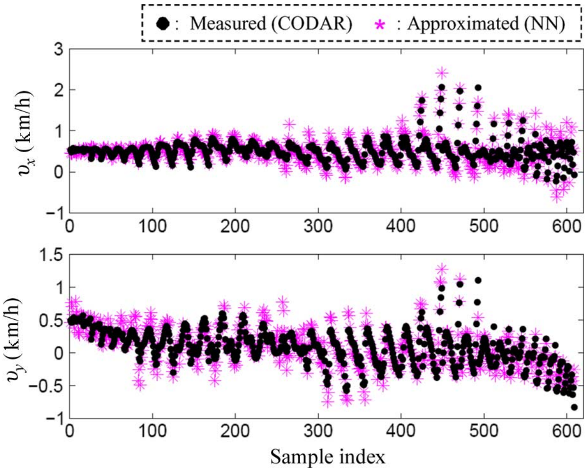


Fig. 6. Comparison of neural network approximation and measured speed components at 600 sample locations in the ROI, at time $t = 122$ h (taken from [54]).

from the sensor operating conditions [35] and are used to compute the sensors' effective ranges, as shown in (12). The system dynamics are simulated by integrating (8) explicitly using a fourth-order Runge–Kutta routine with variable stepsize [92]. This routine was also used to verify that the Euler integration used by the DSM (Section IV-A) brings about a negligible error, with a mean on the order of 1 km over the entire ΔT.

The time-varying ocean current field in and around $\mathcal{A}$ is simulated using real coastal ocean dynamics applications radar (CODAR) data provided by the COOL [38]. A snapshot of the current speed field is shown in Fig. 5 by plotting sample vectors whose length is proportional to the speed's magnitude, at a representative moment in time. The current speed forecast is represented by a feedforward neural network (9), with $s = 100$, obtained by the approach in Section III-A. The database of CODAR measurements was also used to simulate a five-day forecast $\mathcal{F}$ with $p = 12\,525$, via sampling techniques, and the remaining data were used to validate the neural network model. Fig. 6 shows a comparison between the neural network output $\boldsymbol{v}_i(t)$ and the measured speed components drawn from both the training and the validation data at 610 sample locations, and $t\ = 122$ h. The neural network approximation of the speed components v_x and v_y at two sample moments in time is plotted in Fig. 7 together with the CODAR measurements represented by black dots. The effectiveness of the approximation is also validated by comparing the trajectories of point masses placed in the speed field modeled by the neural network (9) to those obtained using the CODAR database combined with a cubic-spline interpolation routine ("trainbr" [93]). Each point mass is randomly placed in the ROI and allowed to drift according to a second-order Newtonian dynamic model [94] for a period of five days. As shown by the three examples in Fig. 5(a), and by the higher resolution example in Fig. 5(b), the trajectories obtained with the neural network approximator are very close to those obtained from the real CODAR measurements.

VI. Results

In this section, DSM is first validated by comparing its performance to GPM, which has been shown effective at solving trajectory optimization problems involving multiple spacecraft [43], [85]. The results in Section VI-A show that DSM trajectories have a slightly higher performance than GPM trajectories, and require significantly less computation time. The influence of the objective function weighting (23) on the sensors' trajectories is illustrated by conducting a parametric study in Section VI-B, where the Pareto front is obtained numerically for a network with $n = 10$ sensors, and $k = 3$ required detections. Then, in Section VI-C, the effectiveness of the OC approach is demonstrated by comparing the quality of service of networks deployed by OC to that of sensors deployed by existing strategies, such as area-coverage optimization [33], [72], [73], and minimum-energy trajectory planning [26]–[28]. Finally, in Section VI-D, the versatility of the approach presented in this paper is demonstrated by modifying the OC formulation to include additional constraints, such as minimum required track coverage over ΔT.

A. Comparison Between DSM and GPM Solution Methods

In this research, several numerical methods were investigated for the solution of the OC problem in (8) and (23)–(27). It was

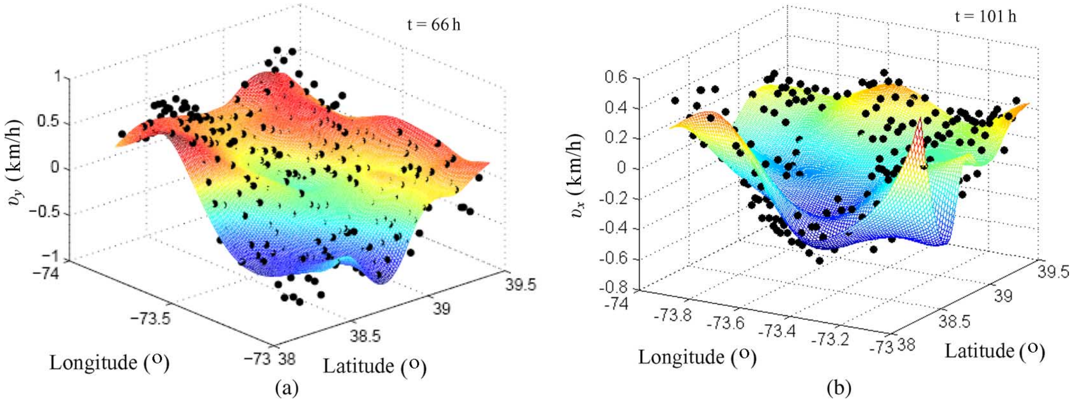


Fig. 7. Neural network approximation of the speed components as a function of position at (a) $t = 66$ h and (b) $t = 101$ h, with available CODAR measurements superimposed in black dots (taken from [54]). Latitude: N; longitude: W.

TABLE II
COMPARISON BETWEEN DSM AND GPM NUMERICAL SOLUTIONS

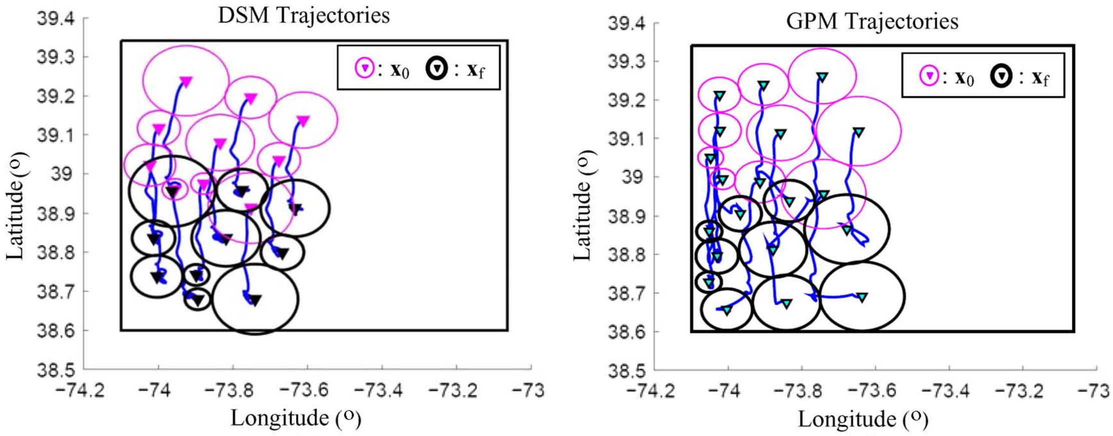
Mission Parameters	Performance	DSM	GPM	DSM Improvement
$(n, k) = (10,3)$ x_0 fixed $\Delta T = 120$ h	Total Track coverage	2.93×10^4	2.81×10^4	4%
	Total Energy	924	1.41×10^3	35%
	Objective Function	2.84×10^4	2.66×10^4	7%
	Computation Time (h)	36	72	50%
$(n, k) = (10,3)$ x_0 free $\Delta T = 72$ h	Total Track coverage	1.82×10^4	1.81×10^4	0.5%
	Total Energy	165	336	51%
	Objective Function	1.81×10^4	1.78×10^4	2%
	Computation Time (h)	48	96	50%

found that, due to the dimensionality of the dynamic equation (8) and to the complexity of the cost function (23), only DSM and GPM were capable of computing near-optimal trajectories for real-size networks, such as those in Table I. Since the sensor network's dynamics (8) are time varying, the optimality of the numerical solutions cannot be verified by the minimum principle. A heuristic approach to evaluating the solutions consists of comparing the results of two independent approximation methods [80]. Thus, in this section, the performance of the DSM trajectories, as expressed by the objective function (23), is compared to that of the GPM trajectories. Also, random initializations are used for both methods to avoid local maxima and obtain trajectories that can be considered to be reasonable approximations to the global solutions.

The performance comparison summarized in Table II is representative of extensive numerical studies involving DSM and GPM, conducted on an Intel (R) Core (TM) 2 Duo Processor T7500 (4M Cache, 2.20 GHz, 800-MHz FSB). Methods to reduce the computation time are currently being investigated for real-time OC implementations, and will be the subject of a separate paper. The objective function J, defined in (23) as a tradeoff between energy and track coverage, and the individual performance functions are all shown in Table II for two examples, and $(W_{\mathcal{E}} : W_{\mathcal{T}}) = (1 : 1)$. In the first example, the time interval is chosen as $\Delta T =$ [0, 120] h, and the initial sensors' placement $\mathbf{x}_0$ is given. In the second example, the time interval is chosen as $\Delta T =$ [0, 72] h, and $\mathbf{x}_0$ is optimized together with the state and control trajectories. These results show that the performance of the DSM solutions is slightly higher than that of the GPM solutions, and that the DSM solutions are computed in approximately half the time. As shown in Fig. 8, the DSM state trajectories are very similar to those obtained by GPM, indicating that they likely are near a global optimum. Similar results are obtained when comparing the time histories of the performance functions obtained by DSM and GPM, plotted in Fig. 9, and those of the OC inputs (omitted for brevity). It can be concluded that for this OC problem DSM is faster than GPM, and converges to solutions that are near a global optimum. Therefore, DSM is implemented in all of the numerical studies presented in the remainder of this paper. The clustered formations near the corners of the ROI in Fig. 8 are obtained because they maximize track coverage by intersecting a large percentage of tracks in $\mathcal{A}$ multiple (k) times (as is also shown in [8]). As shown in Table II, when the initial placement $\mathbf{x}_0$ is optimized, the energy consumption is greatly reduced because OC exploits the current field model to place the sensors such that trajectories with high track coverage (e.g., clustered formations) can be achieved with lesser effort.

B. Numerical Analysis of Objective Function Weighting

According to a classical OC approach [30], competing objectives such as maximizing track coverage and minimizing energy can be combined into a single objective function by changing their influence on J through weights specified by the user (e.g., $W_{\mathcal{T}}$ and $W_{\mathcal{E}}$). Because selecting the correct weights is an important but sometimes difficult step, a parametric study is con-



(a)

(b)

Fig. 8. Comparison of optimal initial positions and state trajectories obtained by (a) DSM and (b) GPM for $n = 10$, $k = 3$, and $\Delta T = 72$ h. Latitude: N; longitude: W.

TABLE III
PARAMETRIC STUDY

$(W_{\mathcal{E}} : W_{\mathcal{T}})$	Total Energy	Total Track Coverage	Objective Function
(1:0)	2.07×10^{-5}	9.41×10^{2}	-2.09×10^{-3}
(100:1)	1.72×10^{-2}	1.33×10^{3}	132
(1:1)	4.85	1.73×10^{3}	8.48×10^{3}
(1:100)	1.65×10^{2}	1.82×10^{3}	1.80×10^{4}
(0:1)	4.60×10^{2}	1.83×10^{3}	1.85×10^{4}

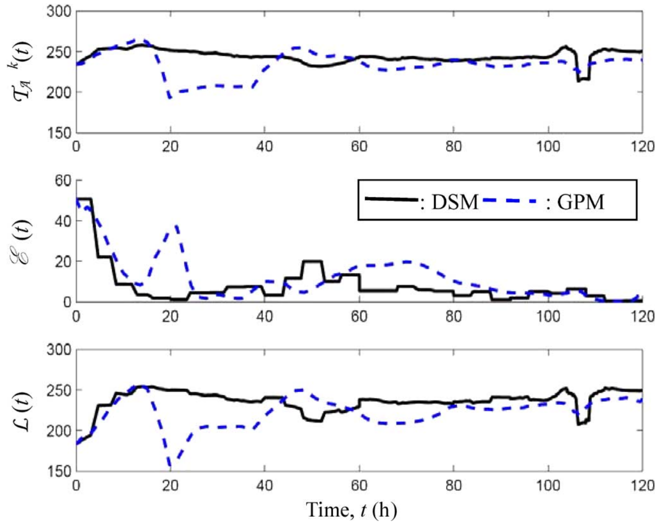


Fig. 9. Comparison of track coverage, energy, and Lagrangian time histories obtained by DSM and GPM for $n = 10$, $k = 3$, $\Delta T = 120$ h, and a fixed $\mathbf{x}_0$.

ducted in this section to determine the sensitivity of the optimal sensor trajectories to the magnitudes of these weights, and to prevent an improper choice of weights that could result in a bias toward either objectives [31], [95]. The parametric study considers multiple combinations of $W_{\mathcal{T}}$ and $W_{\mathcal{E}}$, obtained such that $W_{\mathcal{E}} + W_{\mathcal{T}} = K$, by letting $W_{\mathcal{T}} = \kappa$, and $W_{\mathcal{E}} = K - \kappa$, for a variable parameter $\kappa \in [0, K]$, and a user-defined constant K that is chosen based on the units of the objective functions. The OC solution is controlled by the weights $W_{\mathcal{T}}$ and $W_{\mathcal{E}}$ [95], and a Pareto front can be obtained by plotting the optimal values of energy versus track coverage.

The study is conducted for a sensor network with $n = 10$, $k = 3$, and $K = 101$, optimizing the initial sensor positions $\mathbf{x}(t_0)$ together with the state and control trajectories. For each combination of weights, obtained by varying the parameter κ over its range, the optimal trajectories of the sensor networks are computed by DSM, and the optimal total track coverage and total energy are obtained by integrating $\mathcal{T}_{\mathcal{A}}^{k}$ in (20) and $\mathcal{E}$ in (22) over the time interval ΔT. The results in Table III are interpolated to obtain an approximation of the Pareto front, shown in Fig. 10(a), which represents a set of optimal solutions with different tradeoffs between the two competing objectives [96]. Although in some missions the user may wish to emphasize either energy or track coverage, in many applications, the goal is to find the optimal solution that offers the least objective conflict, also known as Pareto optimum [95]. As shown in Fig. 10(a), this solution can be found from the "knee" of the Pareto front, for which no single objective can be improved upon without deteriorating the other.

The influence of the weights on the OC solutions can be seen in Fig. 10(b)–(d), where the optimal sensors' trajectories are plotted for $(W_{\mathcal{T}} : W_{\mathcal{E}}) = (0 : 1), (1 : 1)$, and $(1 : 0)$. As can be expected, when $(W_{\mathcal{T}} : W_{\mathcal{E}}) = (0 : 1)$, the sensors follow the currents and provide poor track coverage compared to the other solutions, resulting in trajectories that are qualitatively similar to the zero-control solutions (not shown for brevity). When energy and track coverage are equally weighted, the solution is

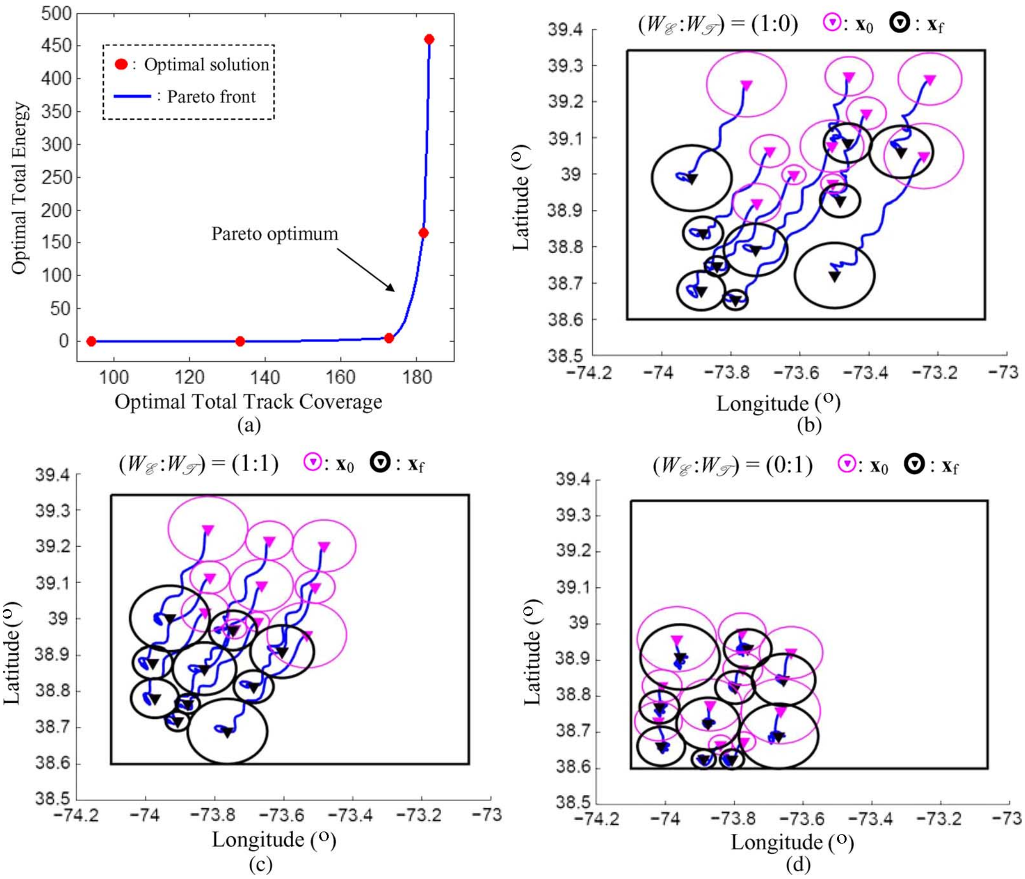


Fig. 10. Numerical approximation of Pareto front and (a) optimal sensors' trajectories for $(W_{\mathcal{E}} : W_{\mathcal{T}})$ equal to (b) $(1:0)$, (c) $(1:1)$, and (d) $(0:1)$. Latitude: N; longitude: W.

close to the Pareto optimum, and the sensors move efficiently by exploiting the ocean currents, but also maneuver to increase track coverage. When $(W_{\mathcal{T}} : W_{\mathcal{E}}) = (1:0)$, the optimal initial positions and trajectories are such that the sensors attempt to remain stationary near a configuration with optimal (static) track coverage, at the expense of the energy required. Due to symmetry, reflections of this configuration also exhibit optimal track coverage. However, the solution in Fig. 10(d) is obtained by virtue of the ocean currents [Fig. 5(a)], which in this region produce an external forcing in the southwest direction and, then, die out in the bottom-left corner of $\mathcal{A}$, allowing sensors to remain there with little or zero energy consumption. Since in simulations the Pareto optimum [Fig. 10(a)] was found to be robust to the OC conditions and parameters, the corresponding set of weights, obtained by interpolation, was adopted for all subsequent trajectory optimizations.

C. Comparison Between OC and Existing Deployment Methods

Although existing sensor-deployment methods do not address the track coverage of a mobile sensor network in a variable ocean environment, very effective methods have been developed for optimizing the AC of static sensor networks [33], [72], [73], and for generating minimum-energy trajectories in a time-invariant ocean current field [26]–[28]. In this section, these methods are extended and implemented on the cooperative sensor networks in Table I to compare them to the OC approach presented in this paper and illustrate its effectiveness.

AC is a well-known performance measure for sensor networks with binary sensor models [33], [72], [73], and is defined as the union of the areas representing the sensors' FOVs divided by the areas of the ROI [33]. The approach of placing sensors in an ROI such that their AC is maximized can be extended to the case of a mobile sensor network by modifying the OC approach in Sections III and IV to compute minimum-energy trajectories that guarantee disjoint FOVs. When the ith sensor's FOV can be represented by the disk $C_i(\mathbf{x}_i, r_i)$, the network's AC is given by

$$A_C(\mathbf{x},\mathbf{r}) \equiv \bigcup_{i=1}^{n} \frac{A_i}{L_1 L_2} \leq \frac{1}{L_1 L_2} \sum_{i=1}^{n} \pi r_i^2 \equiv A_C^* \qquad (46)$$

where A_i denotes the area covered by the ith sensor. Then, the maximum AC A_C^* is attained when the sensors' positions $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are such that their FOVs $C_1, \ldots, C_n$ lie entirely in $\mathcal{A}$ and do not intersect. When the sensors are static, circle packing algorithms can be used to find the sensors' positions in $\mathcal{A}$ that lead to A_C^* [74], [75]. When the sensors are mobile their trajectories must be generated such that these conditions are satisfied at all times. For comparison, minimum-energy trajectories with maximum AC are computed by maximizing (23) with $W_{\mathcal{E}} = 1$ and $W_{\phi} = W_{\mathcal{T}} = 0$, subject to the dynamic constraint (8), and the inequality constraints (25) and (26), using DSM.

TABLE IV
COMPARISON BETWEEN OPTIMAL CONTROL AND OTHER DEPLOYMENT TECHNIQUES

Mission Parameters	Performance	Optimal Control	Area Coverage	Path Planning	Zero Control
$(n, k) = (10,3)$ x_0 fixed $\Delta T = 120$ h	Total Track Coverage	2.93×10^4	1.71×10^4	1.67×10^4	1.27×10^4
	Total Energy	924	5.44×10^{-1}	233	0
	Total Performance (J)	2.84×10^4	1.82×10^4	1.65×10^4	1.27×10^4
	OC Improvement	n/a	56%	72%	123%
$(n, k) = (10,3)$ x_0 free $\Delta T = 72$ h	Total Track Coverage	1.82×10^4	9.73×10^2	1.49×10^4	same as above
	Total Energy	165	1.51×10^{-4}	705	
	Total Performance (J)	1.81×10^4	1.21×10^4	1.42×10^4	
	OC Improvement	n/a	50%	28%	
$(n, k) = (15,3)$ x_0 fixed $\Delta T = 72$ h	Total Track Coverage	1.74×10^4	8.02×10^3	8.87×10^3	1.99×10^3
	Total Energy	1.31×10^3	141	604	0
	Total Performance (J)	1.74×10^4	7.88×10^3	8.27×10^3	1.99×10^3
	OC Improvement	n/a	121%	111%	776%
$(n, k) = (20,4)$ x_0 free $\Delta T = 72$ h	Total Track Coverage	1.79×10^4	6.73×10^3	1.23×10^4	3.14×10^3
	Total Energy	269	9.82×10^{-4}	1.56×10^4	0
	Total Performance (J)	1.77×10^4	6.73×10^3	1.07×10^4	3.14×10^3
	OC Improvement	n/a	163%	65%	464%

PP approaches compute minimum-energy collision-free trajectories for a network of n vehicles that must travel from a set of given initial positions $\mathbf{x}(t_0) = \mathbf{x}_0$ to a set of given final position $\mathbf{x}(t_f) = \mathbf{x}_f$ in $\mathcal{A}$ [26]. Similarly to [26], in this paper, PP is implemented by minimizing the energy required by the vehicles subject to the ocean speed forecast in (9). Collisions between the sensors are avoided by modifying the inequality constraint (25) to include a minimum safety distance in place of the FOV radius r_i. Since the network must perform cooperative track detections, $\mathbf{x}_0$ and $\mathbf{x}_f$ are chosen as sensor network configurations that maximize the instantaneous track coverage function (20). Subsequently, DSM is used to compute noncooperative minimum-energy trajectories by maximizing (23) with $W_{\mathcal{E}} = W_{\phi} = 1$ and $W_{\mathcal{T}} = 0$, subject to (8) and (25). Finally, to demonstrate the advantage of onboard controls and actuators, zero-control trajectories are simulated by setting the thrust inputs in (8) equal to zero at all times, thereby allowing the sensors to drift in $\mathcal{A}$ similarly to point masses.

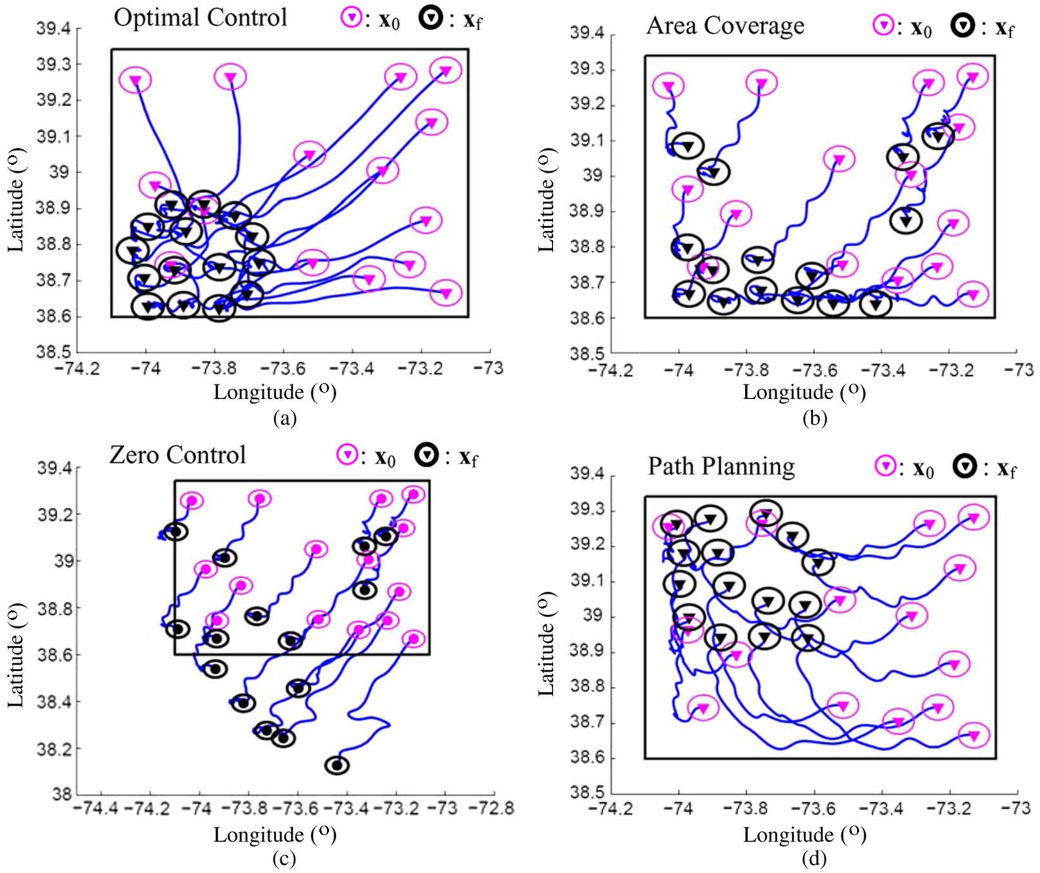
The performance of sensor networks deployed using OC, AC, PP, and zero control (ZC) is summarized in Table IV for four missions involving the networks in Table I. A comparison of sensors' trajectories obtained by the four deployment methods is shown in Fig. 11 for a network with $n = 15$, $k = 3$, a given (fixed) $\mathbf{x}_0$, and $\Delta T = 72$ h. It can be seen from Table IV that by implementing the OC approach presented in this paper, the total sensor network performance (J), defined in (23) as a weighted combination of track coverage and energy, is improved by up to 776% compared to the zero-control solution, and by up to 163% and 111% compared to the area-coverage and path-planning solutions, respectively. Although energy consumption typically is higher for the optimally controlled network, as a result, OC also improves track coverage by up to one order of magnitude (Table IV). Since PP optimizes track coverage at t_0 and t_f, it outperforms the AC and zero-control techniques (Table IV). However, OC always outperforms PP, because it maximizes track coverage throughout the time interval ΔT, and it can simultaneously optimize $\mathbf{x}(t_0)$, $\mathbf{x}(t_f)$, and state and control trajectories, thereby considering extremals with all possible initial and final conditions. As a result, track coverage is considerably improved at all times, as shown by the time histories plotted in Fig. 12.

D. Optimal Sensors' Trajectories Subject to Track Coverage Constraints

Sensor networks' applications often require that a minimum required quality of service, such as track coverage, be maintained at all times using minimum resources. Sequential deployment algorithms based on path exposure or NLP were presented in [67] and [8], respectively, to achieve a required probability of detection in static sensor networks. As shown in Sections VI-A–VI-C, when the sensor network is mobile, its quality of service changes over time, and is a function of the system dynamics and environmental conditions. Hence, maintaining a minimum required quality of service can be more challenging as well as more crucial to the sensor network operability. Another advantage of the OC approach presented in this paper is that different objectives and constraints can be accommodated by the same formalism, without the need for reformulating the problem or for changing the solution method.

As an example, in this section, the sensors' trajectories are computed such that the network is guaranteed to maintain a minimum required track coverage $\hat{\mathcal{T}}_{\mathcal{A}}^k$ at all times, with minimal energy consumption. A similar approach can be used to optimize track coverage subject to maximum allowable energy constraints, e.g., to facilitate recovery of the sensors at t_f. Thanks to the OC problem formulation, this is easily accomplished by including an additional scalar constraint

$$c_{\mathcal{T}} = \hat{\mathcal{T}}_{\mathcal{A}}^k - \mathcal{T}_{\mathcal{A}}^k[\mathbf{x}(t), \mathbf{r}(t)] \leq 0 \tag{47}$$



(a) (b) (c) (d)

Fig. 11. Comparison of sensors' trajectories obtained by (a) OC, (b) AC, (c) ZC, and (d) PP, for a network with $n = 15$, $k = 3$, $\mathbf{x}_0$ fixed, and $\Delta T = 3$ days. Latitude: N; longitude: W.

TABLE V
PERFORMANCE OF TRACK COVERAGE CONSTRAINED TRAJECTORIES

$\hat{\mathcal{T}}_\mathcal{A}^k$	Total Energy	Total Track Coverage
290	0.591	2.19×10^4
300	189	2.26×10^4
315	254	2.35×10^4

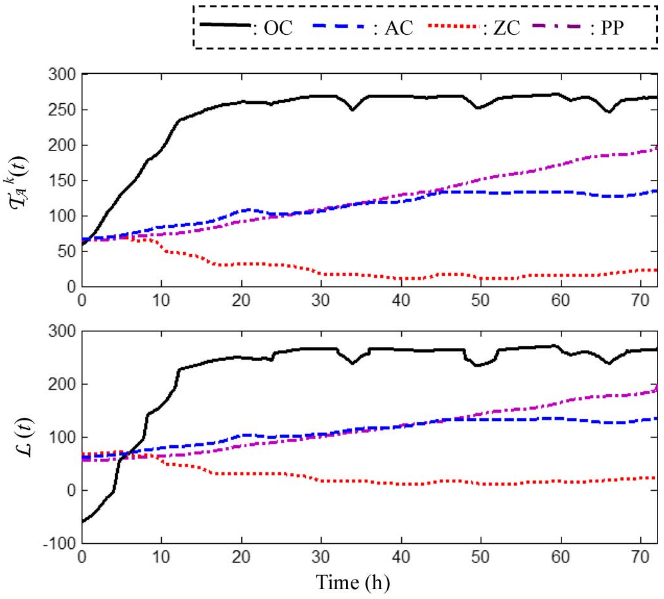


Fig. 12. Comparison of instantaneous track coverage $\mathcal{T}_\mathcal{A}^k(t)$, and objective function $\mathcal{L}(t)$, obtained by OC, AC, ZC, and PP.

in (3), letting $\mathbf{c} \equiv [\mathbf{c}_f^T\ \mathbf{c}_\mathcal{A}^T\ \mathbf{c}_u^T\ c_\mathcal{T}]^T$, $r = c_1 + 4n + nm + 1$, and $W_\mathcal{T} = 0$ in (23). In this study, both $\mathbf{x}_0$ and the sensors' state and control trajectories are optimized subject to both dynamic and inequality constraints. The numerical results are presented in Table V for a sensor network with $n = 20$ and $k = 3$ (Table I), and three levels of required track coverage. The time histories of the optimal track coverage and energy consumption are plotted in Fig. 13. It can be seen from Fig. 13(a) that the track coverage of the sensor network remains above $\hat{\mathcal{T}}_\mathcal{A}^k$ at all times for all three examples. However, the energy required is significantly higher when $\hat{\mathcal{T}}_\mathcal{A}^k$ is equal to 300 and 315, as shown in Fig. 13(b) and Table V. As shown in Fig. 14, the optimal initial positions and state trajectories can be highly influenced by the value of $\hat{\mathcal{T}}_\mathcal{A}^k$. In this example, due to the nature of the ocean currents [Fig. 5(a)], the energy required by the solutions with $\hat{\mathcal{T}}_\mathcal{A}^k$ equal to 300 and 315 is utilized mostly during the first day, while later the sensors require little or no control to remain in the high track coverage configuration located in the bottom-left corner of $\mathcal{A}$ [Fig. 14(b)]. When $\hat{\mathcal{T}}_\mathcal{A}^k$ is equal to 290, the amount of energy required to operate the sensor network is very small, and is more evenly distributed throughout ΔT [Fig. 13(b)].

From these results, it can be concluded that highly efficient network deployments, such as the one shown in Fig. 14(a), can be obtained by optimizing the sensors' initial positions, as well

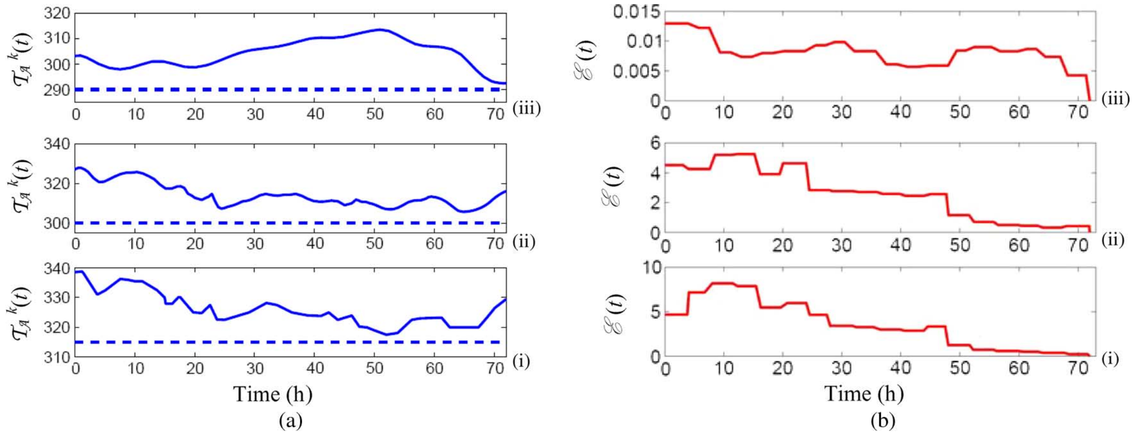


Fig. 13. Time histories of (a) track coverage and (b) energy for $\hat{\mathcal{T}}_{\mathcal{A}}^{k}$ equal to (i) 315, (ii) 300, and (iii) 290.

as their state and control trajectories. Similarly to other OC problems, such as spacecraft trajectory optimization [30], [43], the optimal sensors' trajectories are very sensitive to inequality constraints. Therefore, the approach presented in this section is convenient only when energy is of great concern, and a low but fairly constant quality of service is necessary. When the quality of service is to be maximized, the multiobjective optimization approach utilized in the previous sections computes the trajectories with the best tradeoff between energy and track coverage.

VII. Summary and Conclusion

An approach is presented for optimally controlling an underwater sensor network that is deployed to cooperatively track unauthorized targets in an ROI. An integral objective function representing the sensor network's track coverage over time is obtained using geometric transversals. A differential equation model of the sensor network's dynamics is obtained by considering three dependent subsystems, i.e., underwater vehicles, onboard sensors, and oceanographic fields. Each sensor-equipped vehicle is modeled as a disk, with a time-varying radius, that moves according to underwater-vehicle dynamics. By this approach, sensors' trajectories that maximize track coverage and minimize energy consumption can be computed using a DSM, which transcribes the OC problem into an NLP. The approach is implemented on three simulated sensor networks that are deployed in a simulated ROI near the New Jersey coastline. The numerical results show that OC significantly improves the sensor network's performance compared to AC and PP methods. Also, it can be used to incorporate useful inequality constraints on the sensors' performance, state, and control vectors, and to generate fronts of Pareto optimal trajectories.

Appendix I
Coverage Cone

Let $\mathcal{R}_\theta(b_y)$ denote a ray that intersects a disk $C_i(t) = C_i[\mathbf{x}_i(t), r_i(t)]$ with radius $r_i(t)$, and centered at $\mathbf{x}_i(t)$ at time $t \in \Delta t$, in $\mathbb{R}^2_+$. Consider any two points that lie on $\mathcal{R}_\theta(b_y)$ and inside $C_i(t)$, and let $\mathbf{q}_1, \mathbf{q}_2 \in \mathbb{R}^2_+$ denote their positions relative to the origin b_y of the coverage cone $K[C_i(t), b_y]$. By construction, $\mathbf{q}_1, \mathbf{q}_2 \in C_i(t)$, and a vector $\mathbf{z}$ joining the two points will lie on the ray $\mathcal{R}_\theta(b_y)$. Let c_1 and c_2 denote any two positive constants. By definition of vector sum and subtraction [70], if $\mathbf{z} = c_1\mathbf{q}_1 + c_2\mathbf{q}_2$, then $\mathbf{z}$ has the same origin as $\mathbf{q}_1$ and $\mathbf{q}_2$. Thus, since $\mathbf{z}$ lies on $\mathcal{R}_\theta(b_y)$, $\mathcal{R}_\theta(b_y)$ intercepts the y-axis at the cone's origin b_y. If $\mathbf{z} = \pm c_1\mathbf{q}_1 \mp c_2\mathbf{q}_2$, $\mathbf{z}$ does not have the same origin as $\mathbf{q}_1$ and $\mathbf{q}_2$ and, thus, $\mathcal{R}_\theta(b_y)$ does not intercept the y-axis at b_y. By definition (15), $K[C_i(t), b_y]$ is the set of all nonnegative combinations of the elements in $C_i(t)$. Since $\mathbf{q}_1$ and $\mathbf{q}_2$ are two elements in $C_i(t)$, and any nonnegative combination of these two elements can be written as $c_1\mathbf{q}_1 + c_2\mathbf{q}_2$, with $c_1, c_2 > 0$, it follows that $\mathbf{z} = c_1\mathbf{q}_1 + c_2\mathbf{q}_2 \in K[C_i(t), b_y]$. Finally, since $\mathcal{R}_\theta(b_y)$ denotes any ray with intercept b_y that intersects $C_i(t)$ in $\mathbb{R}^2_+$, and $\mathbf{z} = c_1\mathbf{q}_1 + c_2\mathbf{q}_2$ provided that $\mathcal{R}_\theta(b_y)$ intersects the y-axis at b_y, it also follows that any $\mathcal{R}_\theta(b_y)$ that intersects both $C_i(t)$ and the y-axis at b_y is contained by $K[C_i(t), b_y]$ in (15).

Appendix II
Proof of Proposition 3.1

This proof considers a family of $k = 3$ disks $S_k(t) = \{C_i(t), C_j(t), C_l(t)\}$ with index set $I_{S_k} = \{i, j, l\}$. The results can be extended to higher k by induction. The coverage cone $K[C_\ell(t), b_y]$ contains the set of all rays that intersect $C_\ell(t)$ in $\mathbb{R}^2_+$, at time t, where $\ell \in I_{S_k}$. Then, from set theory, the set of tracks intersecting all circles in the family $S_k(t)$ is given by the following intersection:

$$\begin{aligned} K_k\left[S_k(t), b_y\right] &= \bigcap_{\ell \in I_{S_k}} K\left[C_\ell(t), b_y\right] \\ &= K\left[C_i(t), b_y\right] \cap K\left[C_j(t), b_y\right] \\ &\quad \cap K\left[C_l(t), b_y\right]. \end{aligned} \tag{48}$$

From the properties of cones [69, p. 70], the intersection of a collection of cones is also a cone and, thus, $K_k[S_k(t), b_y]$ is a cone. A vector $\mathbf{z}$ representing a ray $\mathcal{R}_\theta$ with the same slope and origin lies in a cone K if and only if $\mathcal{R}_\theta$ lies in K, since any point on $\mathcal{R}_\theta$ can be written as $c\mathbf{z}$, with $c > 0$.

Consider a ray $\mathcal{R}_\theta \in K[C_\ell(t), b_y]$, where $K[C_\ell(t), b_y] = \text{cone}(\hat{\mathbf{l}}_\ell, \hat{\mathbf{h}}_\ell)$, and thus can be represented by a vector $\mathbf{z}_\ell = c_1\hat{\mathbf{l}}_\ell + c_2\hat{\mathbf{h}}_\ell$ with constants $c_1, c_2 > 0$. Then, $\mathbf{z}_\ell \in K[C_\ell(t), b_y]$ and, by the properties of the vector sum, $\hat{\mathbf{l}}_\ell \prec \mathbf{z}_\ell \prec \hat{\mathbf{h}}_\ell$. Next,

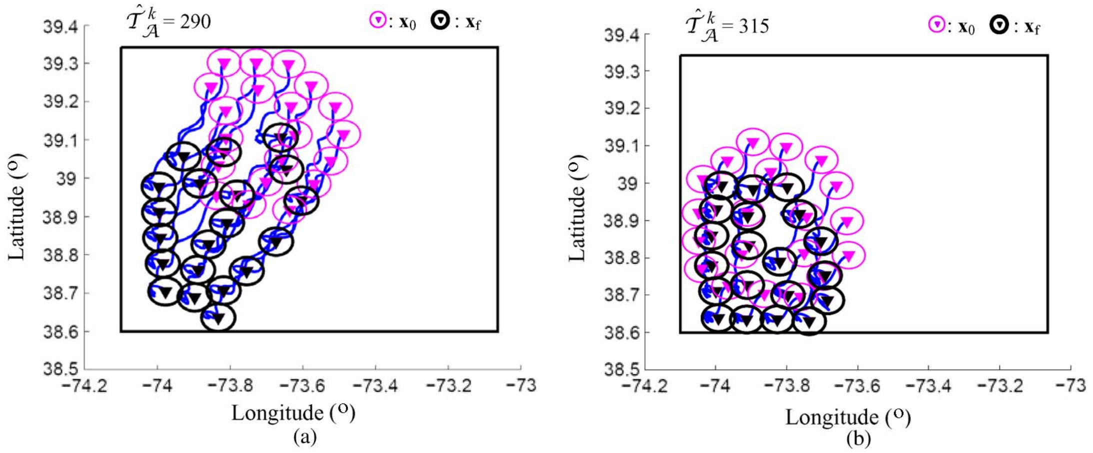


Fig. 14. Optimal sensors' trajectories and initial positions for $\hat{\mathcal{T}}^k_{\mathcal{A}}$ equal to (a) 290 and (b) 315. Latitude: N; longitude: W.

consider a cone $K^* = \mathrm{cone}(\hat{\mathbf{l}}^*, \hat{\mathbf{h}}^*)$ that is finitely generated by two unit vectors $\hat{\mathbf{h}}^* = \hat{\mathbf{h}}_\jmath$ and $\hat{\mathbf{l}}^* = \hat{\mathbf{l}}_\imath$ with $\jmath, \imath \in I_{S_k}$, and assume $\hat{\mathbf{l}}_\imath \prec \hat{\mathbf{h}}_\jmath$. By the properties of finitely generated cones [69], defined as in (15), any vector $\mathbf{z}^* = b_1\hat{\mathbf{l}}^* + b_2\hat{\mathbf{h}}^*$ with constants $b_1, b_2 > 0$ must lie in K^*. It follows that a ray $\mathcal{R}^*_\theta$ with the same slope and origin as $\mathbf{z}^*$ must also lie in K^*, since any point on $\mathcal{R}^*_\theta$ can be written as $c\mathbf{z}^*$, with $c > 0$. Since $\mathbf{z}^*$ is a linear combination of $\hat{\mathbf{l}}^*$ and $\hat{\mathbf{h}}^*$, it also follows that $\hat{\mathbf{l}}^* \prec \mathbf{z}^* \prec \hat{\mathbf{h}}^*$.

According to Proposition 3.1, choose $\hat{\mathbf{h}}^* = \hat{\mathbf{h}}_\jmath \preceq \hat{\mathbf{h}}_\ell$ and $\hat{\mathbf{l}}^* = \hat{\mathbf{l}}_\imath \succeq \hat{\mathbf{l}}_\ell \; \forall \ell \in I_{S_k}$. Suppose the unit vectors of $S_k(t)$ can be ordered as $\hat{\mathbf{h}}_l \prec \hat{\mathbf{h}}_j \prec \hat{\mathbf{h}}_i$ and $\hat{\mathbf{l}}_i \prec \hat{\mathbf{l}}_l \prec \hat{\mathbf{l}}_j$. Then, the unit vectors and $\mathbf{z}^*$ can be ordered as follows:

$$\hat{\mathbf{l}}_\ell \preceq \hat{\mathbf{l}}_j = \hat{\mathbf{l}}^* \prec \mathbf{z}^* \prec \hat{\mathbf{h}}^* = \hat{\mathbf{h}}_l \preceq \hat{\mathbf{h}}_\ell \qquad \forall \ell \in \{i, j, l\} = I_{S_k} \tag{49}$$

or, more explicitly

$$\hat{\mathbf{l}}_i \prec \hat{\mathbf{l}}_l \prec \hat{\mathbf{l}}_j = \hat{\mathbf{l}}^* \prec \mathbf{z}^* \prec \hat{\mathbf{h}}^* = \hat{\mathbf{h}}_l \prec \hat{\mathbf{h}}_j \prec \hat{\mathbf{h}}_i. \tag{50}$$

Since the above order also implies $\hat{\mathbf{l}}_\ell \prec \mathbf{z}^* \prec \hat{\mathbf{h}}_\ell, \forall \ell \in I_{S_k}$, then $\mathbf{z}^*, \mathcal{R}^*_\theta \in K[C_\ell(t), b_y] \; \forall \ell \in I_{S_k}$. Thus, from (48), $\mathbf{z}^*, \mathcal{R}^*_\theta \in K_k[S_k(t), b_y] = K^* = \mathrm{cone}(\hat{\mathbf{l}}^*, \hat{\mathbf{h}}^*)$, provided $\hat{\mathbf{h}}^*$ and $\hat{\mathbf{l}}^*$ are chosen subject to (49).

So far it was assumed that $\hat{\mathbf{l}}_\imath \prec \hat{\mathbf{h}}_\jmath$. If the unit vectors of $S_k(t)$ are such that $\hat{\mathbf{l}}_\imath \succ \hat{\mathbf{h}}_\jmath$, then there are no vectors that can satisfy the order $\hat{\mathbf{l}}_\imath = \hat{\mathbf{l}}^* \prec \mathbf{z}^* \prec \hat{\mathbf{h}}^* = \hat{\mathbf{h}}_\jmath$, and $K_k[S_k(t), b_y] = K^* = \emptyset$.

Appendix III
Derivation of Track Coverage Function

First, we seek a Lebesgue measure μ on the set of tracks that are detected by at least k sensors in $S(t)$ at time t given by

$$\mathcal{K}_k\left[S(t), b_y\right] = \bigcup_{j=1}^{q} K_k\left[S^j_k(t), b_y\right], \quad q = \binom{n}{k} \tag{51}$$

where $S^j_k(t)$ denotes the $\jmath$th k-subset of $S(t)$ at time t, and the number q of possible k-subsets is given by the binomial coefficient $\binom{n}{k}$, as shown in (51). Since $\mathcal{K}_k$ is a union of possibly disjoint cones, defined in (16), it may not be a cone [69], and is computed using the principle of inclusion–exclusion [97], such that

$$\begin{aligned}\mu\left\{\mathcal{K}_k\left[S(t), b_y\right]\right\} &= \mu\left\{\bigcup_{j=1}^{q} K_k\left[S^j_k(t), b_y\right]\right\} \\ &= \sum_{j=1}^{q}(-1)^{j+1} \sum_{1\leq i_1<\ldots<i_j\leq q} \mu\Big\{K_k\left[S^{i_1}_k(t), b_y\right] \\ &\qquad \cap \ldots \cap K_k\Big[S^{i_j}_k(t), b_y\Big]\Big\}\end{aligned} \tag{52}$$

where

$$q = \binom{n}{k} = \frac{n!}{(n-k)!\,k!} \quad \text{and} \quad \sum_{1\leq i_1<\ldots<i_j\leq q}$$

is a sum over all the $[q!/(q-j)!j!]$ distinct integer j-tuples $(i_1, \ldots, i_j)$ satisfying $1 \leq i_1 < \ldots < i_j \leq q$. Also, $\mu\{\cdot\}$ denotes a measure on the set. Since the right-hand side of (52) is an intersection of cones, it also is a cone on which we can impose the Lebesgue measure μ.

Now, consider the intersection of cones $\{K_k[S^{i_1}_k(t), b_y] \cap \ldots \cap K_k[S^{i_j}_k(t), b_y]\}$ inside the inner summation in (52). $S^{i_l}_k(t)$ denotes the i_lth k-subset of $S(t)$ at time t, where i_l is a positive integer between 1 and $i_j \leq q$, and q is the total number of k-subsets in $S(t)$. By the properties of cones, this intersection is also a cone, and represents the set of tracks through b_y that intersect all sensors in the family $\{S^{i_1}_k(t) \cup \ldots \cup S^{i_j}_k(t)\} \equiv S^{i_1,j}_p(t)$. Based on the properties of k-subsets, this set must contain $k \leq p \leq n$ elements of $S(t)$ and, thus, is a p-subset of $S(t)$. From Proposition 3.1, the set of line transversals of $S^{i_1,j}_p(t)$ through b_y can be represented by the p-coverage cone $K_p[S^{i_1}_k(t) \cup \ldots \cup S^{i_j}_k(t), b_y]$. Using the Lebesgue measure (18) on k-coverage cones, (52) can be written as

$$\begin{aligned}\mu\{\mathcal{K}_k[S(t), b_y]\} \\ = \sum_{j=1}^{q}(-1)^{j+1} \sum_{1\leq i_1<\ldots<i_j\leq q} \mu\Big\{K_p\Big[S^{i_1}_k(t) \cup \ldots \cup S^{i_j}_k(t), b_y\Big]\Big\}\end{aligned} \tag{53}$$

$$=\sum_{j=1}^{q}(-1)^{j+1}\sum_{1\le i_1<\ldots<i_j\le q}\psi\left[S_k^{i_1}(t)\cup\ldots\cup S_k^{i_j}(t),b_y\right] \tag{54}$$

$$=\sum_{j=1}^{q}(-1)^{j+1}\sum_{1\le i_1<\ldots<i_j\le q}\psi\left[S_p^{i_1,j}(t),b_y\right] \tag{55}$$

where p is the number of elements in the union of j k-subsets of $S(t)$, and $\psi[\cdot]$ is given by (18).

The set of tracks that traverse $\mathcal{A}$ and are detected by at least k sensors is given by the union of all k-coverage cones with origins $b_y, b_x, b_{y'}, b_{x'} \in \partial\mathcal{A}$, and with opening angles denoted by ψ, ζ, φ, and ρ, respectively (Fig. 4). To obtain representations that are computationally tractable, $\partial\mathcal{A}$ is discretized into increments of size δb, and $2(L_1+L_2)/\delta b$ intercept values, indexed by ℓ, are considered. Using an appropriate transformation [8], (13)–(18) can be used to compute all opening angles, $\psi[S_p^{i_1,j}(t),b_y]$, $\rho[S_p^{i_1,j}(t),b_x]$, $\xi[S_p^{i_1,j}(t),b_{y'}]$, and $\zeta[S_p^{i_1,j}(t),b_{x'}]$, as a function of their origins and of the sensors' positions $\mathbf{x}(t)$ and ranges $\mathbf{r}(t)$ in the xy-frame. By considering that every track in this union intersects two sides of $\mathcal{A}$ and belongs to two k-coverage cones, and by shifting the indices to consider intercepts at the corners only once, the final track coverage function in (20) is obtained.

ACKNOWLEDGMENT

The authors would like to thank Dr. J. T. Kohut at the Coastal Ocean Observation Laboratory (COOL), Rutgers—The State University of New Jersey, Piscataway, for providing the coastal ocean dynamics applications radar (CODAR) data used in this research, and Dr. T. A. Wettergren at the Naval Undersea Warfare Center, Washington, DC, and Prof. P. F. J. Lermusiaux at the Massachusetts Institute of Technology (MIT), Cambridge, for their helpful guidance and suggestions.

REFERENCES

[1] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein, "Energy efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet," in *Proc. 10th Int. Conf. Archit. Support Programm. Lang. Operat. Syst.*, 2002, pp. 96–107.

[2] J. R. Sibert and J. L. Nielsen, *Electronic Tagging and Tracking in Marine Fisheries*. Norwell, MA: Kluwer, 2001.

[3] V. Isler, S. Khanna, J. Spletzer, and C. Taylor, "Target tracking with distributed sensors: The focus of attention problem," *Comput. Vis. Image Understand.*, vol. 100, pp. 225–247, 2005.

[4] E. M. Sozer, M. Stojanovic, and J. G. Proakis, "Underwater acoustic networks," *IEEE J. Ocean. Eng.*, vol. 25, no. 1, pp. 72–83, Jan. 2000.

[5] T. A. Wettergren, "Performance of search via track-before-detect for distributed sensor networks," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 44, no. 1, pp. 314–325, Jan. 2008.

[6] S. Coraluppi and C. Carthel, "Recursive track fusion for multi-sensor surveillance," *Inf. Fusion*, vol. 5, no. 1, pp. 23–33, 2004.

[7] J.-P. LeCadre and G. Souris, "Searching tracks," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 36, no. 4, pp. 1149–1166, Oct. 2000.

[8] K. C. Baumgartner and S. Ferrari, "A geometric transversal approach to analyzing track coverage in sensor networks," *IEEE Trans. Comput.*, vol. 57, no. 8, pp. 1113–1128, Aug. 2008.

[9] Y. Zou and K. Chakrabarty, "A distributed coverage- and connectivity- centric technique for selecting active nodes in wireless sensor networks," *IEEE Trans. Comput.*, vol. 54, no. 8, pp. 978–991, Aug. 2005.

[10] L. M. Kaplan, "Global node selection for localization in a distributed sensor network," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 42, no. 1, pp. 113–135, Jan. 2006.

[11] B. Koopman, *Search and Screening: General Principles with Historical Applicaitons*. New York: Pergamon, 1980.

[12] H. Cox, "Cumulative detection probabilities for a randomly moving source in a sparse field of sensors," in *Proc. Asilomar Conf. Circuits Syst. Comput.*, 1989, pp. 384–389.

[13] K. Chakrabarty, S. S. Iyengar, H. Qi, and E. Cho, "Grid coverage for surveillance and target location in distributed sensor networks," *IEEE Trans. Comput.*, vol. 51, no. 12, pp. 1448–1453, Dec. 2002.

[14] X.-Y. Li, P.-J. Wan, and O. Frieder, "Coverage in wireless ad-hoc sensor networks," *IEEE Trans. Comput.*, vol. 52, no. 6, pp. 753–763, Jun. 2003.

[15] S. Megerian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, "Worst and best-case coverage in sensor networks," *IEEE Trans. Mobile Comput.*, vol. 4, no. 2, pp. 84–92, Jan.–Feb. 2005.

[16] J. Ai and A. A. Abouzeid, "Coverage by directional sensors in randomly deployed wireless sensor networks," *J. Combinat. Optim.*, vol. 11, pp. 21–41, 2006.

[17] H. Choset, "Coverage for robotics: A survey of recent results," *Ann. Math. Artif. Intell.*, vol. 31, no. 1-4, pp. 113–126, 2001.

[18] E. U. Acar, "Path planning for robotic demining: Robust sensor-based coverage of unstructured environments and probabilistic methods," *Int. J. Robot. Res.*, vol. 22, pp. 441–466, 2003.

[19] X. Liao and L. Carin, "Application of the theory of optimal experiments to adaptive electromagnetic-induction sensing of buried targets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 8, pp. 961–972, Aug. 2004.

[20] J. R. Spletzer and C. J. Taylor, "Dynamic sensor planning and control for optimally tracking target," *Int. J. Robot. Res.*, vol. 22, no. 1, pp. 7–20, 2003.

[21] J. Corté, S. Martínez, T. Karataş, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Trans. Robot. Autom.*, vol. 20, no. 2, pp. 243–255, Apr. 2004.

[22] P. Ögren, E. Fiorelli, and N. Leonard, "Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment," *IEEE Trans. Autom. Control*, vol. 49, no. 8, pp. 1292–1302, Aug. 2004.

[23] J. C. Latombe, A. Lazanas, and S. Shekhar, "Robot motion planning with uncertainty in control and sensing," *Artif. Intell.*, vol. 52, no. 1, pp. 1–47, Nov. 1991.

[24] N. E. Leonard, D. A. Paley, F. Lekien, R. Sepulchre, D. M. Fratantoni, and R. E. Davis, "Collective motion, sensor networks, and ocean sampling," *Proc. IEEE*, vol. 95, no. 1, pp. 48–74, Jan. 2007.

[25] T. Curtin, J. G. Bellingham, J. Catipovic, and D. Webb, "Autonomous oceanographic sampling networks," *Oceanography*, vol. 6, pp. 86–94, 1989.

[26] A. Alvarez, A. Caiti, and R. Onken, "Evolutionary path planning for autonomous underwater vehicles in a variable ocean," *IEEE J. Ocean. Eng.*, vol. 29, no. 2, pp. 418–429, Apr. 2004.

[27] T. Inanc, S. C. Shadden, and J. E. Marsden, "Optimal trajectory generation in ocean flows," in *Proc. Amer. Control Conf.*, Portland, OR, 2005, pp. 674–679.

[28] I. Spangelo and O. Egeland, "Trajectory planning and collision avoidance for underwater vehicles using optimal control," *IEEE J. Ocean. Eng.*, vol. 19, no. 4, pp. 502–511, Oct. 1994.

[29] P. van de Ven, C. Flanagan, and D. Toal, "Neural network control of underwater vehicles," *Proc. SPIE—Int. Soc. Opt. Eng.*, vol. 18, Detection and Remediation Technologies for Mines and Minelike Targets VIII, no. 5, pp. 533–547, 2005.

[30] R. F. Stengel, *Optimal Control and Estimation*. New York: Dover, 1986.

[31] V. Coverstone-Carroll, J. W. Hartmann, and W. J. Mason, "Optimal multi-objective low-thrust spacecraft trajectories," *Comput. Methods Appl. Mech. Eng.*, vol. 186, pp. 387–402, 2000.

[32] J. Barraquand and J. C. Latombe, "Nonholonomic multibody mobile robots: Controllability and motion planning in the presence of obstacles," *Algorithmica*, vol. 10, no. 2–4, pp. 121–155, 1993.

[33] N. Heo and P. K. Varshney, "Energy-efficient deployment of intelligent mobile sensor networks," *IEEE Trans. Syst. Man Cybern. A, Syst. Humans*, vol. 35, no. 1, pp. 78–92, Jan. 2005.

[34] M. Marengoni, B. A. Draper, A. Hanson, and R. A. Sitaraman, "System to place observers on a polyhedral terrain in polynomial time," *Image Vis. Comput.*, vol. 18, pp. 773–780, 1996.

[35] C. Cai and S. Ferrari, "Bayesian network modeling of acoustic sensor measurements," in *Proc. IEEE Sensors Conf.*, Atlanta, GA, 2007, pp. 345–349.

[36] D. MacKay, "Bayesian interpolation," *Neural Comput.*, vol. 4, no. 3, pp. 415–447, 1992.

[37] P. F. J. Lermusiaux, "Evolving the subspace of the three-dimensional multiscale ocean variability: Massachusetts Bay," *J. Mar. Syst.*, vol. 29, 2001.

[38] Coastal Ocean Observation Laboratory (COOL), Rutgers—The State University of New Jersey, Piscataway, NJ, 2002 [Online]. Available: http://marine.rutgers.edu/

[39] M. Athans and P. Falb*, Optimal Control*. New York: McGraw-Hill, 1966.

[40] J. Betts, "Survey of numerical methods for trajectory optimization," *J. Guid. Control Dyn.*, vol. 21, no. 2, pp. 193–207, 1998.

[41] P. Gill, W. Murray, and M. Wright*, Practical Optimization*. London, U.K.: Academic, 1981.

[42] The MathWorks, Inc., Matlab Optimization Toolbox, Natick, MA, 2004 [Online]. Available: http://www.mathworks.com, function: fmincon

[43] G. Huntington and A. Rao, "Optimal reconfiguration of spacecraft formations using a gauss pseudospectral method," *J. Guid. Control Dyn.*, vol. 31, no. 3, pp. 689–698, 2007.

[44] P. Williams, "A comparison of differentiation and integration based direct transcription methods," in *Proc. AAS/AIAA Spaceflight Mech. Meeting*, Copper Mountain, CO, Jan. 2005, pp. 389–408.

[45] C. R. Hargraves and S. W. Paris, "Direct trajectory optimization using nonlinear programming and collocation," *J. Guid. Control Dyn.*, vol. 10, no. 4, pp. 338–342, 1987.

[46] F. H. Todd and D. Taylor, "Resistance and propulsion," in *Principles of Naval Architecture*, J. P. Comstock, Ed. New York: The Society of Naval Architects and Marine Engineers, 1967, pp. 288–462.

[47] T. I. Fossen and S. I. Sagatun, "Adaptive control of nonlinear systems: A case study of underwater vehicles," *J. Robot. Syst.*, vol. 8, no. 3, pp. 393–412, 1991.

[48] T. I. Fossen*, Guidance and Control of Underwater Vehicles*. New York: Wiley, 1994.

[49] D. Smallwood and L. Whitcomb, "Model-based dynamic positioning of underwater robotic vehicles: Theory and experiment," *IEEE J. Ocean. Eng.*, vol. 29, no. 1, pp. 169–186, Jan. 2004.

[50] A. Liu, Y. Zhao, and M.-K. Hsu, "Ocean surface drift revealed by synthetic aperture radar images," *EOS Trans. Amer. Geophys. Union*, vol. 87, no. 24, pp. 233–239, Jun. 13, 2006.

[51] A. R. Robinson, "Physical processes, field estimation and an approach to interdisciplinary ocean modeling," *Earth-Science Rev.*, vol. 40, pp. 3–54, 1996.

[52] D. P. Bertsekas*, Dynamic Programming and Optimal Control, Vols. I and II*. Belmont, MA: Athena Scientific, 1995.

[53] F. Lekien, C. Coulliette, R. Bank, and J. Marsden, "Open-boundary modal analysis: Interpolation, extrapolation and filtering," *J. Geophys. Res.—Oceans*, vol. 109, no. C1, 2004, DOI:10.1029/2004JC002323.

[54] K. C. Baumgartner, S. Ferrari, and T. Wettergren, "Robust deployment of dynamic sensor networks for cooperative track detection," *IEEE Sensors*, vol. 9, no. 9, pp. 1029–1048, Sep. 2009.

[55] T. Lyche, K. Mørken, and E. Quak, "Theory and algorithms for nonuniform spline wavelets," in *Multivariate Approximation and Applications*, N. Dyn, D. Leviatan, D. Levin, and A. Pinkus, Eds. Cambridge, U.K.: Cambridge Univ. Press, 2001, ch. 6, pp. 152–187.

[56] D. P. Bertsekas and J. N. Tsitsiklis*, Neuro-Dynamic Programming*. Belmont, MA: Athena Scientific, 1996.

[57] J. H. Friedman, "Multivariate adaptive regression splines," *Ann. Statist.*, vol. 19, pp. 1–141, 1991.

[58] F. Girosi, M. Jones, and T. Poggio, "Regularization theory and neural networks architectures," *Neural Comput.*, vol. 7, no. 2, pp. 219–269, 1995.

[59] The MathWorks, Inc., MATLAB Neural Network Toolbox, Natick, MA, 2006 [Online]. Available: http://www.mathworks.com, function: trainbr

[60] F. Foresee and M. Hagan, "Gauss-Newton approximation to Bayesian regularization," in *Proc. Int. Joint Conf. Neural Netw.*, 1997, pp. 1930–1935.

[61] M. Chu, H. Haussecker, and F. Zhao, "Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks," *Int. J. High Performance Comput. Appl.*, vol. 16, no. 3, pp. 293–313, 2002.

[62] T. A. Wettergren, R. L. Streit, and J. R. Short, "Tracking with distributed sets of proximity sensors using geometric invariants," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 40, no. 4, pp. 1366–1374, Oct. 2004.

[63] R. Urick*, Principles of Underwater Sound*, 3rd ed. New York: McGraw-Hill, 1983.

[64] S. Jenkins, D. Humphreys, J. Sherman, J. Osse, C. Jones, N. Leonard, J. Graver, and R. Bachmayer, "Alternatives for enhancement of transport economy in underwater gliders," in *Proc. OCEANS Conf.*, 2003, vol. 2, pp. 948–950.

[65] G. Thatte and U. Mitra, "Sensor selection and power allocation for distributed estimation in sensor networks: Beyond the star topology," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 2649–2661, Jul. 2008.

[66] C. M. Traweek and T. A. Wettergren, "Efficient sensor characteristic selection for cost-effective distributed sensor networks," *IEEE J. Ocean. Eng.*, vol. 31, no. 2, pp. 480–486, Apr. 2006.

[67] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan, and K. Saluja, "Sensor deployment for detection of targets traversing a region," *Mobile Netw. Appl.*, vol. 8, pp. 453–461, Aug. 2003.

[68] J. Goodman, R. Pollack, and R. Wenger, "Geometric transversal theory," in *New Trends in Discrete and Computational Geometry*, J. Pach, Ed. New York: Springer-Verlag, 1991, pp. 163–198.

[69] D. P. Bertsekas*, Convex Analysis and Optimization*. Belmont, MA: Athena Scientific, 2003.

[70] H. F. Davis and A. D. Snider*, Vector Analysis*. Dubuque, IA: William C. Brown, 1987, ch. 1 and 2.

[71] S. Skiena, "Generating k-subsets," in *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematics*. Reading, MA: Addison-Wesley, 1990, pp. 44–46.

[72] V. Isler, S. Khanna, and K. Daniilidis, "Sampling based sensor-network deployment," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2004, vol. 100, pp. 1780–1785.

[73] L. Benyuan and D. Towsley, "A study of the coverage of large-scale sensor networks," in *IEEE Int. Conf. Mobile Ad-Hoc Sensor Syst.*, 2004, pp. 475–483.

[74] W. Huang, Y. Li, and C. Li, "Greedy algorithms for packing unequal circles into a rectangular container," *J. Operat. Res. Soc.*, vol. 56, pp. 539–548, 2005.

[75] J. A. George, J. M. George, and B. Lamer, "Packing different-sized circles into a rectangular container," *Eur. J. Operat. Res.*, vol. 84, pp. 693–712, 1995.

[76] D. S. Bernstein*, Matrix Mathematics*. Princeton, NJ: Princeton Univ. Press, 2005.

[77] R. F. Stengel, R. Ghigliazza, N. Kulkarni, and O. Laplace, "Optimal control of innate immune response," *Optimal Control Appl. Methods*, vol. 23, pp. 91–104, 2002.

[78] J. Betts and P. Frank, "A sparse nonlinear optimization algorithm," *J. Optim. Theory Appl.*, vol. 82, pp. 519–541, 1994.

[79] D. G. Hull, "Conversion of optimal control problems into parameter optimization problems," *J. Guid. Control Dyn.*, vol. 20, no. 1, pp. 57–60, 1997.

[80] D. P. Bertsekas*, Nonlinear Programming*. Belmont, MA: Athena Scientific, 2007.

[81] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty*, Nonlinear Programming: Theory and Algorithms*. Hoboken, NJ: Wiley, 2006.

[82] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang*, Spectral Methods in Fluid Dynamics*. New York: Springer-Verlag, 1988.

[83] B. Fornberg*, A Practical Guide to Pseudospectral Methods*. New York: Cambridge Univ. Press, 1998.

[84] G. W. Reddien, "Collocation at Gauss points as a discretization in optimal control," *SIAM J. Control Optim.*, vol. 17, no. 2, pp. 298–306, 1979.

[85] G. Huntington and A. Rao, "Design of optimal tetrahedral spacecraft formations," *J. Astronaut. Sci.*, vol. 55, no. 2, pp. 141–169, 2007.

[86] D. Benson, G. Huntington, T. Thorvaldsen, and A. Rao, "Direct trajectory optimization and costate estimation via an orthogonal collocation method," *J. Guid. Control Dyn.*, vol. 29, no. 6, pp. 1435–1440, 2006.

[87] G. Huntington, "Advancement and analysis of a Gauss pseudospectral transcription for optimal control," Ph.D. dissertation, Dept. Aeronaut. Astronaut., Massachusetts Inst. Technol., Cambridge, MA, May 2007.

[88] P. Davis, Ed.*, Interpolation and Approximation*. New York: Dover, 1975.

[89] D. Benson, "A Gauss pseudospectral transcription for optimal control," Ph.D. dissertation, Dept. Aeronaut. Astronaut., Massachusetts Inst. Technol., Cambridge, MA, Nov. 2004.

[90] A. V. Rao, D. A. Benson, C. L. Darby, C. Francolin, M. A. Patterson, I. Sanders, and G. T. Huntington, "User's manual for GPOPS: A Matlab software for solving multiple-phase optimal control problems using the Gauss pseudospectral method 2009 [Online]. Available: http://gpops.svn.sourceforge.net/viewvc/gpops/Manual/gpopsManual.pdf

[91] A. V. Rao, D. A. Benson, C. L. Darby, C. Francolin, M. A. Patterson, I. Sanders, and G. T. Huntington, "Algorithm xxx: GPOPS, a Matlab software for solving multiple-phase optimal control problems using the Gauss pseudospectral method," *ACM Trans. Math. Softw.*, Jun. 2009 [Online]. Available: http://vdol.mae.ufl.edu/tomsgpm.pdf, to be published

[92] The MathWorks, Inc., Matlab Optimization Toolbox, Natick, MA, 2004 [Online]. Available: http://www.mathworks.com, function: ode45

[93] The MathWorks, Inc., Matlab, Natick, MA, 2004 [Online]. Available: http://www.mathworks.com

[94] D. A. Paley, F. Zhang, and N. E. Leonard, "Cooperative control for ocean sampling: The glider coordinated control system," *IEEE Trans. Control Syst. Technol.*, vol. 16, no. 4, pp. 735–744, Jul. 2008.

[95] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evol. Comput.*, vol. 2, no. 3, pp. 221–248, 1995.

[96] K. M. Miettinen*, Nonlinear Multiobjective Optimization*. Norwell, MA: Kluwer, 1998.

[97] S. Ross*, Introduction to Stochastic Dynamic Programming*. New York: Academic, 1983.

Kelli A. C. Baumgartner (M'09) received the B.S. degree in aerospace engineering from Embry-Riddle Aeronautical University, Daytona Beach, FL, in 2003 and the M.S. and Ph.D. degrees in mechanical engineering and materials science from Duke University, Durham, NC, in 2005 and 2007, respectively.

Currently, she is a Flight Design and Controls Engineer at the System Development and Integration Division, Analex Corporation, Brook Park, OH, a subsidiary of QinetiQ North America. While at Duke University, her principal research interests included distributed sensor networks and intelligent systems using Bayesian networks.

Dr. Baumgartner is a member of the American Institute of Aeronautics and Astronautics (AIAA) and the American Society of Mechanical Engineers (ASME), and a Ronald E. McNair scholar.

Silvia Ferrari (S'01–M'02–SM'08) received the B.S. degree in aerospace engineering from Embry-Riddle Aeronautical University, Daytona Beach, FL, in 1997 and the M.A. and Ph.D. degrees in mechanical and aerospace engineering from Princeton University, Princeton, NJ, in 1999 and 2002, respectively.

Currently, she is an Assistant Professor of Mechanical Engineering and Materials Science at Duke University, Durham, NC, where she directs the Laboratory for Intelligent Systems and Controls (LISC). Her principal research interests include robust adaptive control of aircraft, learning and approximate dynamic programming, and optimal control of mobile sensor networks.

Dr. Ferrari is a member of the American Society of Mechanical Engineers (ASME), The International Society for Optical Engineers (SPIE), and the American Institute of Aeronautics and Astronautics (AIAA). She is the recipient of the U.S. Office of Naval Research (ONR) Young Investigator Award (2004), the National Science Foundation (NSF) CAREER Award (2005), and the Presidential Early Career Award for Scientists and Engineers (PECASE) award (2006).

Anil V. Rao received the B.S. degree in mechanical engineering and the A.B. degree in mathematics from Cornell University, Ithaca, NY, both in 1988, the M.S.E. degree in aerospace engineering from The University of Michigan, Ann Arbor, in 1989, and the M.A. and Ph.D. degrees in mechanical and aerospace engineering from Princeton University, Princeton, NJ, in 1992 and 1996, respectively.

After receiving his Ph.D. degree, he joined the Flight Mechanics Department, the Aerospace Corporation, Los Angeles, CA, where he was a Member of the Technical Staff and a Senior Member of the Technical Staff. During his employment at the Aerospace Corporation, he led projects in the areas of orbital and reentry debris estimation, aeroassisted orbital transfer, and launch vehicle ascent. Subsequently, he joined the Guidance and Navigation Division, the Charles Stark Draper Laboratory, Cambridge, MA, as a Senior Member of the Technical Staff. While at Draper, he led projects in numerical methods for solving optimal control problems, guidance and navigation of small munitions, and advanced numerical methods for accuracy analysis of inertial instruments for the U.S. Navy. Simultaneous with his employment at Draper, he was a Lecturer/Senior Lecturer at Boston University, Boston, MA, where for five consecutive years he taught the core engineering dynamics course at the Department of Aerospace and Mechanical Engineering. Since July 2006, he has been an Assistant Professor of Mechanical and Aerospace Engineering, the University of Florida, Gainesville, where he runs the Flight Dynamics and Optimization Laboratory. His current research interests include trajectory optimization, nonlinear control, guidance, and navigation. He has published many conference and journal articles during his research career, has supervised several masters and doctoral students, won numerous teaching awards at Boston University and the University of Florida, and has published a book on engineering dynamics.