International Journal of Control

ISSN: 0020-7179 (Print) 1366-5820 (Online) Journal homepage: http://www.tandfonline.com/loi/tcon20

Path Planning in GPS-denied Environments via Collective Intelligence of Distributed Sensor Networks

Devesh K. Jha, Pritthi Chattopadhyay, Soumik Sarkar & Asok Ray

To cite this article: Devesh K. Jha, Pritthi Chattopadhyay, Soumik Sarkar & Asok Ray (2015): Path Planning in GPS-denied Environments via Collective Intelligence of Distributed Sensor Networks, International Journal of Control, DOI: 10.1080/00207179.2015.1110754

To link to this article: http://dx.doi.org/10.1080/00207179.2015.1110754

Accepted author version posted online: 29 Oct 2015.

Publisher: Taylor & Francis

Journal: *International Journal of Control*

DOI: http://dx.doi.org/10.1080/00207179.2015.1110754

Paper ID: TCON-2015-0387R1 accepted for publication in ***International Journal of Control***

Path Planning in GPS-denied Environments via Collective Intelligence of Distributed Sensor Networks

Devesh K. Jha† dkj5042@psu.edu
Pritthi Chattopadhyay† pxc271@psu.edu
Soumik Sarkar‡ soumiks@iastate.edu
Asok Ray† axr2@psu.edu

† Mechanical & Nuclear Engineering Department, Pennsylvania State University, University Park, PA 16802, USA
‡ Mechanical Engineering Department, Iowa State University, Ames, IA 50011, USA

Keywords: Target Detection; Path Planning; Language Measure; Mobile Sensor Network; Collective Intelligence

***Abstract*—This paper proposes a framework for reactive goal-directed navigation without global positioning facilities in unknown dynamic environments. A mobile sensor network is used for localizing regions of interest for path planning of an autonomous mobile robot. The underlying theory is an extension of a generalized gossip algorithm that has been recently developed in a language-measure-theoretic setting. The algorithm has been used to propagate local decisions of target detection over a mobile sensor network and thus, it generates a *belief map* for the detected target over the network. In this setting, an autonomous mobile robot may communicate only with a few mobile sensing nodes in its own neighborhood and localize itself relative to the communicating nodes with bounded uncertainties. The robot makes use of the knowledge based on the *belief* of the mobile sensors to generate a sequence of way-points, leading to a possible goal. The estimated way-points are used by a sampling-based motion planning algorithm to generate feasible trajectories for the robot. The proposed concept has been validated by numerical simulation on a mobile sensor network test-bed and a Dubin's car-like robot.**

I. Motivation and Introduction

Autonomous robots are becoming ubiquitous and are envisaged to play an increasingly important role in both civilian and military applications such as intelligence, surveillance & reconnaissance (ISR), weather monitoring, fighting wildfire, health care, and logistics, to name a few. As such, the ability of robots to make complex decisions is becoming an increasingly commonplace requirement for such missions. Autonomous robots operating in unknown and unstructured environments often have limited or unreliable long-range communication and GPS capabilities. Examples include a team of mobile autonomous robots for long endurance military applications (e.g., mine-hunting) and non-military applications (e.g., weather monitoring). These missions are often limited due to unavailability of global information and imposition of communication constraints resulting from energy requirements and environmental uncertainties. However, with the recent advances in low-complexity signal processing algorithms, sensing systems can locally detect regions of interest with high accuracy, which allow information extraction at the sensor site for significant reduction of the communication overhead. Even though reduction of communication overhead improves the life of such autonomous sensing networks, it limits the network performance and capacity. For example, a search and rescue operation in an urban scenario in an apartment complex may require sequential collaboration between the '*search agents*' and the '*rescue agents*', where the rescue operation is usually triggered based on the real-time information collected in a network environment by the independent '*search agents*'. The efficacy of such missions depends on how quickly the network can react to the sensed targets and guide a '*rescue agent*' to the target of interest under the constraints of limited communication and global positioning. [Note: The term *agent* has been used for a mobile sensor in this paper and it should not be confused with an autonomous robot that is navigated to the target location.]

Recently much work has been reported on source seeking in sensor fields [1]–[9], where the objective is to identify the possible location of the source as the minimal point of an unknown signal field by using a stochastic gradient descent algorithm. Authors in [7] [4] [3] [9] present

★This work has been supported in part by the U.S. Office of Naval Research under Grant No. N00014-14-1-0545, and by the U.S. Army Research Laboratory under Cooperative Agreement No. W911NF-14-2-0068. Any opinions, findings and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the sponsoring agencies.

a multi-agent coordination framework where the mobile agents together estimate the peaks of sensor field and all the agents collectively move to the peak of the field following the estimated gradient of the field. However, the agents have to communicate their sensor measurements and an artificial potential function is required to estimate the gradient of the sensor field. Also, in [3], the agents need to maintain a particular formation for accurate estimation of the gradient which constrains the motion of the individual agents who might have complicated dynamics. Authors in [1] [2] [5] [6] present algorithms for estimating the source location by using a stochastic gradient descent algorithm that takes into account the robot dynamics. In contrast, the focus of the present work is to use a mobile sensor network for guiding an autonomous robot to a source of interest in environments, where the knowledge of global positioning is not always feasible.

Several attempts have been made in literature to make use of static sensor networks to guide a robot. Li et al. [10] present the use of sensor networks as adaptive repositories of information for guiding an autonomous robot by creating a map to dangerous areas (e.g., obstacles, or populated areas). Deshpande et al. [11] use a pseudo-gradient, calculated based on sensor readings, for localization and directed navigation of an autonomous robot in unknown environments. However, a sensor network with static *search agents* has been used and the underlying algorithm cannot be easily extended for navigation with a mobile sensor network that may have a potential advantage over static counterparts in terms of spatial coverage and time-criticality [12]. It is noted that gradient-based algorithms may not be operable in unstructured environments if the sensing model is not sufficiently accurate. Furthermore, global sensing of targets is not usually applicable for surveillance in large (e.g., littoral underwater) sensor networks, because exchange of global information could be severely restricted due to communication constraints for limited energy availability. Another set of relevant literature is from motion planning in the presence of various uncertainties. Motion planning has a very rich literature and the current state-of-the-art algorithms are sampling-based and can guarantee optimality [13]. Some work in sampling-based motion planning literature for planning with uncertainties could be found in [14]–[17]. However, the main idea in these papers is to compensate for state estimation error, when the goal position is known along the trajectories that are generated by using sampling-based algorithms.

The current paper proposes a framework, where a mobile sensor network is used to generate way-points for an autonomous robot based on the sensed regions of interest, which are exploited by sampling-based motion planning (e.g., rapidly-exploring random tree (RRT) [18]) to generate feasible trajectories for the robot; a preliminary version of this work has been presented earlier as a conference paper [19]. A different perspective is presented for solving source seeking problems in sensor fields where, based on their limited detection capabilities (i.e., only a small fraction of the sensor population actually detects a target due to their physical proximity to the target) and collaborative information exchange, low-level sensors can create a belief about a point of interest (or target). The belief map could then be used for navigation by autonomous robots. The pertinent problem here is navigation of an autonomous robot, assisted by a mobile sensor network, in an unknown and dynamic environment without a global positioning system (GPS); the sensor network serves the two-fold purpose of target localization and way-point generation. In this setting, a low-level motion planning algorithm makes use of the information provided by the sensor network to generate feasible paths for the robot motion [18], which is conceptually similar to emergent sensing of complex environments by animal groups [20].

The work, reported in the current paper, is built upon the concept of distributed decision propagation in mobile ad-hoc sensor networks [21], where a proximity network of sensing agents is modeled as a probabilistic finite state automaton (PFSA) [22]; a major objective here is to assist a search-and-rescue-type mission. An '*agent measure function*' is defined, based on the recently reported language measure theory [22] [23], for all agents in the network, which signify their '*level of awareness*' regarding a locally sensed '*target*' in the operational area.

While the bulk of the related work considers structured data that are generated from a well-defined sensing model, the underlying algorithm reported in the current paper relies on unstructured data and makes use of the concept of model-free source-seeking. Specifically, the '*agent measure function*' is generated by a mobile sensor network, which can be used to guide an autonomous robot through an unknown and unstructured environment [24]. The proposed framework of sensor-network-assisted robot navigation has the following potential advantages over those reported in the current literature:

- There is no need to communicate the actual signals sensed by the sensing agents. The algorithm only requires exchange of local beliefs about the sensed targets. This method has the potential of significantly reducing the communication overhead and makes the network more robust to communication flips.
- As opposed to typical source seeking scenarios, the current setup does not require the sensing agents to move towards the source. Apart from computational advantages, this is potentially more suited in an adverse environment. In such a case, the network may need to handle multiple non-collocated sources including possibly dummy sources. Another advantage is that the network does not need to be strongly connected; this is particularly important in large sensor networks used for distributed surveillance.
- No artificial potential function is required to guide an agent to the locally detected goal; a gradient is automatically generated by the *agent measure function* which is maximized at the location of sensed target.
- The proposed approach is closely related to the general class of model-free source seeking [25] i.e., where no sensing model is used. Under this umbrella, the

proposed approach doesn't require any formation of the sensing agents to estimate a gradient to the locally detected target. This has the potential benefit of decoupling the dynamics of sensing agents as they can move independently without any constraint on synchronizing their movement with others.

II. Decision Propagation in Sensor Networks

This section introduces the mathematical preliminaries as needed for analysis of distributed decision propagation in sensor networks. It briefly describes the salient concepts of the real measure [22] [23] of probabilistic regular languages generated by a PFSA [22], followed by those of a generalized gossip algorithm [21].

A. Language-Measure Theory

This subsection succinctly presents the theory of language measure, restricted to irreducible stationary Markov chains (i.e., where each state can be reached from another state in finitely many transitions). Further details are reported in [22] [23].

Definition 2.1: (Real Measure of a Markov Chain) An irreducible stationary Markov chain is denoted by the three-tuple $(Q, \mathbf{\Pi}, \boldsymbol{\chi})$, where

- Q is the set of states with cardinality $|Q|$,
- $|Q| \times |Q|$ is a stochastic matrix $\mathbf{\Pi}$ (i.e, each entry is non-negative and each row sum is equal to 1) represents the state transition function $\boldsymbol{\pi} : Q \times Q \to [0, 1]$ for the Markov chain,
- $\boldsymbol{\chi} : Q \to \mathbb{R}$ is the vector-valued characteristic function that assigns a signed real weight to each state $q_i \in Q$.

Then, a real measure ν_i for the state q_i is defined as

$$\nu_i(\theta) \triangleq \sum_{k=0}^{\infty} \theta(1-\theta)^k \mathbf{\Delta}_i \mathbf{\Pi}^k \boldsymbol{\chi}, \quad i = 1, 2, \cdots, |Q| \qquad (1)$$

where $\theta \in (0, 1)$ is a user-specified scalar parameter and $\mathbf{\Delta}_i$ is the $(1 \times |Q|)$ vector $[\delta_{i1}, \delta_{i2} \ldots \delta_{i|Q|}]$ for which the elements δ_{ij} are defined as

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

The real measure in Eq. (1) for the Markov chain is vectorially represented as

$$\boldsymbol{\nu}(\theta) = \theta(\mathbb{I} - (1-\theta)\mathbf{\Pi})^{-1}\boldsymbol{\chi} \qquad (2)$$

where the inverse is guaranteed to exist for $\theta \in (0, 1)$.

Remark 2.1: (Significance of States of a Markov Chain) The set of states Q is an abstract representation of the set of mobile sensors (or agents) in the context of language-measure-theoretic information management in the network. In other words, a mobile sensor, designated as the agent i, is represented as the state $q_i \in Q$ in the setting of a Markov chain.

Remark 2.2: (Significance of Real Measure) If the current state of the Markov chain is $q_i \in Q$, then the expected value of the characteristic function after n time steps in the future is given by $\mathbf{\Delta}_i \mathbf{\Pi}^n \boldsymbol{\chi}$. It follows from Eq. (1) that the measure of the state q_i represents the weighted expected value of $\boldsymbol{\chi}$ over all time steps in future for a Markov chain that begins in state q_i. The weight at the k^{th} time-step is $\theta(1-\theta)^k$ (see Eq. (1)); and these weights form a decreasing geometric series whose sum equals to 1. Consequently, the measure $\nu_i(\theta)$ is a convex combination of all the elements of $\boldsymbol{\chi}$.

B. Generalized Gossip Policy

This subsection briefly describes the formulation of the generalized gossip policy in the context of proximity networks as proposed in [21]. The proximity network [26] is a particular formulation of time-dependent mobile-agent networks, inspired from social networks, where only proximal agents communicate at any given time epoch [27].

In the present context, proximal agents exchange information related to their beliefs regarding the environment. After the expiry of a message lifetime L_m, agents possibly update their beliefs based on their own observation and messages from other agents. There are two time-scales involved in this problem setup. In contrast to the faster time-scale (t) of agent motion, the algorithm for updating the agents' beliefs runs on a (possibly) slower time-scale (denoted by τ). The time-scale for updating the belief is chosen to be slower as it allows for sufficient interactions among the agents, especially if the density of agents is low. If the message lifetime L_m is very small, then the network may not be able to build up over time and possibly remains sparse. On the other hand, the network would eventually become fully connected as $L_m \to \infty$. Thus, to capture temporal effects in a realistic setting, L_m should be appropriately chosen based on other network parameters. In this setting, a time-dependent (in the slow-scale τ) graph is denoted as G and a few related terms are defined as follows.

Definition 2.2: (Adjacency Matrix [28]) The adjacency matrix A of a graph G is defined such that its element a_{ij} in the ij^{th} position is unity if the agent i communicates with the agent j within the time period of the message life time L_m; otherwise the matrix element a_{ij} is zero. To eliminate self-loops, each diagonal element of the adjacency matrix is constrained to be zero.

Definition 2.3: (Laplacian Matrix [28]) The Laplacian matrix ($\mathcal{L}$) of a graph G is defined as:

$$\mathcal{L} = D - A$$

where the degree matrix D is a diagonal matrix whose i^{th} diagonal element is d_i. [Note: d_i is called the degree of the node i that may be considered as the state q_i in the Markov setting (see Definition 2.1).]

Definition 2.4: (Interaction Matrix [28]) The agent interaction matrix (which is a restricted version of the state transition matrix in Definition 2.1) is defined as:

$$\mathbf{\Pi} = I - \beta\mathcal{L}$$

where the scalar parameter β is chosen such that $\mathbf{\Pi}$ becomes a stochastic matrix and its second largest eigenvalue

satisfies the condition $|\lambda_2(\mathbf{\Pi})| < 1$. In other words, $\mathbf{\Pi}$ is a primitive (i.e., acyclic and irreducible) stochastic matrix.

In the context of proximity networks, the requirement of keeping $\mathbf{\Pi}$ as a stochastic matrix in Definition 2.4 is achieved by setting $\beta = 1/(\bar{d}+1)$, where $\bar{d}$ is a (positive integer) parameter that is pre-determined off-line. The physical significance of the parameter $\bar{d}$ is explained below.

In order to maintain the stochastic matrix properties of $\mathbf{\Pi}$ for on-line operation, an agent ignores communications with other agents that are beyond the $\bar{d}$ time steps within the message lifetime L_m. However, the expected degree distribution of the network can be obtained off-line at the design stage (see [21] for details); therefore, $\bar{d}$ is chosen to be large enough such that the probability that the degree $d_i > \bar{d}$ for any node i (i.e., state q_i in the Markov setting in Definition 2.1) is very low, i.e., $Pr(d_i > \bar{d}) \leq \epsilon \; \forall i$ (for simulation exercises reported in this paper, ϵ has been taken to be 0.001). Note that, in this case, the stochastic matrix $\mathbf{\Pi}$ is further restricted to be symmetric (i.e., also doubly stochastic) due to the above construction procedure.

The generalized gossip strategy involves two system variables associated with each agent (i.e., each state in the Markov setting), namely the *state characteristic vector* $\boldsymbol{\chi}$ and the *agent measure vector* $\boldsymbol{\nu}$, where each element of the these two vectors is restricted as: $\nu_i \in [0,1]$ and $\chi_i \in \{0,1\}$. The restriction $\nu_i \in [0,1]$ signifies the level of awareness or belief of the agent i regarding the presence of a *target* in the surveillance region. The restriction $\chi_i \in \{0,1\}$ signifies whether the agent i has detected a *target* or not (i.e., 1 for detection, 0 for no detection).

Based on current values of $\boldsymbol{\chi}$ and $\boldsymbol{\nu}$ of the agent population, the measures are updated for all agents synchronously after the expiry of one message lifetime L_m. It is noted that, based on the discussion up to this point, $\mathbf{\Pi}$, $\boldsymbol{\chi}$ and $\boldsymbol{\nu}$ are functions of the slow time-scale τ. In the above setting, a decentralized strategy of updating the measure $\nu_i(\theta)|_\tau$ in the mobile-agent population at a (slow time scale) epoch τ is introduced below in terms of a user-defined control parameter $\theta \in (0,1)$.

$$\nu_i(\theta)|_{\tau+1} = (1-\theta) \sum_{j \in \{i\} \cup \texttt{Nbhd(i)}} \Pi_{ij}|_\tau \, \nu_j(\theta)|_\tau + \theta \chi_i|_\tau \quad (3)$$

where `Nbhd(i)` denotes the set of agents in the neighborhood of agent i (or state q_i). that is, the agents (states) that communicate with the agent i (state q_i) during the time span between τ and $\tau+1$. It is noted that while computing the future (awareness or belief) measure of an agent, the parameter θ controls the trade-off between the effects of current self-observation and current measures of all agents. In the vector notation, the dynamics can be expressed as:

$$\boldsymbol{\nu}(\theta)|_{\tau+1} = (1-\theta)\mathbf{\Pi}|_\tau \boldsymbol{\nu}(\theta)|_\tau + \theta \boldsymbol{\chi}|_\tau \quad (4)$$

Thus, this policy is simply a gossip algorithm with varying input $\boldsymbol{\chi}$ and varying network topology represented by $\mathbf{\Pi}|_\tau$. The memory of a past input fades as a function of the parameter θ. Due to this notion, the above policy is called a *generalized gossip algorithm* with θ as the generalizing parameter.

Remark 2.3: The *agent measure function* (ν_i) of the agent i (state q_i) serves as the degree of awareness or *belief* of an agent regarding a locally sensed target. In the following sections, the agent measure function is also referred to as belief.

III. Problem Formulation

This section formulates the problem of path planning for an autonomous robot in the absence of global positioning facilities to find routes to a locally detected target by a distributed sensor network.

Assumptions: For simplicity of exposition, certain simplifying assumptions are made to unambiguously present the efficacy of the proposed framework for reactive navigation in the absence of global positioning systems. Major assumptions in the problem formulation are outlined below

1) An autonomous robot can locally estimate relative positions of mobile sensors using state-of-art positioning techniques in sensor networks [29].
2) Mobile sensors and the autonomous robot are locally able to coordinate for collision avoidance.
3) Communication of the robot with other mobile sensors is considered in the time scale $\mathrm{T} >> \tau$.

Assumption 1 enables the autonomous robot to locate itself (with bounded uncertainty) with respect to its proximal sensors (which is a small fraction of the population) that are communicating their beliefs. Assumption 2 is used so that the idea of the paper could be unambiguously presented without focusing on the actual collision avoidance strategies of moving sensors. Assumption 3 is used to ensure that the robot always communicates with its local network at instants when every sensor in its local network has a steady belief about the target or has steady agent measure. Also, the robot doesn't need to communicate while moving between way-points.

Mobile Sensor Network: Under these assumptions, let a set of mobile sensors $Q = \{q_1, q_2, \ldots, q_n\}$ perform surveillance in a region $\mathbb{X}$, where the task is to detect targets in the given region. For simplicity, the target (i.e., the goal for the autonomous robot) is modeled as a local region of interest in the surveillance area such that only a few sensors that come within that region of interest have a non-zero probability of detecting it. For clarity, a simplistic model for target detection is followed which is described next. A region of interest, say $X^G \subset \mathbb{X}$, is modeled as a map for probability of detection of a target. Let the probability of detection of a target be denoted by P_D, which attains the maximum at the center of the target's physical location and decays to zero linearly with distance from the center in a radially symmetric manner. A region of interest is then, characterized by the following parameters:

- The maximum probability of detection of the target, P_{Dmax}
- The effective radius (r_{hs}) of the circular region within which $P_D > 0.5$, i.e., agents further than a distance

of r_{hs} from the center of the hot-spot have less than 0.5 probability of detecting the threat.

In general, sensor networks are designed for a constant false alarm rate (CFAR), where each sensor is capable of detecting a target within its own radius r_{hs} with probability P_D. (Details are available in [30].) For CFAR detection models, a $k-$detection strategy [31] is followed to achieve the desired level of search performance in the surveillance region [30]. In this paper, the sensor network is assumed to be capable of handling false alarms so that a desired search performance is maintained in the surveillance region.

Autonomous Robot: Consider a single robot in an arbitrary surveillance environment where the dynamics of the robot is governed by the following equations.

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, t) + \eta(\mathbf{x}, \mathbf{u}, t) \quad (5a)$$

$$\mathbf{y} = h(\mathbf{x}) \quad (5b)$$

where $\mathbf{x}$ is the state of the robot, $\mathbf{u}$ is the local control input, $\mathbf{y}$ is the output. The functions f and h describe the known mathematical abstraction of the system dynamics. The function η indicates the physical uncertainties in the dynamics, including modeling errors, noises and potential physical failures. Let $\mathbb{X}$ and $\mathbb{U}$ denote the constraint sets for the state and the input, respectively, i.e., $\mathbf{x}(t) \in \mathbb{X}$ and $\mathbf{u}(t) \in \mathbb{U}$ must hold, where $\mathbb{X}$ is the surveillance region, which means the robot is always constrained to stay in the surveillance region. Further details of the input constraint set $\mathbb{U}$ are provided in Section V. As described earlier, the distributed decision propagation algorithm proposed in [21] is adopted here for dissemination of the knowledge of sensed target throughout the mobile sensor network.

The primary contribution of the current work is development of a distributed navigation algorithm that uses the disseminated information to help guide an autonomous robot to the detected region of interest (i.e., goal for the autonomous robot). Hence, the autonomous robot only relies on the collective intelligence of the mobile sensor network and does not use any global positioning system. Note that none of the sensing agents is actually aware of any sensed location of the target and hence cannot directly provide such information to the robot. Furthermore, the robot has only a finite sensing and communication radius. Under this constraint, the robot can only be aware of the beliefs of its neighboring mobile sensors. The problem of reactive navigation to the global target is then reduced to the recursive estimation of a sequence of way-points which the robot can follow to finally reach the goal. More formally, reachability task is considered, where the robot has to reach a goal set $X^G \subset \mathbb{X}$ by estimating the location of X^G using the belief of a mobile sensor network deployed in $\mathbb{X}$.

The work, reported in this paper assumes no knowledge of global positioning coordinates of the sensors and the targets; in other words, the sensors (that detect the targets) have no knowledge of their own global positions as well as those of the targets. Furthermore, not all the sensors receive signals from the targets; only a few sensors in the local neighborhood of the target can sense and hence detect the target. Thus, target detection is a local event in contrast to other reported work where target detection is a global event (i.e., all sensors can sense the targets and a path is found based on the gradient of the sensed signal). This local target detection problem requires creation of an artificial gradient towards the sensed target which can then be used to calculate the sequence of waypoints to reach the region of the detected target. This is more suitable and appropriate for networks used in unstructured environments, where the assumption of a reliable sensing model is generally not applicable.

IV. Proposed Approach

In the current settings as explained in Section III, not all the sensors detect the target. Hence, an artificial *awareness* about the presence of a target is created by using gossip among the sensors. To ensure goal-directed behavior of the artificial *awareness* towards the sensed target, the gossip algorithm presented in this section ensures a gradient in the belief function of sensors towards the sensed target. The problem of target detection is GPS-denied in the sense that the sensing system is unaware of the global positions or the sensor(s) and the target. The idea is to use the sensors as dynamic landmarks and identify a sequence of such landmarks so that a path is found to the sensor that detects the target. This is an event-triggered phenomenon and depends on the ability of the network to detect a target. Subsequent sections show that the proposed framework allows such intelligent behavior under appropriate assumptions.

This section first presents an algorithm for decentralized belief map generation in a mobile sensor network to propagate an *awareness* about the locally sensed target in the distributed sensor network. The algorithm is presented under the generalized gossip framework (briefly described in Section II-B) which guarantees a unique maxima and a gradient towards the same in the network. The key idea here is that if the autonomous robot moves in a way so that its belief (based on the belief of its nearby mobile sensors) monotonically improves (or increases) with movement, then under the condition that the belief of the network is maximized at the physical location of the goal, the robot will eventually reach the goal. Under the constraints of limited communication and sensing horizon, the robot has access to beliefs of only its proximal mobile sensors. However, due to the presence of a gradient towards the goal, the robot is able to estimate a way-point where the belief is greater than its current belief. To this end, the robot can learn an implicit correspondence between a geographical location and the belief in the network by using an interpolation scheme (e.g., neural nets etc.). Note that the network at any time instant is sparse when compared to the actual physical space. The interpolation algorithm is trained based on the set of beliefs corresponding to local mobile sensors (i.e., within the communication radius) of the robot. The maximum of the implicit surface is the way-point the robot moves to, over a certain time horizon till the next communication with the network is established. This

is achieved by following a feasible trajectory obtained by sampling from the local configuration space using rapidly exploring random trees (RRTs). These steps are recursively followed till the robot reaches the sensed region of interest i.e., the goal. The idea is similar to the commonly studied receding horizon motion planning framework, where a reactive plan is followed by the robot over a finite time horizon as a reaction to real-time information. The goal is to design a hierarchical distributed data-driven motion planing framework for navigation to unknown areas of interest, in a receding horizon fashion and the challenge is to relax the computational and communication requirements by intelligently aggregating information of a mobile sensor network while ensuring correct behavior. Figure 1 shows the flowchart of the framework. It is noted that the sensors use the generalized gossip framework earlier proposed in [21] for information propagation which is explained in the next section.

A. Decentralized Belief Map Generation

Based on the generalized gossip framework, this subsection presents an algorithm which creates a belief map in the mobile sensor network with a bias towards the sensed region of interest, i.e., the goal for the autonomous robot. The idea is based on optimal control control theory of a Probabilistic Finite State Automata [23] [32]. Under this umbrella, the belief of every sensor is maximized by averaging only over the set of its neighbors that have belief greater than the sensor. In the original gossip strategy (see equation 3), a sensing agent is influenced by all other agents in its neighborhood (or adjacency) set. However, to maximize its measure, an agent can follow a strategy where it is only influenced by agents that have a higher belief than its own belief (i.e, a higher measure than its own measure). Therefore, every agent ignores the influence of the neighboring agents that have a lower belief about the target, thereby maximizing its own belief about the target by averaging over the better informed neighbors. This strategy is succinctly presented in Algorithm 1. The key point is that the elements of the interaction matrix corresponding to agents with a lower measure are made zero (i.e., they do not have any influence on an agent with a higher belief). However, to keep the interaction matrix stochastic, those elements are adjusted as a self-loop to the agent (see steps 5 through 11 in Algorithm 1). Based on the results in [23] [32], this strategy ensures a maximum in the belief network at the goal region for the autonomous agent and at the same time, it creates a gradient towards the same. This biased approach ensures that a mobile sensor which is closer to the sensed region of interest will have a higher belief as compared to those further away from it.

To analyze the belief map generation with movement of the sensors, a frozen network is assumed at every instant τ of the slow time scale; then, the measure for every sensor via Algorithm 1 is updated. All the sensors in the network move till the next time instant (measured in the slow time scale τ), and the process of measure update is repeated again. Figure 2 shows a frozen sensor network, where all the sensors have a steady belief about the presence of the target after it has been detected. It shows a voronoi partition of the surveillance region $\mathbb{X}$ based on the sensor locations. It shows a gradient towards the locally detected target as obtained over the network by the biased gossip. This shows the correctness of the gossip in the network to create a goal directedness. Detailed proofs of optimality and convergence of distributed algorithm for language measure computation could be found in [32] (see Proposition 2 through 5). However, for the completeness of the paper and clarity of presentation, The details are as follows.

Let $Q = \{q_1, q_2, \ldots, q_n\}$ be the set of mobile sensors. Consider a frozen network of sensors at any instant τ. Let us consider the sensor which has detected the target to be denoted as q_{TG}. It is assumed that, at any instant τ, there is only one sensor which has detected the target.

Proposition 4.1: There exists a sequence of hops from any sensor $q_i \in Q$ to q_{TG} if $\nu_{q_i} > 0$.

Proof: Consider the fact that $\chi^{q_{\text{TG}}} = 1$ and $\chi^{q_j} = 0$, $\forall\ q_j \in Q \setminus q_{\text{TG}}$. Then, it is obvious that there exists a directed path from sensor q_i to the sensor which has detected the target i.e., q_{TG} (constituted by intermediate sensors) else ν_{q_i} would be identically equal to 0. Existence of paths from every sensor of the network to the one that detects the target implies a connected network. That is, if $\nu_i > 0\ \forall\ q_i \in Q$, then the network is implied to be connected. ■

Proposition 4.2: There exists a sequence of hops $q_i \rightarrow q_j \rightarrow \cdots \rightarrow q_{\text{TG}}$, where every sensor in the sequence is a neighbor of the preceding member in the sequence such that $\nu_{q_i} \leq \nu_{q_j} \leq \cdots \leq \nu_{q_{\text{TG}}}$ iff $\nu_{q_i} > 0$.

Proof: Algorithm 1 implies that if $\nu_{q_i} > 0$, there exists at least one $q_j \in \texttt{Nbhd}(\texttt{q}_\texttt{i})$ such that $\Pi_{ij} \neq 0$ and $\nu_{q_j} \geq \nu_{q_i}$. This follows from the fact that $\Pi_{ij} > 0$ iff $\nu_{q_j} \geq \nu_{q_i}$ (see Algorithm 1). The same argument is valid for q_j. And hence, if $\nu_{q_i} > 0\ \exists$ a sequence of hops $q_i \rightarrow q_j \rightarrow \cdots \rightarrow q_{\text{TG}}$ such that $\nu_{q_i} \leq \nu_{q_j} \leq \cdots \leq \nu_{q_{\text{TG}}}$. The converse is straightforward as existence of a directed path from q_{TG} suggests $\nu_{q_i} > 0$. ■

The above two propositions show that, for a connected sensor network where only one of the sensors has detected the region of interest or the target, there exists a sequence of sensors from every sensor to the one that has detected the target. The sequence could be found by finding the monotonic sequence of measure function for the sensors which is maximized at the sensor which detects the target. Thus, Algorithm 1 ensures a direction towards the region of the interest through a decentralized gossip. Hence, as long as there is a sensor in the network detecting the target of interest, algorithm 1 will ensure a gradient in the measure function of the sensors towards the detected target (or the sensor detecting the target). In the case of more than one sensor detecting a target, there would not be a unique maxima in the belief map for the network; however, Propositions 4.1 and 4.2 still ensure paths to one of the sensors detecting the target from every sensor in the network. Figure 2 shows a frozen network where the target is located at $[400, 200]$ (shown as a red circle) along with

the voronoi partition of the surveillance region based on the sensor locations. Also shown are a few sequences of cells that could be followed based on a simple rule of moving to the best neighboring cell to reach the target-detecting sensor from different corners of the region. The best neighbor is the one with the maximum value of ν. The asymptotic runtime complexity of Algorithm 1 is bounded by $O(Nk^2)$ where N is the total number of sensors and k is the number of nearest neighbors for a sensor (see Proposition 6 in [32]).

B. Implicit Surface based Interpolation for Navigation

The robot uses the directed belief generated by the mobile sensor network to recursively estimate way-points; the way-points finally converge to the goal region (i.e., the region of interest sensed by the mobile sensor network). Under the assumption that the robot can localize mobile sensors in its neighborhood [29], beliefs of the mobile sensors in the robot's neighborhood are used to learn an implicit correspondence between a physical location relative to the robot and belief. There could be several ways to do so. The most simplest way was shown in the last subsection IV-A where the voronoi cells were assigned a constant belief equal to the belief of the sensor in that cell (at any time T). However, a smoother implicit surface can also be created by first summing a collection of Radial Basis Functions (RBFs) [33]. The weights for the RBFs are then learned by solving a set of linear equations using the set of observations as the boundary constraints. The functional value at any physical location can then be obtained using the functional form learned in the last step. The local estimates are made at time instants $\mathrm{T}_i, i \in \mathbb{N}$ in a much slower time scale (see Assumption 3 in Section III). At this point, we would like to clarify that the robot estimates the position of the local sensors relative to its own position, as opposed to estimating the absolute position. The set of way-points, estimated by using the proposed regression, considers the relative coordinate system, and the same coordinate system is used by the motion planning module to find feasible trajectories for motion of the robot.

Let $X^R \in \mathbb{X}$ be the location of the robot at some time instant $\mathrm{T}_i, i \in \mathbb{N}$ and $X^G \in \mathbb{X}$ be the location of the target detected by the mobile sensor network. Let $\mathtt{Nbhd(R)}$ be the local neighborhood of the robot in which it can locally estimate the relative positions of mobile sensors within its communication range. Denoting $\tilde{x}$ as the relative coordinate of a physical location measured with respect to the robot in the region $\mathtt{Nbhd(R)}$, let $\tilde{x}_i$, $i = 1, 2, \ldots, M$ be the relative positions of the mobile sensors with respect to the robot. Figure 3 shows the example of a scenario for a local neighborhood of the robot with relative sensor locations at a time instant along with the corresponding estimated way-points.

It is assumed that the robot can estimate $\tilde{x}_i$'s with bounded uncertainties in sensor network by using localization techniques. Then, the interpolation problem is formally stated as follows:

Given the approximate locations of the neighbors of the robot, $\{\tilde{x}_i \in \mathbb{R}^2,\ i = 1, \cdots, M\}$ and their corresponding beliefs $\{\nu_i \in \mathbb{R}\}$, a function $\mathfrak{F} : \mathbb{R}^2 \rightarrow \mathbb{R}$ is estimated, such that it satisfies the boundary constraints $\mathfrak{F}(\tilde{x}_i) = \nu_i$. Then, $\mathfrak{F}(\tilde{x})$ has the following form

$$\mathfrak{F}(\tilde{x}) = \sum_{i=1}^{M} w_i \phi(||\tilde{x} - \tilde{x}_i||) \tag{6}$$

where $\phi(\bullet)$ is a radial basis function (RBF) and w_i's are the weights assigned to the individual RBF's that are centered at the respective $\tilde{x}_i$'s. By making use of Stone-Weierstrass theorem that states "any continuous function with a compact support can be approximated with arbitrary accuracy by a polynomial," Eq. (6) becomes valid if a sufficiently large number of RBFs (i.e., sufficiently large positive integer M) is selected [33]. The function $\mathfrak{F}$ represents an implicit correspondence between the local physical locations and belief about the region of interest. An analogy can be drawn with value functions from the optimal control literature which is often used for state-based feedback in motion planning. Based on this analogy, the interpolated functional values can be treated as value functions which the robot can use for an intelligent navigation to reach the goal. In this setting,

$$\tilde{x}_W = \arg \max_{\tilde{x} \in \mathtt{Nbhd(R)}} \mathfrak{F}(\tilde{x}) \ \text{ and } \ \nu_W = \max_{\tilde{x} \in \mathtt{Nbhd(R)}} \mathfrak{F}(\tilde{x}) \tag{7}$$

Then, $\tilde{x}_W(\mathrm{T}_i)$ is the estimated way-point to which the robot needs to move, over the next time horizon $(\mathrm{T}_i, \mathrm{T}_{i+1}]$. Let $\{\tilde{x}_W(\mathrm{T}_1), \tilde{x}_W(\mathrm{T}_2), \ldots, \tilde{x}_W(\mathrm{T}_n)\}$ be the sequence of way-points estimated by the robot in the slow time scale at instants $\mathrm{T}_1, \mathrm{T}_2, \ldots, \mathrm{T}_n$. Then, if the robot moves in a way such that $\{\nu_W(\mathrm{T}_i),\ i = 1, \ldots, n\}$ is a monotonically increasing set i.e., $\nu_W(\mathrm{T}_1) \leq \nu_W(\mathrm{T}_2) \leq \ldots, \nu_W(\mathrm{T}_n)$, then the following result will holds:

$$\lim_{n \rightarrow \infty} ||\tilde{x}_W(\mathrm{T}_n) - X^G||_2 < \epsilon \tag{8}$$

where $\| \bullet \|_2$ is the standard Euclidean norm. Equation (8) follows from the fact that the measure function of agents is bounded above and thus the monotonicity of the estimated values ensure convergence. If the robot can communicate with sensors having a strictly positive measure function, then Proposition 4.2 suggests that there exists a sequence of sensors that finally leads to the region of interest.

There are several popular choices for RBFs such as Gaussian, inverse multi-quadric and thin spline. In general, the degree of smoothness of the estimated implicit surface can be controlled by changing the shape of RBFs. In this work, three different RBFs are explored for implicit surface estimation. The functional form of the RBFs are listed in Table I (where $r = ||\tilde{x} - \tilde{x}_i||_2$) .

In order to determine $\{w_i, i = 1 \text{ to } M\}$, a multiple regression algorithm is used. Details of the regression algorithm are being skipped for brevity. Interested readers are referred to [34]. Different steps for the implicit surface interpolation-based way-point generation are succinctly presented in Algorithm 2.

Remark 4.1: The kernel-based regression algorithm takes as input the position and belief of the individual mobile sensors and finds a statistical function using the kernel-function, where an objective function (e.g., an expected error minimization) could be used to solve the system of equations. This function creates a mapping from the configuration space of the robot to the belief space; the functional value yields an estimate of the belief at any particular position in the configuration space of the robot, based on the observations of the neighboring sensors. In the presence of obstacles, the domain of the function becomes limited to the free configuration space of the robot, $\mathbf{X}_{\text{free}} = \mathbf{X} \setminus \mathbf{X}_{\text{obs}}$, where $\mathbf{X}_{\text{obs}}$ denotes the part of the configuration space of the robot occupied by obstacles, and the properties of RBF-based regression functions still hold. The computational complexity of the algorithm for belief map construction is similar to that for solving linear regression problems that involve matrix inversion. It is well known that the worst-case computational complexity for matrix inversion is $O(n^3)$ where n is the order of the data matrix [35].

C. Rapidly Exploring Random tree (RRT)-based Open-Loop Controller Synthesis

In the last step, the robot gets an estimate of the way-point it should move to. This is used as an input to a low level continuous-time controller to find a feasible trajectory for the robot. A sampling-based algorithm is used to tackle the dynamics of the robot. In particular, based on the current location and the way-point found in the last step, a rapidly exploring random tree (RRT) is built in an anytime fashion to find a feasible trajectory for the robot. Specifically, the new estimate of the way-point is assigned as the new goal for tree expansion; the initial point is the current location of the robot. RRT is used to synthesize collision-free (with static obstacles) trajectories and the corresponding control inputs for moving the robot from the initial point to the target set. Since the estimate of the new way-point is provided relative to the robot, the RRT algorithm also operates in a relative coordinate system in a receding horizon fashion. When the robot receives a new estimate for the way-point, a new tree is grown for reaching the goal. This process terminates when the robot reaches the target set. The robot can, however, avoid other mobile sensors by locally communicating with them, and hence, they aren't considered while finding feasible trajectories to avoid extra computation and complexity. For completeness of the paper, RRT has been succinctly explained in Algorithm 3; it is well known that the computation complexity of the RRT algorithm is $O(m \log(m))$, where m is the number of sampled points. Thus, the overall complexity of the proposed algorithm is of the polynomial order. For more detailed information, interested readers are referred to [18].

This module considers the dynamics of the robot and based on the way-point estimated, it provides the sequence of control inputs that can navigate the robot to the way-point. It is noted that the RRT in Algorithm 3 finds only a feasible solution to the kinodynamic motion planning; finding optimality with RRT* [13] requires a solution to the steering function that has not been considered in the current work.

Remark 4.2: **Correctness**: The plan will always give the robot a path to the sensed goal. This is argued by making some observations. Due to the biased gossip algorithm based on the optimal control of a weighted PFSA, it is ensured that there is a gradient towards the goal region i.e., there always exists a sequence of hops from any sensor with a positive measure function to the sensor detecting the target. If the robot can communicate with sensors with positive measure function, then under the assumption of bounded uncertainties, in the relative localization estimates of the mobile sensors made by the robot within its communication radius, the robot can always locate a way-point which has a higher *belief* (as found by the interpolation function) than its current belief (corresponding to its current physical location). This follows from the fact that a non-zero measure function for a sensor implies that there exists at least one neighbor that has a higher measure than the sensor itself. The *agent measure function* is maximized for the sensor which detects the target. Therefore, if the robot moves in such a way such that its *measure function* (i.e., belief about the presence of a goal) monotonically increases, it will end up at the goal. Thus, the algorithms presented in Sections IV-A through IV-C can always search a path for the robot, if there exists a sensor in the network that could detect the target of interest at any time under consideration and the network is connected (i.e., the agent measure function ν is strictly positive for all sensors).

Remark 4.3: While the individual sensors participate in the generalized gossip for information propagation, the robot performs the kernel-based regression and motion planning; the motion planning algorithm is dependent on the regression algorithm. These two algorithms are performed at two different time-scales; the robot estimates the new way-point after it reaches the last way-point predicted by the estimator. The two-different algorithms have different time-complexity for which the respective problem size is different. The estimation-complexity grows with the number of near sensors robot can communicate with, while estimating the local belief map; the complexity of the motion planning algorithm grows with the number of points-sampled during trajectory planning.

V. Results and Interpretation

This section presents results of numerical experiments for an example problem of surveillance and reconnaissance which involves a mobile sensor network and an autonomous robot which needs to navigate to a target detected by the sensor network. A surveillance region of area A is monitored by N mobile sensors, where each mobile sensor has a communication radius R_s. The robot has a communication radius R_r and a sensing radius r_{hs}. The individual mission of each sensing agent is to detect any target and communicate its belief to its neighbors. The

global mission objective of the sensor network is to direct a robot with greater capabilities to the target region for the purpose of threat neutralization of or service delivery. For the simulation study, the parameters are chosen as: $A = 500^2, N = 150, R_s = 50,$ and $R_r = 150$. For modeling of target (see Section III), the value of P_D was chosen to be 0.9 and r_{hs} was chosen to be 20. The robot motion kinematics for the robot is given by the following equations.

$$\dot{x} = v\cos(\phi) \quad (9a)$$
$$\dot{y} = v\sin(\phi) \quad (9b)$$
$$\dot{\phi} = \omega \quad (9c)$$

where, $v \in [v_{\min}, v_{\max}]$ and $\omega \in [-\psi, \psi]$. The velocity of the mobile sensors in the network was chosen to be 5. The mobile sensors are moving in the region with a 2-D random walk fashion with the constant velocity. A slower velocity for the mobile sensors might result in a slower information propagation but, it results in a more stable local dynamics for the robot. Target is located at $[400, 200]$ while the robot is at $[1, 1]$ to begin with. ϵ (see equation 8) for mission termination is chosen to be equal to r_{hs}. The value of θ chosen for results shown in Figures 4 through 6 is 0.02.

The robot starts moving towards the goal as soon as its local neighborhood becomes aware of the target detection through gossip. Once the robot becomes aware of the detection, it makes use of the disseminated distributed belief about the target to find a path to the target. Figure 4 shows the implicit surfaces learned by the robot by communicating with the mobile sensors in its communication range, at different time instants in the slow time scale T using the inverse multiquadric RBF. It gives the estimated correspondence between a physical location and the belief (or awareness) about the target based on the measure function of the mobile sensors in the robot's neighborhood. The next way-point is then estimated by finding the maxima for the implicit surface. The robot then moves to the estimated way-point by following a trajectory found by sampling from the environment in an anytime fashion. Communication is re-established with neighboring mobile sensors after the robot reaches the estimated way-point. These plots together show the goal-directed navigation of the robot using the distributed information in the slow time scale T.

Figure 5(a) shows the monotonic increment in the belief of the estimated way-points during the navigation to the unknown goal. It can be seen in Figure 5(a) that as soon as the robot becomes aware of a sensed target (belief $>$ 0), it is able to move in such a fashion that its awareness about the presence of the target monotonically increases and finally converges to its maximum value as it reaches the goal. The belief of the way-points could also be used as a measure for degree of completion of the mission; convergence of the belief to its maximum value suggests mission completion. Figure 5(b) shows the relative change in robot's position w.r.t. the goal in the slow time scale T. It shows a monotonic convergence of the robot's position to the goal under the proposed framework. Figure 5(b) also shows the inherent goal-directedness in the robot's motion once it *sniffs* (i.e., belief > 0) the presence of a target in the surveillance region.

Figure 6 shows the actual trajectory found by the robot using RRT and the surface interpolation at T_5 (see Figure 5) using the Gaussian RBF. The contour plot in Figure 6 shows the level surfaces for the interpolation using the Gaussian RBF. For brevity the graphic details of the remaining two RBFs are skipped in this paper. Figure 7 shows planning in the presence of obstacles, where the domain of kernel-based regression (see Section IV-B) has been restricted to the free configuration space, $\mathbb{X}_{\text{free}}$, of the robot, because the robot's state at any instant is constrained to lie in $\mathbb{X}_{\text{free}}$. The obstacles are again considered while building the motion tree and a sampled point is added to the tree only if the corresponding edge is collision free (see Algorithm 3).

Figure 8 shows a scenario of robot navigation with three different generalized gossip parameters θ. The idea is to show the effect of the gossip parameter θ on the robot navigation. It was shown in [21] that θ controls the localization of information in a mobile sensor network. The network reaches consensus with θ very close to 0, i.e., all sensors have the same *agent measure function* ν. On the other hand, a value of θ closer to 1 results in localization of information around the target of interest. Under the biased gossip setting presented here, the gradient in the network is still controlled by θ. For θ close to 0 (see θ=0.02), there will be a uniform gradient across the network. For higher values of θ (0.2 and 0.8), the information is more localized in the sense shown in Figure 8. As θ is increased, there are more variations in the belief of the sensors. As a result, the robot will experience a steep gradient in the belief directed to the region of interest in the local neighborhood as compared to areas further away from it (see the plots of ν_W vs T for different θ in Figure 8). It is seen that the steepest gradient is found for $\theta = 0.8$ followed by $\theta = 0.2$. But, at the same time, information is more localized with increasing θ (notice the slow increase in ν_W for $\theta = 0.8$ and 0.2 before the sharp increase). The existence of a gradient is, however, independent of the values of θ; it only controls nature of the gradient. The robot should be able to navigate to the goal $\forall\theta \in (0, 1)$. It is noted that the results in Figure 8 correspond to the Inverse Multiquadric RBF.

Figure 9 shows the effect of bounded uncertainty in the sensor location estimates on the robot navigation. In this example, the x- and y-coordinate of the sensor locations w.r.t. the robot are modeled as independent Gaussian random variables with the mean as the actual location and standard deviation d. Hence, in other words, the robot is able to localize the sensors within a region d^2 ($d = 10$ for the example shown), with high confidence. With this uncertainty in sensor location estimates, the Maximum Likely (ML) paths are obtained for the robot to study its convergence behavior. To obtain the ML path estimate, a Monte-Carlo simulation is done where while estimating a way-point for the robot, the neighboring sensor locations, $\tilde{x}_i$ where $i = 1, 2, \ldots, M$ are sampled from the assumed distribution. A way-point is then estimated using the sampled locations.

This process is repeated N times to obtain an approximate distribution for the way-points and a ML estimate of the way-point is obtained from the distribution. Results for an example ($\theta = 0.8$, $N = 1000$) are shown in Figure 9, which show convergence of the ML paths, with steady-state deviation from the paths obtained with perfect information about the sensor location estimates.

VI. Summary, Conclusions, and Future Work

A framework for hierarchical planning for reactive navigation of autonomous robots is presented in this paper, where a mobile sensor network serves the dual-purpose of: (i) information exchange among the mobile sensors, and (ii) feedback control of robot motion to find feasible paths to follow. Specifically, a robot makes use of the collective intelligence of the distributed mobile sensor network to localize the goal by sequentially estimating the way-points converging to the goal point. Making use of a controlled gossip algorithm and sequential estimation of the way-points locally, it is shown that the robot is capable of finding a path to the goal point. However, efficacy of the proposed path planning algorithm is contingent upon the accuracy of localization techniques that are executed over the sensor network.

The work reported in this paper is different from those presented in current literature [25] in the sense that not all sensors in the network are required to detect the target. Such requirements of long-range sensing may become unrealistic in real-life scenarios such as those in the undersea environment. In the present formulation, an awareness about the presence of a local target is developed via the gossip algorithm and this information is fed back to the robot controller to find a path. The collective intelligence of the sensor network is used to generate a sequence of way-points, which finally converges to the sensed location of the region of interest. A sampling-based algorithm is used to tackle the dynamics of the robot at the control level. Under the prevalent conditions of limited sensing range and communication capabilities, the robot recursively estimates the way-points, based on the belief of its neighbors; and this process is repeated until the robot reaches a close vicinity of the sensed region of interest.

This paper presents initial results on use of unstructured data for source-seeking in large sensor networks, where the network is not strongly connected. While there are numerous research directions for path planning with distributed information, the following topics are recommended for future research.

1) Extension of the path planning algorithm in the presence of multiple targets and for multiple regions of interest as well as for large-scale high-dimensional environments.
2) Quantization of error bounds on navigation of the robot due to imperfections of target localization by the sensor network and relative localization of mobile sensors.
3) Identification of explicit relationships between network topology parameters and errors in robot motion control.
4) Use of closed-loop control for motion planning of the robot as open-loop planning and then online tracking of trajectories might be expensive and inefficient especially in GPS denied environments.
5) Experimental validation of the algorithm in laboratory settings.
6) While the motion planning module considers presence of obstacles during synthesis of trajectories, modeling the effects of obstacles in communication and sensing would require more detailed analysis with more accurate sensing and communication models. Analysis of the algorithm under such environments with more detailed models is a topic of future research.

References

[1] N. Atanasov, J. Le Ny, N. Michael, and G. Pappas, "Stochastic source seeking in complex environments," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 3013–3018, May 2012.

[2] S.-J. Liu and M. Krstic, "Stochastic source seeking for nonholonomic unicycle," *Automatica*, vol. 46, no. 9, pp. 1443–1453, 2010.

[3] L. Brinon-Arranz and L. Schenato, "Consensus-based source-seeking with a circular formation of agents," in *Control Conference (ECC), 2013 European*, pp. 2831–2836, July 2013.

[4] M. Jadaliha, J. Lee, and J. Choi, "Adaptive control of multiagent systems for finding peaks of uncertain static fields," *Journal of Dynamic Systems, Measurement, and Control*, vol. 134, no. 5, p. 051007, 2012.

[5] S.-i. Azuma, M. S. Sakar, and G. J. Pappas, "Stochastic source seeking by mobile robots," *Automatic Control, IEEE Transactions on*, vol. 57, no. 9, pp. 2308–2321, 2012.

[6] A. S. Matveev, H. Teimoori, and A. V. Savkin, "Navigation of a unicycle-like mobile robot for environmental extremum seeking," *Automatica*, vol. 47, no. 1, pp. 85–91, 2011.

[7] J. Choi, S. Oh, and R. Horowitz, "Distributed learning and cooperative control for multi-agent systems," *Automatica*, vol. 45, no. 12, pp. 2802 – 2814, 2009.

[8] M. S. Stankovic and D. M. Stipanovic, "Discrete time extremum seeking by autonomous vehicles in a stochastic environment," in *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pp. 4541–4546, IEEE, 2009.

[9] P. Ogren, E. Fiorelli, and N. Leonard, "Cooperative control of mobile sensor networks:adaptive gradient climbing in a distributed environment," *Automatic Control, IEEE Transactions on*, vol. 49, pp. 1292–1302, Aug 2004.

[10] Q. Li, M. De Rosa, and D. Rus, "Distributed algorithms for guiding navigation across a sensor network," in *Proceedings of the 9th annual international conference on Mobile computing and networking*, pp. 313–325, ACM, 2003.

[11] N. Deshpande, E. Grant, and T. Henderson, "Target localization and autonomous navigation using wireless sensor networks- A pseudogradient algorithm approach," *Systems Journal, IEEE*, vol. 8, pp. 93–103, March 2014.

[12] Y. Peres, A. Sinclair, P. Sousi, and A. Stauffer, "Mobile geometric graphs: Detection, coverage and percolation," in *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 412–428, SIAM, 2011.

[13] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[14] D. Levine, B. Luders, and J. P. How, "Information-rich path planning with general constraints using rapidly-exploring random trees," in *AIAA Infotech@ Aerospace Conference, Atlanta, GA*, 2010.

[15] A. Bry, A. Bachrach, and N. Roy, "State estimation for aggressive flight in GPS-denied environments using onboard sensing," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 1–8, IEEE, 2012.
[16] A. Bry and N. Roy, "Rapidly-exploring random belief trees for motion planning under uncertainty," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 723–730, IEEE, 2011.
[17] D. Li, Q. Li, N. Cheng, and J. Song, "Sampling-based real-time motion planning under state uncertainty for autonomous micro-aerial vehicles in GPS-denied environments," *Sensors*, vol. 14, no. 11, pp. 21791–21825, 2014.
[18] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.
[19] P. chattopadhyay, D. Jha, S. Sarkar, and A.Ray, "Path planning in gps-denied environments: A collective intelligence approach," in *American Control Conference, Chicago, IL, USA*, pp. 3082–3087, July 2015.
[20] A. Berdahl, C. J. Torney, C. C. Ioannou, J. J. Faria, and I. D. Couzin, "Emergent sensing of complex environments by mobile animal groups," *Science*, vol. 339, no. 6119, pp. 574–576, 2013.
[21] S. Sarkar, K. Mukherjee, and A. Ray, "Distributed decision propagation in mobile-agent proximity networks," *International Journal of Control*, vol. 86, no. 6, pp. 1118–1130, 2013.
[22] A. Ray, "Signed real measure of regular languages for discrete event supervisory control," *International Journal of Control*, vol. 78, no. 12, pp. 949–967, 2005.
[23] I. Chattopadhyay and A. Ray, "Language-measure-theoretic optimal control of probabilistic finite-state systems," *International Journal of Control*, vol. 80, no. 8, pp. 1271–1290, 2007.
[24] D. K. Jha, Y. Li, T. A. Wettergren, and A. Ray, "Robot path planning in uncertain environments: A language-measure-theoretic approach," *ASME Journal of Dynamic Systems, Measurement and Control*, vol. 137, no. 1, pp. 034503–1, 2015.
[25] N. A. Atanasov, J. Le Ny, and G. J. Pappas, "Distributed algorithms for stochastic source seeking with mobile robot networks," *ASME Journal of Dyn. Sys., Meas., Control*, vol. 137, p. 031011, 2015.
[26] Z. Toroczkai and H. Guclu, "Proximity networks and epidemics," *Physica A: Statistical Mechanics and its Applications*, vol. 378, no. 1, pp. 68–75, 2007.
[27] M. C. González, P. G. Lind, and H. J. Herrmann, "Networks based on collisions among mobile agents," *Physica D: Nonlinear Phenomena*, vol. 224, no. 1, pp. 137–148, 2006.
[28] S. Patterson, B. Bamieh, and A. El Abbadi, "Convergence rates of distributed average consensus with stochastic link failures," *Automatic Control, IEEE Transactions on*, vol. 55, pp. 880–892, April 2010.
[29] G. Mao, B. Fidan, and B. Anderson, "Wireless sensor network localization techniques," *Computer networks*, vol. 51, no. 10, pp. 2529–2553, 2007.
[30] D. K. Jha, T. A. Wettergren, A. Ray, and K. Mukherjee, "Topology optimisation for energy management in underwater sensor networks," *International Journal of Control*, no. just-accepted, pp. 1–14, 2015.
[31] T. A. Wettergren, "Performance of search via track-before-detect for distributed sensor networks," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 44, no. 1, pp. 314–325, 2008.
[32] I. Chattopadhyay, "Scalable ϵ-optimal decision-making and stochastic routing in large networks via distributed supervision of probabilistic automata," *SIAM Journal on Control and Optimization*, vol. 52, no. 4, pp. 2512–2547, 2014.
[33] G. Turk, H. Q. Dinh, J. F. O'Brien, and G. Yngve, "Implicit surfaces that interpolate," in *Proceedings of the International Conference on Shape Modeling & Applications*, SMI '01, IEEE Computer Society, 2001.
[34] C. M. Bishop *et al.*, *Pattern recognition and machine learning*, vol. 1. Springer, New York, NY, USA, 2006.
[35] M. Sipser, *Introduction to the Theory of Computation*. Cengage Learning, 2012.

Algorithm 1: Belief updating strategy for mobile sensors

while true do
  for all sensors 'i' in the network **do**
    if $\texttt{Nbhd(i)} \neq \texttt{0}$ **then**
      $d_i = \text{CARD}(\texttt{Nbhd(i)})$
      {Begin Infinite Asynchronous loop}
      {Query $\nu(\theta)_j$}
      if $\nu_j(\theta)|_\tau \leq \nu_i(\theta)|_\tau$ **then**
        $\Pi_{ii}|_\tau = \Pi_{ii}|_\tau + \Pi_{ij}|_\tau$
        $\Pi_{ij}|_\tau = 0$
      end if
      if $\nu_j(\theta)|_\tau > \nu_i(\theta)|_\tau \ \ \& \ \ \Pi_{ij}|_\tau = 0$ **then**
        $\Pi_{ij}|_\tau = 1/d_i$
        $\Pi_{ii}|_\tau = \Pi_{ii}|_\tau - 1/d_i$
      end if
    end if
    $\nu_i(\theta)|_\tau = (1-\theta)\sum_{j\in\{i\}\cup\texttt{Nbhd(i)}} \Pi_{ij}|_\tau \nu_j(\theta)|_\tau + \theta\chi^i|_\tau$
  end for
end while

Algorithm 2: Navigation of the Robot

1: **while** $\|X^R - X^G\|_2 > \epsilon$ **do**
2: Solve $\mathfrak{F}(\tilde{x}) = \sum_{i=1}^{M} w_i \phi(||\tilde{x} - \tilde{x}_i||)$ using boundary constraints $\{\tilde{x}_i, \nu_i\}, \tilde{x}_i \in \texttt{Nbhd}(\mathbb{R})$
3: Use $\mathfrak{F}(\tilde{x})$ to estimate $\tilde{x}_W = \arg\max_{\tilde{x} \in \texttt{Nbhd(R)}} \mathfrak{F}(\tilde{x})$ and $\nu_W = \max_{\tilde{x} \in \texttt{Nbhd(R)}} \mathfrak{F}(\tilde{x})$
4: RRT$(X^R, K, \Delta t, \tilde{x}_W)$ {For the function RRT, see Algorithm 3}
5: **end while**

Algorithm 3: Rapidly Exploring Random Tree (RRT)

Input:$(x_i, K, \Delta t, x_f)$
Output: Tree $G = (V, E)$ with a path P from $x_{\texttt{init}}$ to $x_{\texttt{final}}$
$V(0) = x_{\text{init}}$
$E(0) = \emptyset$
for k = 1 to K **do**
  $x_{\texttt{rand}} \leftarrow \texttt{RandConf}()$
  {Pick a point randomly in the configuration space of the robot}
  $x_{\texttt{near}} \leftarrow \texttt{NearestVertex}(\texttt{x}_{\texttt{rand}}, \texttt{G})$
  {Calculate the nearest vertex of the tree to x_{rand}}
  u_k : $\texttt{SelectInput}\,(\texttt{x}_{\texttt{rand}}, \texttt{x}_{\texttt{near}})$
  {select the input that takes the robot closest to $x_{\texttt{rand}}$}
  $x_{\texttt{new}} \leftarrow \texttt{NewState}\,(\texttt{x}_{\texttt{near}}, \texttt{u}_{\texttt{k}}, \Delta\texttt{t})$
  if $\texttt{CollisionFree}(\texttt{x}_{\texttt{near}}, \texttt{x}_{\texttt{new}})$ **then**
    $V \leftarrow V \cup x_{\texttt{new}}$
    $E \leftarrow E \cup (x_{\texttt{near}}, x_{\texttt{new}})$
  end if
end for
$x_{\texttt{final,near}} \leftarrow \texttt{NearestVertex}(\texttt{x}_{\texttt{final}}, \texttt{G})$
Retrace a path P from $x_{\texttt{final,near}}$ to $x_{\texttt{init}}$ over G.
return P

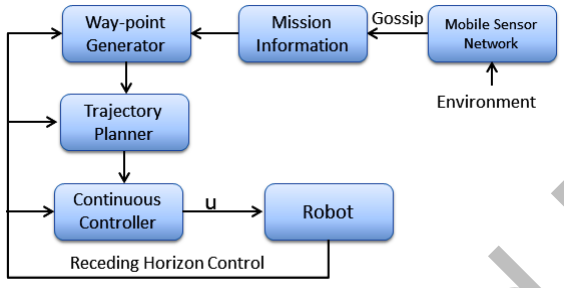


Fig. 1. Receding-horizon path planning on a distributed sensor network

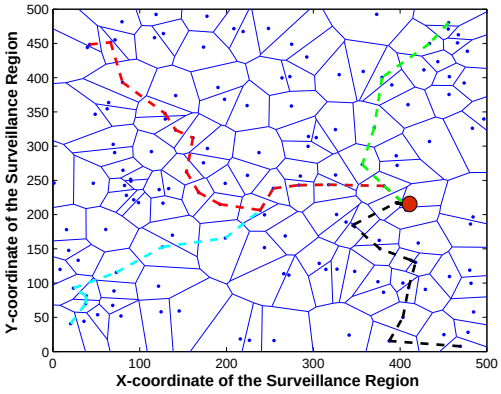


Fig. 2. Voronoi Partitioning of the network: A Frozen network in the slow time scale τ showing the Voronoi partitioning according to the mobile sensor location. In this particular setting, the individual cells of the partition are assigned the belief of the corresponding sensor and in a sense, is the simplest interpolation of the belief map. The figure also shows the sequence of cells that could be traversed, based on the gradient of the spatial belief, to reach the sensed target cell (shown as a red circle).

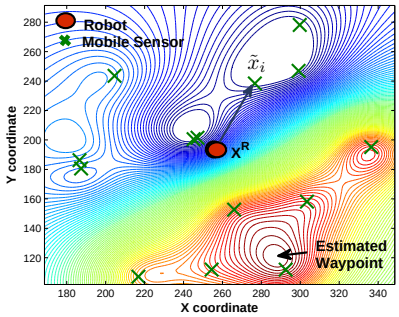


Fig. 3. Typical scenario for sensor localization: Relative locations of sensors are used by the robot to estimate a local implicit correspondence between a physical location in its neighborhood and the measure function ν.

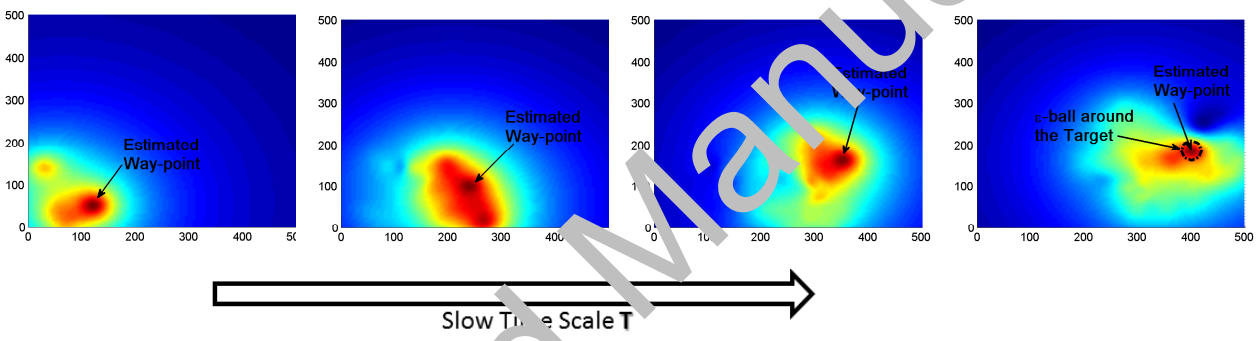


Fig. 4. Sequential estimation of way-points: Four plates show the sequential way-points estimated based on the communication of the robot with the neighboring sensors using the Inverse Multiquadric RBF. The robot moves to the way-point following the path found by RRT and then estimates the next way-point until it reaches the ϵ-neighborhood of the target.

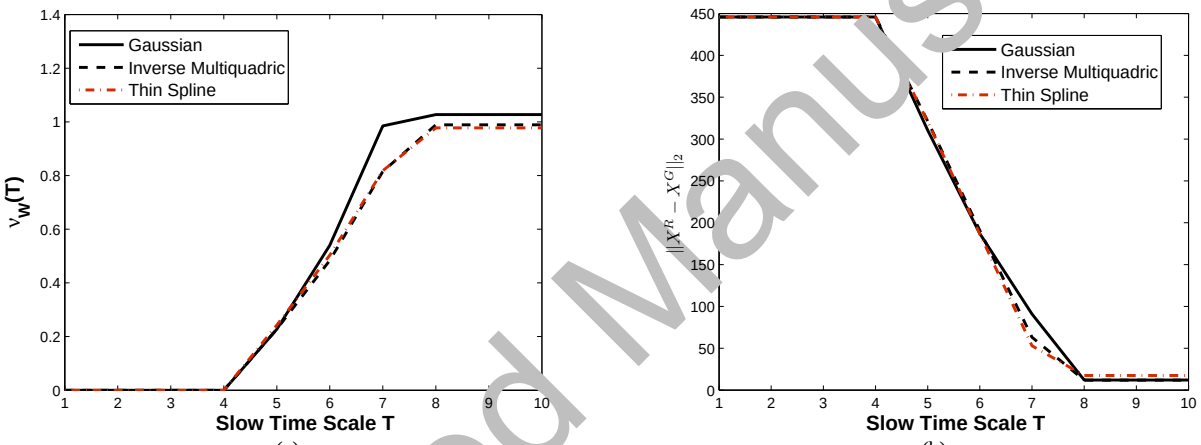


Fig. 5. Performance of way-point estimation on the slow time scale: Plate (a) shows the monotonic improvement in belief of the estimated way-points ν_W in the slow time scale (T). It is also considered as the degree of completion of the mission. Plate (b) shows the monotonic decrease in the Euclidean norm between the robot's location and the goal, measured in the slow time scale T.

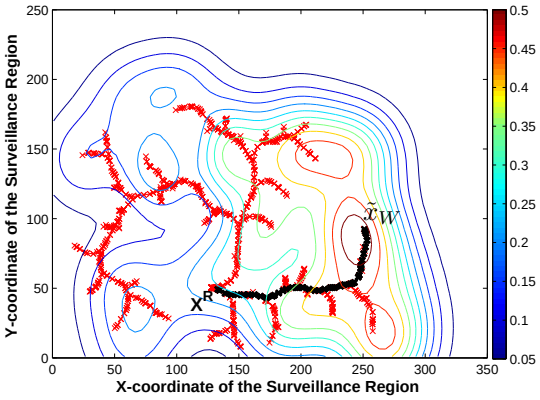


Fig. 6. Path found between two consecutive way-points: Rapidly-exploring random tree (RRT) (shown in black) and the dynamic model in Eq. (9) have been used. The contour shows the surface interpolated by using the Gaussian RBF. This is the actual trajectory followed by the robot between the time instants $\mathrm{T} = 5$ and $\mathrm{T} = 6$ in Figure 5.

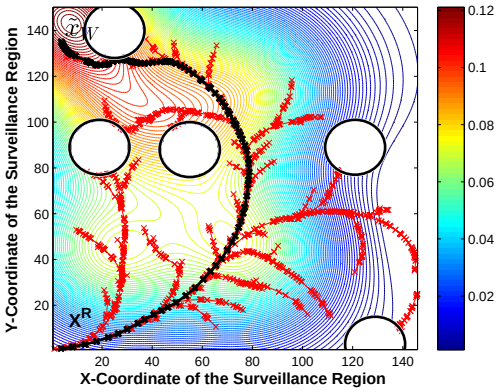


Fig. 7. Path found between two consecutive way-points in the presence of obstacles: Rapidly-exploring random tree (RRT) (shown in black) and the dynamic model in Eq. (9) have been used. The contour shows the surface interpolated by using the Gaussian RBF. The big circles denote the obstacles. The trajectory is shown in black.

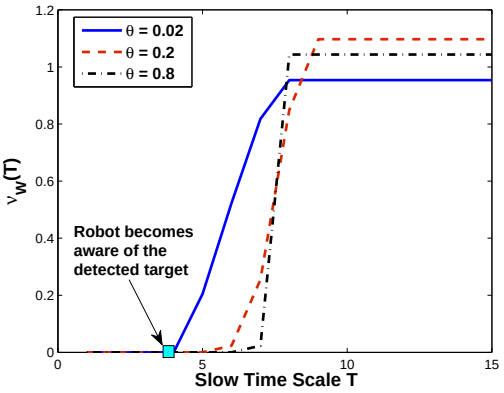
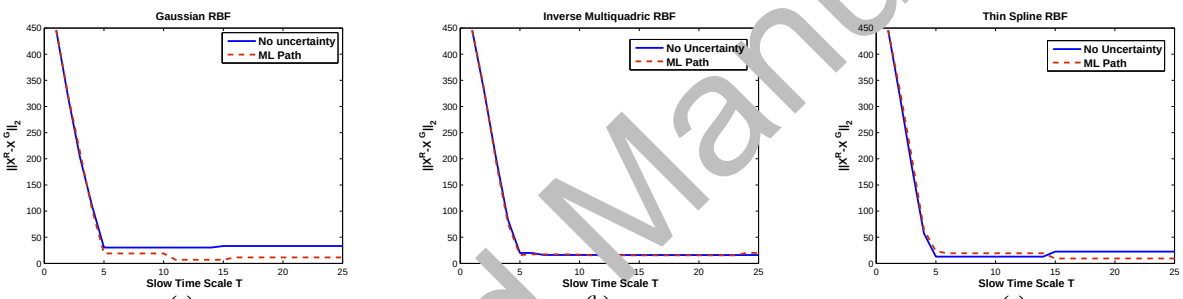


Fig. 8. Effect of parameter θ on performance of the distributed algorithm: Increasing the value of θ localizes the information of target detection in a small neighborhood around the target location resulting in high beliefs around the region of interest and comparatively smaller values away from the region.



(a) (b) (c)

Fig. 9. Maximum likely (ML) paths followed by the robot under imperfect localization of the mobile sensors in its local neighborhood.

TABLE I
THE RADIAL BASIS FUNCTIONS USED FOR INTERPOLATION

Function Name	Functional Form
Inverse Multiquadric	$\frac{1}{\sqrt{r^2+\sigma^2}}$
Gaussian	$\exp\left(-\frac{r^2}{2\sigma^2}\right)$
Thin Plate Spline	$r^2 \cdot \log r$