On a Decentralized Active Sensing Strategy using Mobile Sensor Platforms in a Network

Timothy H. Chung, Vijay Gupta, Joel W. Burdick and Richard M. Murray

Abstract— In this paper, we consider the problem of active sensing using mobile nodes as a sensor network to estimate the state of a dynamic target. We propose a gradient-search-based decentralized algorithm that demonstrates the benefits of distributed sensing. We then examine the task of tracking multiple targets, and address it via a simple extension to our algorithm. Simulation results show that our simple decentralized approach performs quite well and leads to interesting cooperative behavior.

I. INTRODUCTION

In recent years, much focus has been devoted to the theory and applications of sensor networks. Typical advantages of using sensor networks include relatively lower costs, inherent robustness and greater coverage area, as well as possibly heterogeneous sensing. These advantages are further enhanced if sensors are mobile.

One of the basic questions that arises in (mobile) sensor networks from an estimation perspective is: *How should the sensors move to attain the best estimates of the target(s)?* Robustness, communication complexity and performance requirements demand that solutions to this question not require a central computation node, especially as the number of nodes in the network increases. In this paper, we are specifically concerned with the problem of optimal positioning and motion of mobile sensors. This problem arises when the quality of observations of some target(s) varies with the location of the sensors. This can be due to the fact, e.g., that the observation noise varies with distance between target and sensor. Thus the objective is to plan the sensors' trajectories to obtain the best estimates of dynamic targets.

The area of research addressing the control of actuated or configurable sensors in the presence of uncertainty is called active sensing [1], [2]. Many scenarios where active sensing is useful, such as active vision [3], [4] and mobile robot navigation [5], [6], are well-surveyed in [7]. Sensor placement also has received considerable attention, for applications of vision [8], sensor coverage [9], and parameter estimation [10], [11], [12].

Attempts have been made over the years to address the problem of optimal motion control of sensors in various contexts, including application-specific examples such as

odor detection [13], collision avoidance [14], as well as estimation and tracking [15], [16], [17]. Further, [18] provides a general mathematical formulation of the problem, but notes that computing the optimal solution is computationally expensive.

However, many of these approaches are developed for single-sensor situations, whereas the methods applicable for multi-sensor teams, as in [15], require the optimization to be done centrally in general. On the other hand, [16] examines a distributed optimization algorithm along the lines presented in [19], and suggests that computational savings speak in favor of sub-optimal approaches. Further, while the decentralized algorithm presented in [17] is most relevant to the work discussed in this paper, the sensors in that work are assumed to be able to measure multiple targets simultaneously, and hence the problem of optimally assigning sensors to targets is not addressed.

The main contribution of this paper is the demonstration of how a simple decentralized algorithm can be used for motion-planning of sensors in a network to achieve significantly better estimates of the target state. The performance of the algorithm has been demonstrated through simulation examples where interesting cooperative behavior such as splitting into sub-teams is observed, even though no centralized optimizing command or signal is specifically transmitted to the sensors. This points to a large number of promising distributed sensing applications to which the algorithm and the approach can be applied, where decisions that affect the collective team performance need to be made by "intelligent" agents in a decentralized fashion.

The paper is organized as follows. In the next section, we set up the problem and define the conventions used. Then we propose the algorithm. We begin with the case of a single target for which certain analytical results can also be derived. We then extend to the case of multiple targets. The problem of multiple targets becomes more interesting if we consider sensors that can observe only one target at a time. In this case we have the additional problem of assigning sensors to targets. We then demonstrate the performance of the algorithm with a few simulation studies, concluding finally with some encouraging and interesting avenues for future work.

II. PROBLEM FORMULATION

In this section we formulate the problem of selecting sensor positions to realize the best observations of targets that move with time. For the purpose of this discussion, we assume that the targets are being modeled by a constant

position model while the sensors are capable of taking only range measurements with the measurement noise depending on the distance from the sensor to the target. In this we are motivated by standard sonar models [20]. The discussion can be easily generalized to more general target and sensor models. Consider m targets doing a random walk in a plane. The motion of the j-th target evolves according to

$$X_j[k + 1] = X_j[k] + w_j[k], \qquad (1)$$

where $X[k] \in \mathbb{R}^2$ is the state of the target at time k consisting of the x and y positions. $w_j[k] \in \mathbb{R}^2$ represents the process noise acting on j-th target that is assumed zero-mean, Gaussian and white with covariance matrix $Q_j[k] \in \mathbb{R}^{2 \times 2}$. Further the process noises of different targets are assumed independent of each other. Each target j is tracked by n sensors with the i-th sensor making observations $Y_{ij}[k] \in \mathbb{R}^2$ described by

$$Y_{ij}[k] = X_j[k] + T(\theta_{ij})v_{ij}[k]. \qquad (2)$$

$T(\theta_{ij}) \in SO(2)$ is the rotation matrix that transforms the noise from the local sensor coordinates to the global coordinate system. The noise $v_{ij}[k] \in \mathbb{R}^2$ is also assumed zero-mean, Gaussian and white. The noises of various sensors and targets are assumed mutually independent. The covariance matrix of $v_{ij}[k]$ is denoted by $R_{ij}[k] \in \mathbb{R}^{2 \times 2}$ and in keeping with the usual sonar models (see, e.g., [20]) it is assumed to be a diagonal matrix of the form

$$R_{ij}[k] = \begin{bmatrix} \left(\sigma_{\text{range}}^{ij}\right)^2 & 0 \\ 0 & \left(\sigma_{\text{bearing}}^{ij}\right)^2 \end{bmatrix}. \qquad (3)$$

$\left(\sigma_{\text{range}}^{ij}\right)^2$ is the range measurement noise variance and is represented by a function $f(r_{ij})$ of the distance r_{ij} from sensor i to target j. A common model of $f(r_{ij})$ is as a quadratic dependence on range, with the minimum value being achieved at a particular distance from the target, namely the "sweet spot" of the sensor. The bearing noise variance $\left(\sigma_{\text{bearing}}^{ij}\right)^2$ is often modeled (e.g. see [21]) as a fixed multiple α of the range noise variance. Thus we obtain that

$$R_{ij}[k] = \begin{bmatrix} f(r_{ij}) & 0 \\ 0 & \alpha f(r_{ij}) \end{bmatrix}. \qquad (4)$$

The time index k will be implied in the remainder of this paper, except where explicit indication is necessary.

To process the observations and generate an estimate, a Kalman filter (KF) is used. However a centralized KF for all the observations would be computationally very expensive. Instead, every node has a local KF that produces an estimate based only on local observations. Then these estimates and their covariances are exchanged and combined to yield a global estimate. The estimates are combined by assuming that there is no cross-covariance between local estimates.

Thus we use the relations [22]

$$P_{\text{global}}^{-1} \hat{X}_{\text{global}} = \sum_i^n P_{i,\text{local}}^{-1} \hat{X}_{i,\text{local}} \qquad (5)$$

$$P_{\text{global}}^{-1} = \sum_i^n P_{i,\text{local}}^{-1}. \qquad (6)$$

Note that the assumption of no cross-covariance between the local estimates is not strictly true [23] and hence the global estimate is sub-optimal. However, this algorithm is much simpler than its alternatives and seems not to incur huge performance penalties [24]. Note further that the sensor fusion algorithm itself is stable in the sense that the global error covariance matrix stays bounded. This is so since under the usual observability constraints, each sensor's local estimate will converge to a steady state and hence the global estimate will converge to a steady state whether or not cross-covariance between the different local estimates is assumed.

P_{global} refers to the covariance of the error in the global estimate and hence is an indicator of the quality of the estimate. Since the sensor noise covariance matrix is a function of the distance between the sensor and the target, the quality of the estimate depends on the distances between the various sensors and the targets. Thus by varying the positions of the sensors, we can vary the error covariance. The problem we pose is how to do so in a decentralized way. As a cost function, we seek to minimize the determinant of the error covariance matrix P_{global}. This is referred in the literature [7] as the D-optimal design.

A key feature of this work is the decentralized nature of the algorithm. Note that if the positions of all nodes at time step $k + 1$ (and hence their respective $R_{ij}[k + 1]$'s) were known, $P_{\text{global}}[k+1]$ could be computed by fusing the results of the Riccati recursion of the local error covariance matrices. Then the optimal positions of all nodes to minimize the cost function can be calculated. This is the essence of centralized methodologies, which is itself a challenging problem. Instead, in the distributed case, the responsibility of the fusion of estimates and the optimization of sensor trajectories and assignments is given to each sensor, rather than relying on a central computation node.

III. CASE I : SINGLE TARGET

To begin with, we assume that only a single target is present and every sensor is observing it. As mentioned in the previous section, at each time step, every sensor takes measurements, communicates local estimates, and fuses information from other sensors to obtain a global estimate $\hat{X}_{\text{global}}$ and a global error covariance matrix P_{global}.

The task that remains for each sensor is to identify its optimal location for the next time step. However, obtaining a solution for all sensors in closed form or even numerically is challenging, in general. In many practical cases, the number of active sensors may change, further complicating the optimization problem. Thus we use a gradient descent

algorithm which defines the optimal control action (i.e. direction of movement) as that which will position the sensors to minimize $\det(P_{\text{global}})$ in the next time step. Recall that the calculation of the gradient is intrinsically decentralized [25], and thus is quite suitable for our purposes. The optimization of sensor positions in the algorithm involves the following steps:

1) Each sensor node assumes other nodes remain at their present positions, allowing for calculation of their local error covariances at the next time step.
2) For each possible control action it can take, the node calculates its own error covariance at the next time step and generates the fused global error covariance matrix for each possible action.
3) It chooses the action that minimizes the cost.

The set of possible control actions is discrete and finite, and in this way the gradient descent algorithm is reduced to a discrete gradient search algorithm.

Thus the optimization procedure is decentralized, requiring only the communication of local information between nodes for data fusion. Note that when each sensor receives the local estimates from other nodes, it will also have implicitly received position information of the other sensor nodes. Further, if another node stops communicating, the sensor can simply disregard it in the sensor fusion step.

Investigation of Minima

Given the properties of descent algorithms, under the usual constraints of observability (for the Kalman filter), the error estimates will reach a steady state. Further, the sensor configuration will be such that the cost function reaches *a* minimum (provided the step size is small enough [25]).

However, as is the nature of gradient methods, a local, rather than global, minimum may be encountered. To examine the nature of minima, we take a closer look at the nature of the cost function. Assuming the system has reached the steady state described above, observe that the local error covariance matrices are independent of time. Only changes in sensor positions (through the measurement noise covariance matrices) have any further effect on the error. Thus, all other things being equal, we can relate changes in error covariance with changes in measurement noise covariance for the purposes of examining the qualitative behavior of the cost expression [10].

Note that, for a particular minimizing sensor configuration, an infinite number of equivalent minima can be found by planar rotation about the target. We choose to study the one that has sensor 1 on the x-axis, without any loss of generality. The location of other sensors $i = 2, \ldots, n$ are given by $(r_i \cos(\theta_i), r_i \sin(\theta_i))$. The cost function to be minimized has the form given by (6). This is equivalent to maximizing the expression $\det\left(\sum_i P_{i,\text{local}}^{-1}\right)$. Denoting $f(r_{ij})$ as f_i (i.e. single-target case), the i-th sensor's error estimate covariance matrix, $P_{i,\text{local}}$, is related, as per the previous discussion, to the measurement noise covariance,

$T(\theta_{ij})R_{ij}T(\theta_{ij})^T$:

$$\begin{bmatrix} \alpha f_i \sin^2(\theta_i) + f_i \cos^2(\theta_i) & (1-\alpha)f_i \cos(\theta_i)\sin(\theta_i) \\ (1-\alpha)f_i \cos(\theta_i)\sin(\theta_i) & \alpha f_i \cos^2(\theta_i) + f_i \sin^2(\theta_i) \end{bmatrix}.$$

It follows that the overall expression to be maximized has the form $AD - B^2$, where

$$A = \left[\frac{1}{f_1} + \frac{1}{\alpha}\sum\left[\frac{\alpha}{f_i} + \frac{(1-\alpha)\sin^2(\theta_i)}{f_i}\right]\right]$$

$$D = \left[\frac{1}{\alpha f_1} + \frac{1}{\alpha}\sum\left[\frac{\alpha}{f_i} + \frac{(1-\alpha)\cos^2(\theta_i)}{f_i}\right]\right]$$

$$B = \frac{1-\alpha}{\alpha}\left[\sum \frac{\cos(\theta_i)\sin(\theta_i)}{f_i}\right]$$

where the summation index i runs from 2 through n. Algebraic manipulation yields the following form:

$$\frac{1}{\alpha}\left[\sum_{i=1}^{n}\frac{1}{f_i}\right]^2 + \frac{(1-\alpha)^2}{\alpha^2 f_1}\left[\sum_{i=2}^{n}\frac{\sin^2(\theta_i)}{f_i}\right]$$
$$+ \frac{(1-\alpha)^2}{\alpha^2}\left[\sum_{2 \leq i < j \leq n}\frac{\sin^2(\theta_i - \theta_j)}{f_i f_j}\right], \qquad (7)$$

from which the following conclusions can be readily drawn:

1) There are, in general, many local maxima.
2) Since all the terms are positive and f_i appear only in the denominator, maxima are achieved when the f_i's (which depend only on the range between sensor and target) are minimized. Thus, all sensors end up at their respective optimal distances from the target, regardless of angles θ_i.
3) For the special case of only two sensors, we see that θ_2 must either be $\pi/2$ or $3\pi/2$, independent of the minimum values of f_1 and f_2. This agrees with our intuition of having the two sensors pointing in orthogonal directions.
4) For more than 2 sensors, assuming that the minimum values of f_i are equal, the different local maxima are found by solving for the angles θ_i, $i = 2, \ldots, n$ that maximize the expression

$$\sum_{i=2}^{n}\sin^2(\theta_i) + \sum_{2 \leq i < j \leq n}\sin^2(\theta_i - \theta_j).$$

5) Further, in this case, all local maxima have the same value of the cost expression.

We conclude from this investigation that we need not worry about distinguishing between local and global minima of the original cost function, at least for this simplified case. Simulation examples suggest that even in more general cases, nearly optimal performance is usually obtained.

IV. CASE II : MULTIPLE TARGETS

The algorithm proposed above can readily be extended for the tracking of multiple targets. This situation arises frequently in surveillance, computer vision, signal processing, etc. where a number of targets, features, or signals

appear and disappear from the field. The sensors aim to cooperatively obtain the best possible estimate of all these targets.

With sensors capable of taking measurements of all targets simultaneously, the tracking of multiple targets is exactly the case considered in the previous section, if we simply redefine the new target state to be a stacked vector of all the individual target states. We assume the ability to handle the data association issues [26]. The tracking problem is significantly more interesting (and realistic) if the sensor can observe only one target at a time, with access to a low-resolution observation (e.g. from overhead UAV's [27]) to indicate the presence of other targets on the field. In this case, it is clear that target assignment at each time step also affects the estimates of the target locations. This reflects an underlying problem of optimally forming sub-teams of sensors, which will be discussed in this section. As before, we seek a decentralized solution where this optimization is done efficiently at the nodal level.

To begin addressing this problem, we note that minimization of the cost function involves not only determining sensor motions but also assigning which (and how many) sensors observe each target at each time step. This addition of new assignment variables to optimize over, however, only augments the set of possible control actions, due to, in part, the structure of the discrete gradient search method. In other words, we can extend our algorithm in a simple way. In the optimization step, each node assumes both position and current sensing assignments of other sensors are fixed for the next time step, and optimizes its own position and assignment to minimize the next time step's global cost. Moreover the algorithm remains decentralized, as before, since communication only occurs during data fusion, and the optimization is done locally. The algorithm, as executed by each sensor, is represented in Table I.

V. Examples

In this section, we illustrate the proposed simple algorithm with the help of some demonstrative examples.

A. Single Target Tracking Problem

The first example compares the tracking performance of the decentralized method proposed in this paper against a fully centralized algorithm in which the optimization is done by a central node. We seek to show that the performance loss due to the decentralized approach is not substantial and well-worth the tremendous savings in computation.

Our test case is constructed as follows. A single target must be localized and tracked by three sensing agents. The target is subject to random walk in a plane, described by (1), with process noise covariance matrix given by

$$Q[k] = \left[\begin{array}{cc} 0.01 & 0 \\ 0 & 0.01 \end{array} \right]$$

The i-th sensor observes the target according to (2). The measurement noise covariance matrix is given by (4), with

TABLE I
DECENTRALIZED ALGORITHM PSEUDO-CODE FOR EACH SENSOR IN MULTI-TARGET TRACKING SCENARIOS

```
For k=1:sim_time

    % —— Local Observation ——
    Take local measurement;
    Update local estimate X̂_local and error covariance matrix P_local;

    % —— Sensor Fusion ——
    Transmit local information to other sensors;
    Receive information from other sensors;
    Fuse all local estimates to get global estimate X̂_global;

    % — Optimization of target assignment and sensor position —
    Assume other sensors do not move;
    Assume other sensors do not change target assignment;
    Propagate P_local of other sensors by one time step;
    For all own target assignments
        For all own allowable motion actions
            Propagate own P_local;
            Fuse with propagated P_local of other sensors;
            Obtain cost function estimate;
        end
    end
    Identify cost-minimizing position and target assignment;

    % —— Update position and assignment ——
    Update position for next time step;
    Update target assignment;
end
```

$\alpha = 5$ and the range-dependent quadratic function is the same for all three sensors, given by $f(r_i) = 0.0008r_i^2 - 0.0250r_i + 0.3481$.

For this setup, the resulting sensor configuration is illustrated in Fig. 1. As predicted by our analysis of (7) as well as intuition, we see the three sensors at their "sweet spot" range from the target and also $120°$ apart from each other.
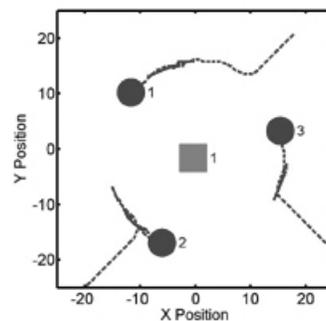

Fig. 1.   Optimal sensor configuration

Moreover, even though our decentralized algorithm is sub-optimal, we see (Fig. 2) that in comparison to the completely centralized exhaustive search method over all possible control actions for all sensors, the performance loss is very little ($< 2\%$). In fact at steady state, the decentralized algorithm effectively attains the minimal cost as found by the centralized method.

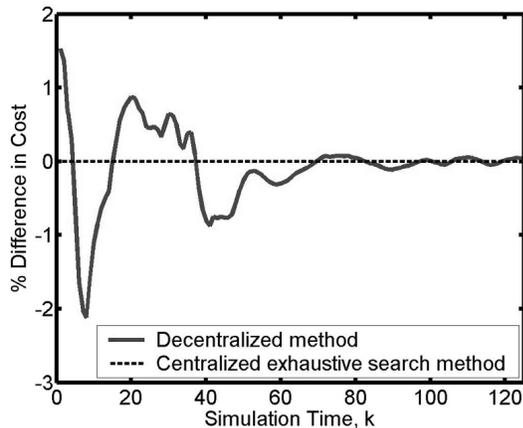We see an initial drop in performance, which indicates that the convergence of the decentralized method to the opti-

Fig. 2. Performance comparison between exhaustive search optimization and proposed decentralized method.

Fig. 3. (a) Initial optimal sensor configuration. (b) Sub-team forms and observes second target, while remaining sensors adjust formation. (c) Sub-team rejoins original group. The dotted lines represent the tracks made by the sensors.

mal sensor locations (and thus, the target position estimate) is somewhat slower than that of the centralized algorithm. Nevertheless, the steady-state performance is generally most relevant to the problem of optimal estimation of the target state. Thus, we see that the proposed decentralized algorithm offers not only the benefit of distributed-ness, but also of performance in cost minimization and in computation.

B. Patrolling Problem

The second example illustrates the use of our algorithm in a distributed surveillance application (e.g. RoboFlag [28], [29]), and demonstrates the assignment methods of matching sensors with targets. The sensor motions in this simulated experiment are governed by our decentralized algorithm, by optimizing over possible local control actions and assignments for each sensor.

The initial setup is illustrated in Fig. 3(a). We assume sensors can observe only one target at a time. Multiple such sensors (depicted by the circles) are initially assigned to a target (represented by the lower square), much like vehicles assigned to patrol a defensive zone. Another target (e.g. an opponent vehicle) enters the playing field (Fig. 3(b)), and is observed by a UAV or an arbiter [29] which alerts the mobile sensors. A sub-team of sensors is automatically formed to track this second target. The remaining sensors around the first target maneuver themselves to optimally cover the first target. When the second target disappears, these sensors readjust to accommodate the returning sub-team so that all the sensors are again covering only the first target, as shown in Fig. 3(c).

Thus we observe several interesting behaviors exhibited by the system, some surprisingly complex given the simplicity of the algorithm. Firstly, we find that the division of the sensing task over multiple targets is a consequence of the distributed nature of our algorithm, rather than any prescribed method or heuristic approach. The sensors are able to optimally split into sub-teams without needing to explicitly address the issues of consensus, communication
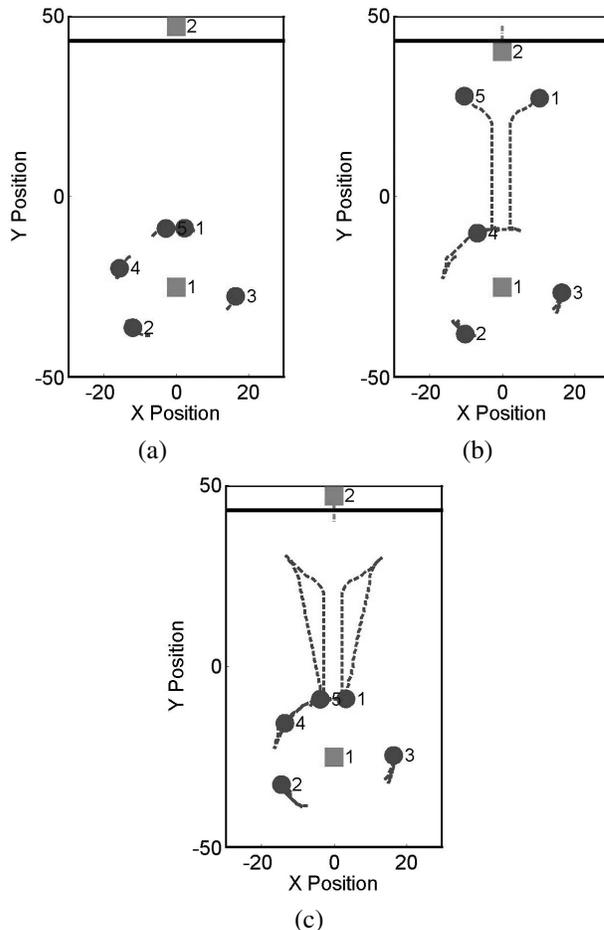
(except during the data exchange step), and coordination of motion for formation control.

Additionally, each member of a sensor sub-team maneuvers optimally with respect to other members, modifying the formation dynamically with the addition/removal of sensing agents. Again, what makes this behavior interesting is that it is simply due to the decentralized optimization of the cost function done in our algorithm. In other words, *a priori* designation of formations types, division of sensing tasks, and optimal estimation trajectories is not an input to the decentralized algorithm. Nevertheless, the behaviors and benefits of such a designation emerge.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we studied the problem of active sensing using multiple cooperative sensor nodes. The objective of the sensor team is to jointly estimate the state of some dynamic targets. We first investigated the single-target scenario, and examined some properties of the cost function in terms of its minima.

We then proposed a simple and intuitive algorithm that is

decentralized in nature, with the ultimate goal of applying it to the case of observing multiple targets. This distributed sensing task additionally introduced the problem of sensor-target assignment which was addressed by a simple extension to the algorithm. After identifying some illustrative examples, we showed by simulation that the sub-optimal approach is nearly optimal at steady state. Finally, we sought to apply the decentralized algorithm to tracking multiple targets, and discovered that the algorithm exhibits several interesting and promising behaviors for distributed sensing objectives.

Although most of the work presented here so far has largely been of an exploratory nature, the results point to a number of interesting avenues for future work. The problem of distributed sensing is of significant interest due to the resulting complexity, yet this decentralized algorithm allows us, in a fairly simple manner, to address and observe some of the pertinent issues, such as the split/rejoin maneuvers for formations. An immediate extension of our work is to formulate further analysis of these issues and to understand the role they play in mobile sensor networks in general.

Additionally, we can begin to investigate the relaxation of assumptions such as perfect localization of the sensors [30], which will reflect the existence of a more realistic, albeit noisy, world. Also, we hope to examine the behavior of our algorithm under network constraints such as network connectedness [31] and other issues pertaining to communication in sensor networks. Another assumption that might be relaxed is perfect data association in the case of multiple target tracking.

Further, we are interested in understanding the role of distributed sensing in "intelligent team" scenarios, where a collection of sensing agents may observe, classify, and/or even learn the strategies and behavior of opponents. Considering pursuit-evasion applications such as discussed in [27], the distributed sensing algorithm presented in this paper may be extended to address scenarios where targets now move to counter the motion of sensors, thereby attempting to increase the uncertainty in the sensor measurements.

REFERENCES

[1] M. Shimojo and M. Ishikawa, "An Active Touch Sensing Method Using a Spatial Filtering Tactile Sensor," in *Proc. of 1993 IEEE Intl. Conf. on Robotics and Automation*, vol. 1, May 1993, pp. 948–954.

[2] J. J. Clark and R. P. Hewes, "Active Sensing at a Microscopic Scale," in *Proc. of the 5t IEEE Int. Symp. on Intelligent Control*, vol. 1, Philadelphia, PA, September 1990, pp. 246–251.

[3] M. Sznaier and O. Camps, "Control issues in active vision: open problems and some answers," in *Proc. IEEE Conf. on Decision and Control*, vol. 3, Dec 1998, pp. 3238–3244.

[4] J. Aloimonos, I. Weiss, and A. Bandyopadhyay, "Active Vision," *Int. J. Computer Vision*, pp. 333–356, 1988.

[5] H. S. Oh, C. W. Lee, and I. Mitsuru, "Navigation control of a mobile robot based on active vision," in *Proc. Int. Conf. Industrial Electronics, Control and Instrumentation*, vol. 2, 1991, pp. 1122–1126.

[6] P. Mowforth, "Active sensing for mobile robots," in *Proc. Intl. Conf. on Control*, vol. 2, 1991, pp. 1141–1146.

[7] L. Mihaylova, T. Lefebvre, H. Bruyninckx, K. Gadeyne, , and J. D. Schutter, "Active Sensing for Robotics - A Survey," in *Proc. of the 5th International Conference on Numerical Methods and Applications*, Borovets, Bulgaria, August 2002, pp. 316–324.

[8] J. M. D. Luis E. Navarro-Serment and P. K. Khosla, "Optimal sensor placement for cooperative distributed vision," in *Proc. of the 2004 IEEE Conf. on Robotics and Automation*, New Orleans, LA, USA, April 2004, pp. 939–944.

[9] S. Dhillon and K. Chakrabarty and S. Iyengar, "Sensor Placement for Grid Coverage under Imprecise Detections," in *Proc. Intl. Conf. on Information Fusion*, 2002, pp. 1571–1587.

[10] D. Ucinski, *Measurement Optimization for Parameter Estimation in Distributed Systems*. Technical University Press, 1999.

[11] ——, "Optimal sensor location for parameter estimation of distributed processes," *Int. J. Control*, vol. 73, no. 13, pp. 1235–1248, 2000.

[12] A. Arbel, "Sensor Placement in Optimal Filtering and Smoothing Problems," *IEEE. Trans. Automatic Control*, vol. 27, no. 1, pp. 94–98, Feb 1982.

[13] T. Nakamoto, H. Ishida, and T. Moriizumi, "Active odor sensing system," in *Proc. Intl. Symposium on Industrial Electronics*, vol. 1, July 1997, pp. SS128–SS133.

[14] T. Heng, Y. Kuno, and Y. Shirai, "Active sensor fusion for collision avoidance," in *Proc of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 3, 1997, pp. 1244–1249.

[15] J. R. Spletzer and C. J. Taylor, "Dynamic Sensor Planning and Control for Optimally Tracking Targets," *International Journal of Robotics Research*, vol. 22, no. 1, pp. 7–20, January 2003.

[16] T. Mukai and I. Ishikawa, "An active sensing method using estimated errors for multisensor fusion systems," *IEEE Transactions on Industrial Electronics*, vol. 43, no. 3, pp. 380–386, June 1996.

[17] A. Makarenko, E. Nettleton, B. Grocholsky, S. Sukkarieh, and H. Durrant-Whyte, "Building a decentralized active sensor networks," in *11th Intl Conf on Advanced Robotics*, Coimbra, Portugal, 2003.

[18] O. A. Basir and H. C. Shen, "Informational maneuvering in dynamic environment," in *IEEE Int. Conf. on Systems, Man and Cybernetics*, vol. 2, October 1995, pp. 999–1004.

[19] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *Proc. Of the 3rd International Workshop on Information Processing in Sensor Networks*, Berkeley, CA, April 2004.

[20] K. Ramachandra, *Kalman Filtering Techniques for Radar Tracking*. New York, NY: Marcel Dekker, Inc., 2000.

[21] K. Umeda, J. Ota, and H. Kimura, "Fusion of multiple ultrasonic sensor data and imagery data for measuring moving obstacle's motion," in *Proc. of Intl. Conf. on Multisensor Fusion and Integration for Intelligent Systems*, December 1996, pp. 742–748.

[22] Y. Bar-Shalom and T. E. Fortman, *Tracking and Data Association*. Academic Press, 1988.

[23] Y. Bar-Shalom, "On the Track-to-Track Correlation Problem," *IEEE Trans. on Automatic Control*, vol. 26, no. 2, pp. 571–572, April 1981.

[24] Y. Bar-Shalom and L. Campo, "The effect of the common process noise on the two-sensor fused-track covariance," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 22, no. 6, pp. 803–805, 1986.

[25] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation : numerical methods*. Englewood Cliffs, NJ: Prentice Hall, 1989.

[26] M. Chu, S. Mitter, and F. Zhao, "Distributed multiple target tracking and data association in ad hoc sensor networks," in *Proc. of the Sixth Intl. Conf. of Information Fusion*, vol. 1, 2003, pp. 447–454.

[27] R. Vidal, O. Shakernia, H. Kim, D. Shim, and S. Sastry, "Probabilistic pursuit-evasion games: Theory, implementation and experimental evaluation," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 662–669, Oct 2002.

[28] A. Chaudhry, R. D'Andrea, and M. Campbell, "Roboflag - a framework for exploring control, planning, and human interface issues related to coordinating multiple robots in a realtime dynamic environment," in *Proc. of Intl. Conf. on Advanced Robotics*, 2003.

[29] R. D'Andrea and R. Murray, "The roboflag competition," in *Proc. of the American Control Conference*, June 2003, pp. pp. 650–655.

[30] S. Roumeliotis and G. Bekey, "Distributed multi-robot localization," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 781–795, Oct 2002.

[31] D. Spanos and R. Murray, "Robust connectivity of networked vehicles," in *IEEE Conference on Decision and Control*, 2004.