This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE Globecom 2010 proceedings.

LQR Formulation of Sensor Deployment for Decision Fusion Based Detection

Ahmad Ababnah, *IEEE Student Member* and Balasubramaniam Natarajan, *IEEE Senior Member*
WiCom Group, Department of Electrical & Computer Engineering
Kansas State University, Manhattan, Kansas 66506-5204
Telephone: (785) 532-4664, Email: ababnah@ksu.edu, bala@ksu.edu

***Abstract*—Assuming that decision fusion is employed for detection in a distributed sensor network (DSN), what is the best sensor deployment strategy to satisfy specific detection and false alarm rate requirements? This is the fundamental question we address in this paper. Towards this end, we propose a novel treatment of the deployment problem in the context of optimal control theory. Specifically, we model the deployment problem as a linear quadratic regulator problem, with the deployment locations serving as control parameters. Within this framework, we propose a sequential sensor deployment algorithm. We compare the performance of the proposed algorithm against that of a greedy algorithm. Simulation results, show that the proposed algorithm can save up to 40% in the number of sensors used relative to the greedy algorithm.**

I. Introduction

In DSN based detection systems, a number of sensors (with limited sensing and communication capabilities) are deployed in an area of interest with the goal of detecting an anomalous phenomena or target. For a detection system to be reliable, it has to meet certain minimum detection and maximum false alarm requirements. One can use fusion techniques combining data or decisions from multiple sensors in order to meet false alarm/detection performance requirements. Due to limited sensing capabilities, the performance of any DSN based detection system is dependent on the number of available sensors, the fusion technique employed as well as the sensor locations within the area of interest.

In a DSN , information fusion [1] can be used to meet false alarm and detection requirements simultaneously. In [2], sensor deployment in a DSN employing value fusion was investigated. In value fusion, the average of unprocessed measurements from multiple sensors is compared to a decision threshold. The authors use the Constrained Simulated Annealing (CSA) algorithm in developing low complexity deployment algorithms. In decision fusion, sensors detect a target/phenomena and send their decisions to a fusion center. Decision fusion offers the advantages of being more power and bandwidth efficient than value fusion. In [3] and [4], a detection decision is made if at least one sensor reported a detection decision. However, the overall false alarm requirements were not included in the proposed algorithms. In [5], instead of detection probability, the authors use the path exposure criteria decision fusion based DSN. However, their deployment algorithm is heuristic in nature. In [6], deployment occurs along a one dimensional line and the deployment positions are uniformly spaced. The authors find the optimal spacing distance between sensors so as to satisfy detection and false alarm requirements. However, deployment in a 1-D space is not realistic. Additionally, mandating that the sensors are equally spaced might require more sensors than are really needed to satisfy requirements. In summary, prior efforts in sensor deployment strategies that account for decision fusion are either heuristic or overly simplistic.

In this paper, we study the sensor deployment problem in a decision fusion based detection system. Here, sensors then send their decisions on the presence/absence of a target to a fusion center (FC) that employs a majority rule in order to make an overall detection decision. Given a fixed number of sensors, our goal is to develop a deployment strategy that will result in satisfying the false alarm requirements while minimizing the difference between achieved and required detection probabilities in such a system. Under this scenario, we propose a novel framework for the solution of the deployment problem. We model the deployment problem as an optimal control problem. Specifically, the deployment problem is modeled as a linear quadratic regulator (LQR). In an LQR problem, a system's state value evolves linearly with respect to its previous value and with respect to a control vector. The goal in an LQR problem is to find the optimal control vector that results in minimizing a weighted squared error cost function. Towards this end, we derive an linear approximation of the effect of adding a sensor on the difference between achieved and required overall detection probability, which corresponds to our

This work was partially funded through the award of a contract from the Marine Corps Systems Command (MCSC) to M2 Technologies Inc.

978-1-4244-5638-3/10/$26.00 ©2010 IEEE

system state. The control vectors at each discrete step in the evolution of the system correspond to the position of the sensor to be deployed. Furthermore, the squared error between achieved and required detection probabilities serves as the cost function in the LQR problem. Solving for the optimal control vector is equivalent to solving for the optimal sensor deployment positions. One can sequentially solve for the control vector (i.e., sequentially deploy sensors), by solving the set of Karush-Kuhn-Tucker (KKT) conditions associated with the LQR problem. We employ the sweep method, which is commonly used to solve for the control vectors in an LQR problem. The performance of the proposed algorithm is compared against that of a greedy deployment algorithm. Simulation results show that using the proposed algorithm it is possible to save as much as 40% in the number of sensors compared to the greedy algorithm to meet the same requirements.

II. System Model

The area of interest is modeled as a grid $\mathcal{G}$ of $N_x \times N_y$ points. The required false alarm/detection probabilities at all points in the grid are arranged in two $N_x N_y \times 1$ vectors denoted by $\mathbf{p}_f^{req}$/ $\mathbf{p}_d^{req}$, respectively. We assume that sensors have an exponential sensing model [7]. Specifically, if the distance between a sensor and a point of interest is d meters, then the probability of the sensor detecting a target located at that point p_{detect} is given as

$$p_{\text{detect}} = \begin{cases} e^{-\alpha d} & \text{if } d \leq R \\ 0 & \text{if } d > R \end{cases} \quad (1)$$

where, α is a decay rate that depends on sensor design and R is the detection radius. We assume that all sensors to be deployed are identical and have a common α and R. We further assume that all sensors have a common false alarm rate of p_f^s.

In a general decision fusion based system, sensors make local decisions regarding the existence (hypothesis H_1) or absence (hypothesis H_0) of a phenomena/target. The local decision u_i made by the i-th sensor corresponds to the index of the hypothesis decided upon. Local decisions from multiple sensors are communicated to a fusion center where a global decision u_0 regarding the two hypothesis is made. In our model, the fusion rule we employ is the counting rule. Assuming k sensors report their decisions to the FC, then the decision statistics U is given as $U = \sum_{i=1}^{k} u_i$. The decision rule corresponds to

$$u_0 = \begin{cases} 1 \text{ (i.e., } H_1 \text{ is true)} & \text{if } U \geq T \\ 0 \text{ (i.e., } H_0 \text{ is true)} & \text{if } U < T \end{cases} \quad (2)$$

where, T is the decision threshold. Therefore, the system's overall false alarm and detection probabilities at the j-th grid point with k sensor decisions ($p_f^k(j)$ and $p_d^k(j)$, respectively) are given as

$$p_f^k(j) = Pr(U \geq T(j) \mid H_0 \text{ is true}) \quad (3)$$

$$p_d^k(j) = Pr(U \geq T(j) \mid H_1 \text{ is true}) \quad (4)$$

The decision threshold $T(j)$ is chosen so as to satisfy the false alarm requirement $p_f^{req}(j)$. We arrange the achieved (after sensor deployment) false alarm and detection probabilities in two $N_x N_y \times 1$ vectors $\mathbf{p}_f^k$ and $\mathbf{p}_d^k$, respectively.

The problem of interest in this paper, can now be stated as follows: Determine the positions on the grid where a given number of sensors (K) are to be deployed in order to minimize the squared difference between achieved and required detection probabilities, while not exceeding the permissible false alarm requirements. Mathematically, it can be stated as follows

$$\underset{\mathbf{u}}{\text{argmin}} \sum_{j:p_d^K(j)<p_d^{req}(j)} (\mathbf{p}_d^K(j) - \mathbf{p}_d^{req}(j))^2$$
$$\text{subject to} \begin{cases} \mathbf{p}_f^K \leq \mathbf{p}_f^{req} \\ \mathbf{1}^T \mathbf{u} = K \end{cases} \quad (5)$$

where, $\mathbf{u}$ is the deployment vector. The deployment vector is an $N_x N_y \times 1$ vector. Its entries indicate the number of sensors at each point on the grid, and take values of either 0 or 1. $\mathbf{1}^T$ indicates the transpose of an $N_x N_y \times 1$ vector, with all entries set to 1. Note that in Eqn.(5), the squared error is taken at the grid points where detection requirements are not satisfied.

III. Optimal Control Formulation

Noting that the deployment vector $\mathbf{u}$ in Eqn.(5) is a binary vector, one can view the deployment problem as an integer programming problem. However, integer programming problems are NP-hard and are computationally expensive to solve especially for a large number of optimization variables ($N_x \times N_y$ in our case). Our approach to solving the deployment problem relies on modeling the approximate effect of sequential deployment of sensors on the overall false alarm and detection probabilities at the different grid point. However, we note that this approximation is only used for deployment purposes and after the sensor is deployed, the actual resulting overall detection probability is calculated, and used in our proposed system evolution model. This means that any discrepancy between using the approximate value and the actual one will be accounted for and corrected with the evolution of the system.

978-1-4244-5638-3/10/$26.00 ©2010 IEEE

A. System approximation and linearization

Since sensors have a common false alarm rate p_f^s, the decision statistics U under hypothesis H_0 follows a binomial distribution with parameters p_f^s and k where k is the number of sensors. Therefore, the overall false alarm probability at the j-th point with $k(j)$ sensors covering this point is given as

$$p_f^k(j) = Pr(U \geq T(j)|H_0) \quad (6)$$

$$= \sum_{n_j=T(j)}^{k} \binom{k}{n_j} (p_f^s)^{n_j} (1-p_f^s)^{k-n_j}. \quad (7)$$

On the other hand, the overall detection probability can not be easily characterized. This is due to the dependence of a sensor's detection probability on the distance separating it from the point of interest. For illustration purposes, suppose that the point of interest (j) is observed by 3 sensors; s_1, s_2 and s_3. Denoting the detection probability at the i-th sensor as p_d^i, and assuming that the threshold T is set to 2, then the overall detection probability at the j-th point is given as

$$p_d(j) = p_d^1 p_d^2 q_d^3 + p_d^1 q_d^2 p_d^3 + q_d^1 p_d^2 p_d^3 + p_d^1 p_d^2 p_d^3 \quad (8)$$

where, $q_d^i = 1 - p_d^i$. However, since the sensors' detection probabilities depend on the distance to the point (j), $p_d(j)$ is a stochastic r.v. with different possible realizations. Considering all different realizations is a computationally intensive process. Therefore, for mathematical tractability, we consider the ensemble average of $p_d(j)$ denoted by $\overline{p}_d(j)$. Assuming that the sensor decisions are independent, it is possible to express $\overline{p}_d(j)$ as

$$\overline{p}_d(j) = \overline{p}_d^1 \overline{p}_d^2 \overline{q}_d^3 + \overline{p}_d^1 \overline{q}_d^2 \overline{p}_d^3 + \overline{q}_d^1 \overline{p}_d^2 \overline{p}_d^3 + \overline{p}_d^1 \overline{p}_d^2 \overline{p}_d^3 \quad (9)$$

where $\overline{p}_{d_i}$ refers to the average detection probability for the i-th sensor and $\overline{q}_{d_i} = 1 - \overline{p}_{d_i}$. The average is calculated using the detection probabilities at all discrete points within the circle of radius R. Since we assume that sensors are identical, it then follows they share a common average detection probability which we denote by $\overline{p}_d^s$. The average overall detection probability can now be approximated as the ccdf of a binomial distribution with parameters $\overline{p}_d^s$ and (k). Let $X \sim B(k,p)$ denote a r.v. X, that follows a binomial distribution with success rate p and k trials. It is possible to approximate the ccdf of X as [8]

$$Pr(X \geq x) \approx Q(\frac{x-0.5+np}{\sqrt{npq}}) \approx 1 - \frac{1}{1+e^{-\sqrt{2}(\frac{x-0.5+np}{\sqrt{npq}})}} \quad (10)$$

where $q = 1-p$, and $Q(\cdot)$ is the ccdf of a standard Gaussian distribution. Therefore, it is possible to approximate $p_d^{k(j)}(j)$, the average overall detection probability at the j-th point when $k(j)$ sensors are involved in the decision process as

$$\overline{p}_d^{k(j)}(j) \approx Q(\frac{T(j)-0.5-n\overline{p}_d^s}{\sqrt{k(j)\overline{p}_d^s\,\overline{q}_d^s}}) \quad (11)$$

where $\overline{q}_d^s = 1-\overline{p}_d^s$. Using the approximation in (10) and rearranging the terms in Eqn.(11), we get

$$\ln(\frac{1}{1-\overline{p}_d^{k(j)}(j)} - 1) = \sqrt{2}(\frac{k(j)\overline{p}_d^s + 0.5 - T(j)}{\sqrt{k(j)\overline{p}_d^s\,\overline{q}_d^s}}). \quad (12)$$

Let $m^{k(j)}(j) = \ln(\frac{1}{1-\overline{p}_d^{k(j)}(j)} - 1)$. Denoting the decision threshold as $T^{k(j)}(j)$, it is possible using Eqn.(12) to express $m^{k(j)+1}(j)$ as follows

$$m^{k(j)+1}(j) = m^{k(j)}(j) + \sqrt{2}(\frac{\overline{p}_d^s - \delta^{k(j)+1}(j)}{\alpha_{k(j)}\sqrt{k(j)\overline{p}_d^s\,\overline{q}_d^s}}) - \gamma_{k(j)} m^{k(j)}(j) \quad (13)$$

where, $\delta^{k(j)+1}(j)$ is the change in the decision threshold (i.e., $T^{k(j)+1}(j) = T^{k(j)}(j) + \delta^{k(j)+1}(j)$), $\alpha_{k(j)} = \sqrt{\frac{k(j)+1}{k(j)}}$, and $\gamma_{k(j)} = \frac{1}{\alpha_{k(j)}} - 1$. In essence, Eqn.(13) approximates the effect of adding of a sensor on the average overall detection probability. The false alarm probability requirement is always met by a proper choice of the decision threshold. Let $p_d^{req}(j)$ denote the required detection probability at point (j), then $m^{req}(j) = \ln(\frac{1}{1-p_d^{req}(j)} - 1)$. Furthermore, define $x^{k(j)}(j)$ as

$$x^{k(j)}(j) = m^{k(j)}(j) - m^{req}(j) \quad (14)$$

then it is possible to express the change in $x(j)^{k(j)}$ after adding an additional sensor as

$$x^{k(j)+1}(j) = (1-\gamma_{k(j)})x^{k(j)}(j) + \gamma_{k(j)} m^{req}(j) - \sqrt{2}(\frac{\overline{p}_d^s - \delta^{k(i)+1}(j)}{\alpha_{k(j)}\sqrt{k(j)\overline{p}_d^s\,\overline{q}_d^s}}). \quad (15)$$

Considering all points on the grid, then it is possible to express Eqn.(15) in matrix form as

$$\mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k + \mathbf{B}_k \mathbf{u}_k \quad (16)$$

where, the matrix subscripts refer to the total number of sensors deployed in the grid. The matrix $\mathbf{A}_k$ is a square diagonal matrix of dimension $N_x N_y$, where the j-th diagonal element corresponds to the j-th point on the grid and is given as $\mathbf{A}_k(j,j) = (1-\gamma_{k(j)})$. The matrix $\mathbf{B}_k$ is also a square matrix of dimension $N_x N_y$. The (j,i)-th entry corresponds to the change in the average overall detection probability at the j-th point on the grid if a sensor were to be deployed at the i-th point. If the distance between points (i) and (j) is less than R, then $\mathbf{B}_k(j,i) = \sqrt{2}(\frac{\overline{p}_d^s - \delta(j)^{k(j)+1}}{\alpha_{k(j)}\sqrt{k(j)\overline{p}_d^s\overline{q}_d^s}}) - \gamma_{k(j)} m^{req}(j)$, otherwise it is set to 0. Here, $\mathbf{u}_k$ is the deployment

vector, which indicates the positions of sensors that are to be deployed at each point of the grid.

Note that the squared error (SE) between achieved and required detection probabilities, can be described as the weighted quadratic norm of the state ($\mathbf{x}_k$) of the system described in Eqn.(16). Therefore, the deployment problem in Eqn.(5) can be restated as

$$\underset{\mathbf{u}_k}{\text{argmin}}\, J = \frac{1}{2}\mathbf{x}_K^T\mathbf{Q}_f\mathbf{x}_K + \frac{1}{2}\sum_{k=0}^{K-1}(\mathbf{x}_k^T\mathbf{Q}\mathbf{x}_k + \mathbf{u}_k^T\mathbf{R}\mathbf{u}_k)$$
$$\text{subject to} \begin{cases} \mathbf{p}_f^K \preceq \mathbf{p}_f^{req} \\ \mathbf{x}_{k+1} = \mathbf{A}_k\mathbf{x}_k + \mathbf{B}_k\mathbf{u}_k,\ k = 0, \ldots, K-1 \\ \mathbf{1}^T\mathbf{u} = K \end{cases} \tag{17}$$

where, $\mathbf{Q}$, $\mathbf{Q}_f$ and $\mathbf{R}$ are symmetric positive definite weighing matrices. To avoid incurring a penalty for satisfying/exceeding detection requirements, the error terms corresponding to a point where the detection requirement has been met/exceeded is set to zero in J. The optimal control problem corresponding to our system is the linear quadratic regulator (LQR) problem. For the system described in Eqn.(16) and the cost function J in Eqn.(17), it is both necessary and sufficient for the optimal vectors to satisfy the set of Karush-Kuhn-Tucker conditions associated with the LQR problem [9]. One can solve the for the sequence of optimal vectors using the sweep method [9], in which the optimal control vector is given as

$$\mathbf{u}_k = -\mathbf{S}^{-1}\mathbf{R}^{-1}\mathbf{B}_k^T\mathbf{P}_{k+1}\mathbf{A}_k\mathbf{x}_k \tag{18}$$

where, the matrix $\mathbf{S}$ is of dimension $N_xN_y \times N_xN_y$ and is defined as

$$\mathbf{S} = \mathbf{I} + \mathbf{R}^{-1}\mathbf{B}_k^T\mathbf{P}_{k+1}\mathbf{B}_k \tag{19}$$

where, $\mathbf{P}_k$ is a square matrix of dimension $N_x \times N_y$ and is the solution of the Riccati equation associated with the LQR problem [9] and $\mathbf{I}$ is the identity matrix. In order to have a binary integer solution, a 1 is placed at the index where $\mathbf{u}_k$ is maximum and a 0 is placed at the remaining positions.

IV. Deployment Algorithm

In order to guarantee that both false alarm and detection requirements are met, we propose the optimal control based algorithm illustrated in Algorithm 1. Giving more importance to satisfying the false alarm requirements, in the first while loop the algorithm determines the points (I_δ) at which false alarm requirements are not met. Giving a higher priority for sensor deployment at these points, will give more freedom in choosing the suitable decision threshold which will facilitate satisfying the false alarm requirements at those points.

Algorithm 1 Optimal Control Based Algorithm

1: **Inputs**: $\mathbf{p}_f^{req}$, $\mathbf{p}_d^{req}$ and K
2: **Outputs**: $\mathbf{u}$, $\mathbf{p}_f^K$ and $\mathbf{p}_d^K$
3: **Initialization**: $\mathbf{u} = \mathbf{0}$ and $k = 0$
4: **while** $\mathbf{p}_f^k \succeq \mathbf{p}_f^{req}$ and $k < K$ **do**
5: Find I_δ s.t. $\mathbf{p}_f^{req}(I_\delta) < \mathbf{p}_f^k(I_\delta)$
6: Solve for $\mathbf{u}_k$ using Eqn.(18)
7: Find $i_{\max}$ s.t. $\mathbf{u}_k(i_{\max}) \geq \mathbf{u}_k(i),\ \forall i \in I_\delta$
8: Deploy a sensor at grid point $i_{\max}$ (i.e., set $\mathbf{u}(i_{\max}) = 1$)
9: Increment k
10: Change detection thresholds
11: Calculate $\mathbf{p}_f^k$ and $\mathbf{p}_f^k$
12: **end while**
13: **while** $\mathbf{p}_d^k \preceq \mathbf{p}_d^{req}$ and $k < K$ **do**
14: Find I_δ s.t. $\mathbf{p}_d^{req}(I_\delta) > \mathbf{p}_d^k(I_\delta)$
15: Set $\mathbf{x}(j) = 0,\ \forall j \nsubseteq I_\delta$ (equivalent to taking the SE at unsatisfied points)
16: Solve for $\mathbf{u}_k$ using Eqn.(18)
17: Find $i_{\max}$ s.t. $\mathbf{u}_k(i_{\max}) \geq \mathbf{u}_k(i),\ \forall i \in I_\delta$
18: Repeat steps 8-11.
19: **end while**

After calculating the deployment vector $\mathbf{u}_k$, a sensor is deployed at the the point in I_δ that corresponds to the entry with the largest value in $\mathbf{u}_k(I_\delta)$. This will have the additional effect of satisfying the detection requirements at these points. In the case that false alarm requirements have been met at all points, the second while loop is concerned with meeting the detection requirements. The contribution of the points I_δ, at which detection requirements are not met, is considered in calculating the deployment vector $\mathbf{u}_k$. This is done by setting entries in $\mathbf{x}$ corresponding to points with satisfied requirements to 0. After each sensor deployment, the decision thresholds are modified in order to satisfy the false alarm requirements. The algorithm terminates when all available sensors are deployed or, if K is sufficiently large, when all detection requirements are met.

V. Simulation Results

In this section, we compare the performance of two deployment algorithms, namely: a greedy algorithm and the optimal control based algorithm. In the greedy algorithm, sensors deployed at the grid point where the maximum difference between achieved and required detection probabilities occur. In the first experiment, the area of interest is modeled as a uniformly spaced 25×25 grid. Sensors to be used are identical and share the parameters $\alpha = 0.1$, $R = 6$, and $p_f^s = 0.05$. The false alarm and detection requirements are uniform over

TABLE I
MINIMUM NUMBER OF SENSORS REQUIRED BY EACH ALGORITHM

p_d^{req}	Greedy	Optimal
0.6	21	16
0.7	32	21
0.9	60	47

TABLE II
MINIMUM NUMBER OF SENSORS REQUIRED BY EACH ALGORITHM

p_f^{req}	Greedy	Optimal
0.05	27	22
0.01	33	23
0.005	48	41

the grid. The number of sensors needed to satisfy the requirements as reported by each algorithm is for different detection probabilities. The false alarm requirement is set to $p_f^{req} = 0.01$. Numerical results are shown in Table I, where it is evident that the optimal control based deployment algorithm use a fewer number of sensors than the greedy algorithm. The proposed algorithm employs as 34% fewer sensors than the greedy algorithm to satisfy required detection/false alarm requirements. In the second experiment, we study the performance of the deployment algorithms as the false alarm rate is varied. The setup of the second experiment is similar to that of the first experiment. The detection probability requirement is set to $p_d^{req} = 0.8$. Using the optimal control based algorithm, the savings in the number of sensors used can be as high as 25% in comparison to the greedy algorithm as shown in Table II .

We finally compare the performance of the algorithms when the detection requirements are not uniform over the grid as in Fig. 1. In this case, while the greedy algorithm requires 37 sensors, the optimal control based algorithm only needs 22 in order to satisfy the same requirements. This corresponds to about 40% reduction in number of sensors relative to the greedy algorithm.

Simulation results illustrate that the proposed deployment algorithm outperforms the greedy algorithm with respect to the number of sensors needed to satisfy the detection/false alarm requirements. This is due to the fact that the proposed algorithm incorporates more information in making the deployment decision than the greedy algorithm. In the greedy algorithm, sensors are deployed based on the effect the addition of a sensor will have on a single point. In contrast, the proposed algorithm incorporates the approximate effect the deployment of a sensor at any candidate position will have on all the points that fall within its detection radius R.

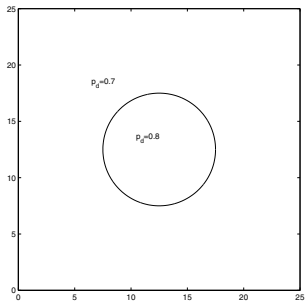

Fig. 1. Required Detection Probability Profile

VI. CONCLUSION

In this paper, we investigate the sensor deployment problem in a decision fusion based DSN system. Specifically, we determine the minimum number of sensors to be deployed, as well as their positions in a grid in order to meet both false alarm and detection requirements. We use optimal control theory principles to model deployment as an LQR problem, and develop a sweep method based deployment algorithm. Performance comparison against a greedy deployment algorithm illustrates the advantages of the proposed algorithm. For given detection/false alarm requirements, the savings in the number of sensors required can be as much as 40% relative to a greedy algorithm.

REFERENCES

[1] R. Viswanathan and P. Varshney, "Distributed detection with multiple sensors i. fundamentals," *Proceedings of the IEEE*, vol. 85, no. 1, pp. 54–63, Jan 1997.

[2] Z. Yuan, R. Tan, G. Xing, C. Lu, Y. Chen, and J. Wang, "Fast sensor deployment for fusion-based target detection," in *IEEE Real-Time Systems Symposium (RTSS'08)*, Dec 2008.

[3] J. Zhang, T. Yan, and S. H. Son, "Deployment strategies for differentiated detection in wireless sensor networks," in *Sensor and Ad Hoc Communications and Networks, 2006. SECON '06. 2006 3rd Annual IEEE Communications Society on*, vol. 1, 28 Sept 2006, pp. 316–325.

[4] Y. Zou and K. Chakrabarty, "Uncertainty-aware and coverage-oriented deployment for sensor networks," *Journal of Parallel and Distributed Computing*, vol. 64, no. 7, pp. 788–798, July 2004.

[5] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan, and K. K. Saluja, "Sensor deployment strategy for target detection," in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, 2002, pp. 42–48.

[6] S. K. Jayaweera, "Optimal node placement in decision fusion wireless sensor networks for distributed detection of a randomly-located target." Military Communications Conference, 2007. MILCOM 2007. IEEE, 29-31 Oct 2007, pp. 1–6.

[7] K. Chakrabarty, S. Iyengar, H. Qi, and E. Cho, "Grid coverage for surveillance and target location in distributed sensor networks," *IEEE Transactions on Computers*, vol. 51, pp. 1448–1453, 2002.

[8] V. C. Raykar, R. Duraiswami, and B. Krishnapuram, "A fast algorithm for learning a ranking function from large-scale data sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 3, no. 7, pp. 1158–1170, July 2008.

[9] M. Mahmoud and M. Singh, *Discrete Systems: Analysis, Control and Optimization*. Springer-Verlag, 1984.