ON A STOCHASTIC ALGORITHM FOR SENSOR SCHEDULING

Vijay Gupta Timothy H. Chung Babak Hassibi Richard M. Murray

California Institute of Technology
Pasadena, California

Abstract: A stochastic algorithm for solving the sensor selection problem is presented. The problem arises when many sensors are jointly trying to estimate a process but only a subset of them can take measurements at any time step. The proposed stochastic sensor selection strategy is easy to implement and is computationally tractable. The algorithm is illustrated through simple examples of sensor scheduling and dynamic sensor coverage.

Keywords: Scheduling algorithms, Sensors, Stochastic systems, Trajectory planning, Ricatti equations

1. INTRODUCTION AND MOTIVATION

Recently there has been a lot of interest in networks of sensing agents which act cooperatively to obtain the best estimate possible, e.g., see (Roumeliotis and Bekey 2002, Hall and Llinas 1997, Viswanathan and Varshney 1997) and the references therein. While such a scheme admittedly has higher complexity than the strategy of treating each sensor independently, the increased accuracy often makes it worthwhile. If all the sensors exchange their measurements, the resulting estimate can be even better than the sensor with the least measurement noise (were no information exchange happening). Some of the basic problems in sensor networks from an estimation perspective are fusion of data from multiple nodes, data association, sensor scheduling in case all sensors cannot measure or transmit simultaneously, optimal positioning of sensors etc. All these problems become more complicated when multiple targets are present. As the number of nodes in the network grows, it also becomes desirable for purposes of robustness and communication complexity that the solutions to these problems should not involve a central computation node. In this paper, an algorithm to solve the problem of sensor scheduling is presented. The algorithm can also be extended for use in the problem of optimal positioning and trajectory generation of sensors.

The problem of sensor scheduling arises when one (or multiple) sensors have to be selected out of N given sensors at every time step for taking measurements. This might be the case if, e.g., there are echo-based sensors like sonars which can interfere with each other. Another situation where sensor scheduling is useful is in tracking and discrimination problems, where a radar can make different types of measurements by transmitting a suitable waveform each of which has a different power requirement. There might be shared communication resources (e.g., broadcast channels or a shared communication bus) that constrain the usage of many sensors at the same time. Such a situation arises, e.g., in telemetry-data aerospace systems. Because of its importance, this problem has received considerable attention in literature. The seminal work in (III *et al.* 1967) proved a separation property between the optimal plant control policy and the measurement control policy for LQ control. The measurement control problem, which is the sensor scheduling problem, was cast as a non-linear deterministic control problem and shown to be solvable by a tree-search in general. Forward dynamic programming and a gradient method were proposed for the purpose. To deal with the complexity of a tree-search, greedy algorithms have been proposed many times, some examples being (Oshman 1994, Kagami and Ishikawa 2004, Gupta *et al.* 2004). Allied contributions have dealt with robust sensor scheduling (A. Savkin and R. Evans and E. Skafidas 2000), a greedy algorithm with an information based cost measure (Zhao *et al.* 2002) and the works of (Miller and Runggaldier 1997, Krishnamurthy 2002, Rago *et al.* 1996) etc. A different numerical approach to solve the problem was provided in (Athans 1972) who cast the problem as a two-point boundary value problem. This approach was further considered in (Kerr and Oshman 1995, Herring and Melsa 1974). Our algorithm differs from these approaches in that it is based on the idea of letting the sensors switch randomly according to some optimal probability distribution to obtain the best *expected* steady-state performance. Besides being computationally more tractable than the other solutions proposed in the literature, it does not rely on the sensors having considerable computational power or knowledge about other sensors. There are numerous other advantages as will be pointed out later in the paper.

The paper is organized as follows. The next section deals with the problem formulation. Then the random sensor selection algorithm is presented. The algorithm is analyzed by using some of the tools developed in the pioneering work (Sinopoli *et al.* 2004) and extended in (Liu and Goldsmith n.d.). A few simple numerical examples are used to illustrate the algorithm and some of its advantages. Finally, the conclusions and avenues for future research are presented.

2. MODELING AND PROBLEM FORMULATION

Consider a system evolving as

$$x[k+1] = Ax[k] + Bw[k]. \qquad (1)$$

$x[k] \in \mathbf{R}^n$ is the process state at time step k and $w[k]$ is the process noise. The process noise is assumed white, Gaussian and zero mean with covariance matrix R_w. The process state is being observed by N sensors $S_1, S_2, \cdots, S_N$ with the measurement equation for the i-th sensor being

$$y_i[k] = C_i x[k] + v_i[k], \qquad (2)$$

where $y_i[k] \in \mathbf{R}^s$ is the measurement. The measurement noises $v_i[k]$'s for the sensors are assumed independent of each other and of the process noise. Further the noise $v_i[k]$ is assumed to be white, Gaussian and zero mean with covariance matrix R_i. At every time step, one sensor is chosen to take the measurement and the measurement is communicated to all the sensors in an error-free manner.[1] Since all measurements are being shared, all the sensors have the same estimate of the process state $x[k]$, denoted by $\hat{x}[k]$, that is given by a Kalman filter assuming a time-varying sensor. Assuming that the i-th sensor takes the measurement at time step k, the estimate error covariance evolves according to the equation

$$\begin{aligned} P[k+1] &= AP[k]A^T + BQB^T \\ &- AP[k]C_i^T \left(C_i P[k] C_i^T + R_i\right)^{-1} C_i P[k] A^T. \qquad (3) \end{aligned}$$

If the initial state $x[0]$ has covariance Π_0, the initial condition is given by $P[0] = \Pi_0$.

It is obvious from (3) that error covariance is a function of the sensor schedule. The goal is to find the sensor schedule that minimizes the steady state error covariance. Obviously all the possible sensor schedule choices can be represented by a tree structure. The depth of any node in the tree represents time instants with the root representing time zero. The branches correspond to choosing a particular sensor to be active at that time instant. Each node is associated with the cost function evaluated using the sensor schedule corresponding to the path from the root to that node. Obviously, finding *the* optimal sequence requires traversing all the paths from the root to the leaves in the tree. If the leaves are at a depth d, a total of 2^d schedules need to be compared. This procedure might place too high a demand on the computational and memory resources of the system. In the next section, an alternative algorithm that does not involve traversing the tree is presented.

The problem of optimal sensor trajectory generation can also be cast in the above framework. For simplicity, assume that only one sensor is present. The area to be covered is discretized into a grid and it is assumed that the discretization is fine enough so that only the evolution of the process at these points needs to be observed. The limited sensing range is modeled by assuming (for example) that if the sensor is at a particular point, it generates measurements corresponding to that point only. As a simple example consider that the area to be monitored has been discretized into N points, denoted by $l_1, l_2, \cdots, l_n$. At location l_i, the process evolves according to the equation

$$x_i[k+1] = A_i x_i[k] + \sum_{j \neq i} A_{ij} x_j[k] + B_i w_i[k],$$

[1] Note that the assumption of one sensor being allowed per time step is without loss of generality.

where the process state at location l_i is assumed to be also affected by the process states at other locations. Thus by stacking the process states at all these locations into a single vector $x[k]$, it can be seen that for the entire process, the evolution is of the form given in (1). Similarly assume that if the sensor is at location l_i, its measurement is described by the equation

$$y_i[k] = H_i x_i[k] + v_i[k].$$

By defining C_i to be a matrix of the form $\begin{bmatrix} 0 & \cdots & 0 & H_i & 0 & \cdots & 0 \end{bmatrix}$, this can easily be recast in the form of (2). Clearly there are N such virtual sensors. Thus the sensor trajectory problem is equivalent to the sensor scheduling problem described earlier where N sensors are present but only one can be selected to take the measurement. Physical constraints on the sensor motion can be modeled, e.g., by assuming that the sensor can move from its current location only to its immediate neighbors. This constraint can be easily modeled by assuming that the sensors are selected with transition probabilities described by a Markov chain. The states of the Markov chain represent the current location. Thus the probability of moving from one location to a location far away in a single time step is zero. For the special case of $A_{ij} = 0$ and movement according to i.i.d. random variable, slightly stronger results can be derived, as in (Tiwari *et al.* 2005).

3. DESCRIPTION OF THE ALGORITHM

Our algorithm consists of choosing sensors randomly according to some probability distribution. The probability distribution is chosen to minimize the expected steady-state error covariance. We will present the results for the case of probabilities being given by a Markov chain and then specialize for the case of sensors being chosen in an i.i.d. manner. Denote

$$\begin{aligned} f_C(P) &= APA^T + BQB^T \\ &\quad - APC^T\left(CPC^T + R\right)^{-1} CPA^T, \\ f_C^k(P) &= \underbrace{f_C\left(f_C\left(\cdots\left(f_C(P)\right)\right)\right)}_{f_C \text{ applied } k \text{ times}}. \end{aligned}$$

Thus the evolution of the estimate error covariance of system (1) when a sensor of the form

$$y[k] = Cx[k] + v[k]$$

is chosen at time k is given by $P[k+1] = f_C\left(P[k]\right)$.

Suppose that at time k, the sensor S_i is chosen from the set $S_1, S_2, \cdots, S_N$ according to a Markov chain with probability transition matrix $Q = [q_{ij}]$, where q_{ij} is the probability of choosing sensor j at time step $k+1$ given that sensor i was chosen at time step k. Then the expected error covariance evolves as

$$E\left[P[k+1]\right] = E\left[f_{C_i}\left(P[k]\right)\right]. \quad (4)$$

Explicit evaluation of the right hand side appears to be intractable. Instead bounds are presented and then the steady-state upper bound is sought to be minimized. Detailed proofs for the following results are given in (Gupta and Chung 2004).

Theorem 1. Denote $q_i = \max_j q_{ji}$. Then an upper bound for $E\left[P_k\right]$ is X_k where

$$X_{k+1} = \begin{cases} \sum_i q_i f_i\left(X_k\right) & k \geq 1 \\ \sum_i \pi_0 f_i\left(P_0\right) & k = 0, \end{cases}$$

where π_0 is the initial probability of being in state i. Note that this condition holds for time-varying probabilities as well. However in the case when the probabilities are time-invariant, a sufficient condition for covergence of the error covariance is that X_k converges as k progresses. The convergence of this recursion for all initial conditions $X[0] \geq 0$ is equivalent to the existence of a positive definite matrix P and matrices $K_1, K_2, \cdots, K_N$ such that

$$\begin{aligned} P > \sum_{i=1}^{N} & q_i(BR_wB^T + K_iR_iK_i^T) + \\ & \left(A + K_iC_i\right) X \left(A + K_iC_i\right)^T. \end{aligned}$$

Further the limit $\bar{X}$ is the unique positive semi-definite solution of the equation

$$X = \sum_{i=1}^{N} q_i f_{C_i}\left(X\right). \quad (5)$$

Note that if the sensors are chosen independently from one time step to next with sensor i chosen with probability $\pi_i[k]$ at time step k, then $q_i = \pi_i[k]$. Thus this case is a special application of the theorem given above.

Theorem 2. Denote the probability of being in Markov state j at time step k by π_k^j. Then a lower bound for $E\left[P_k\right]$ is Y_k where

$$\begin{aligned} Y_k &= q_{jj}^{k-1}\pi_0^j f_0^k\left(P_0\right) \\ &+ \sum_{i=1}^{k} q_{jj}^{i-1}\left(\pi_{k+1-i}^j - q_{jj}\pi_{k-i}^j\right) f_j^i\left(BR_wB^T\right). \end{aligned}$$

Note that one such lower bound exists for each j. Thus a necessary condition for divergence of the error covariance is that

$$q_{jj}|\lambda_{\max}\left(\bar{A}_j\right)|^2 > 1,$$

where $\lambda_{\max}\left(\bar{A}_j\right)$ is the maximum magnitude among the unobservable eigenvalues of A when $\left(A, C^j\right)$ is put in observer cannonical form.

Again note that the theorem can be easily specialized to the iid case where $\pi_k^j = \pi_j[k]$ defined above.

The algorithm thus consists of choosing the probabilities π_i's (in the case of i.i.d. choice of sensors) or the transition probability matrix (in the case of choice being done according to a Markov chain) to optimize the steady state upper bound as a means of optimizing the expected steady state value of P_k itself. The problem can be solved by a gradient search algorithm or even by brute force search for a reasonable value of N. The lower bound is chiefly used to determine sufficient conditions for the expected error covariance to diverge. After determining the probability values, the sensors are turned on and off with their corresponding probabilities. Note that the implementation assumes some shared randomness among the sensors so that two sensors are not turned on at the same time. This can readily be achieved, e.g., through a common seed for a pseudo-random number generator available to all the sensors. Alternatively a token-passing mechanism to implement the scheme can readily be implemented. Some mechanism for sensor synchronization is also assumed.

4. APPLICATION EXAMPLES AND SIMULATION RESULTS

In this section, the algorithm is applied to a few sample problems and it is shown that the algorithm offers a new, interesting and powerful tool in several problems. Assume a vehicle moving in 2-D space. Denoting the position of the vehicle in the two dimensions by p_x and p_y, and the velocities by v_x and v_y, the state of the system can be modeled by the vector $X = \begin{bmatrix} p_x & p_y & v_x & v_y \end{bmatrix}^T$. With a discretization step size of $h = 0.2$, the dynamics of the vehicle are assumed to be

$$X[k+1] = AX[k] + Bw[k], \qquad (6)$$

where

$$A = \begin{bmatrix} 1 & 0 & h & 0 \\ 0 & 1 & 0 & h \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} h^2/2 & 0 \\ 0 & h^2/2 \\ h & 0 \\ 0 & h \end{bmatrix}.$$

The term $w[k]$ is the perturbation term in acceleration and is modeled as a zero mean white Gaussian noise with covariance matrix Q given by

$$R_w = \begin{bmatrix} 1 & 0.25 \\ 0.25 & 1 \end{bmatrix}.$$

Consider two sensors, each with a sonar-like model (Ramachandra 2000). Measurements taken by the two sensors, y_1 and y_2 can be described by

$$y_i[k] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} X[k] + v_i[k]. \qquad (7)$$

The terms $v_i[k]$ model the measurement noise, again assumed white, zero mean and Gaussian and also independent from each other and from $w[k]$. The sensor noise covariances are

$$R_1 = \begin{bmatrix} 2.4 & 0 \\ 0 & 0.4 \end{bmatrix} \qquad R_2 = \begin{bmatrix} 0.7 & 0 \\ 0 & 1.4 \end{bmatrix}. \qquad (8)$$

On optimizing the upper bound in (5) over π_1 and π_2, the optimal probability for sensor 1 turns out to be $\pi_1 = 0.395$. Indeed, if the optimal sequence is found by a complete tree search, it turns out that in the steady state, the percentage of sensor 1 in the sequence is about 37%. For this probability distribution, the steady state value of the upper bound of the sum of the traces of the expected error covariance matrices for the two sensors turns out to be 2.3884, which compares well with the value of about 2.3 obtained by the optimal strategy obtained by the complete tree search. Note that our algorithm results in orders of magnitude less calculation than tree search algorithms and finds a near-optimal schedule in the steady state.

In addition, there are several unique advantages that our algorithm offers over the conventional algorithms. A very important one is the issue of sensor costs. Frequently, there are other considerations beyond estimation accuracy in using one sensor over another. As an example, it might be more costly to use a very accurate sensor at every time step. Similarly, there might be some sort of fairness requirement such that one sensor is not used all the time and drains all its power. Usually, it is not clear how to appropriately weigh the sensor costs with estimation costs. Thus it is not clear how to even generate a tree for the sensor schedule choices and thus the conventional algorithms do not offer an easy way to take such issues into consideration. However it is easy to take sensor costs into account with our algorithm. As an example, consider three sensors of the form of (7) being present with the measurement noise covariances being given by

$$R_1 = \begin{bmatrix} 3.24 & 0 \\ 0 & 1.04 \end{bmatrix} \qquad R_2 = \begin{bmatrix} 0.25 & 0 \\ 0 & 1.36 \end{bmatrix}$$
$$R_3 = \begin{bmatrix} 0.56 & 0 \\ 0 & 0.56 \end{bmatrix}.$$

For the optimal probability distribution, the sensor 2 should be chosen with a probability of 0.2 and the sensor 3 with a probability of 0.8. However, such a strategy would lead to sensor 3 draining away all its power and thus an additional constraint might be imposed such that on an average, no sensor is used more than twice as much as any other sensor. The search is restricted to the relevant $\pi_1 - \pi_2$ space and the optimal probabilities satisfying the additional constraint

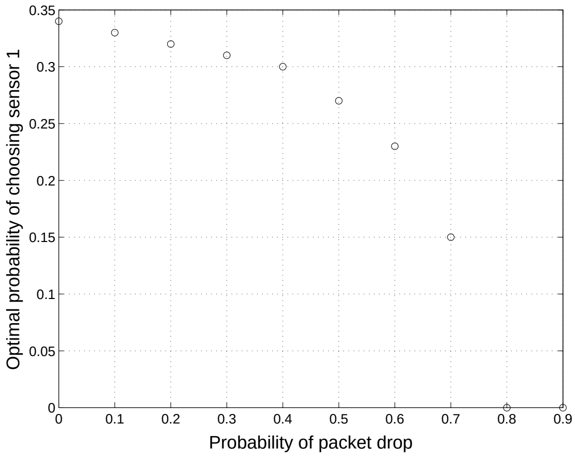

Fig. 1. Optimal probability of use of sensor 1 varies if the channel is dropping packets.

are obtained as the sensor 1 being used with a probability of 0.2 and the sensors 2 and 3 being used each with a probability of 0.4.

Another situation in which our algorithm is much more easily used is when there is some randomness imposed on the system. As an example, consider the case of two sensors with measurement noise covariances given by the values in (8). Suppose that the sensors are communicating over a communication channel that randomly drops packets with probability λ with a base station that fuses the measurements. Compared to the conventional methods, it is easy to take the channel into account while using our algorithm. (5) is set up assuming there are three sensors present. The first two sensors have covariance matrices given above and they are chosen with probabilities $\pi_1(1-\lambda)$ and $\pi_2(1-\lambda)$. The third sensor corresponds to the packet being dropped (and hence no measurement being taken) and it is chosen with a probability of $(\pi_1+\pi_2)\,\lambda$. Then the upper bound is optimized over the parameters π_1 and π_2. Figure 1 shows the change in the optimal probability of choosing sensor 1 as the packet drop probability λ is varied. The plot shows that the packet drop probability indeed plays a role in determining the optimal sensor schedule.

The lower bound derived in Theorem 2 is useful for obtaining the region in the sensor usage probability space when the expected error covariance in (4) diverges. Consider the same example with the second sensor now of the form

$$y[k] = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} X[k] + v[k],$$

with the sensor noise covariances given by (8). Clearly the plant is unobservable while using the second sensor alone and hence as the probability of using the second sensor increases, the error covariance would diverge. It can be shown that although there is a huge gap between the lower and upper bounds, both the bounds diverge at $\pi_1 = 0.56$ which is thus the critical probability for error divergence. This value also matches the value given in Theorem 2 since the largest eigenvalue of the unobservable part of A is 1.5. It may be noted that in general, the probabilities when the bounds diverge will not match and they serve as lower and upper bounds on the critical probability.

To consider a representative example for sensor coverage, consider an area gridded into $N = 2$ points being surveyed by one sensor. Denoting the process at the i-th point by x_i and the state of the entire system by $X = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$, consider the system evolving as

$$X[k+1] = \begin{bmatrix} \alpha & 0 \\ 0.2 & 1.5 \end{bmatrix} X[k] + w[k],$$

where $w[k]$ is white noise with mean zero and covariance equal to the identity matrix. When the sensor is at point i, it can measure the value of x_i corrupted by a Gaussian zero mean noise. Thus, as explained earlier, there are 2 virtual sensors taking measurements according to

$$y_i[k] = x_i[k] + v_i[k],$$

where $v_i[k]$ are all independent of each other and their variances are given by $R_1 = 1$ and $R_2 = 10$. The sensors are modeled as switching according to the transition probability matrix

$$\begin{bmatrix} p & 1-p \\ 1-q & q \end{bmatrix}.$$

Immediately we obtain the conditions that for stability of the observation error covariances, a necessary condition is that $p < 0.44, q < \frac{1}{\alpha^2}$. When the upper bound given in (1) is optimized as a function of α, the optimal values of the parameters are given in figure 2. It can be seen that when the process at the first point is stable, the sensor tends to stay a long time at the second point. As α increases, however, the probability of going to the first point increases. Because of the high measurement noise at the second point, the sensor tends to remain at the second point however.

5. CONCLUSIONS AND FUTURE WORK

An algorithm for stochastically selecting sensors to minimize the expected error covariance was presented. Upper and lower bounds on the error covariance were obtained and their convergence was studied. This algorithm offers many advantages over conventional algorithms for sensor selection. The algorithm was applied to the problems of sensor scheduling and sensor coverage.

The work can potentially be extended in many ways. Finding out how tight the bounds are and

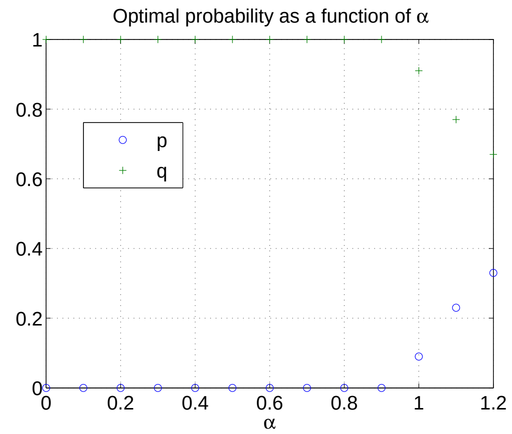

Fig. 2. Optimal probability of use of the sensors.

coming up with tighter bounds is one avenue. Understanding of the spread of the actual value of covariance would also be interesting.

Acknowledgments Research supported in part by the AFOSR grant F49620-01-1-0460 and in part by NSF grant CCR-0326554 for the first author and by the Engineering Research Centers Program of the National Science Foundation under Award Number EEC-9402726 and also a grant from NASA for the second author.

REFERENCES

A. Savkin and R. Evans and E. Skafidas (2000). The Problem of Optimal Robust Sensor Scheduling. In: *Proceedings of the 39th Conference on Decision and Control*. Vol. 1. CDC. Sydney, Australia.

Athans, M. (1972). On the determination of optimal costly measurement strategies. *Automatica* **18**, 397–412.

Gupta, V. and T. H. Chung (2004). A stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage. Technical Report CaltechCDSTR:2004-008. Control and Dynamical Systems, California Institute of Technology. http://resolver.caltech.edu/CaltechCDSTR:2004.008.

Gupta, V., T. Chung, B. Hassibi and R. M. Murray (2004). Sensor scheduling algorithms requiring limited computation. In: *Proc. of the International Conference on Acoustics, Speech and Signal Processing*.

Hall, D. L. and J. Llinas (1997). An introduction to multisensor data fusion. *Proceedings of the IEEE* **85**(1), 6–23.

Herring, K. and J. Melsa (1974). Optimum measurements for estimation. *IEEE Transactions on Automatic Control* **19**(3), 264–266.

III, L. Meier, J. Peschon and R. Dressler (1967). Optimal control of measurement subsystems. *IEEE Transactions on Automatic Control* **12**, 528–536.

Kagami, S. and M. Ishikawa (2004). A sensor selection method considering communication delays. In: *Proc. of the 2004 IEEE Intl. Conf. on Robotics and Automation*. Vol. 1. pp. 206–211.

Kerr, T. H. and Y. Oshman (1995). Further comments on 'optimal sensor selection strategy for discrete-time estimators'[and reply]. *IEEE Transactions on Aerospace and Electronic Systems* **31**, 1159 – 1167.

Krishnamurthy, V. (2002). Algorithms for optimal scheduling and management of hidden markov modelsensors. *IEEE Transactions on Signal Processing* **60**(6), 1382–1397.

Liu, X. and A. J. Goldsmith (n.d.). Kalman filtering with partial observation losses. *IEEE Transactions on Automatic Control*. submitted.

Miller, B. M. and W. J. Runggaldier (1997). Optimization of observations: a stochastic control approach. *SIAM Journal of Control and Optimization* **35**(5), 1030–1052.

Oshman, Y. (1994). Optimal sensor selection strategy for discrete-time state estimators. *IEEE Transactions on Aerospace and Electronic Systems* **30**, 307–314.

Rago, C., P. Willet and Y. Bar-Shalom (1996). Censoring sensor: a low communication rate scheme for distributed detection. *IEEE Transactions on Aerospace and Electronic Systems* **28**(2), 554–568.

Ramachandra, K. (2000). *Kalman filtering techniques for radar tracking*. Marcel Dekker Inc.. New York.

Roumeliotis, S.I. and G.A. Bekey (2002). Distributed multi-robot localization. *IEEE Transactions on Robotics and Automation* **18**(5), 781–795.

Sinopoli, B., L. Schenato, M. Franceschetti, K. Poolla, M. Jordan and S. Sastry (2004). Kalman filtering with intermittent observations. *IEEE Transactions on Automatic Control*.

Tiwari, A., M. Jun, D. E. Jeffcoat and R. M. Murray (2005). The dynamic sensor coverage problem. In: *Proceedings of the 2005 IFAC World Congress (to be presented)*.

Viswanathan, R. and P. K. Varshney (1997). Distributed detection with multiple sensors: Part I - fundamentals. *Proceedings of the IEEE* **85**(1), 54–63.

Zhao, F., J. Shin and J. Reich (2002). Information-driven dynamic sensor collaboration for tracking applications. *IEEE Signal Processing Magazine* **19**(2), 61–72.