SENSOR SCHEDULING ALGORITHMS REQUIRING LIMITED COMPUTATION

Vijay Gupta[*], *Timothy Chung*[†], *Babak Hassibi and Richard M. Murray*

Division of Engineering and Applied Science
California Institute of Technology
{gupta,timothyc,hassibi,murray}@caltech.edu

ABSTRACT

In this paper, we consider the scenario where many sensors co-operate to estimate a process. Only one sensor can take a measurement at any time step. We wish to come up with optimal sensor scheduling algorithms. The problem is motivated by the use of sonar range-finders used by the vehicles on the Caltech Multi-Vehicle Wireless Testbed. We see that this problem involves searching a tree in general and propose and analyze two strategies for pruning the tree to keep the computation limited. The first is a sliding window strategy motivated by the Viterbi algorithm, and the second one uses thresholding. We also study a technique that employs choosing the sensors randomly from a probability distribution which can then be optimized. The performance of the algorithms are illustrated with the help of numerical examples.

1. INTRODUCTION AND MOTIVATION

Sensor networks can significantly improve the estimates of a process (see, e.g., [1] and the references therein). The estimate obtained by fusing measurements from many sensors can be even better than the sensor with the least measurement noise (were no information exchange happening). Thus, many systems have been built utilizing such networks (as an example, see [2]). The assumption usually made in the analysis of such systems is that all the sensors take measurements at the same time. Thus the main issue is multi-sensor data fusion (see, e.g., [3]). Sensor management issues, if any, arise out of concerns like efficient networking and communication protocols [4].

However, in some applications, the use of one sensor places restrictions on the use of other sensors, e.g., simultaneous use of sensors may cause interference in measurements. We face this situation in our own work related to the Caltech Multi-Vehicle Wireless Testbed (MVWT) [5]. When the individual vehicles are using sonar range-finding devices, only one sensor can be active at any time. In such cases the main issue is that of optimally scheduling the sensor measurements so as to minimize the state estimate error covariance. In this paper, we study this problem of coming up with the optimal sensor schedule when only one sensor is allowed to take the measurement at every time step. Some sensor schedule optimization techniques exist in the literature, e.g., [6] examines the problem using stochastic control theory techniques. We pursue two simpler methods, sliding window and thresholding. These methods trade computation/memory requirements for suboptimality; however, they seem to work well on the simulation examples. In addition, we also study a method that involves choosing the sensors randomly according to some optimal probability distribution.

The paper is organized as follows. In the next section, we set up the problem. Then we consider the question of choosing the optimal sensor schedule and present the algorithms mentioned above. We demonstrate these algorithms with the help of examples and end with conclusions and some directions for future work.

2. MODELING AND PROBLEM FORMULATION

Consider the system evolving as follows.

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

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

$$y_i[k] = C_i x[k] + v_i[k]. \quad (2)$$

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. It is assumed that only one sensor can be used at any time. However the measurements are communicated to all the sensors in an error-free manner. It is fairly obvious that since all the nodes have access to

[*]Work supported in part by AFOSR grant F49620-01-1-0460

[†]Work supported in part by the Engineering Research Centers Program of the National Science Foundation under Award Number EEC-9402726 and also a grant from NASA

the same measurements, all the state estimates are the same. Moreover, given a particular sensor schedule, the optimal estimate is obtained by a Kalman filter assuming a time-varying sensor. If we denote the estimate at time step k given measurements till time steps k-1 by $\hat{x}[k]$, we can write

$$\begin{aligned}\hat{x}[k+1] &= A\hat{x}[k] + K_k(y_j[k] - C_j\hat{x}[k]) \\ K[k] &= AP[k]C_j^T(C_jP[k]C_j^T + R_j)^{-1} \\ P[k+1] &= (A - K[k]C_j)P[k](A - K[k]C_j)^T \\ &\quad + BQB^T + K[k]R_jK[k]^T,\end{aligned}$$

where we have assumed that the j-th sensor takes the measurement at time step k. Assuming the initial state $x[0]$ has mean zero and covariance Π_0, the initial condition for above recursions is given by $P[0] = \Pi_0$ and $\hat{x}[0] = 0$.

It is obvious that the minimum error covariance $P[k]$ achievable is a function of the sensor schedule. Thus, a more general problem is that of finding the optimal switching sequence. For simplicity and without loss of generality, we consider only two sensors and define the cost function to be the sum of the traces of the error covariance matrices for the two sensors over the running time of the system.

3. OPTIMIZATION ALGORITHMS

We can represent all the possible sensor schedule choices by a tree structure. The depth of any node in the tree represents time instants with the root at time zero. The branches correspond to choosing a particular sensor to be active at that time instant. Thus, the path from the root to any node at depth d represents a particular sensor schedule choice for time steps 0 to d. We can associate with each node 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. This procedure might place too high a demand on the computational and memory resources of the system. Moreover, in practical applications N might not be fixed a-priori. Hence we need some sort of on-line optimization procedure. We present some approximations which address these difficulties. The first two approximations aim at pruning the tree so as to keep it to a manageable size while trying not to lose the optimal sequence. The third algorithm aims at doing away with traversing the tree altogether although it tries to minimize only the expected steady state error covariance. We now consider these three schemes.

Algorithm 1: Sliding Window: This algorithm is similar to a pseudo real time version of the Viterbi algorithm. We define a window size d where $d < N$. The algorithm proceeds as follows:

1. Start from root node with time $k = 0$.
2. (a) Traverse all the possible paths in the tree for the next d levels from the present node.
   (b) Identify the sensor sequence S_k,S_{k+1},S_{k+2},$\ldots$, S_{k+d-1} that yields the minimum cost at the end of this window of size d.
   (c) Choose the first sensor S_k from the sequence.
3. (a) If $k = N$ then quit, else go to the next step.
   (b) Designate the sensor S_k as the root.
   (c) Update time $k = k + 1$.
   (d) Repeat the traversal step 2.

The window size d is an arbitrary parameter. If it is large enough, the sequence yielding the lowest cost will resemble the optimal sequence for the entire time horizon. Also note that when we slide the window, we already have the error covariances for the first $d - 1$ time steps stored; hence they do not need to be recalculated.

Algorithm 2: Thresholding: This algorithm is similar to that presented in [7], in the context of choosing the optimal controller from a set of many possible choices. We define a cut-off factor $f \geq 1$. The algorithm proceeds as follows:

1. Start from root node with cost 0.
2. (a) Traverse the tree by one step through all possible paths from the current node.
   (b) Calculate the minimum cost till that time step.
   (c) Prune away any branch that yields cost greater than f times the minimum.
   (d) For the remaining branches, denote the cost of the nodes as the cost achieved by moving down the tree till the node.
3. Consider each node in the next time step as the root node and repeat the pruning step 2.
4. After N time steps or a sufficiently large time interval, declare the optimal sequence to be the one yielding the minimum cost till that time step.

The intuition behind the method is that any sequence which yields too high a cost at any intermediate time step would probably not be the one that yields the minimum cost overall. By playing with the factor f, we obtain a trade-off between the certainty that we would not prune away the optimal sequence and the number of branches of the error covariance tree that need to be traversed.

Algorithm 3: Randomly Chosen Sensors: In this algorithm, at each step, the sensors are chosen randomly according to some probability distribution, such that the i-th sensor is chosen with probability q_i. The probability distribution

is then chosen so as to minimize the expected steady state error covariance. Note that we can not calculate the exact value of the error covariance since that will depend on the specific sensor schedule chosen. Hence we optimize the expected steady-state value of the error covariance. Thus we are interested in

$$E\left[P[k+1]\right] = E\left[BQB^T + AP[k]A^T\right] - \Delta,$$

where Δ equals

$$E\left[AP[k]C[k]^T(R[k] + C[k]P[k]C[k]^T)^{-1}C[k]P[k]A^T\right].$$

The quantity $R[k]$ is the sensor noise that depends on the particular sensor chosen at time step k. Explicitly evaluating this expectation appears to be intractable. We can however obtain an upper bound as follows [8]. First note that the quantities $P[k]$ and $C[k]$ are independent. Moreover, since P is positive semi-definite and R is positive definite, $APC^T(R + CPC^T)^{-1}CPA^T$ is convex in P and we can apply Jensen's inequality. Using these facts yields the upper bound

$$E\left[P[k+1]\right] \leq BQB^T + AE\left[P[k]\right]A^T - \sum q_i\Delta_i, \quad (3)$$

where Δ_i equals

$$A\left[E[P[k]]C_i^T(R_i + C_iE[P[k]]C_i^T)^{-1}C_iE[P[k]]\right]A^T.$$

Following [9], we can prove that as long as A is stable, the modified Riccati recursion in equation (3) converges and the expected value of error covariance is the unique positive semi-definite solution of the correspoding modified algebraic Riccati equation. Note that A being stable is the usual case in practical applications of estimation.

The algorithm thus consists of choosing q_i's so as to optimize the upper bound as a means of optimizing the expected steady state value of P_k itself. The optimization problem is solved under the constraint that all q_i's should be non-negative and should sum up to 1. The problem can be solved by the gradient search algorithm or even by brute force search for small N.

4. SIMULATION RESULTS

In this section, we walk through an example demonstrating the use of algorithms developed above. We assume three sensing vehicles trying to locate a non-cooperating target. We model the target vehicle with the standard constant acceleration model [10] in two dimensions. Thus with a discretization step size of h, the dynamics of the vehicle can be modeled as in equation (1) with

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

The sensor model is the usual sonar model [11]. If the sensor is oriented at an angle θ to the global x-axis the vehicle's measurement in the global frame is given by

$$y(k) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} x(k) + \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} v(k).$$

The sensor noise has two components, range noise and the bearing noise. The range noise increases with the distance of the sensor from the target and the bearing noise variance can usually be modeled as a fixed multiple of the range noise variance for a given sensor. For simplicity, the two noises can be assumed independent.

In the numerical example, we consider $h = 0.2$. and the noise covariance matrices equal to

$$Q = \begin{bmatrix} 1 & 0.25 \\ 0.25 & 1 \end{bmatrix} \qquad R_1 = \begin{bmatrix} 3.24 & 0 \\ 0 & 0.04 \end{bmatrix}$$
$$R_2 = \begin{bmatrix} 2.25 & 0 \\ 0 & 0.36 \end{bmatrix} \qquad R_3 = \begin{bmatrix} 2.56 & 0 \\ 0 & 0.16 \end{bmatrix}.$$

The first sensor is placed at position corresponding to $\theta = 0°$, the second sensor at $\theta = 90°$ and the third at $\theta = 45°$. We compare the algorithm performances over a time horizon of 15 steps. The cost function is simply the sum of the trace of the error covariance matrices of the three sensors from time $k = 0$ to time $k = 15$.

Note that the simple strategy of always choosing any one sensor is not optimal. In fact we can easily verify that even a random sensor switching strategy usually reduces the cost. Calculating an optimal sequence by the use of algorithms discussed earlier can lead to significant improments. Figure 1 shows the improvement in the cost due to the predicted (sub)-optimal sensor sequence over using only the sensor 1 as a function of varying window sizes. The improvements over the strategy of using only sensor 2 or sensor 3 are even bigger. It can be seen from the figure that even a window size of $k = 1$ leads to more than 20% improvement in the cost by predicting a good sensor switching strategy. Figure 2 shows the improvement in cost due to the optimal sensor sequence predicted by the thresholding protocol as the cut-off factor f is varied. Again, a large improvement can be obtained by using a fairly small thresholding factor. The optimal probability distribution for the random choice algorithm by optimizing the upper bound in equation (3) is $q_1 = 0.45$ and $q_2 = 0.23$. If we find the optimal sequence by the thresholding algorithm, it turns out that in the steady state, the percentage of sensor 1 and sensor 2 in the sequence is each about 40%. For the probability distribution $q_1 = 0.45$ and $q_2 = 0.23$, the steady state value of the upper bound of the sum of traces of the expected error covariance matrices for the three sensors turns out to be 2.51, which compares well with the value of about 2.2 obtained by the optimal strategy found from the thresholding algorithm.

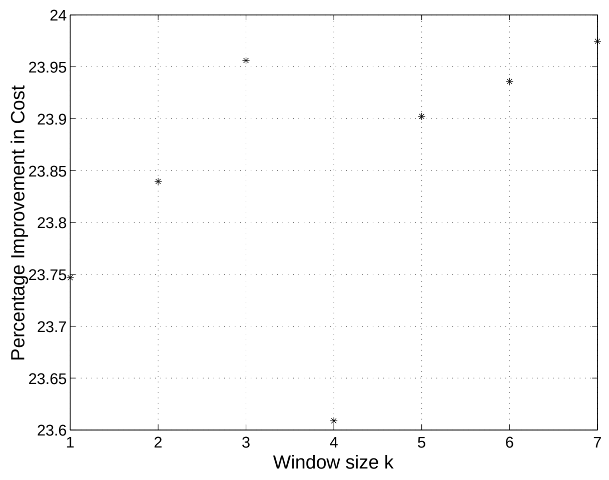

Fig. 1. Percent improvement in cost due to the optimal sensor switching strategy as predicted by the sliding window scheme.

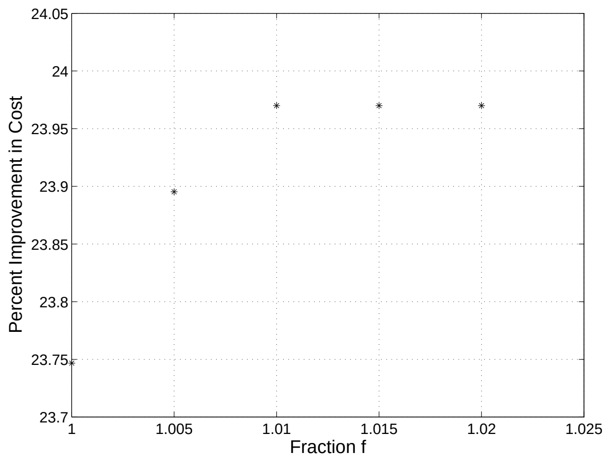

Fig. 2. Percent improvement in cost due to the optimal sensor switching strategy as predicted by the thresholding scheme.

5. CONCLUSIONS AND FUTURE WORK

In this paper, motivated by the use of sonar range-finders on the vehicles on the Caltech MVWT, we investigated the problem of determining an optimal sensor switching strategy when only one sensor is allowed to take a measurement per time step. We saw that this problem involves searching a tree in general and proposed two strategies for pruning the tree to keep the computation limited. We also considered an algorithm which simply chooses sensors at random according to an optimal probability distribution. Some examples demonstrating these algorithms were presented.

The work can potentially be extended in many ways. Obviously there exist better strategies for the case when communication noise is present, although they might entail more amount of data transmitted. Another avenue that can be investigated is the effect of data loss due to fading in the wireless channel.

6. REFERENCES

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

[2] H. Karl, "Making sensor networks useful: Distributed services - the EYES project," ESF Workshop, La Spezia, Italy, 2002.

[3] B.S.Y. Rao, H.F. Durrant-Whyte, and J.A. Sheen, "A fully decentralized multi-sensor system for tracking and surveillance," *Intl. Journal of Robotics Research*, vol. 12, pp. 20–44, 1993.

[4] C. Shen, C. Srisathapornphat, and C. Jaikaeo, "Sensor information networking architecture and applications," *IEEE Personel Communication Magazine*, vol. 8, no. 4, pp. 52–59, August 2001.

[5] L. Cremean, W. Dunbar, D. van Gogh, J. Hickey, E. Klavins, J. Meltzer, and R. M. Murray, "The Caltech multi-vehicle wireless testbed," in *Proc. of the IEEE Conf on Decision and Control*, 2002.

[6] A. Savkin, R. Evans and E. Skafidas, "The Problem of Optimal Robust Sensor Scheduling," in *Proc. of the 39th Conf. on Decision and Control*, Sydney, Australia, December 2000, number 1.

[7] Bo Lincoln and Bo Bernhardsson, "LQR optimization of linear system switching," *IEEE Tran. on Automatic Control*, vol. 47, pp. 1701–1705, Oct. 2002.

[8] V. Gupta, T. Chung, B. Hassibi, and R. M. Murray, "Scheduling for Distributed Sensor Networks," in *Information Processing in Sensor Networks*, Berkeley, CA, 2004, submitted.

[9] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. Jordan, and S. Sastry, "Kalman filtering with intermittent observations," in *Proc. of the IEEE Conf on Decision and Control*, Dec 2003, To Appear.

[10] Y. Bar-Shalom and X. R. Li, *Estimation and Tracking: Principles, Techniques and software*, Artech House, 1993.

[11] K. Ramachandra, *Kalman filtering techniques for radar tracking*, Marcel Dekker Inc., New York, 2000.