THE USE OF PARTICLE FILTERING WITH THE UNSCENTED TRANSFORM TO SCHEDULE SENSORS MULTIPLE STEPS AHEAD

Amit S. Chhetri, Darryl Morrell and Antonia Papandreou-Suppappola

Department of Electrical Engineering, Arizona State University, Tempe, Arizona 85287, USA
amit.chhetri@asu.edu, morrell@asu.edu, papandreou@asu.edu

ABSTRACT

In a multisensor network, sensor scheduling can be used to minimize the cost of resources and improve system performance. In this paper, we propose multisensor scheduling algorithm using a particle filter and the unscented transform for a target tracking application. Under the constraint that only one sensor may be used at each time step, we predict the expected cost multiple steps ahead. We achieve this using several sets of particles for each sequence of sensors and then choose the sequence that minimizes the predicted cost. An advantage of the proposed algorithm is that it can incorporate arbitrary cost functions. Monte Carlo simulations, using squared error as the cost function, demonstrate the improved target tracking performance achieved with sensor scheduling.

1. INTRODUCTION

A critical aspect of a multisensor system is the constraint on system resources such as sensor-lifetime, bandwidth or computational complexity. Sensor scheduling, which is the allocation of sensing resources over time, can minimize the cost of resources and improve the performance of the system under such constraints.

Sensor scheduling is a stochastic control problem that involves optimization of an expected cost function over time. Although this optimization can be performed using dynamic programming [1], in practice computing optimal solutions may be prohibitively expensive, and sub-optimal algorithms are used instead [2–4]. Some cost functions that have been used to schedule the sensors include sensor usage cost, mean squared state estimate error [5], desired estimate covariance [6], and information theoretic costs [2, 3, 7].

In this paper, we propose a scheduling algorithm that computes expected costs one or multiple steps ahead using a particle filter and the unscented transform (UT) [8]. We formulate the algorithm such that different cost functions can be chosen based on the application. The proposed algorithm is an extension of the sensor management approach in [7], in which an information theoretic measure (Kullback-Leibler distance) is used to schedule sensors one step ahead using only a particle filter.

This work was supported by a Raytheon contract through the DARPA Integrated Sensing and Processing program.

We implement the new scheduling algorithm in the context of the following target tracking scenario. A target moves in a 2-dimensional (2-D) plane. An infrared (IR) sensor and a radar sensor are located at the origin, and we assume that we can use only one sensor at a given time k. We schedule the IR and radar sensors to obtain measurements, and track the target based on the measurements using a particle filter. The objective of the scheduling algorithm is to find the sensor sequence that minimizes the expected predicted cost one or multiple steps in the future. The predicted cost is computed using state and observation samples obtained from the particle filter and the UT. Monte Carlo simulations show that the tracking performance improves significantly with sensor scheduling.

2. TARGET TRACKING

We consider a target moving in a 2-D Cartesian coordinate system. The target state at time k is defined as $\mathbf{x}_k = [x_k\ \dot{x}_k\ y_k\ \dot{y}_k]^T$, where x_k and y_k are the target positions and $\dot{x}_k$ and $\dot{y}_k$ are the velocities. We model dynamics with a linear constant-velocity model driven by white Gaussian noise $\mathbf{v}_{k-1}$ [5]

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{v}_{k-1}\,. \quad (1)$$

The radar and IR sensor provide three measurements: range r, range rate $\dot{r}$ and azimuth angle ϕ. The measurements are arranged as a vector $\mathbf{y}_k = [r_k\ \dot{r}_k\ \phi_k]^T$ at time k. They are nonlinearly related to the state as

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{n}_k^{s_k} \quad (2)$$

where s_k denotes the sensor used to obtain measurements at k and may be either $s_k = I$ (for IR sensor) or $s_k = R$ (for radar sensor). The IR sensor provides an accurate measurement of the azimuth angle, while the radar provides accurate measurements of the range and range rate. We model the relative accuracy by using different observation noise covariance matrices for each sensor. The conditional densities $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ and $p(\mathbf{y}_k|\mathbf{x}_k, s_k)$ can be derived from (1) and (2). Note that full details of this model are given in [5].

The target state is estimated by a particle filter [9, 10]. The particle filter is an asymptotically optimal implementation of a recursive Bayesian filter based on samples (particles) and associated importance weights. The N samples and weights at time k are denoted by $\mathbf{x}_k^i$ and w_k^i, where $i = 1, \cdots, N$. A significant advantage of the particle filter is that it can be used for nonlinear systems with non-Gaussian noise. It is used in this work to handle the nonlinearity of the measurement model.

3. SENSOR SCHEDULING

We select the sequence of future sensor uses that minimizes an expected future cost; we make this selection by (approximately) computing the expected future cost for each possible sequence of sensor uses. We use samples of future states and observations to perform these computations. We first investigated Monte Carlo methods to generate all of the samples, but found them to be computationally intractable. Hence, in this paper, we use the UT to generate samples of future states and observations.

In this section we first derive expressions for the expected cost at $k+1$; expressions for $k+m$ are similar. We then describe the method of generating samples and the approximate computation of the expected cost with these samples. The cost at $k+1$ is a function of the true state $\mathbf{x}_{k+1}$, and the history of the selected sensors $s_{1:k+1}$ and observations $\mathbf{y}_{1:k+1}$; the state estimate $\hat{\mathbf{x}}_{k+1|k+1}$ is a function of $s_{1:k+1}$ and $\mathbf{y}_{1:k+1}$. Thus, we denote the cost as $c(\mathbf{x}_{k+1}, \hat{\mathbf{x}}_{k+1|k+1})$. The expected future cost is

$$\begin{aligned} J(s_{k+1}) &= E_{\mathbf{x}_{k+1},\mathbf{y}_{k+1}|\mathbf{y}_{1:k},s_{1:k+1}}\left[c(\mathbf{x}_{k+1}, \hat{\mathbf{x}}_{k+1|k+1})\right] \\ &= \int p(\mathbf{y}_{k+1}|\mathbf{y}_{1:k}, s_{1:k+1}) \int p(\mathbf{x}_{k+1}|\mathbf{y}_{1:k+1}, s_{1:k+1}) \cdot \\ &\qquad c(\mathbf{x}_{k+1}, \hat{\mathbf{x}}_{k+1|k+1}) d\mathbf{x}_{k+1} d\mathbf{y}_{k+1} \end{aligned} \tag{3}$$

Using the model properties gives

$$p(\mathbf{y}_{k+1}|\mathbf{y}_{1:k}, s_{1:k+1}) = \int p(\mathbf{y}_{k+1}|\mathbf{x}_{k+1}, s_{k+1}) \cdot p(\mathbf{x}_{k+1}|\mathbf{y}_{1:k}, s_{1:k}) d\mathbf{x}_{k+1}$$

$$p(\mathbf{x}_{k+1}|\mathbf{y}_{1:k+1}, s_{1:k+1}) = p(\mathbf{y}_{k+1}|\mathbf{x}_{k+1}, s_{k+1}) \cdot \frac{p(\mathbf{x}_{k+1}|\mathbf{y}_{1:k}, s_{1:k})}{p(\mathbf{y}_{k+1}|\mathbf{y}_{1:k}, s_{1:k})}$$

In this work we use the squared error as the cost function.

We approximate the computation of $J(s_{k+1})$ in (3) using several sets of particles. To schedule one step ahead, we generate the sets of particles shown in Figure 1 as follows.

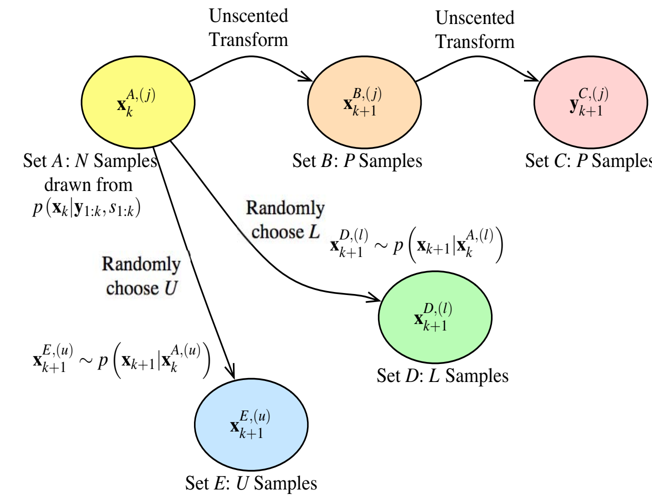

Fig. 1. Sets of particles used to compute the objective function.

1. Let $A_k = \left\{\mathbf{x}_k^{A,(i)}\right\}_{i=1}^N$ be the set of resampled particles at time k.

2. Using the UT [8], deterministically draw P sigma points from A_k and project them to time $k+1$ to form a set of particles $B_{k+1} = \left\{\mathbf{x}_{k+1}^{B,(j)}\right\}_{j=1}^P$ and two sets of measurements $C_{k+1} = \left\{\mathbf{y}_{k+1}^{C,(j)}\right\}_{j=1}^P$ (one set for $s_{k+1} = R$ and another for $s_{k+1} = I$). An advantage of using the UT is that the projected sigma points approximate the statistics of the future states and measurements very closely up to second order.

3. Randomly choose (with replacement) L particles from A_k; predict each of these particles to $k+1$ by sampling from the distribution: $\mathbf{x}_{k+1}^{D,(l)} \sim p(\mathbf{x}_{k+1}|\mathbf{x}_k^{A,(l)})$. This gives us the set $D_{k+1} = \left\{\mathbf{x}_{k+1}^{D,(l)}\right\}_{l=1}^L$. The set $E_{k+1} = \left\{\mathbf{x}_{k+1}^{E,(u)}\right\}_{u=1}^U$ is obtained similarly.

These sets of particles are then used to compute the approximate expected future cost $\hat{J}(s_{k+1})$ using steps (i) through (v) shown in Table 1 with $M = 1$.

When scheduling M steps ahead, the number of possible sensor sequences is 2^M. We must compute the total expected future cost for each possible sequence of sensor uses. In the following, we describe the process for a given sequence $s_{k+1:k+m}$ with the understanding that it is applied for each possible sequence.

To predict two steps ahead (given that $B_{k+1}, C_{k+1}, D_{k+1}$ and E_{k+1} are computed), project the particles in A_k to time $k+1$ using the system model in (1), compute the mean of the measurement particles as $\bar{\mathbf{y}}_{k+1}^C = \sum_{j=0}^P \mathcal{W}^{(j)} \mathbf{y}_{k+1}^{C,(j)}$

(where $\mathcal{W}^{(j)}$ are obtained as a part of UT in step 2 above), and assign to each projected particle a weight using $\bar{\mathbf{y}}_{k+1}^{C}$ as the observation. These weighted particles are then resampled to form the set A_{k+1}. Steps 2 and 3 are performed at time $k+2$ to obtain $B_{k+2}, C_{k+2}, D_{k+2}$ and E_{k+2}. This procedure can be iterated up to time $k+M$ to obtain the five sets of particles at each time step. With these sets, we can predict the cost at each stage using steps (i) through (v) of the algorithm in Table 1. The total cost for a sequence of sensors is calculated by summing up the cost at each stage. We then choose the sequence of sensor uses that gives the minimum total cost. The proposed multiple step sensor scheduling algorithm is summarized in Table 1.

Once the optimal sequence of sensors is selected, measurements are obtained and the target state estimate is computed using the particle filter.

For each possible sequence of sensors $s_{k+1:k+M}$

- $C_k = \mathbf{\Phi}$ (empty set) and
  $J^{(j)}(s_{k+m}) = E_{\mathbf{x}_{k+m}|\mathbf{y}_{k+m}^{C,(j)}}\left[c\left(\mathbf{x}_{k+m}, \hat{\mathbf{x}}_{k+m|k+m}\right)\right]$
- For m=1 to M,
  - Obtain sets $A_{k+m}, B_{k+m}, C_{k+m}, D_{k+m}$ and E_{k+m} from A_{k+m-1} and C_{k+m-1}
  - Compute the conditional objective function $J^{(j)}(s_{k+m})$ using steps (i) through (v)

(i) Compute $w_{k+m}^{(j,l)}$ using the particles in D_{k+m}

$$w_{k+m}^{(j,l)} \propto p(\mathbf{y}_{k+m}^{C,(j)}|\mathbf{x}_{k+m}^{D,(l)}, s_{k+m})$$

(ii) Compute the state estimate using $\mathbf{x}_{k+m}^{D,(l)}$

$$\hat{\mathbf{x}}_{k+m|k+m}^{(j)} = \sum_{l=1}^{L} w_{k+m}^{(j,l)} \mathbf{x}_{k+m}^{D,(l)}$$

(iii) Compute the approximate conditional objective function $\hat{J}^{(j)}(s_{k+m})$ using the particles in D_{k+m}

$$\hat{J}^{(j)}(s_{k+m}) = \sum_{l=1}^{L} w_{k+m}^{(j,l)} c(\mathbf{x}_{k+m}^{D,(l)}, \hat{\mathbf{x}}_{k+m|k+m}^{(j)})$$

(iv) Compute the approximate conditional density of $\mathbf{y}_{k+m}^{C,(j)}$ using the particles in E_{k+m}

$$\hat{p}(\mathbf{y}_{k+m}^{C,(j)}|\mathbf{y}_{1:k}, s_{1:k-1}) = \sum_{u=1}^{U} p(\mathbf{y}_{k+m}^{C,(j)}|\mathbf{x}_{k+m}^{E,(u)})$$

(v) Compute the approximate objective function

$$\hat{J}(s_{k+m}) = \frac{\sum_{j=1}^{P} \hat{p}(\mathbf{y}_{k+m}^{C,(j)}|\mathbf{y}_{1:k}, s_{1:k-1}) \hat{J}^{(j)}(s_{k+m})}{\sum_{j'=1}^{P} \hat{p}(\mathbf{y}_{k+m}^{C,(j')}|\mathbf{y}_{1:k}, s_{1:k-1})}$$

- Calculate the approximate total cost for $s_{k+1:k+m}$

$$\hat{\mathbf{J}}_{s_{k+1:k+M}} = \sum_{m=1}^{M} \hat{J}(s_{k+m})$$

End

Choose the optimal sequence of sensors as

$$s_{k+1:k+M}^{opt} = \arg\min_{s_{k+1:k+M}} \left\{\hat{\mathbf{J}}_{s_{k+1:k+M}}\right\}$$

Table 1. Multiple step sensor scheduling algorithm.

4. SIMULATIONS AND RESULTS

We simulate a target trajectory in a 2-D plane which starts at (x, y)=(8000, 3000) m and ends at $(-2442, 2812)$ m. The initial velocities of the target in the x and y directions are -300 m/s and -50 m/s, respectively. The sensors are fixed at the origin, and the target moves for 35 s with each step corresponding to 1 s. The measurement error covariance matrices for the IR and radar sensors are the same as in [5]. For the particle filter algorithm, we used $N = 800$ particles, and a total of 200 Monte Carlo simulations were run.

The values for L, U and P were chosen to be 700,700 and 23 respectively. We compare the tracking results of one and two step sensor scheduling (which we denote as $M^{PF} = 1$ and $M^{PF} = 2$, respectively) with the case of no-scheduling (NS). We also compare these results with the one step extended Kalman filter (EKF) assisted scheduling algorithm that we proposed in [5] (which we denote as $M^{EKF} = 1$). For the NS case, we use only the radar sensor. A comparison of variances (in dB) of the position and velocity estimates for the $M^{PF} = 1, M^{EKF} = 1, M^{PF} = 2$ and NS cases can be seen in Figure 2. We observe that for the scheduling cases, but not for the NS case, the variances decrease steadily with time. This is more evident for the variance in x (top plot in Figure 2) since the NS radar case does not provide an accurate azimuth angle measurement. Figure 3 illustrates a comparison of the mean square error (MSE) (in dB) for different scheduling cases including the three step sensor scheduling denoted as $M^{PF} = 3$. It can be seen that the MSE decreases with time for the scheduling cases while it levels out for the NS case. At time $k = 35$, for example, the difference in MSE between the NS and scheduling cases is about 14 dB. We should note that the results for the one, two and three step scheduling are similar which complies with the results presented in [5]. This suggests that for the current dynamic model, looking ahead just one step will suffice. Figure 4 compares the tracked trajectory for the various scheduling cases. It can be seen that the tracking performance for the scheduling cases is much better than the NS case. Similar results were obtained when an IR sensor was used instead of the radar sensor for the NS case. We thus conclude that by scheduling the sensors we obtain improved tracking results with low sensor usage costs. It should also be noted that by using the new algorithm, we can obtain as good results as with the algorithm in [5]. The added advantage here is that the proposed algorithm can be applied to different cost functions.

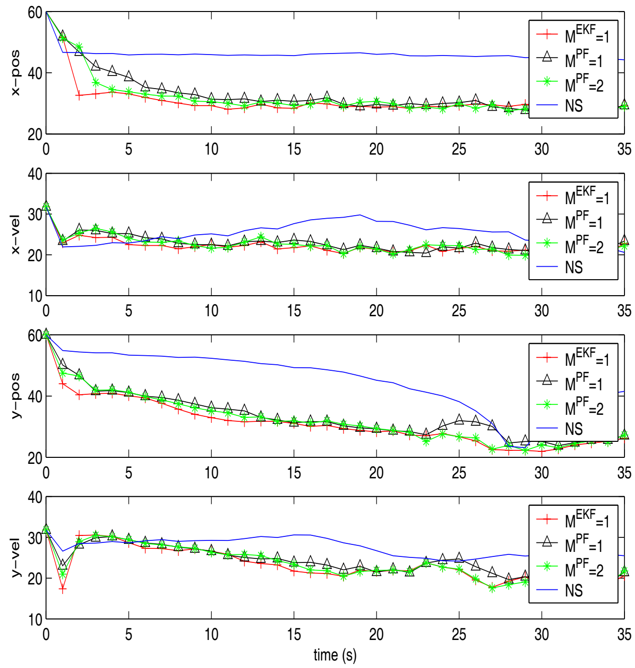

Fig. 2. Comparison of the variances for the NS case and the $M^{PF} = 1, 2$ and $M^{EKF} = 1$ scheduling cases.

5. CONCLUSIONS

We have developed a sensor scheduling algorithm for target tracking using a particle filter and the unscented transform. We schedule the sensors by predicting the expected cost multiple steps ahead and minimizing the expected cost obtained for all sensor sequences. Monte Carlo simulations of our algorithm reveal that the tracking performance using sensor scheduling is superior to the no-scheduling case. We also observe that the tracking performance with the new algorithm is compatible to the one we proposed in [5]. Notice, however, that although the squared error cost function was used here, the algorithm extends without modifications to general cost functions as shown in [11].

6. REFERENCES

[1] D.A. Castanon, "Approximate dynamic programming for sensor management," in *Proceedings of the 36th IEEE Conf. on Decision and Control*, 1997, vol. 2, pp. 1202–1207.

[2] F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration for tracking applications," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 61–72, March 2002.

[3] C. Kreucher, K. Kastella, and A. Hero, "A Bayesian method for integrated multitarget tracking and sensor management," in *Proc. of 6th Int. Conf. on Information Fusion*, 2003.

[4] A. Logothetis and A. Isaksson, "On sensor scheduling via information theoretic criteria," in *Proceedings of the 1999 American Control Conference*, 1999, vol. 4, pp. 2402–2406.

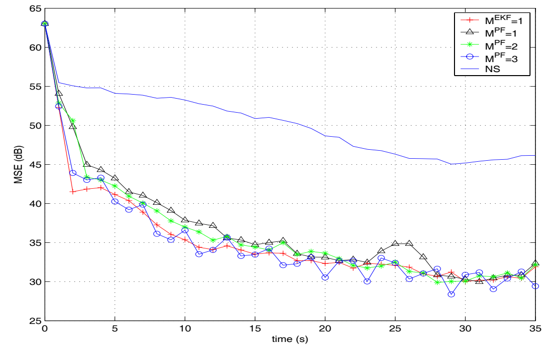

Fig. 3. Comparison of the MSE for the NS and the $M^{PF} = 1, 2, 3$ and $M^{EKF} = 1$ scheduling cases.

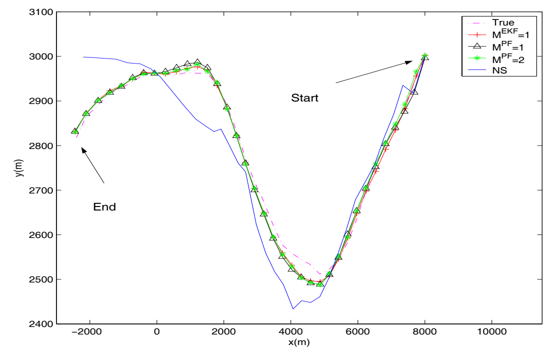

Fig. 4. Tracked target for the NS and the $M^{PF} = 1, 2$ and $M^{EKF} = 1$ scheduling cases.

[5] A. Chhetri, D. Morrell, and A. Papandreou-Suppappola, "Scheduling multiple sensors using particle filters in target tracking," in *Proceedings IEEE Statistical and Signal Processing Workshop*, September 2003, pp. 529–532.

[6] M. Kalandros and L.Y. Pao, "Covariance control for multi-sensor systems," *IEEE. Trans. on Aerospace and Electronic Systems*, vol. 38, pp. 1138–1157, October 2002.

[7] A. Doucet, B.N. Vo, C. Andrieu, and M. Davy, "Particle filtering for multi-target tracking and sensor management," in *Int. Conf. on Information Fusion*, 2002, vol. 1, pp. 474–481.

[8] S.J. Julier and J.K. Uhlmann, "Reduced sigma point filters for the propagation of means and covariances through non-linear transformations," in *Proceedings of the 2002 American Control Conference*, 2002, vol. 2, pp. 887–892.

[9] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. on Signal Processing*, vol. 50, pp. 174–188, February 2002.

[10] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*, Springer-Verlag, 2001.

[11] A. Chhetri, D. Morrell, and A. Papandreou-Suppappola, "Non-myopic cost-based scheduling of multiple sensors for sequential monte-carlo target tracking algorithms," submitted to *IEEE Trans. Signal Proc.*, Jan 2004.