A GENERALIZED PARTICLE FILTERING BASED SENSOR SCHEDULING ALGORITHM FOR TARGET TRACKING

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

1. INTRODUCTION

One of the critical components of a multisensor system is the constraint on system resources such as sensor-lifetime, bandwidth, computation etc. Sensor scheduling, which is the allocation of sensing resources over time, can serve to 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 expected cost function over time. In principle, solutions to this problem can be computed using dynamic programming [1, 2], however in practice computing optimal solutions may be prohibitively expensive, and sub-optimal algorithms are used instead [3–5]. Cost functions that can be used to schedule the sensors include sensor usage cost, mean squared state estimate error [6], desired estimate covariance [7], information theoretic costs [3, 4, 8] etc.

In this paper, we propose a generalized scheduling algorithm by computing expected costs one or multiple steps ahead using a particle filter and the unscented transform (UT). The proposed algorithm is an extension of the sensor management approach in [8], in which an information theoretic measure (Kullbach-Leibler distance) is used to schedule sensors. It is also an improvement over the one in [6], as it (unlike the latter) can be applied to different cost functions including the information theoretic cost.

For this paper, we implement the scheduling algorithm in the context of the following target tracking scenario. A target moves in a 2-dimensional plane whose kinematics are unknown to us. At the origin are located an infrared (IR) sensor and a radar sensor. We assume that we can use only one sensor at any given time k. We schedule the infrared and radar sensors to obtain measurements, and track the target based on the measurements using a particle filter. The sensors are scheduled by predicting multiple steps ahead the expected state, measurements (for both radar and IR sensors) and cost function using the particle filter and the UT. The objective of the scheduling algorithm is to find the sensor sequence that minimizes the predicted cost at those steps. Monte Carlo simulations show that the tracking performance improves significantly with sensor scheduling.

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

2. PROBLEM FORMULATION

We consider a target moving in a 2-dimensional Cartesian space. The target state at time k is defined as

$$\mathbf{x}_k = \left[\begin{array}{cccc} x_k & \dot{x}_k & y_k & \dot{y}_k \end{array} \right]^T$$

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 [6]

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

The radar and IR sensor provide three measurements: range r, range rate $\dot{r}$ and azimuth angle ϕ. While the IR sensor provides a good measurement of the azimuth angle, the radar sensor provides a good measurement of the range and range rate of the target. The accuracy of the measurements depends on the sensor type; we model the relative accuracy by using different observation noise covariance matrices for each sensor. The measurements are arranged as a vector $\mathbf{y}_k$ at time k

$$\mathbf{y}_k = \left[\begin{array}{ccc} r_k & \dot{r}_k & \phi_k \end{array} \right]^T$$

The measurements are nonlinearly related to the state as

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

The conditional density of $\mathbf{y}_k$ given $\mathbf{x}_k$ and s_k is denoted by $p(\mathbf{y}_k|\mathbf{x}_k, s_k)$. The details of this model are given in [6].

Tracking is the problem of inferring the motion of an object from observations. Given the conditional densities $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ and $p(\mathbf{y}_k|\mathbf{x}_k, s_k)$, the state can be estimated recursively using 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 samples and weights at time k are denoted by $\mathbf{x}_k^i$ and w_k^i respectively, where $i = 1, \cdots, N$ (N being the number of particles). 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 as well as to allow for non-Gaussian noise.

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. We then describe the method of generating samples and the approximate computation of the expected cost using these samples. We denote the cost as $c(\mathbf{x}_{k+1}, \hat{\mathbf{x}}_{k+1|k+1})$, where $\hat{\mathbf{x}}_{k+1|k+1}$ is a function of s_{k+1} and $\mathbf{y}_{k+1}$. In the notation $k+1|k+1$, the first and second $k+1$ denote the current time and the time up to which the measurements have been received. The overall goal of the sensor selection is to minimize the expected future cost given as

$$J(s_{k+1}) = E_{\mathbf{x}_{k+1}, \mathbf{y}_{k+1}|\mathbf{y}_{1:k}, s_{1:k}} \left[c(\mathbf{x}_{k+1}, \hat{\mathbf{x}}_{k+1|k+1}) \right] \quad (2)$$

We can express the above cost in terms of the available densities as

$$J(s_{k+1}) = \int p(\mathbf{y}_{k+1}|\mathbf{y}_{1:k}, s_{1:k+1}) \int p(\mathbf{x}_{k+1}|\mathbf{y}_{1:k+1}, s_{1:k+1}) c(\mathbf{x}_{k+1}, \hat{\mathbf{x}}_{k+1|k+1}) d\mathbf{x}_{k+1} d\mathbf{y}_{k+1} \quad (3)$$

Using the model properties, we have the following expressions for the densities in (3):

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

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

In this work we use the predicted mean squared error as the cost function. We denote the squared error at time $k+1$ as

$$c(s_{k+1}, \mathbf{x}_{k+1}, \mathbf{y}_{k+1}) = \left(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1|k+1}\right)^T \left(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1|k+1}\right) \quad (4)$$

We approximate the computation of $J(s_{k+1})$ in (3) using particles. In the following, we consider a case where the compuation of the cost $c(\mathbf{x}_{k+1}, \hat{\mathbf{x}}_{k+1|k+1})$ requires an estimate of the state $\hat{\mathbf{x}}_{k+1|k+1}$. We use several sets of particles as shown in Figure 1. For a one step sensor scheduling, we proceed as follows to generate the sets of particles.

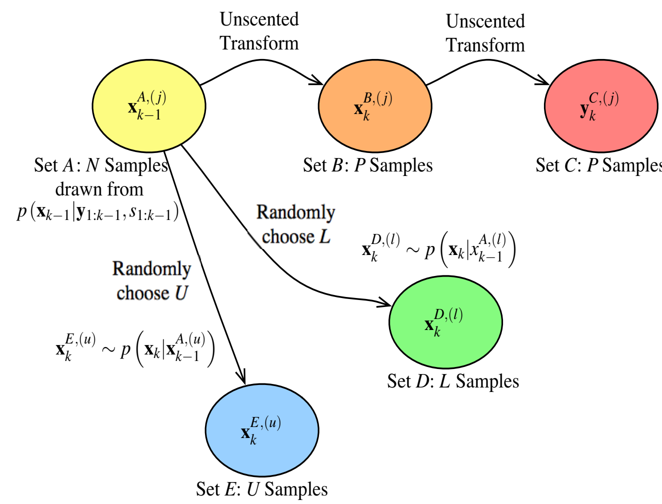

Fig. 1. Sets of particles used to compute the objective function.

1. The resampled particles at time k are assumed to form a set $A_k = \left\{\mathbf{x}_k^{A,(i)}\right\}_{i=1}^N$.
2. Using the UT [11], draw deterministically P sigma points from set A 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 $C_{k+1}^{s_{k+1}} = \left\{\mathbf{y}_{k+1}^{C,(j)}\right\}_{j=1}^P$ (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 upto second order. The sigma points being small in number also help in reducing the computational load of the scheduling algorithm.
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 $D_{k+1} = \left\{\mathbf{x}_{k+1}^{D,(l)}\right\}_{l=1}^L$.
4. Randomly choose (with replacement) U particles from A_k; predict each of these particles to $k+1$ by sampling from the distribution: $\mathbf{x}_{k+1}^{E,(u)} \sim p(\mathbf{x}_{k+1}|\mathbf{x}_k^{A,(u)})$. This gives us the set $E_{k+1} = \left\{\mathbf{x}_{k+1}^{E,(u)}\right\}_{u=1}^U$.

For M step sensor scheduling, the total number of possible sensor sequences is 2^M. To obtain $A_{k+1}^{s_{k+1}}$ from A_k, we proceed as follows. Project the particles in A_k to time $k+1$ using the system model in (1). Compuite the mean of the

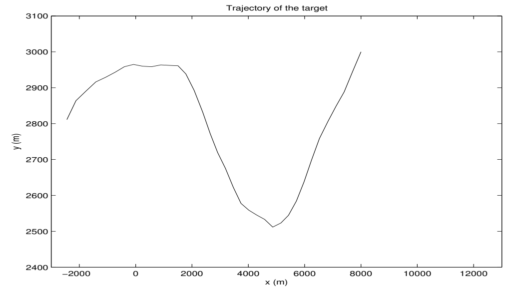

Fig. 2. Simulated trajectory of the target.

measurements in set $C_{k+1}^{s_{k+1}}$ 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 step 2). Assign each projected particle a weight using $\bar{\mathbf{y}}_{k+1}^{C}$ as the observation. We resample these weighted particles to form the set $A_{k+1}^{s_{k+1}}$. The steps 2,3 and 4 (from above) are repeated at time $k+2$ to obtain $B_{k+2}^{s_{k+1}}, C_{k+2}^{s_{k+1:k+2}}, D_{k+2}^{s_{k+1}}$ and $E_{k+2}^{s_{k+1}}$. Note that there will be two sets each for $A_{k+1}^{s_{k+1}}, B_{k+2}^{s_{k+1}}, D_{k+2}^{s_{k+1}}$ and $E_{k+2}^{s_{k+1}}$ (corresponding to the two values of s_{k+1}). Also, since $C_{k+2}^{s_{k+1:k+2}}$ depends on the choices of sensors at times $k+1$ and $k+2$, we will have four sets for it. 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 squared error cost at each stage using steps (i) through (v) of the algorithm in Table 1. The total cost for a sequence of sensor is calculated by summing up the squared error cost at each stage. We then choose the sequence of sensor uses that gives the minimum total squared error. The proposed multiple step sensor scheduling algorithm is summarized in Table 1. For the ease of notation, we use A_{k+m}, B_{k+m}, C_{k+m}, D_{k+m} and E_{k+m} (where $m = 1, \cdots, M$) in place of $A_{k+m}^{s_{k+1:k+m}}, B_{k+m}^{s_{k+1:k+m}}, C_{k+m}^{s_{k+1:k+m+1}}, D_{k+m}^{s_{k+1:k+m}}$, and $E_{k+m}^{s_{k+1:k+m}}$ respectively. Similarly, we use $J^{(j)}(s_{k+m})$, $w_{k+m}^{(jl)}$ and $\hat{J}(s_{k+m})$ in place of $J^{(j)s_{k+1:k+m}}(s_{k+m})$, $w_{k+m}^{(jl)s_{k+1:k+m}}$ and $\hat{J}^{s_{k+1:k+m}}(s_{k+m})$ respectively.

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 $s_{k+1:k+M}$

- Define $A_k^{s_{k+1:k}} = A_k$, $C_k^{s_{k+1:k+1}} = \mathbf{\Phi}$ (empty set) and $J^{(j)}(s_{k+m}) = E_{\mathbf{x}_{k+m}|\mathbf{y}_k = \mathbf{y}_{k+m}^{C,(j)}} \left[c\left(\mathbf{x}_{k+m}, \hat{\mathbf{x}}_{k+m|k+m}\right)\right]$
- 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 funtion $J^{(j)}(s_{k+m})$ using steps (i) through (v)

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

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

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

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

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

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

  (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 funtion

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

  - 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-dimensional plane as shown in Figure 2. The target starts at (x, y)=(8000, 3000) m and ends at $(-2442, 2812)$ m. The initial velocity of the target in the x and y directions is -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 1s. The measurement error covariance matrices for the IR and radar sensors were chosen as [6]

$$\mathbf{R}^I = \begin{bmatrix} 2.5 \times 10^5 & 0 & 0 \\ 0 & 10^4 & 0 \\ 0 & 0 & 7.61 \times 10^{-5} \end{bmatrix}$$

$$\mathbf{R}^R = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 0.0247 \end{bmatrix}$$

For the particle filter algorithm we chose $N = 800$ particles, and a total of 200 Monte Carlo simulations were run. The initial particles are generated as samples from a Gaussian distribution whose mean is the true location and whose

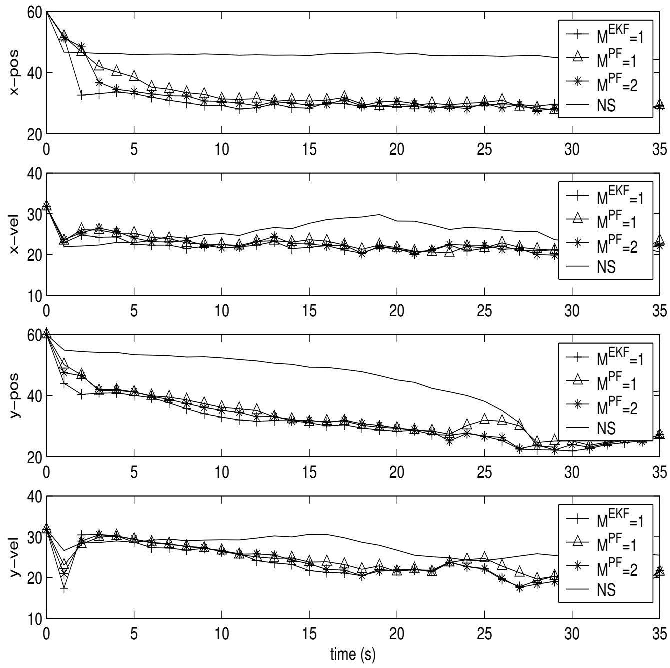

Fig. 3. Comparison of the variances for the NS case and the $M^{PF} = 1$ $M^{PF} = 2$, $M^{EKF} = 1$ scheduling cases.

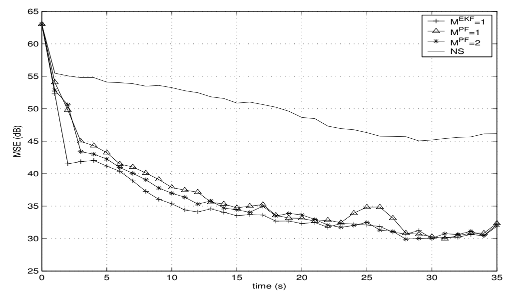

Fig. 4. Comparison of the MSE for the NS case and the $M = 1, 2$, and 3 step scheduling cases.

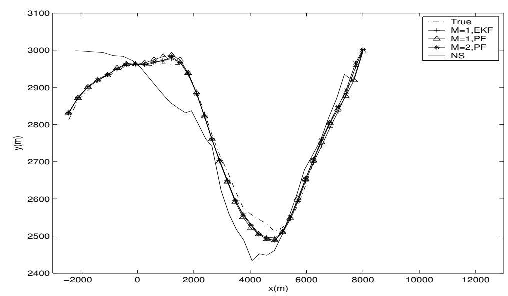

Fig. 5. Comparison of the tracked target for the NS case and the $M = 1, 2$, and 3 step scheduling cases.

covariance matrix is

$$\mathbf{P}_0 = \begin{bmatrix} 10^6 & 0 & 0 & 0 \\ 0 & 1500 & 0 & 0 \\ 0 & 0 & 10^6 & 0 \\ 0 & 0 & 0 & 1500 \end{bmatrix}$$

The values for L, U and P were chosen to be 700,700 and 23 respectively. We compare the tracking results of 1 and 2 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 EKF assisted scheduling algorithm in [6] (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 3. We observe that for the scheduling cases, the variances decrease steadily with time which does not hold true for the NS case. This is more evident for the variance in x (top plot in Figure 3) since the NS radar case does not provide an accurate azimuth angle measurement. Figure 4 illustrates a comparison of the MSE for different scheduling cases. It can be seen that the MSE decreases with time for the scheduling cases while it levels out for the NS case. At time $k = 35$, the difference in MSE between the NS and scheduling cases is about 14 dB. Figure 5 compares the tracked trajectory for the various scheduling cases. It is 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 get improved tracking results with low sensor usage costs. It should also be noted that by using the proposed algorithm, we can get as good results as with the algorithm in [6]. The added advantage here is that the proposed algorithm can be applied to different cost functions.

5. CONCLUSIONS

We have developed a sensor scheduling algorithm for target tracking using a particle filtering approach. Specifically, we used the particle filter and UT to predict squared error cost multiple steps ahead and minimize the predicted cost obtained from all sensor sequences. Monte Carlo simulations of our algorithm reveal that the tracking performance using sensor scheduling is superior to the no-scheduling case. Note that the use of the MSE as a cost function does not take into account the non-diagonal elements of the error covariance matrix. Thus, we plan to further improve our scheduling performance using the difference between predicted and desired error covariance matrices as suggested in [7] for non-particle filter applications.

6. REFERENCES

[1] V. Krishnamurthy, “Algorithms for optimal scheduling and management of hidden Markov model sensors,”

IEEE Transactions on Signal Processing, vol. 50, pp. 1382–1397, June 2002.

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

[3] 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.

[4] C. Kreucher, K. Kastella, and A. Hero, "A Bayesian method for integrated multitarget tracking and sensor management," in *Proceedings of 6th International Conference on Information Fusion*, 2003.

[5] 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.

[6] A. Chhetri, D. Morrell, and A. Papandreou-Suppappola, "Scheduling multiple sensors using particle filtersin target tracking," in *Proceedings IEEE Statistical and Signal Processing Workshop*, September 2003.

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

[8] A. Doucet, B.N. Vo, C. Andrieu, and M. Davy, "Particle filtering for multi-target tracking and sensor management," in *Proceedings of the Fifth International Conference on Information Fusion*, 2002, vol. 1, pp. 474–481.

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