SENSOR SCHEDULING AND TARGET TRACKING USING EXPECTATION PROPAGATION

Travis J. Hestilow, Tao Wei, Yufei Huang

Department of Electrical and Computer Engineering
The University of Texas at San Antonio
San Antonio, TX 78249 USA
thestilo@lonestar.utsa.edu, twei@lonestar.utsa.edu, yhuang@utsa.edu

ABSTRACT

Multiple-sensor scheduling for target tracking applications using expectation propagation (EP) is examined. The method is an alternative to that of [1] wherein an Extended Kalman Filter (EKF) was used to predict the next state for sensor scheduling purposes, and a sequential Monte Carlo particle filter (PF) method was used to implement the target tracking. In this application, EP is used instead of PF to estimate the unobserved state variable. Initial simulations show the EKF+EP (with scheduling) algorithm performs at least as well as EKF+PF, with a shorter run time and less programmatic complexity. EKF+EP (with scheduling) also performs better than EKF+EP (without scheduling).

1. INTRODUCTION

Over the last few years, a frequently examined method for producing accurate state tracking of dynamic systems has been particle filtering (PF) [2]. In particle filtering, random samples from a known distribution are produced and used, in conjunction with the tracking state equations and the observed data, to estimate the position and velocity of the tracked target. When coupled to a predictive method such as the Extended Kalman Filter (EKF), this not only allows the tracking of a target, but the efficient use of sensor assets to accomplish the tracking. Reference [1] explored the improvement in the error covariance matrix when individual sensors were scheduled subject to the constraint that the trace of the predicted error covariance matrix was chosen to be minimum over a set of sensors.

We consider here target tracking in sensor networks, where different types of sensors are available for use. These sensors may be able to measure a particular parameter very accurately (range, range rate, angle); however, no one sensor can measure everything well. Thus, an effort is made to combine sensor measurements in such a way as to increase the accuracy of the target position estimate.

Unfortunately, the problem of fusing multiple sensors very quickly becomes intractable. Tracking even a single target requires estimation of position very rapidly. This puts stringent requirements on the speed of not only the sensor package but the computational engine used to fuse the measurements into a single estimate. The communication and control links between the sensor package and the computational engine must also be able to handle a high data rate.

Despite its the flexibility and good performance in target tracking [3], PF is computationally very intensive and requires excessive memory to store the particles and weights. The generation, tracking, and update of several hundred trajectories requires extremely efficient, highly-complex coding. Recently, a computationally more efficient alternative known as Expectation Propagation (EP) was proposed in [4]. When coupled to an EKF, the method is also known as the iterative EKF smoother [5]. EP has a very straightforward implementation, which results in much less complex coding, faster execution, and less stringent requirements on storage. In addition to its computational advantage, EP has shown to provide performance comparable or closer to that of PF [4, 5, 6, 7], making it a potentially favorable choice for many of the tracking applications. Motivated by the advantages of EP, we investigate in this paper an EP solution to target tracking and sensor scheduling.

This paper is organized as follows. A brief description of the tracking problem will be in Section 2. A background discussion of EP follows in Section 3. The application of EP to the problem is presented in Section 4. The results of our study of this problem are given in Section 5. Finally, Section 6 gives a brief summary and some conclusions.

2. PROBLEM FORMULATION

The scheduling and tracking problem is formulated in a similar fashion to that in [1]. A single target is moving through two-dimensional Cartesian space, tracked by a sensor suite located at the origin. We let $\mathbf{x}_k$ be a real-valued random vector representing the target state at time index k. We then define $\mathbf{z}_k$ as the observation vector at time k, also a real-valued random variable. The state consists of the position and velocity in the Cartesian frame while the observation vector consists of the range, range rate, and bearing to the target:

$$\mathbf{x}_k = \begin{bmatrix} x_k & \dot{x}_k & y_k & \dot{y}_k \end{bmatrix}^{\mathbf{T}} \quad (1)$$

$$\mathbf{z}_k = \begin{bmatrix} r_k & \dot{r}_k & \phi_k \end{bmatrix}^{\mathbf{T}} \quad (2)$$

The dynamic state system is a linear model with an additive Gaussian noise driver. The measurement is a nonlinear model, but again with additive Gaussian noise

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{w}_{k-1} \quad (3)$$

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{n}_k \quad (4)$$

In this model, $\mathbf{F}$ is the state transition matrix and models the target state kinematics, while $\mathbf{h}$ is the non-linear relationship between the observation and state vectors.

$$\mathbf{F} = \begin{bmatrix} 1 & \Delta\mathrm{t} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta\mathrm{t} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

$$\mathbf{h}(\mathbf{x}_k) = \begin{bmatrix} \sqrt{x_k^2 + y_k^2} & \frac{x_k x_k' + y_k y_k'}{\sqrt{x_k^2+y_k^2}} & \tan^{-1}\frac{y_k}{x_k} \end{bmatrix}^{\mathbf{T}} \quad (6)$$

The vectors $\mathbf{w}$ and $\mathbf{n}$ represent the process and measurement noise drivers, respectively. Both are modeled as zero-mean Gaussian noise and are uncorrelated with each other.

Rather than merge or "fuse" the data from several sensors, a single sensor is chosen to conduct each measurement. This is accomplished by estimating the covariance likely to be present after a measurement by a particular sensor. Since the error covariance matrices for each sensor are constant, this estimation is carried out by exhaustively updating the total state estimation covariance using the current estimation of the state error covariance and the (constant) error covariance of each sensor. The sensor which results in the smallest error covariance (measured by the trace of the predicted error covariance matrix) is chosen for the the next measurement.

To summarize, the problem consists of scheduling multiple sensors so as to minimize the estimated error covariance, estimating the unobserved state variable $\mathbf{x}$ through propagation of prior estimates through the state equations, estimating the posterior distribution $p(\mathbf{x}_k|\mathbf{z}_k)$, and iteratively updating and smoothing prior posterior distributions $p(\mathbf{x}_{k-n}|\mathbf{z}_{k-n})$.

3. BACKGROUND ON EXPECTATION PROPAGATION

EP is an extension to belief propagation (BP) and is specially developed for distributions outside of the exponential family [7, 4]. EP has relatively less complexity and may produce comparable performance to particle filtering. Moreover, EP is potentially more powerful and applicable to wider problem than BP.

Consider K independent observations generated from a statistical parametric model $\mathbf{z}_k = f(\mathbf{x}, \mathbf{n}_k)$, where $f(\cdot)$ is a parametric function, $\mathbf{x}$ the unknown parameter, and $\mathbf{n}_k$ random noise. Given the prior distribution $p(\mathbf{x})$, our object is to obtain the posterior distribution $p(\mathbf{x}|\mathbf{z}_{1:K})$. Unfortunately, except for limited cases where, for instance, $f(\cdot)$ is linear and $\mathbf{n}_k$ is Gaussian, the posterior distribution cannot be derived analytically. Instead, EP can be applied to approximate the desired posterior distribution.

EP consists of two major parts: initial density estimation and iterative refinement. In the first part, an initial estimate on $p(\mathbf{x}|\mathbf{z}_{1:k})$ is constructed by sequentially incorporating the observations from $k = 1$ to K. Note that the order is imposed for convenience of our composition and would be natural for dynamic systems, but it is not necessary in EP nevertheless. To see the procedure in detail, we assume that, at step $k-1$, we have obtained an approximation of $p(\mathbf{x}|\mathbf{z}_{1:k-1})$, say, $q(\mathbf{x}|\mathbf{z}_{1:k-1})$. We, however, restrict $q(\mathbf{x}|\mathbf{z}_{1:k-1})$ to be from the exponential family, a key requirement of EP when approximating posterior distribution. This is because that, with a distribution from the exponential family, only a fixed number of expectations (the sufficient statistics) need to be propagated. Now, to incorporate the new likelihood $p(\mathbf{z}_k|\mathbf{x})$ at step k and to obtain the new approximation $q(\mathbf{x}|\mathbf{z}_{1:k})$ from $p(x|z_{1:k})$, we start from the following relationship

$$\begin{aligned} p(\mathbf{x}|\mathbf{z}_{1:k}) &= \frac{p(\mathbf{z}_k|\mathbf{x})p(\mathbf{x}|\mathbf{z}_{1:k-1})}{Z} \\ &\approx \frac{p(\mathbf{z}_k|\mathbf{x})q(\mathbf{x}|\mathbf{z}_{1:k-1})}{Z} \\ &= \hat{q}(\mathbf{x}|\mathbf{z}_{1:k}) \end{aligned} \quad (7)$$

where Z is the normalizing constant. Since $\hat{q}(\mathbf{x}|\mathbf{z}_{1:k})$ may not be in the exponential family, we need to project $\hat{q}(\mathbf{x}|\mathbf{z}_{1:k})$ to the exponential family distribution to obtain the required approximation $q(\mathbf{x}|\mathbf{z}_{1:k})$. Based on the criterion that the Kullback-Leibler (KL) distance between the original and the projected is minimized, it is shown in [4] that the projection is equivalent to moment matching. For example, if $q(\mathbf{x}|\mathbf{z}_{1:k})$ is chosen to be Gaussian, then the moment matching matches the mean and variance of $q(\mathbf{x}|\mathbf{z}_{1:k})$ to those of $\hat{q}(\mathbf{x}|\mathbf{z}_{1:k})$. When $f(\cdot)$ is not linear, the moments of $\hat{q}(x|z_{1:k})$ cannot be obtained analytically. Techniques including quadratic approximation and unscented transformation can be used instead to approximate these moments. Note, with $q(\mathbf{x}|\mathbf{z}_{1:k})$ and $q(\mathbf{x}|\mathbf{z}_{1:k-1})$, the corresponding approximated likelihood function can be also obtained as

$$q(\mathbf{z}_k|\mathbf{x}) \propto \frac{q(\mathbf{x}|\mathbf{z}_{1:k})}{q(\mathbf{x}|\mathbf{z}_{1:k-1})} \quad (8)$$

When the above steps are finished at $k = K$, we obtain an initial estimate on $p(x|z_{1:K})$.

In the second part of EP, the approximation $q(\mathbf{x}|\mathbf{z}_{1:K})$ is refined iteratively. In each iteration, refinement is also performed sequentially from 1 to K by recycling the K likelihoods. Specifically, at the kth step, two sub-steps are included:

1. *Removal of the approximated likelihood*: The approximated likelihood is removed according to
$$q(\mathbf{x}|\mathbf{z}_{1:K;-k}) \propto \frac{q(\mathbf{x}|\mathbf{z}_{1:K})}{q(\mathbf{z}_k|\mathbf{x})} \quad (9)$$
where $\mathbf{z}_{1:K;-k}$ represents the collection of the observations except $\mathbf{z}_k$.
2. *True likelihood recycling and moment matching*: The true likelihood $p(\mathbf{z}_k|x)$ is then combined with $q(\mathbf{x}|\mathbf{z}_{1:K;-k})$ by the same fashion as in (7) and the refined approximation $q(\mathbf{x}|\mathbf{z}_{1:K})$ is obtained through moment matching.

After one sweep from $k = 1$ to K, this refinement iterates again until the convergence of $q(\mathbf{x}|\mathbf{z}_{1:K})$ and EP outputs the converged $q(\mathbf{x}|\mathbf{z}_{1:K})$ as the final approximation to $p(\mathbf{x}|\mathbf{z}_{1:K})$.

4. SENSOR SCHEDULING AND TRACKING

In Section 1 the tracking problem was described as being one of sensor scheduling. Of a suite of (in this case) three separate sensors, one is chosen for the measurement so as to minimize the estimated total MSE for that measurement. To choose the proper sensor, an EKF prediction step is carried out using the previous value for the state variable $\mathbf{x}$ and the previous estimate for the covariance matrix $\mathbf{P}$. Once the sensor for the measurement has been chosen, the measurement is executed using that sensor, a full EKF prediction and update process is executed to predict the next value of the state variable. This provides sufficient information to calculate the mean and variance of the estimated likelihood function at time k. Finally, the EP algorithm is begun to incorporate that measurement (and the estimate) into a new estimate for the state variable.

As stated in Section 3, the first objective of EP is to calculate the posterior distribution $p(\mathbf{x}|\mathbf{z}_{1:K})\ \forall k$. Since the measurement measurement $\mathbf{z_k}$ is related to the state vector $\mathbf{x_k}$ through a nonlinear function $\mathbf{h}(\mathbf{x_k})$, this distribution must be approximated by another distribution, $q(\mathbf{x}|\mathbf{z}_{1:K})$. This approximate posterior can be expressed (through Bayes' theorem) in terms of the estimated likelihood and the previous prediction of the posterior:

$$q(\mathbf{x}_k|\mathbf{z}_{1:K}) \propto q(\mathbf{z}_k|\mathbf{x}_k)q(\mathbf{x}_k|\mathbf{z}_{1:k-1}) \tag{10}$$

From the dynamic state system, we can write the following estimated distributions:

$$\begin{aligned} q(\mathbf{x}_k|\mathbf{x}_{k-1}) &= \mathcal{N}(\mathbf{F}\mathbf{x}_{k-1}, \mathbf{Q}) && (11)\\ q(\mathbf{z}_k|\mathbf{x}_k) &= \mathcal{N}(\mathbf{h}(\mathbf{x}_k), \mathbf{N}) && (12)\end{aligned}$$

where $\mathbf{Q}$ and $\mathbf{N}$ are the covariance matrices of the process and the measurement. Further, we can also write:

$$q(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int q(\mathbf{x}_k|\mathbf{x}_{k-1})q(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})d\mathbf{x}_{k-1} \tag{13}$$

Let us assume we have an estimate for the posterior at time $k-1$:

$$q(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) = \mathcal{N}(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}) \tag{14}$$

Then we can write the estimate for the integral in 13 as:

$$\begin{aligned} & q(\mathbf{x}_k|\mathbf{z}_{1:k-1}) \\ &= \int \mathcal{N}(\mathbf{F}\mathbf{x}_{k-1}, \mathbf{Q})\mathcal{N}(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k-1|k-1})d\mathbf{x}_{k-1} \\ &= \mathcal{N}(\hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}) && (15)\end{aligned}$$

where

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= \mathbf{F}\hat{\mathbf{x}}_{k-1|k-1} && (16)\\ \mathbf{P}_{k|k-1} &= \mathbf{F}\mathbf{P}_{k-1|k-1}\mathbf{F}^{\mathbf{T}} + \mathbf{Q} && (17)\end{aligned}$$

Since $\mathbf{z}_k$ is related nonlinearly to $\mathbf{x}_k$, we estimate the posterior using an EKF step:

$$\begin{aligned} q(\mathbf{x}_k|\mathbf{z}_{1:k}) &= \mathcal{N}(\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}) \\ \mathbf{G} &= \mathbf{P}_{k|k-1}\mathbf{H}_k^{\mathbf{T}}\left[\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^{\mathbf{T}} + \mathbf{N}_k\right]^{-1} && (18)\\ \hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{G}\left[\mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_{k|k-1})\right] && (19)\\ \mathbf{P}_{k|k} &= [\mathbf{I} - \mathbf{G}\mathbf{H}_k]\,\mathbf{P}_{k|k-1} && (20)\end{aligned}$$

$\mathbf{H}_k$ is the Jacobian matrix of $\mathbf{h}(\mathbf{x}_k)$ evaluated at $\mathbf{x}_k = \hat{\mathbf{x}}_{k|k-1}$ and can be written as:

$$\mathbf{H}_k = \begin{bmatrix} \frac{\partial r_k}{\partial x_k} & \frac{\partial r_k}{\partial \dot{x}_k} & \frac{\partial r_k}{\partial y_k} & \frac{\partial r_k}{\partial \dot{y}_k} \\ \frac{\partial \dot{r}_k}{\partial x_k} & \frac{\partial \dot{r}_k}{\partial \dot{x}_k} & \frac{\partial \dot{r}_k}{\partial y_k} & \frac{\partial \dot{r}_k}{\partial \dot{y}_k} \\ \frac{\partial \phi_k}{\partial x_k} & \frac{\partial \phi_k}{\partial \dot{x}_k} & \frac{\partial \phi_k}{\partial y_k} & \frac{\partial \phi_k}{\partial \dot{y}_k} \end{bmatrix} \tag{21}$$

Since the estimated posterior in exponential form, no moment matching is required. With the parameters thus estimated, we can now calculate the estimated likelihood:

$$\begin{aligned} q(\mathbf{z}_k|\mathbf{x}_k) &\propto \frac{q(\mathbf{x}_k|\mathbf{z}_{1:k})}{q(\mathbf{x}_k|\mathbf{z}_{1:k-1})} \\ &\propto \mathcal{N}(\tilde{\mu}_k, \tilde{\lambda}_k) \\ \tilde{\lambda}_k &= (\mathbf{P}_{k|k}^{-1} - \mathbf{P}_{k|k-1}^{-1})^{-1} && (22)\\ \tilde{\mu}_k &= \tilde{\lambda}_k(\mathbf{P}_{k|k}^{-1}\hat{\mathbf{x}}_{k|k} - \mathbf{P}_{k|k-1}^{-1}\hat{\mathbf{x}}_{k|k-1}) && (23)\end{aligned}$$

Now that we have calculated an estimate of the posterior at time k, we begin the second half of the EP process. As discussed in Section 3, the smoothing process requires two steps. First, if the time index i under review is less than the most recent K, an EKF smoothing step is accomplished:

$$\begin{aligned} \mathbf{J}_i &= \mathbf{P}_{i|i}\mathbf{F}^{\mathbf{T}}\mathbf{P}_{i+1|i}^{-1} && (24)\\ \hat{\mathbf{x}}_{i|K} &= \hat{\mathbf{x}}_{i|i-1} + \mathbf{J}_i\ \ \hat{\mathbf{x}}_{i+1|K} - \hat{\mathbf{x}}_{i+1|i} && (25)\\ \mathbf{P}_{i|K} &= \mathbf{P}_{i|i} + \mathbf{J}_i(\mathbf{P}_{i+1|K} - \mathbf{P}_{i+1|i})\mathbf{J}_i^{\mathbf{T}} && (26)\end{aligned}$$

Next, we remove the estimated likelihood from the smoothing densities, over the length of the smoothing window. This is accomplished using the equations:

$$\begin{aligned} q(\mathbf{x}_i|\mathbf{z}_{1:K,-i}) &= \mathcal{N}(\hat{\mathbf{x}}_{-i|K}, \mathbf{P}_{-i|K}) \\ \mathbf{P}_{-i|K} &= (\mathbf{P}_{i|K}^{-1} - \tilde{\lambda}_i^{-1})^{-1} && (27)\\ \hat{\mathbf{x}}_{-i|K} &= \mathbf{P}_{-i|K}(\mathbf{P}_{i|K}^{-1} - \tilde{\lambda}_i^{-1}\tilde{\mu}_i) && (28)\end{aligned}$$

where the notation $\{\text{par}\}_{-i|K}$ means the estimate of the parameter $\{\text{par}\}$ based on inputs from time $1:K$, with time i removed, and $\{\text{par}\}_{i|K}$ means the estimate of the parameter $\{\text{par}\}$ based on inputs from time $1:K$, **at** time i. Next, another EKF prediction is accomplished, and a new mean and covariance for the likelihood function is calculated. The EKF process in (24) through (26) is then repeated for the next prior point, and so on, until the entire smoothing window is covered.

The second step of the smoothing process begins with the most prior point in the smoothing window. First, the

mean and error covariance of the predicted density at time i are calculated:

$$\hat{\mathbf{x}}_{i|i-1} = \mathbf{F}\hat{\mathbf{x}}_{i-1|i-1} \quad (29)$$

$$\mathbf{P}_{i|i-1} = \mathbf{F}\mathbf{P}_{i-1|i-1}\mathbf{F}^{\mathbf{T}} + \mathbf{Q} \quad (30)$$

Then, the true likelihoods are incorporated into the posterior density at that point:

$$\mathbf{P}_{i|i} = (\mathbf{P}_{i|i-1} - \tilde{\lambda}_i^{-1})^{-1} \quad (31)$$

$$\hat{\mathbf{x}}_{i|i} = \mathbf{P}_{i|i}(\mathbf{P}_{i|i-1}^{-1}\hat{\mathbf{x}}_{i|i-1} + \tilde{\mu}_i) \quad (32)$$

The algorithm for the sensor scheduling and tracking using EP can be stated as follows:

At time index $k = K$:

1. Calculate the mean and error covariance of the predicted density using equations (16) and (17)
2. Evaluate the Jacobian of the non-linear measurement function at $\mathbf{x}_k = \hat{\mathbf{x}}_{K|K-1}$ using equation (21)
3. Calculate the estimated covariance matrices of the individual sensors ($\mathbf{N}_s$ is the covariance matrix for sensor $\mathbf{s}$):

$$\hat{\mathbf{P}}_{K|K}^{s} = \left[\mathbf{P}_{K|K-1}^{-1} + \mathbf{H}^{\mathbf{T}}\mathbf{N}_s^{-1}\mathbf{H}\right]^{-1} \quad (33)$$

4. Choose the sensor for observation K by choosing the covariance matrix $\mathbf{N}_s$ that minimizes the trace of $\hat{\mathbf{P}}_{K|K}$
5. Perform EKF moment matching using equations (18) through (20) where $\mathbf{N}$ is the covariance matrix of the chosen sensor
6. Calculate the mean and covariance of the estimated likelihood function $p(\mathbf{z}_K|\mathbf{x}_K)$ using equations (22) and (23)

•• For time index $i = K : -1 : K - (windowsize) + 1$

7. Perform Kalman smoothing (for $i < K$) using equations (24) through (26)
8. Remove the likelihood using equations (27) and (28)
9. Calculate the Jacobian of the non-linear measurement function at $\mathbf{x}_k = \hat{\mathbf{x}}_{-i|K}$ using equation (21)
10. Perform new moment matching using equations (18) through (20), but evaluated thus:

$$\mathbf{G} = \mathbf{P}_{i|K}\mathbf{H}_k^{\mathbf{T}}\left[\mathbf{H}_k\mathbf{P}_{i|K}\mathbf{H}_k^{\mathbf{T}} + \mathbf{N}_i\right]^{-1} \quad (34)$$

$$\hat{\mathbf{x}}_{i|i} = \hat{\mathbf{x}}_{i|K} + \mathbf{G}\left[\mathbf{z}_i - \mathbf{h}(\hat{\mathbf{x}}_{i|K})\right] \quad (35)$$

$$\mathbf{P}_{i|i} = [\mathbf{I} - \mathbf{G}\mathbf{H}]\,\mathbf{P}_{i|K} \quad (36)$$

11. Obtain the new mean and covariance for the likelihood function at i using equations (22) and (23) evaluated thus:

$$\tilde{\lambda}_i = (\mathbf{P}_{i|i}^{-1} - \mathbf{P}_{-i|K}^{-1})^{-1} \quad (37)$$

$$\tilde{\mu}_K = \tilde{\lambda}_i(\mathbf{P}_{i|i}^{-1}\hat{\mathbf{x}}_{i|i} - \mathbf{P}_{-i|K}^{-1}\hat{\mathbf{x}}_{-i|K}) \quad (38)$$

•• For time index $i = K - (windowsize) + 1 : K$

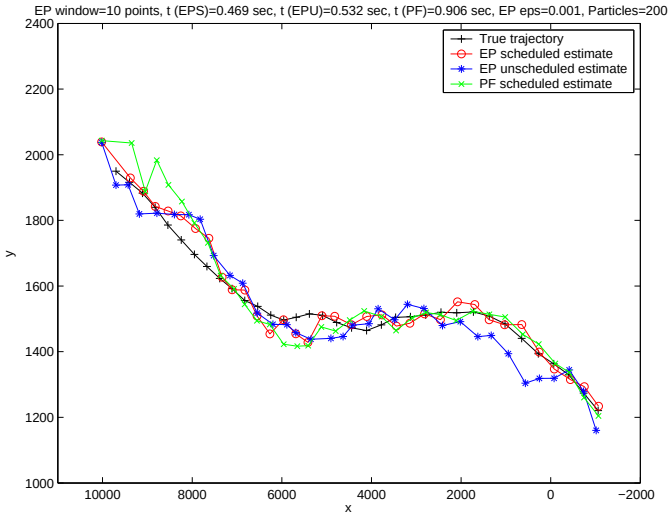

Fig. 1. True vs. estimated trajectories ("good" PF)

12. Calculate the mean and error covariance of the predicted density using equations (29) and (30)
13. Incorporate the true likelihoods into the posterior density using equations (31) and (32)
14. Evaluate the convergence. Return to step 7 if convergence is not reached; otherwise, return to step 1 and begin with next time index

---

In evaluating the convergence, the Kullback-Leibler (KL) divergence of the two most recent estimated filtering distributions of the state variable is calculated at the end of an EP smoothing/update loop. Smoothing and updating is terminated when this difference falls below a predefined threshold. If we assume the estimated posterior density $\mathbf{q}(\mathbf{x}_k|\mathbf{z}_{1:k})$ at the end of EP iteration i to be:

$$q^i(\mathbf{x}_k|\mathbf{z}_{1:k}) = \mathcal{N}(\mathbf{x}_{k|k}^i, \mathbf{P}_{k|k}^i) \quad (39)$$

then the KL divergence at the end of iteration $i+1$ is given as:

$$\begin{aligned} &KL(q^{i+1}|q^i) \\ &= (\mathbf{x}_{k|k}^i - \mathbf{x}_{k|k}^{i+1})^{\mathbf{T}}(\mathbf{P}_{k|k}^{i+1})^{-1}(\mathbf{x}_{k|k}^i - \mathbf{x}_{k|k}^{i+1}) \\ &\quad + \mathbf{tr}(\mathbf{P}_{k|k}^i(\mathbf{P}_{k|k}^{i+1})^{-1} - \mathbf{I}) \\ &\quad - \ln\ \mathbf{P}_{k|k}^i(\mathbf{P}_{k|k}^{i+1})^{-1} \end{aligned} \quad (40)$$

5. SIMULATION RESULTS

The algorithm was tested against a notional 35-second data set, with 3 separate sensors. The smoothing window size for EP was set at 10, and the convergence threshold was 10^{-3}. Each sensor was modeled as accurate in one of the three observed variables (range, range rate, and angle) and could

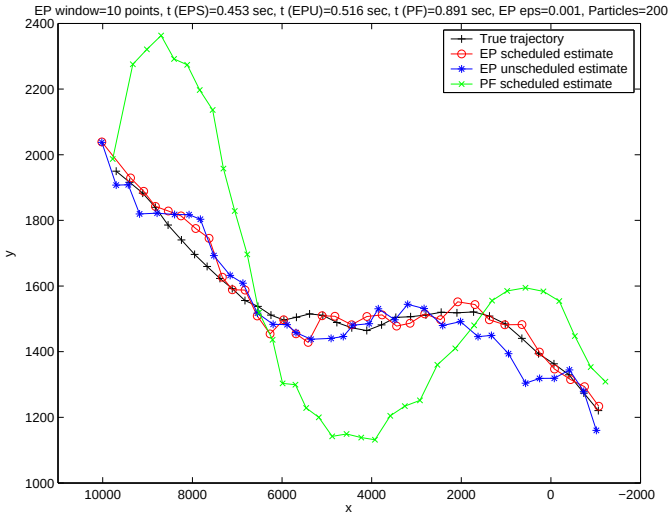

Fig. 2. True vs. estimated trajectories ("bad" PF)

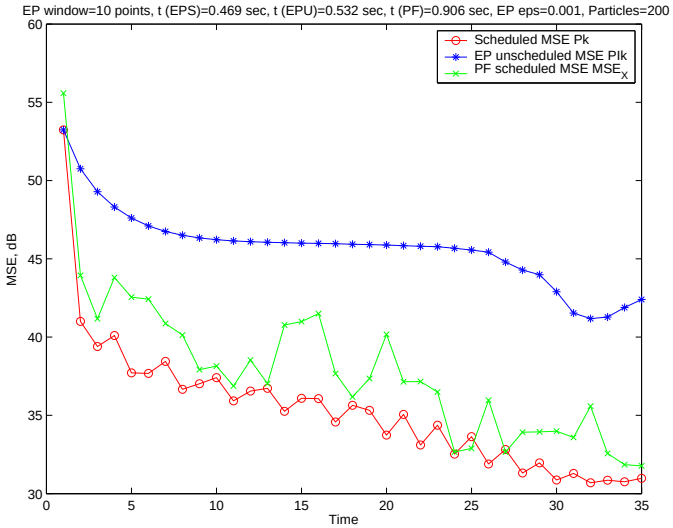

Fig. 3. Total MSE comparisons ("good" PF)

be thought of as generic radar, Doppler, and infra-red (IR) sensors. The sensor measurement covariance matrices were similar to those used in [1], although in [1] only 2 sensors were used.

In addition, the same EP algorithm was tested without the scheduling step, using only the angle sensor. Finally, a scheduled PF algorithm (with 200 particles) was also tested against this data.

Figures 1 and 2 show show two different results of the test. Since particle filtering is a stochastic process, the results can vary from run to run, depending on the initialization as well as the random draws by which the particles are populated. EP, however, does not vary significantly from run to run. Thus, the only difference seen in figures 1 and 2 is in the particle filtering result.

As can be seen, all three algorithms tracked the true trajectory to some degree. However, the scheduled EP algorithm provided the closest track, followed by the scheduled PF algorithm. Further, the PF algorithm had significantly different track results, depending on the run, while the EP algorithm converged to a single track.

Figures 3 and 4 show the estimated error covariance matrices. Plotted are the total MSE versus time, in logarithmic units. The total MSE is calculated from the trace of the error covariance matrix. As would be expected, in figure 4 the total MSE for the PF implementation is much higher than for the scheduled EP implementation. However, figure 3 shows a total MSE of somewhat the same order as the scheduled EP implementation (although still slightly worse).

Note, however, that the total MSE for the scheduled EP implementation does not change from run to run. Also note that the total MSE for the unscheduled EP implementation is relatively constant over a long period of time. This is due to the fact that a single sensor (with its associated covariance matrix) is used in this implementation.

Besides a consistency from simulation to simulation, what was purchased by switching to EP over PF? As noted

Table 1. Average run times for each implementation

Method	Parameters	Run time $\pm 1\sigma$
EP (sch)	W = 10, KL = $\mathbf{10^{-3}}$	$\mathbf{0.449 \pm 0.016}$
EP (unsch)	W = 10, KL = $\mathbf{10^{-3}}$	$\mathbf{0.509 \pm 0.025}$
PF	Particles = 200	$\mathbf{0.908 \pm 0.016}$

in Section 1, EP is a more straightforward implementation than PF. It is less complex to program, and it is less stringent in terms of memory requirements. Most importantly, however, it is a faster implementation.

Table 1 shows the average runtimes for each implementation, at the parameters given, as well as the standard deviation. A total of 30 runs were averaged for these times. As can be seen, the scheduled EP algorithm ran in half the time as the scheduled PF algorithm. Further, the scheduled EP algorithm actually ran slightly faster than the unscheduled EP version. This is most likely due to the fact that, since the scheduled EP algorithm chooses which observation to use based on predicted error covariances, the smoothing and update iterations were likely fewer than with the unscheduled EP. The computation time lost in the scheduling algorithm was more than made up in the smoothing and update.

6. SUMMARY AND OUTLOOK FOR FUTURE STUDY

A new method for sensor scheduling and target tracking is presented. The method makes use of the EP algorithm rather than the currently-used PF algorithm. Not only is the EP algorithm more efficient in computational time and storage, it also provides better tracking performance than PF. Future study of this (and similar) problems would likely examine the use of different prediction/update methods (such as the unscented Kalman filter), and much more

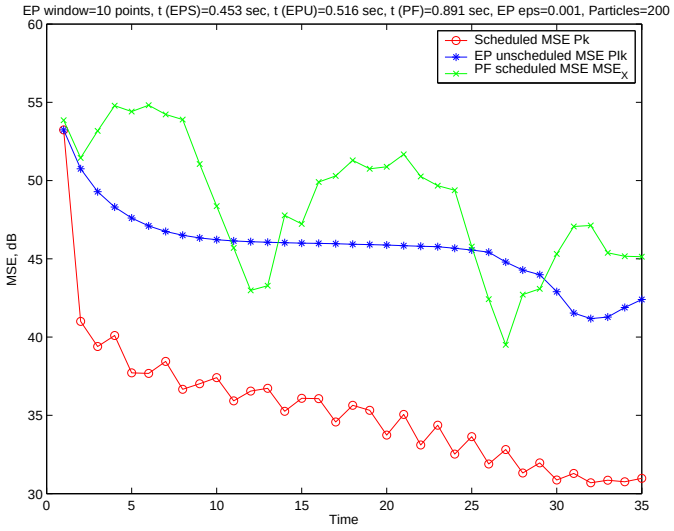

Fig. 4. Total MSE comparisons ("bad" PF)

rigorous programming methods for PF and EP to better validate our run time results.

7. REFERENCES

[1] A. S. Chhetri, D. Morrell, and A. Papandreou-Suppappola, "Scheduling multiple sensors using particle filters in target tracking."

[2] P. Djurić, J. Kotecha, J. Zhang, Y. Huang, M. Bugallo, and J. Miguez, "Particle filter," *IEEE Signal Processing Magazine*, September 2003.

[3] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, vol. 50, no. 2, pp. 174–188, 2002.

[4] T. P. Minka, "A family of algorithms for approximate bayesian inference," Ph.D. dissertation, MIT, Jan 2001.

[5] A. Ypma and T. Heskes, "Iterated extended kalman smoothing with expectation propagation," *Proceedings of NNSP*, 2003.

[6] Y. Qi and T. Minka, "Expectation propagation for signal detection in flat-fading channels," in *proceedings of IEEE International Symposium on Information Theory*, vol. Yokohama, Japan, June 2003.

[7] T. Wei, Y. Huang, and Y. Qi, "Symbol detection with time-varying unknown phase by expectation propagation," in *Proceedings of ICASSP*, 2005.