Scientific Research

Blending Sensor Scheduling Strategy with Particle Filter to Track a Smart Target

Bin LIU[1], Chunlin JI[1], Yangyang ZHANG[2], Chengpeng HAO[3]

[1]*Department of Statistical Science, Duke University, Durham, U. S. A*
[2]*Adastral Park Research Campus, University College London, London, UK*
[3]*Institute of Acoustics, Chinese Academy of Sciences, Beijing, China*
Email: {*bin.liu2, chunlin.ji*}*@duke.edu, y.zhang@adastral.ucl.ac.uk, haochengp@sohu.com*

Abstract

We discuss blending sensor scheduling strategies with particle filtering (PF) methods to deal with the problem of tracking a 'smart' target, that is, a target being able to be aware it is being tracked and act in a manner that makes the future track more difficult. We concern here how to accurately track the target with a care on concealing the observer to a possible extent. We propose a PF method, which is tailored to mix a sensor scheduling technique, called covariance control, within its framework. A Rao-blackwellised unscented Kalman filter (UKF) is used to produce proposal distributions for the PF method, making it more robust and computationally efficient. We show that the proposed method can balance the tracking filter performance with the observer's concealment.

1. Introduction

The problem of target tracking has received considerable attention from both academic and engineering communities. Generally, people formulate this problem as a state estimation or filtering problem, focusing on the tracking filter's performance. A limitation of such works in the literature is that it assumes that any changes in the behavior of the target are unconnected to the action of the target tracking process. However, in many real-world situations, this is an unrealistic assumption. Taking a sonar application as an example, where the observer is an autonomous underwater vehicle (AUV), the target, e.g., a submarine equipped with elaborate detection instruments, is able to detect, and, once it is aware it is being tracked, it can modify its behavior quickly to escape from this track and make the future track more difficult.

A complete solution to the problem of tracking a smart target is still an open problem. However, some initial results are available. Kreucher *et al* perform a reinforcement learning approach to schedule a multi-modality

sensor to detect and track smart targets [1]. For their approach, a multi-step ahead scheduling policy is essential to provide sensible performance. Savage et al. consider an idealized problem where the target has a set of possible motion models and selects the one to best reduce the sensor's tracking performance, and treat this problem in the framework of game theory [2]. Gittins and Roberts use game theory to investigate the case in which a target is trying to escape detection [3,4]. We consider the problem of tracking a smart target with a care on concealing the observer to an extent and propose a smart tracker by mixing a sensor scheduling technique with particle filtering (PF) methods [5].

This paper is an expanded version of [5]. Here we assume there are two sensors to be used by the observer, with passive and active modalities, respectively. The passive sensor measures the energy that has already existed in the environment, without emitting any energy outside. Such a quite mode makes the observer conceal itself well, but cannot guarantee the tracking performance, especially when the SNR is small. Differing from the passive sensor, the active one emits energy to the environment and before collecting reflected energy to do detection. Such an active mode has substantially better detection and tracking capabilities than the passive one,

however, it makes the observer easily detected by the target. So, employing these two sensors, there is a contradiction between the tracking filter's performance and the concealment of the observer. The goal of this paper is actually to design a method, which can both guarantee the tracking filter's performance and conceal the observer to a reasonable extent. We resort to sensor scheduling strategies and particle filtering (PF) methods to seek a balance between these two aspects.

Sensor job (time) scheduling is within the context of multi-sensor management. It has become increasingly important in the research and development of modern multi-sensor systems. Sensor scheduling lies in the first level of a top-down policy of sensor management with the role of assigning each sensor with a detailed schedule on what to do [6]. PF is a Sequential Monte Carlo method which founds great research and applications in the last decade (see [7–9] and references therein). It beats Kalman filter, a classical method used in the target tracking discipline, in dealing with nonlinear dynamical and measurement models and non-Gaussian noises in the model. Theoretically, employing enough particles, PF can provide an approximate optimal Bayesian solution to any state-space based estimation problem. In this paper we mix a specific sensor scheduling technique, namely covariance control [10], with PF methods to deal with the problem of tracking a smart target. We use a Rao-blackwellised unscented Kalman filter (UKF) [11] to produce proposal distributions for the PF, making it more robust and computationally efficient. It is shown that the proposed method provides a balance between the tracking filter's performance and the observer's concealment, hence it satisfies our needs for the problem under consideration.

The remaining of this paper is organized as follows. Section 2 describes the dynamic models involved. Section 3 presents the sensor scheduling technique, covariance control. The proposed PF algorithm is illustrated in Section 4 and its performance is evaluated in Section 5. Finally we conclude this paper in Section 6.

2. Models

In this section, we describe the models involved in this paper. First the dynamic model for the target is presented.

Then the measurement models are derived for both the passive and the active sensors.

The evolution of the target state, $\mathbf{x}_k$, is modeled by a discrete time linear Gaussian:

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + v_k \tag{1}$$

where $v_k \sim N(0, \mathbf{Q})$. Here the target state vector is composed of the position and velocity items in the x and y coordinates and is defined as follows:

$$\mathbf{x}_k = \begin{bmatrix} x_{t,k} & \dot{x}_{t,k} & y_{t,k} & \dot{y}_{t,k} \end{bmatrix}^T \tag{2}$$

where the dot denotes the operation of first order derivative and the superscript T denotes transposition of a matrix. We use a constant-velocity process model for the target, so that

$$\mathbf{F} = \begin{bmatrix} \mathbf{F_s} & 0 \\ 0 & \mathbf{F_s} \end{bmatrix}, \mathbf{F_s} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \tag{3}$$

and

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q_s} & 0 \\ 0 & \mathbf{Q_s} \end{bmatrix}, \mathbf{Q_s} = q_s \begin{bmatrix} T^3/3 & T^2/2 \\ T^2/2 & T \end{bmatrix} \tag{4}$$

where T is the sampling period, q_s is the power spectral density of the acceleration noise in the spatial dimensions.

Defining $\varphi_k = (x_{o,k}, y_{o,k})$ as the observer's position at time step k, we derive measurement functions for both the passive and the active sensors in the following. We consider the case where the passive sensor only provides relative bearings measurements originated from the target, then the associated measurement function is

$$\mathbf{z}_k = \operatorname{atan}\left(\frac{x_{t,k} - x_{o,k}}{y_{t,k} - y_{o,k}}\right) + n_k \tag{5}$$

where $n_k \sim N(0, \mathbf{R}_b)$.

We assume the observer adopts a track-while-scan sensor [10] to do active sensing, which can measure both the bearings and the ranges. The associated measurement function is denoted as

$$\mathbf{z}_k = \left[\operatorname{atan}\left(\frac{x_{t,k} - x_{o,k}}{y_{t,k} - y_{o,k}}\right), \sqrt{(x_{t,k} - x_{o,k})^2 + (y_{t,k} - y_{o,k})^2}\right]^T + \left[n_k, r_k\right]^T \tag{6}$$

where $r_k \sim N(0, \mathbf{R}_d)$ denotes the noise item in the range. So the covariance matrix of the measurement noise is $diag[\mathbf{R}_b, \mathbf{R}_d]$. Here *diag* denotes the operation of diagonalization.

3. A Sensor Scheduling Technique: Covariance Control

In this section we present the sensor scheduling technique, called covariance control, which will be embedded in the PF framework described in Section 4.

Covariance control begins with a desired covariance matrix, which is this approach differs from many other sensor management algorithms. A desired covariance matrix for an n-dimensional state estimate, $\mathbf{P}_D$, is defined by all $n \times n$ elements of that matrix. The goal is to find a specific sensor combination i that produces covariance matrix $\mathbf{P}^i$, assuring the difference $\mathbf{P}_D - \mathbf{P}^i$ is positive semi-definite. To properly evaluate that difference, a scalar metric is needed. A variety of these exist, including functions based on the determinant or the trace of the matrix. However, these metrics rely on the positive definiteness of the matrix to provide accurate evaluations. If a difference is only semi-definite, then the determinant is zero, possibly masking a large difference in a different direction (note that a covariance can be represented as an ellipsoid, whose axes directions can be indicated by the eigenvectors of the covariance matrix). A similar problem exists with the trace, where a large positive difference can mask a large negative difference along a different direction. To avoid these problems, M. Kalandros and L. Y. Pao, examined other techniques, such as the eigenvalue/minimum sensors algorithm, the matrix norm algorithm and the norm/sensors algorithm [10]. The norm/sensors algorithm relaxes the requirements of the matrix norm technique, allowing the norm of the covariance difference to vary within a predefined boundary $\pm \delta$. So we borrow the idea of the norm/sensors algorithm and propose the following sensor scheduling strategy:

• If

$$\left\| \mathbf{P}_D - \mathbf{P}_k \right\|_2 > \delta \qquad (7)$$

select the active sensor to work for next time step;
• Else
select the passive sensor to work for next time step. ($\mathbf{P}_k$ denotes the covariance matrix associated with the estimate for the target state at the k th time step)

Note that the aim of this sensor scheduling strategy is to select an appropriate sensor for use for next iteration of the tracking process, other than to search a sensor combination that can work with the fewest sensors involved, which is the purpose of the methods proposed in [10].

4. Particle Filtering Algorithm

This section presents our proposed PF algorithm. First we give a brief introduction for a basic PF method. Then we describe the Rao-blackwellised UKF [11], which is used to produce proposal distributions for our PF method. Finally we mix the sensor scheduling technique presented in Section 3 with the PF algorithm, leading to the proposed method for tracking a smart target.

Particle filter is a Sequential Monte Carlo method, whose basic idea is very simple: the target distribution is represented by a weighted set of Monte Carlo samples. These samples are propagated and updated using a sequential version of importance sampling as new measurements become available. We summarize a basic PF algorithm as follows, while referring the reader to [7–9] for detail discussions on PF methods.

Algorithm 1: Basic Particle Filter Algorithm
❖   Initialization. Sample N equally weighted particles from the initial pdf of the target state, $p(\mathbf{x}_0)$:
• For $i = 1, \ldots, N$

$$\mathbf{x}_0^i \sim p(\mathbf{x}_0); \quad \omega_0^i = \frac{1}{N}$$

• Set $k \leftarrow 0$
❖   Iteration $k+1$
• Sampling new particles from proposal distribution $q(\cdot)$, i.e.,
For $i = 1, \ldots, N \quad \mathbf{x}_{k+1}^i \sim q(\cdot)$
• Evaluate importance weights:

$$\widetilde{\omega}_{k+1}^i = \frac{p(\mathbf{z}_{k+1} \mid \mathbf{x}_{k+1}^i) p(\mathbf{x}_{k+1}^i \mid \mathbf{x}_k^i)}{q(\mathbf{x}_{k+1}^i)}$$

• Normalize the importance weights such that

$$\omega_{k+1}^i \propto \widetilde{\omega}_{k+1}^i, \text{ and } \sum_{i=1}^N \omega_{k+1}^i = 1$$

• Selection step: Multiply/Suppress particles with high/low importance weights respectively, resulting in a set of equally weighted particles, $\mathbf{x}_{k+1}^i, i = 1, \ldots, N$.
• Output:

$$E(\mathbf{x}_{k+1}) = \frac{1}{N} \sum_{i=1}^N (\mathbf{x}_{k+1}^i)$$
$$Cov(\mathbf{x}_{k+1}) = \frac{1}{N} \sum_{i=1}^N (\mathbf{x}_{k+1}^i - E(\mathbf{x}_{k+1}))(\mathbf{x}_{k+1}^i - E(\mathbf{x}_{k+1}))^T$$

The design of the proposal distribution, i.e., $q(\cdot)$, is of paramount importance for the PF algorithm. It has been shown that UKF can be used to produce good proposal distributions, particularly when the observation model is nonlinear [12]. The idea is that one treats a Guassian distribution outputted by the UKF as the PF's proposal distribution. It is shown that Rao-blackwellization technique can be used to improve the UKF's computational efficiency [11]. So here we adopt the Rao-blackwellised

UKF (RB-UKF) to generate the PF's proposal. An implementation of RB-UKF based on the models described in Section 2 is summarized as follows.

Algorithm 2: RB-UKF Algorithm
Assume we have got the estimate for the target state at time step k, $\mathbf{x}_k$, with its corresponding covariance, $\mathbf{P}_k$, the goal is to solve $\mathbf{x}_{k+1}$ and $\mathbf{P}_{k+1}$, as a new measurement $\mathbf{z}_{k+1}$ arrives.

❖ Linear State Prediction:

- $\mu_p = \mathbf{F}\mathbf{x}_k$

- $\mathbf{P}_p = \mathbf{Q} + \mathbf{F}\mathbf{P}_k\mathbf{F}^T$

❖ *Sigma* points sampling

- $\chi_0 = \mu_p$, $\omega_0^c = \lambda/(n+\lambda) + (1-\alpha^2+\beta)$

- $\chi_i = \mu_p + \left(\sqrt{(n+\lambda)\mathbf{P}_p}\right)_i$, $\omega_i^c = \dfrac{1}{2(n+\lambda)}$,

  for $i = 1,\ldots,n$

- $\chi_i = \mu_p - \left(\sqrt{(n+\lambda)\mathbf{P}_p}\right)_i$, $\omega_i^c = \dfrac{1}{2(n+\lambda)}$,

  for $i = n+1,\ldots,2n$

where n is the dimension of the state vector, and α, β, and λ are parameters prescribed beforehand for the UKF.

❖ Nonlinear measurement update based on Unscented Transform

- $\mathbf{z}_{i,u} = h(\chi_i)$, $i = 0,1,\ldots,2n$. ($h(\cdot)$ denotes the measurement function)

- $\hat{\mathbf{z}}_u = \sum_{i=0}^{2n} \omega_i^c \mathbf{z}_{i,u}$

- $\mathbf{P}_u = \sum_{i=0}^{2n} \omega_i^c \left(\mathbf{z}_{i,u} - \hat{\mathbf{z}}_u\right)\left(\mathbf{z}_{i,u} - \hat{\mathbf{z}}_u\right)^T$

- $\mathbf{P}_c = \sum_{i=0}^{2n} \omega_i^c \left(\chi_i - \mu_p\right)\left(\mathbf{z}_{i,u} - \hat{\mathbf{z}}_u\right)^T$

- $\mathbf{K} = \mathbf{P}_c \mathbf{P}_u^{-1}$

- $\mathbf{x}_{k+1} = \mu_p + \mathbf{K}\left(\mathbf{z}_{k+1} - \hat{\mathbf{z}}_u\right)$

- $\mathbf{P}_{k+1} = \mathbf{P}_p - \mathbf{K}\mathbf{P}_u\mathbf{K}^T$

Next we use such RB-UKF algorithm to generate proposal distributions for the PF, and mix the sensor scheduling technique proposed in Section 3 into the PF framework, leading to the proposed PF algorithm.

Algorithm 3: The Proposed PF Algorithm for Smart Target Tracking
❖ Initialization.
- Sample N equally weighted particles from the initial *pdf* of the target state, $p(\mathbf{x}_0)$

- Assign specific values for the desired covariance

matrix, $\mathbf{P}_d$, and δ

- Set $k \leftarrow 0$
- Sensor scheduling for the next time step: use the active sensor while keep the passive one idle
❖ Iteration $k+1$
- For $i = 1,\ldots,N$
➤ Perform RB-UKF algorithm to $\mathbf{x}_k^i$ to get

$$\left\{\tilde{\mathbf{x}}_{k+1}^i, \mathbf{P}_{k+1}^i\right\}$$

➤ Sample a new particle from the proposal distribution: $\mathbf{x}_{k+1}^i \sim q(\cdot) = N\left(\tilde{\mathbf{x}}_{k+1}^i, \mathbf{P}_{k+1}^i\right)$

- Evaluate importance weights:

$$\tilde{\omega}_{k+1}^i = \frac{p\left(\mathbf{z}_{k+1}\mid\mathbf{x}_{k+1}^i\right)p\left(\mathbf{x}_{k+1}^i\mid\mathbf{x}_k^i\right)}{q\left(\mathbf{x}_{k+1}^i\right)}$$

Normalize the importance weights such that

$$\omega_{k+1}^i \propto \tilde{\omega}_{k+1}^i, \text{ and } \sum_{i=1}^{N}\omega_{k+1}^i = 1$$

- Selection step: Multiply/Suppress particles with high/low importance weights respectively, resulting in a set of equally weighted particles, $\mathbf{x}_{k+1}^i, i = 1,\ldots,N$.
- Output:

$$E\left(\mathbf{x}_{k+1}\right) = \frac{1}{N}\sum_{i=1}^{N}\left(\mathbf{x}_{k+1}^i\right)$$

$$Cov\left(\mathbf{x}_{k+1}\right) = \frac{1}{N}\sum_{i=1}^{N}\left(\mathbf{x}_{k+1}^i - E\left(\mathbf{x}_{k+1}\right)\right)\left(\mathbf{x}_{k+1}^i - E\left(\mathbf{x}_{k+1}\right)\right)^T$$

- Sensor scheduling for the next time step:
➤ If $\left\|\mathbf{P}_D - Cov\left(\mathbf{x}_{k+1}\right)\right\|_2 > \delta$

select the active sensor to work while keep the passive one idle;
➤ Else, select the passive sensor to work while keep the active one idle.

5. Performance Evaluation

In this section, we evaluate the performance of our proposed method in Section 4 by simulations. First, we compare the tracking performance of our method with those of two other trackers, one adopting the passive sensor for detection and the other utilizing the active sensor for detection, based on a set of Monte-Carlo (MC) simulations. The purpose of this comparison is to demonstrate the effect of the sensor scheduling technique in the aspect of concealing the observer. Next, we investigate the effects of the parameter δ on our method's performance. This parameter is used to measure the difference between the desired covariance and the current estimation covariance in the sensor scheduling stage of our method.

The scenario to be investigated is shown in Figure 1. The observer travels at a fixed speed of 10m/s and executes 2 maneuvers. The observation period lasts 40 seconds. The target motion, described by (1) in this simulation, is subjected to an amount of process noise with $q_s = 1$. The initial position and speed of the target are $(300m, 300m)$ and $(12.25\,m/s, -12.25\,m/s)$, respectively. The other parameters for simulation initialization are summarized in Table 1.

For performance comparison, we take the root-mean square (RMS) position error as the index:

$$\text{RMS}_k = \sqrt{\frac{1}{M} \cdot \sum_{i=1}^{M} \left(\bar{x}_{t,k}^i - x_{t,k}^i\right)^2 + \left(\bar{y}_{t,k}^i - y_{t,k}^i\right)^2} \qquad (8)$$

where $(x_{t,k}^i, y_{t,k}^i)$ and $(\bar{x}_{t,k}^i, \bar{y}_{t,k}^i)$ denote the true and the estimated target positions at time step k at the ith MC run, and M is the total number of independent MC runs. Here $M = 50$ runs are done for the following three trackers, the proposed sensor scheduling based PF (SS-PF) tracker, the passive/active mode PF (PaPF/AcPF) tracker which only use the passive/active sensor in the filtering process. As shown in Figure 2, the performance of the proposed SS-PF tracker is comparable to that of the AcPF tracker, and it is much better than that of the PaPF tracker. For the SS-PF tracker, the average number of time epochs, when the active sensor is used during the whole tracking process, is only 13. It means that the SS-PF tracker gets a similar filtering performance as that of the tracker which uses the active sensor all the time, while concealing the observer to an extent by reducing the use of the active sensor. A specific estimation result of this SS-PF for the target's trajectory is shown in Figure 1; the associated sensor scheduling result is also illustrated in Figure 4. As can be seen, at first, the SS-PF tracker selects the active sensor to do detection to get a good enough tracking initialization, then it dynamically switch the uses of the passive and the active sensors online. The sensor switch uses of the passive and the active sensors online. The sen-

Table 1. Parameters used for initialization.

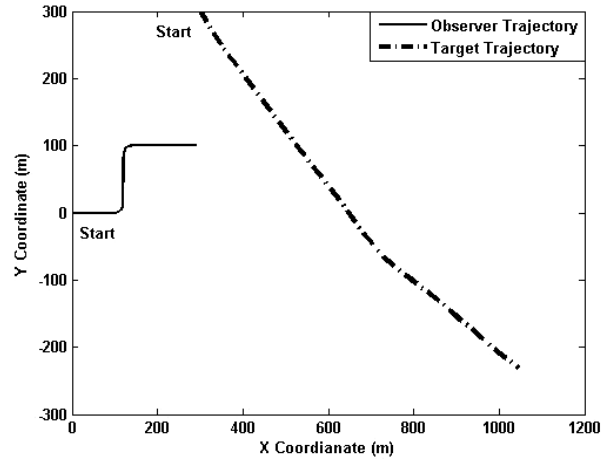| Symbol	Quantity	Value
T	Sampling period	$1s$
σ_b	Standard error of bearing noise	$1°$
σ_d	Standard error of range noise	$5m$
N	Particle Number	200
$\mathbf{P}_D$	desired covariance matrix	diag([5 0.25 0.2])
δ	predefined boundary for the norm of covariance	5

Figure 1. The observer's and the target's movement trajectories in this experiment.
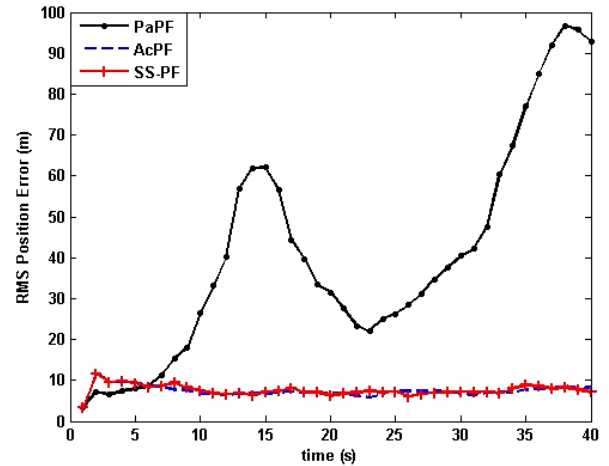

Figure 2. RMS position error versus time. PaPF and AcPF denotes passive mode and active mode PF tracker respectively, and SS-PF denotes the proposed tracker in this paper.
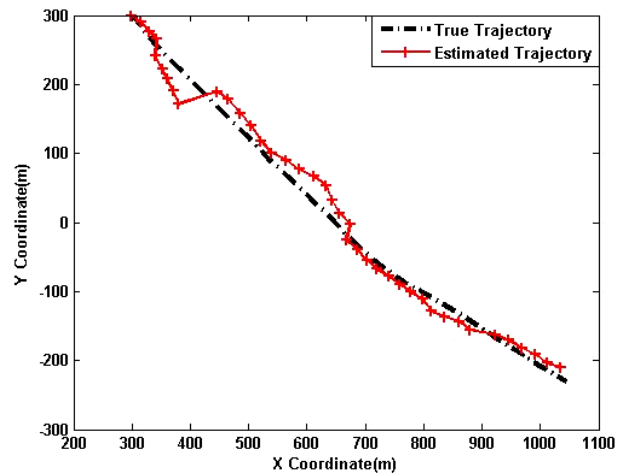

Figure 3. The true target trajectory against the estimated one by the proposed PF tracker.
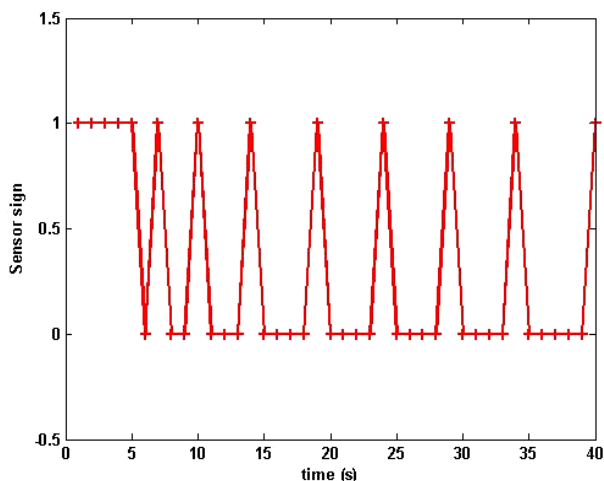
Figure 4. One instance of the sensor scheduling result: 0/1 denotes passive/active sensor being used.

Table 2. Performance evaluation with different δ values.

δ	The averaged number of time epochs when the active senor is used/ total number of time epochs	RTAMS (m)
5	13/40	8.08
10	11.5/40	9.06
50	7/40	9.11
100	6/40	19.66

sor switch process is actually a process of balancing the tracking filter performance with the concealment of the observer.

Next we evaluate the performance of the SS-PF tracker with respect to the value of δ. We use as index the root time averaged mean square (RTAMS) error defined as follows

$$\text{RTAMS} = \sqrt{\frac{1}{(t_{max} - l)M} \sum_{k=l+1}^{t_{max}} \sum_{i=1}^{M} \left(\overline{x}_{t,k}^i - x_{t,k}^i \right)^2 + \left(\overline{y}_{t,k}^i - y_{t,k}^i \right)^2}$$

where t_{max} is the total number of the time epochs for a single run. Here $t_{max} = 40$. l is the time index after which the averaging is carried out. Here $l = 0$. For each case with a specific δ value, $M = 50$ independent MC runs are done. We summarize the result in Table 2.

It is shown that, the proposed SS-PF method actually balances the tracking filter performance with the concealment of the observer, and such balance is controlled by the parameter δ.

6. Conclusions

In this paper, we address the problem of tracking a smart target. This problem requires that the observer conceal itself well, for that once it is detected by the smart target,
the latter may react in a manner that makes the future track more difficult. We analyze the relationship between the tracking filter performance and the observer's concealment. Based on such analysis, we propose a novel tracking method, in which a sensor scheduling technique, covariance control, is blended with an elaborately devised PF algorithm. Both theoretical analysis and simulation results demonstrate the efficiency of this method in dealing with the problem under consideration. It is shown that this method can balance the state filtering performance with the concealment of the observer well.

7. References

[1] C. Kreucher, D. Blatt, A. Hero, and K. Kastella, "Adaptive multi-modality sensor scheduling for detection and tracking of smart targets," Digital Signal Process, Vol. 16, pp. 546–567, 2006.

[2] C. Savage and B. L. Scala, "Sensor management for tracking smart targets," Digital Signal Process, doi:10.1016/j.dsp.2007.10.013 , 2007.

[3] J. C. Gittins and D. M. Roberts, "Search for an intelligent evader concealed in one of an arbitrary number of regions," Naval Research Logistics Quarterly, Vol. 26, No. 4, pp. 657–666, 1979.

[4] D. M. Roberts and J. C. Gittins, "Search for an intelligent evader: strategies for searcher and evader in the two-region problem," Naval Research Logistics Quarterly, Vol. 25, No.1, pp. 95–106, 1978.

[5] B. Liu, X. Ma, and C. Hou, "Smart target tracking using sensor scheduling and particle filter," in Proc. of Inter. Conf. on Signal Processing, Beijing, pp. 2620– 2623, 2008.

[6] N. Xiong and P. Svensson, "Multi-sensor management for information fusion: Issues and approaches," Information Fusion, Vol. 3, No. 2, pp. 163–186, 2002.

[7] B. Ristic, S. Arulampalam, and N. Gordon, Beyond the Kalman Filter: Particle Filters for Tracking Applications, Artech House, 2004.

[8] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonliear/non-gaussian bayesian tracking," IEEE Trans. on Signal Process, Vol. 50, No. 2, pp. 174–188, 2002.

[9] A. Doucet, N. De. Freitas, and N. Gordon, Sequential Monte Carlo in Practice, Springer Verlag, New York, 2001.

[10] M. Kalandros and L. Y. PAO, "Covariance control for multisensor systems," IEEE Trans. on Aerospace and Electronics Systems, Vol. 38, No. 4, pp. 1138–1157, 2002.

[11] M. Briers, S. Maskell, and R. Wright, "A rao-blackwellised unscented kalman filter," in Proc. of the 6th Int. Conf of Info. Fusion, Vol. 1, pp. 55–61, 2003.

[12] R. der Merwe, A. Doucet, N. Freitas, and E. Wan, "The unscented particle filter," Tech. Rep, Department of engineering, University of Cambridge, CB21PZ Cambridge, 2000.