A hybrid weighted interacting particle filter for multi-target tracking

David J. Ballatyne, Jarett Hailes, Michael A. Kouritzin,
Hongwei Long, and Jonathan H. Wiersma

Prediction in Interacting Systems,
Mathematics of Information Technology and Complex Systems,
Department of Mathematics at the University of Alberta,
Edmonton, Canada

ABSTRACT

A hybrid weighted interacting particle filter, the selectively resampling particle filter (SERP), is used to detect and track multiple ships maneuvering in a region of water. The ship trajectories exhibit nonlinear dynamics and interact in a nonlinear manner such that the ships do not collide. There is no prior knowledge on the number of ships in the region. The observations model a sensor tracking the ships from above the region, as in a low observable SAR or infrared problem. The SERP filter simulates particles to provide the approximated conditional distribution of the signal in the signal domain at a particular time, given the sequence of observations. After each observation, the hybrid filter uses selective resampling to move some particles with low weights to locations that have a higher likelihood of being correct, without resampling all particles or creating bias. Such a method is both easy to implement and highly computationally efficient. Quantitative results recording the capacity of the filter to determine the number of ships in the region and the location of each ship are presented. The hybrid filter is compared against an earlier particle filtering method.

Keywords: particle-based filtering, hybrid filter, multiple target, tracking, non-linear filtering, SERP

1. INTRODUCTION

Filtering theory is an active and current research field with wide applications to real world problems in areas such as financial markets, signal processing, communication networks, target detection and tracking, pollution tracking, and search and rescue. Signals are usually modeled and described by stochastic dynamical systems which cannot be observed directly and completely. We can only obtain some partial, distorted, and corrupted observations of their states. The purpose of filtering is to find probabilistic knowledge of the past path, current state, or future changes of the signals based on the back observations.

For signal tracking, most practical models apply to the single target scenario. But in reality, we may face many situations with multiple targets, for example, in surveillance, navigation, financial markets, communication networks, etc., in which it is critical to determine how many targets are located in a space domain based on imperfect observations. From a mathematical modeling point of view, we can describe the motions of multiple targets by a weakly interacting stochastic dynamical system. Thus, the signal can be regarded as a measure-valued Markov process. For a brief introduction to filtering for such a Markov process in a Polish space, we refer to Section 1.1 in Ballantyne, Chan and Kouritzin.[2]

Since our signal is often nonlinear, it is not possible to solve the filtering problems both practically and analytically. Recently, several adaptive particle system methods have been proposed and designed to provide

Further author information: (Send correspondence to J.H.W.)
D.J.B.: E-mail: dballant@math.ualberta.ca, Telephone: 1 780 492 4394
J.H.: E-mail: jhailes@math.ualberta.ca, Telephone: 1 780 492 0215
M.A.K: E-mail: mkouritz@math.ualberta.ca, Telephone: 1 780 492 2704
H.L.: E-mail:long@math.ualberta.ca, Telephone: 1 780 492 4394
J.H.W.: E-mail: jwiersma@math.ualberta.ca, Telephone: 1 780 492 0215
PINTS Website: http://www.math.ualberta.ca/pints

Signal Processing, Sensor Fusion, and Target Recognition XII, Ivan Kadar, Editor,
Proceedings of SPIE Vol. 5096 (2003) © 2003 SPIE · 0277-786X/03/$15.00

approximate solutions to nonlinear filtering problems. Del Moral and Salut[7] introduced the adaptive particle system approach to filtering, Kurtz and Xiong[10] described and analyzed the classical weighted particle method, the "cautious" branching particle method was introduced by Kouritzin and first discussed in Ballantyne, Chan and Kouritzin,[1] and the hybrid weighted particle filter method (SERP) was proposed by Kouritzin and first discussed in Ballantyne, Kim and Kouritzin.[3] Note that the classical weighted particle filter doesn't adapt its particle locations based upon observations. Thus, for this filter, most particle evolutions differ significantly from the real signal and the method assigns significant weights to a very few particles, which will cause the approximate filter to degenerate. The SERP filter overcomes these drawbacks and exhibits much better performance in simulations.

Here, we mention some existing works for multi-target tracking by Mahler and Zajic,[11] Orton and Fitzgerald,[12] Hue, Le Cadre and Pérez[9] and Ballantyne, Chan and Kouritzin.[2] Mahler and Zajic[11] were interested in finding the unknown number of targets in the region of state space rather than the exact location of each target, and used the *probability hypothesis density* (PHD) to design an approximate multi-target filter to estimate the expected number of targets. Orton and Fitzgerald[12] applied independent partition particle filter method for tracking the direction-of-arrival (DOA) of multiple moving targets. In the resampling step, particles with high weights are copied many times, and those with low weights only a few, or not at all. Hue, Le Cadre and Pérez[9] proposed the multi-target particle filter (MTPF), which combines the two major steps (prediction and weighting) of the classical particle filter with a Gibbs sampler-based estimation of the assignment probabilities. They also added two statistical tests to decide if a target has appeared or disappeared from the surveillance area.

In the aforementioned papers, it is assumed that targets are moving according to independent Markovian dynamics. In this paper, similarly to Ballantyne, Chan and Kouritzin,[2] we consider weakly interacting multiple targets (to avoid collisions and to attract distant targets) and regard the (random) unknown number of targets as part of our signal which enables us to handle cases in which the initial number of targets is unknown and that number may change over time. Also, our particle filter methods are different from those used in[12] and.[9] We discuss the implementation of branching particle filter and SERP filter methods for a multiple dinghy tracking problem and compare the computational efficiency of these two methods.

This paper is organized as follows: Section 2 describes in detail the two particle filter methods, MIBR and SERP, which will be used in our multiple dinghy tracking problem. In Section 3, the multiple dinghy tracking problem is described by considering the signal model, as given by different dinghy motion behaviors, and the observation model. In Section 4, the application of the filtering techniques to the problem is described in detail and simulation data is presented comparing the filter performance in terms of computational efficiency and error values. Lastly, in Section 5, conclusions are presented.

2. PARTICLE FILTERS

2.1. Background

For nonlinear filtering problems, it is usually impossible to find exact finitely recursive solutions. To obtain a computationally feasible solution, some kind of approximation is required. Particle filters are particularly appropriate since they can be applied to very general models and are easy to implement. The classical particle filter method (Monte-Carlo method) is composed of two stages: the particles are evolved according to the signal state equation during the prediction step; then, weights are calculated using the likelihood of the new observation combined with the previous weights. Unfortunately, it is well-known that the resulting particle scheme is not effective because the deviation of the particles may be too large and the growth of the weights is difficult to control.

In the past decade, a resampling step has been added to the classical scheme by adapting the particles to observations. Different types of resampling techniques have been developed, some examples of which are: the *bootstrap filter* by Gordon, Salmond and Smith,[8] the *adaptive interacting particle filter* by Del Moral and Salut,[7] the *adaptive branching particle filter* by Crisan and Lyons,[5] the *refining interacting particle filter* by Del Moral, Kouritzin and Miclo,[6] the *refining branching particle filter* by Ballantyne, Chan and Kouritzin,[2] and the *refining hybrid weighted interacting particle filter* by Ballantyne, Kim and Kouritzin.[3] The *adaptive, non-refining* particle filter methods introduce some unnecessary randomness in the resampling step which degrades performance, while the *refining* particle filter methods are designed to undertake a reduced amount of resampling which has been demonstrated to be superior. In this paper, the refining branching particle filter (MIBR) and refining hybrid weighted interacting particle filter (SERP) methods are applied to a multi-target tracking problem.

2.2. Branching particle filter

The MITACS-PINTS Branching Particle Filter, or MIBR Filter, was introduced by Fleischmann and Kouritzin and first published in Ballantyne, Chan and Kouritzin[1] (then known as the *refining branching particle filter*). This particle method selects particles for duplication or removal based on their Bayes update factor from the current observation.

2.2.1. Branching filter update algorithm

The MIBR filter algorithm proceeds as follows:

1. Initialize particles $\{\xi_0^1, \xi_0^2, \ldots, \xi_0^N\}$ by sampling from the initial signal distribution.
2. Repeat for $k = 1, 2, \ldots$
   (a) Simulate the evolution of the particles forward independently of each other over the time interval $t_k - t_{k-1}$. Call the new particles just prior to the next observation $\{\xi_{t_k-}^1, \xi_{t_k-}^2, \ldots, \xi_{t_k-}^N\}$.
   (b) Repeat for i = 1, 2,..., N
      i. At the k^{th} observation, evaluate $\zeta_k^i = \frac{\widehat{W}_k^i N}{\sum_{i=1}^N \widehat{W}_k^i} - 1$, where $\widehat{W}_k^i$ is the Bayes update factor for the i^{th} particle at time t_k.
      ii. If $\zeta_k^i < 0$, remove particle $\xi_{t_k-}^i$ with probability $-\zeta_k^i$.
      iii. Otherwise, if $\zeta_k^i > 0$, add $\lfloor \zeta_k^i \rfloor$ particles at the location $\xi_{t_{k-}}^i$. Then, add one more particle at this site with probability $\zeta_k^i - \lfloor \zeta_k^i \rfloor$.
   (c) Introduce unbiased particle control to bring the number of particles back to N.
   (d) Re-label the resulting particles $\{\xi_{t_k}^1, \xi_{t_k}^2, \ldots, \xi_{t_k}^N\}$.

The approximated conditional distribution for the signal state is calculated via

$$P(X_{t_k} \in A | Y_1, \ldots, Y_k) \approx \frac{1}{N} \sum_{i=1}^{N} 1_{\xi_{t_k}^i \in A}. \tag{1}$$

Note that for many problems, most of the particles are neither branched nor removed, since the value of ζ_k^i is often close to 0.

2.3. Selectively resampling particle filter

The selectively resampling particle (SERP) filter was first mentioned in Ballantyne, Kim and Kouritzin[3] (then known as the *weighted interacting* or *hybrid particle filter*), and further discussed in Bauer, Kim and Kouritzin.[4] In this method, particles are evolved as in the classical weighted method until such a time that the ratio of the largest particle weight to the smallest particle weight exceeds some factor ρ. Then, particles are resampled pairwise until the ratio of the weights falls within the value ρ.

2.3.1. SERP update algorithm

The SERP filter algorithm proceeds as follows:

1. Initialize particles $\{\xi_0^1, \xi_0^2, \ldots, \xi_0^N\}$ by sampling from the initial signal distribution. Set $W_0^i = 1\ \forall i$ (or to a value designed to compensate for a skew in the initial sampling of the particles; see Section 2.3.3).
2. Repeat for $k = 1, 2, \ldots$
   (a) Simulate the evolution of the particles forward independently of each other over the time interval $t_k - t_{k-1}$. Call the new particles just prior to the next observation $\{\xi_{t_k-}^1, \xi_{t_k-}^2, \ldots, \xi_{t_k-}^N\}$.
   (b) At the k^{th} observation, update the weight for all particles by the equation $W_k^i = \widehat{W}_k^i W_{k-1}^i$, where $\widehat{W}_k^i$ is the Bayes update factor for the i^{th} particle at time t_k.
   (c) Resample all particles until $W_k^{i_1} < \rho W_k^{i_2}$, $\forall i_1, i_2$, as described in Section 2.3.2.
   (d) Re-label the resulting particles $\{\xi_{t_k}^1, \xi_{t_k}^2, \ldots, \xi_{t_k}^N\}$.

The approximated conditional distribution of the signal state is calculated via

$$P(X_{t_k} \in A | Y_1, ..., Y_k) \approx \frac{\sum_{i=1}^N W_k^i 1_{\xi_{t_k}^i \in A}}{\sum_{i=1}^N W_k^i}. \tag{2}$$

There are numerous computational details that can be exploited to make this algorithm very efficient. The types of data structures used to store the particles and their weights can significantly affect the fidelity and computational performance of the SERP method. As well, the sum of the particle weights must be maintained within the minimum and maximum computer representation limits on the implementing architecture. Because of this, a particle re-weighting method is introduced after the re-labeling step to scale each particle weight by an identical factor should the sum of the weights overrun some predefined minimum or maximum value.

2.3.2. Resampling procedure

Particles are resampled in a pair-wise fashion in the SERP method. The highest particle weight, say that of particle i_1, and the lowest particle weight, say that of particle i_2, are compared to one another. If $W_k^{i_1} > \rho W_k^{i_2}$, then the particles are resampled. Both particles take the state value of particle ξ^{i_1} with probability $\frac{W_k^{i_1}}{W_k^{i_1}+W_k^{i_2}}$ or the state value of particle ξ^{i_2} with probability $\frac{W_k^{i_2}}{W_k^{i_1}+W_k^{i_2}}$. Afterwards, both $W_k^{i_1}$ and $W_k^{i_2}$ are set to the value $\frac{W_k^{i_1}+W_k^{i_2}}{2}$.

2.3.3. Particle initialization

For the particle methods in which particles retain nonuniform associated weights across observation steps, there is the potential to bias the initial particle sampling in a manner which is counterbalanced by including a compensating factor in the initial particle weight. For example, it may be useful to increase the number of particles that are initially deployed to search some portion of the signal domain. This technique is similar to importance sampling, except that it occurs during the particle initialization. If the particle filter resamples every particle at each observation, then many of the particles that are initialized with low weights will be immediately lost. In contrast, if the parameter ρ takes appropriate values, the SERP algorithm allows for particles to be initialized with weight less than one without the likelihood of immediate relocation through resampling.

2.3.4. Parameter explanation

The parameter ρ determines the amount of resampling that the SERP method performs. When $\rho = 1$, the SERP method is similar to the interacting particle method in that all particles are resampled at each simulation iteration. At the other extreme, when $\rho = \infty$, the SERP method becomes the classical weighted method, since no particles are resampled. This parameter allows for dynamic customization of the extent of resampling and may be a time-dependent value, or may depend on the state of the particle system. An empirical solution to the optimal value of ρ at time t is being investigated in the context of a stochastic control problem.

3. PROBLEM DEFINITION

3.1. Signal description

Suppose that (S_0, d_0) is some complete, separable metric space and that $\mathcal{B}$ is the Borel σ-field for S_0. We let S denote the space of finite, non-negative Borel measures μ on S_0, metrized by the extended Prohorov distance

$$d(\mu, \nu) \doteq \inf\{\varepsilon > 0 : \mu(B) \leq \nu(B^\varepsilon) + \varepsilon, \nu(B) \leq \mu(B^\varepsilon) + \varepsilon \ \forall B \in \mathcal{B}\}, \tag{3}$$

where $B^\varepsilon \doteq \{x \in S_0 : d_0(x, y) < \varepsilon \text{ for some } y \in B\}$. Then, (S, d) is also a complete, separable metric space.

Our signal measure is a homogeneous Markov process $\mathbb{X}$ on S constructed as

$$\mathbb{X}_t(B) = \sum_{j=1}^{M_t} \delta_{X_t^j}(B) = \text{number of targets in } B, \tag{4}$$

where M_t is the random number of targets depending on t. Since it is vital that $\mathbb{X}_t$ is a Markov process, we consider the case in which M_t is decreasing through the removal of targets to a cemetery state when they leave a bounded portion of the target domain, and we limit possible target interactions to weak interactions, meaning those that can be represented in terms of $\mathbb{X}_t$ itself. In this case, $\mathbb{X}$ is a $(P_0, \mathcal{L})$-Markov process, where P_0 is the initial signal measure distribution and the operator $\mathcal{L}$ describes the probabilistic evolution of the signal, including the target interactions. The weak interactions are designed to repel nearby targets in order to avoid collisions and to attract distant targets. This is enforced indirectly by increasing the probability that the orientation change of a target is away from other close targets, and increasing the probability that the orientation change of a target is towards other distant targets. Our model thereby reflects loose coordination of the targets.

3.1.1. Target models with interactions

We assume that there is a random number M_t of targets, where $M_t \in \{0, 1, 2, 3\}$. Each target has seven state variables: $(x_t^j,\ y_t^j)$ and $(\dot{x}_t^j,\ \dot{y}_t^j)$ denote the current planar location and velocity, respectively, of the j^{th} target; θ_t^j and $\dot{\theta}_t^j$ are the orientation and angular velocity; and χ_t^j denotes the current maneuver type from the possible set of {drifting, rowing, motoring}. The domain of (x^j, y^j) is taken to be the set $[0, R] \times [0, R]$, where $R = 192$, and any target which leaves this domain through its stochastic evolution is removed from the signal measure, decrementing M_t by one. We use the notation $X_t^j \doteq (x_t^j, y_t^j, \theta_t^j, \dot{x}_t^j, \dot{y}_t^j, \dot{\theta}_t^j, \chi_t^j)$ for the state of target j and $\mathbb{X}_t(\cdot) = \sum_{j=1}^{M_t} \delta_{X_t^j}(\cdot)$ for the signal measure. In order to write the stochastic equations for these variables we let $\{B^{j,a}, B^{j,b}, B^{j,c}\}_{j=1}^{M_t}$ be independent standard Brownian motions with dimensions three, two, and two. Also, we define the following attraction-repulsion field:

$$\kappa(s_1, s_2) = \begin{cases} 0 & s_1 = s_2 \\ \frac{\Pi(s_1,s_2)}{1000N} - \frac{3}{4(\Pi(s_1,s_2)-\varepsilon)} & s_1 \neq s_2 \end{cases}, \tag{5}$$

$$\kappa_x(s_1, s_2) = \kappa(s_1, s_2)\frac{\pi_x(s_1) - \pi_x(s_2)}{\Pi(s_1, s_2)}, \tag{6}$$

and

$$\kappa_y(s_1, s_2) = \kappa(s_1, s_2)\frac{\pi_y(s_1) - \pi_y(s_2)}{\Pi(s_1, s_2)} \tag{7}$$

for all ships $s_1, s_2 \in S_0$ (the domain of the targets X_t^j), where

$$\Pi(s_1, s_2) = \sqrt{|\pi_x(s_1) - \pi_x(s_2)|^2 + |\pi_y(s_1) - \pi_y(s_2)|^2} \tag{8}$$

is a pseudometric and π_x, π_y denote projection onto the x and y components. The effect of this field, when incorporated into the stochastic Itô equations that describe the motion of the individual targets, is to draw a target slightly towards the pack if it is distant and to repel it from any one other target to which it becomes too close. The value $\epsilon = 10$ represents the closest distance that two targets should come to each other. We define the following variables to enact this attraction-repulsion:

$$\phi_t^{r,j} = \sqrt{\int \left|\kappa(z, X_t^j)\right|^2 \mathbb{X}_t(dz)}, \tag{9}$$

and

$$\phi_t^{\theta,j} = \arctan(\int \kappa_y(z, X_t^j)\mathbb{X}_t(dz), \int \kappa_x(z, X_t^j)\mathbb{X}_t(dz)), \tag{10}$$

which represent the strength and orientation of the deflecting force, and both of which decompose into sums over the finite set of targets.

3.1.2. Maneuver type

The target state variable χ_t^j is a Markov chain with state space $\{1, 2, 3\}$, representing adrift, rowing, and motoring respectively. The rates are given by:

$$\lambda_{j,t}^{1\to 3}(X_t^j, \mathbb{X}_t) = \lambda_{j,t}^{2\to 3}(X_t^j, \mathbb{X}_t) = 100\phi_t^{r,j}, \tag{11}$$

$$\lambda_{j,t}^{1\to 2}(X_t^j, \mathbb{X}_t) = \lambda_{j,t}^{2\to 1}(X_t^j, \mathbb{X}_t) = \frac{3}{10}, \tag{12}$$

and

$$\lambda_{j,t}^{3\to 1}(X_t^j, \mathbb{X}_t) = \lambda_{j,t}^{3\to 2}(X_t^j, \mathbb{X}_t) = \frac{1}{200\phi_t^{r,j}}. \tag{13}$$

Since the attraction-repulsion field only affects the motion of a target if it is motorized, a strong field increases the likelihood that the target will switch into, or stay in, the motorized maneuver.

The differential equations describing the motion of target j depend on the maneuver type χ_t^j, as follows:

1. *Adrift* ($\chi_t^j = 1$) In this case,

$$d\begin{bmatrix}\dot{x}_t^j \\ \dot{y}_t^j \\ \dot{\theta}_t^j\end{bmatrix} = \begin{bmatrix}\mathcal{F}_{\dot{x}}(\dot{x}_t^j, \dot{y}_t^j, \theta_t^j) \\ \mathcal{F}_{\dot{y}}(\dot{x}_t^j, \dot{y}_t^j, \theta_t^j) \\ \mathcal{F}_{\dot{\theta}}(\dot{x}_t^j, \dot{y}_t^j, \dot{\theta}_t^j, \theta_t^j)\end{bmatrix} dt + \begin{bmatrix}1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.5\end{bmatrix} dB_t^{j,a}, \tag{14}$$

   where $B_t^{j,a}$ is a three-dimensional standard Brownian motion mentioned previously. To make the simulation more realistic, frictions $\mathcal{F}_{\dot{x}}$, $\mathcal{F}_{\dot{y}}$, and $\mathcal{F}_{\dot{\theta}}$ are included in the equation. The calculation of this friction is similar for each motion type and is described below in Equations (19 - 23).

2. *Rowing* ($\chi_t^j = 2$) In this case,

$$\begin{bmatrix}\dot{x}_t^j \\ \dot{y}_t^j\end{bmatrix} = \begin{bmatrix}\dot{f}_t^j \cos\theta_t^j \\ \dot{f}_t^j \sin\theta_t^j\end{bmatrix}, \tag{15}$$

   where $\dot{f}_t^j$ represents scalar velocity in the forward direction, with

$$d\begin{bmatrix}\dot{f}_t^j \\ \dot{\theta}_t^j\end{bmatrix} = \begin{bmatrix}(3.5 - \dot{f}_t^j + \mathcal{F}_{\dot{f}}(\dot{x}_t^j, \dot{y}_t^j, \theta_t^j)) \\ \mathcal{F}_{\dot{\theta}}(\dot{x}_t^j, \dot{y}_t^j, \dot{\theta}_t^j, \theta_t^j)\end{bmatrix} dt + \begin{bmatrix}\sqrt{(4 - \dot{f}_t^j)(\dot{f}_t^j - 3)} & 0 \\ 0 & 0.4\end{bmatrix} dB_t^{j,b}, \tag{16}$$

   where $B_t^{j,b}$ is a two-dimensional standard Brownian motion. This Itô equation is designed to maintain the speed of the ship between three and four pixels per time unit.

3. *Motorized* ($\chi_t^j = 3$) In this case,

$$\begin{bmatrix} \dot{x}_t^j \\ \dot{y}_t^j \end{bmatrix} = \begin{bmatrix} \dot{f}_t^j \cos\theta_t^j \\ \dot{f}_t^j \sin\theta_t^j \end{bmatrix}, \tag{17}$$

where $\dot{f}_t^j$ represents scalar velocity in the forward direction, with

$$d\begin{bmatrix} \dot{f}_t^j \\ \dot{\theta}_t^j \end{bmatrix} = \begin{bmatrix} (9.5 - \dot{f}_t^j + \mathcal{F}_{\dot{f}}(\dot{x}_t^j, \dot{y}_t^j, \theta_t^j)) \\ (\phi_t^{\theta,j} - \dot{\theta}_t^j)\phi_t^{r,j} + \mathcal{F}_{\dot{\theta}}(\dot{x}_t^j, \dot{y}_t^j, \dot{\theta}_t^j, \theta_t^j) \end{bmatrix} dt + \begin{bmatrix} \sqrt{(10 - \dot{f}_t^j)(\dot{f}_t^j - 9)} & 0 \\ 0 & 0.4 \end{bmatrix} dB_t^{j,c}, \tag{18}$$

where $B_t^{j,c}$ is a two-dimensional standard Brownian motion.

Friction is calculated according to the following model:

$$\mathcal{F}_{\dot{x}}(\dot{x}, \dot{y}, \theta) = \begin{cases} -\frac{\dot{x}\sqrt{\dot{x}^2+\dot{y}^2}}{\mathcal{F}_r(\dot{x},\dot{y},\theta)} \cdot f_\ell & \text{if } \dot{x} \neq 0, \dot{y} \neq 0 \\ 0 & \text{if } \dot{x} = \dot{y} = 0 \end{cases}, \tag{19}$$

$$\mathcal{F}_{\dot{y}}(\dot{x}, \dot{y}, \theta) = \begin{cases} -\frac{\dot{y}\sqrt{\dot{x}^2+\dot{y}^2}}{\mathcal{F}_r(\dot{x},\dot{y},\theta)} \cdot f_\ell & \text{if } \dot{x} \neq 0, \dot{y} \neq 0 \\ 0 & \text{if } \dot{x} = \dot{y} = 0 \end{cases}, \tag{20}$$

$$\mathcal{F}_r(\dot{x}, \dot{y}, \theta) = \sqrt{(\dot{x}\cos\theta + \dot{y}\sin\theta)^2 + \frac{1}{4}(\dot{y}\cos\theta - \dot{x}\sin\theta)^2}, \tag{21}$$

$$\mathcal{F}_{\dot{f}}(\dot{x}, \dot{y}, \theta) = \sqrt{\mathcal{F}_{\dot{x}}(\dot{x}, \dot{y}, \theta)^2 + \mathcal{F}_{\dot{y}}(\dot{x}, \dot{y}, \theta)^2}, \tag{22}$$

and

$$\mathcal{F}_{\dot{\theta}}(\dot{x}, \dot{y}, \dot{\theta}, \theta) = -\dot{\theta} f_\theta, \tag{23}$$

where f_ℓ and f_θ are constant parameters indicating the magnitude of the planar velocity and change-in-orientation frictions, and are equal to 0.25 and 2 respectively. These formulas have the effect of increasing the planar velocity friction as the direction of motion of the ship becomes more orthogonal to the orientation of the ship, and decreasing this friction as the velocity vector becomes more parallel to the orientation of the ship.

3.2. Observation model

The observation process consists of a sequence of images arriving at an interval of $t_{k+1} - t_k = 0.1$ time units at observation times $\{t_k\}_{k=1}^{\infty}$. Each image is a two-dimensional pixel raster $\{Y_k^{(l,m)}\}_{l,m=1}^{R}$ with height and width R pixels. The value $Y_k^{(l,m)}$ represents the (l, m) pixel in the two-dimensional raster of the observation at time t_k. The observations are constructed by

$$Y_k^{(l,m)} = h^{(l,m)}(\mathbb{X}_{t_k}) + V_k^{(l,m)}, \tag{24}$$

where $h^{(l,m)}(\mathbb{X}) = 1_{(l,m)\in A_{\mathbb{X}}}$ and $V_k^{(l,m)}$ is a zero-mean normal random variable with standard deviation σ_V. The set $A_{\mathbb{X}}$ consists of all points within the polygon representations of the target ships of the signal $\mathbb{X}$. The constituent target polygons of $A_{\mathbb{X}}$ are defined as follows: for each target ship X^j, $1 \leq j \leq M^{\mathbb{X}}$,

1. Place a box with sides of length 8 perpendicular to the raster grid and centered at point (x^j, y^j),
2. Add a triangle of height 4 to the right side of the box so that the base of the triangle is the side of the box,
3. Rotate the resulting polygon by the angle θ^j about (x^j, y^j).

3.3. Objective

The problem is to estimate the conditional distribution of the signal state based solely on the noisy observations and the initial signal distribution $\mathbb{X}_0$. That is, the objective is to calculate

$$P\left(\mathbb{X}_t \in A \;\middle|\; \sigma\left\{Y_k^{(l,m)}, 1 \leq l, m \leq R, k : t_k \leq t\right\}\right). \tag{25}$$

The observations are not preprocessed in any way; they are used directly by the filter.

4. SIMULATIONS

4.1. Simulation description

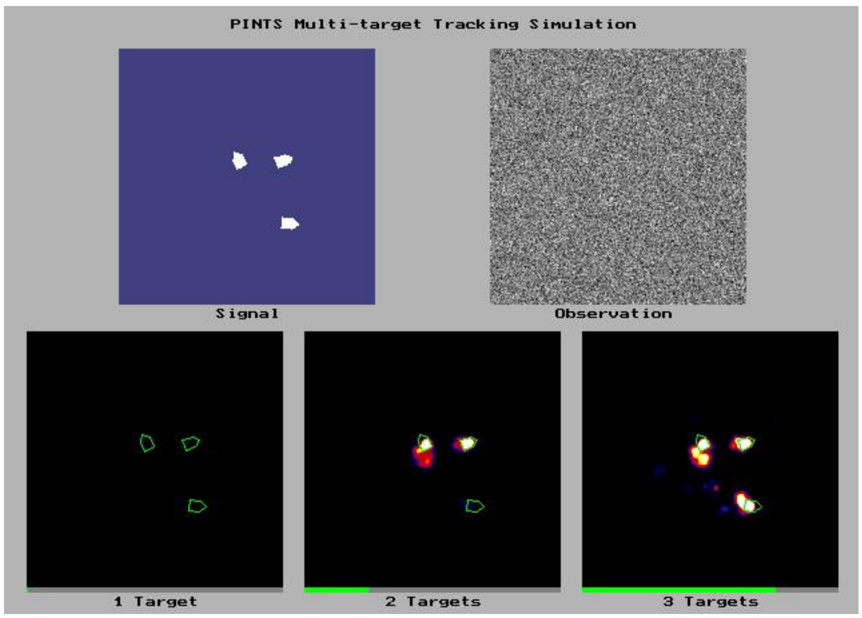

Figure 1. Three target simulation with the SERP filter

Both filters, MIBR and SERP, are run against 50 simulations of signal evolution and observation process. The data from the same observation process simulation is used by both filter types to construct approximate distributions and their error values.

Each simulation starts at time $t_0 = 0$ time units and progresses to $t_{50} = 5$ time units. The signal targets range over a positional domain $(x, y) \in ([0, 192], [0, 192])$ and have an initial distribution constructed as follows: a random number of targets is selected from among the possibilities zero, one, two, and three with uniform distribution, and then each such target is positioned at a random location in the target domain according to a two-dimensional Gaussian random variable, truncated to have support only within the target domain, with standard deviation 19.2 (the width of the x and y coordinates in the target domain divided by ten). If two targets are located less than the (x, y)-Euclidean distance of 15 pixels from each other then one is randomly repositioned. Each target is given a uniform random orientation. It has a one third probability to be exhibiting each motion type and the initial velocity is randomly determined based on this motion type. The initial change in orientation is zero for all targets.

Observations have width and height $R = 192$ pixels, are constructed at time intervals $t_{k+1} - t_k = 0.1$ time units, and the pixel-by-pixel independent noise standard deviation is $\sigma_V = 3$.

The filters use particles of the form $\xi_t^j = \mathbb{X}_t^j$ as their sample measures from the domain of signal measures. Each of the two types of filters uses $N = 400,000$ particles in its approximation. In the MIBR filter, the particles are initialized with uniform distribution over the possible number of targets, in accordance with the initial signal distribution. The SERP filter has the advantage of being able to retain a weight for particles that is not unity, and can evolve through observation updates without necessarily resampling particles with weights other than 1. For this reason SERP particles can be initialized with a distribution among target counts that differs from that of the signal, as described in Section 2.3.3, as long as this is balanced by multiplying the initial weights by a compensating factor. For these simulations, the SERP particles are distributed among zero, one, two, and three target cases with probabilities $\frac{1}{80{,}000}$, $\frac{999}{80{,}000}$, $\frac{13}{80}$, and $\frac{33}{40}$, providing many more particles to explore the more extensive portions of the signal domain corresponding to high target numbers. The constant, non-optimized value $\rho = 10^{24}$ is used in the SERP filter to determine the extent of resampling.

As depicted in Figure 1, the upper left image in our simulation represents the current state of the signal. The upper right image represents the observation created from the current signal state. The bottom three panels depict the filter's conditional distribution of the signal given that the signal has one, two, or three targets. Particles that contain zero targets are present in the filter, but do not display position information and are therefore omitted from the simulation image.

The scenario is simulated 50 times over the time frame $t = 5$ time units for each of the possible initial number of targets in the signal.

4.2. Comparison method

After every observation update the filters are compared in terms of an estimate error value. Normally, the error of a particle filter approximation would be measured with a MSE defined by:

$$\mathrm{MSE}(t_k) = \frac{1}{r_{\max}} \sum_{r=1}^{r_{\max}} \mathrm{d}(X_{t_k}^r, E[X_{t_k}^r | Y_1^r, \ldots, Y_k^r])^2, \tag{26}$$

where X^r, Y_k^r are the signal path and observations from run number r, $r_{\max}$ is the total number of simulation runs, and d is some distance function defined on the signal domain. However, in the case of multiple targets in which each particle $\mathbb{X}^j$ is a counting measure rather than a single point, and in which a distance between two counting measures or the mean of a set of counting measures which may contain differing number of points has no usual definition, no standard MSE calculation is possible and a different value for filter error must be defined.

Therefore, we define $\Upsilon_{t_k}(\left\{\mathbb{X}_{t_k}^j\right\}_{j=1}^N, \mathbb{X}_{t_k})$, the error of the filter measure with respect to signal truth, to be

$$\Upsilon_{t_k}(\left\{\mathbb{X}_{t_k}^j\right\}_{j=1}^N, \mathbb{X}_{t_k}) = \sum_{i=1}^N \frac{W_k^i}{\sum_{i'=1}^N W_k^{i'}} \mathrm{d}(\mathbb{X}_{t_k}, \mathbb{X}_{t_k}^i), \tag{27}$$

where for $\mathbb{X} = \frac{1}{M^{\mathbb{X}}} \sum_{j=1}^{M^{\mathbb{X}}} \delta_{X^j}$ and $\mathbb{Y} = \frac{1}{M^{\mathbb{Y}}} \sum_{j=1}^{M^{\mathbb{Y}}} \delta_{Y^j}$,

$$\mathrm{d}(\mathbb{X}, \mathbb{Y}) = \begin{cases} \min_{\sigma \in \mathrm{perm}\{1,\ldots,M^{\mathbb{Y}}\}} \sqrt{\sum_{j=1}^{M^{\mathbb{X}}} \left(\Pi(X^j, Y^{\sigma(j)})\right)^2 + (M^{\mathbb{Y}} - M^{\mathbb{X}})\, D^2} & M^{\mathbb{X}} \le M^{\mathbb{Y}} \\ \min_{\sigma \in \mathrm{perm}\{1,\ldots,M^{\mathbb{X}}\}} \sqrt{\sum_{j=1}^{M^{\mathbb{Y}}} \left(\Pi(X^{\sigma(j)}, Y^j)\right)^2 + (M^{\mathbb{X}} - M^{\mathbb{Y}})\, D^2} & M^{\mathbb{X}} > M^{\mathbb{Y}} \end{cases}. \tag{28}$$

Here $\Pi(X^{j_1}, Y^{j_2})$ is the distance between the (x, y) locations of the targets X^{j_1} and Y^{j_2} as defined by Equation (8), and D is the diagonal length of the target domain with respect to this distance function Π. For MIBR, $W_k^i = 1\ \forall i, k$. Comparisons of accuracy between the two filters will be made in terms of this value Υ_{t_k}.

4.3. Filter comparison

The values Υ_{t_k} have been averaged over 50 runs using signals with 0, 1, 2 and 3 targets and at times $t_k, 0 \le t_k \le 5$. Figure 2, Figure 3, Figure 4, and Figure 5 show these values for 0, 1, 2, and 3 target signals respectively with the SERP and MIBR filters.

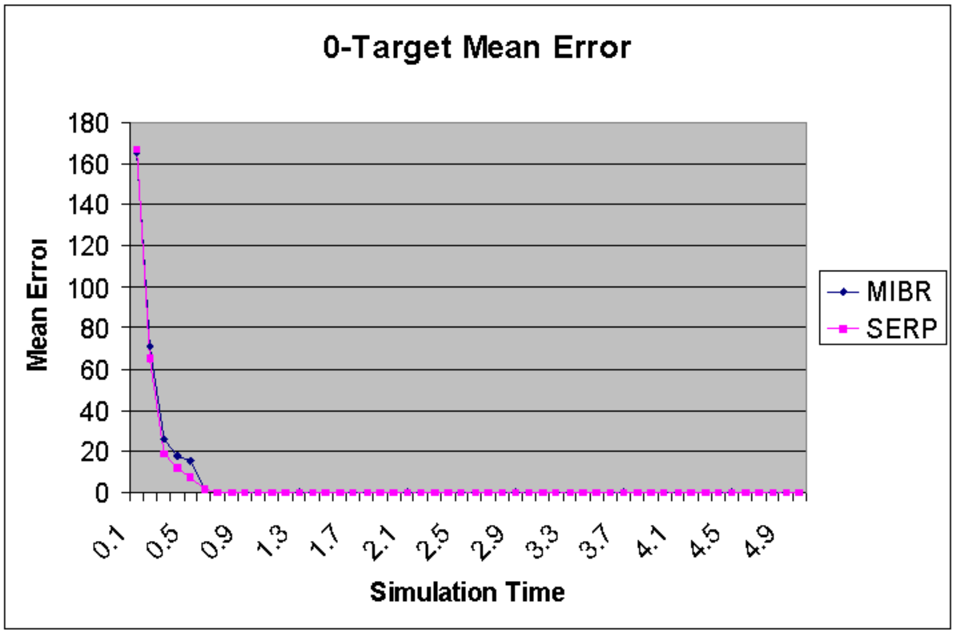

Figure 2. 0 Target Mean Error

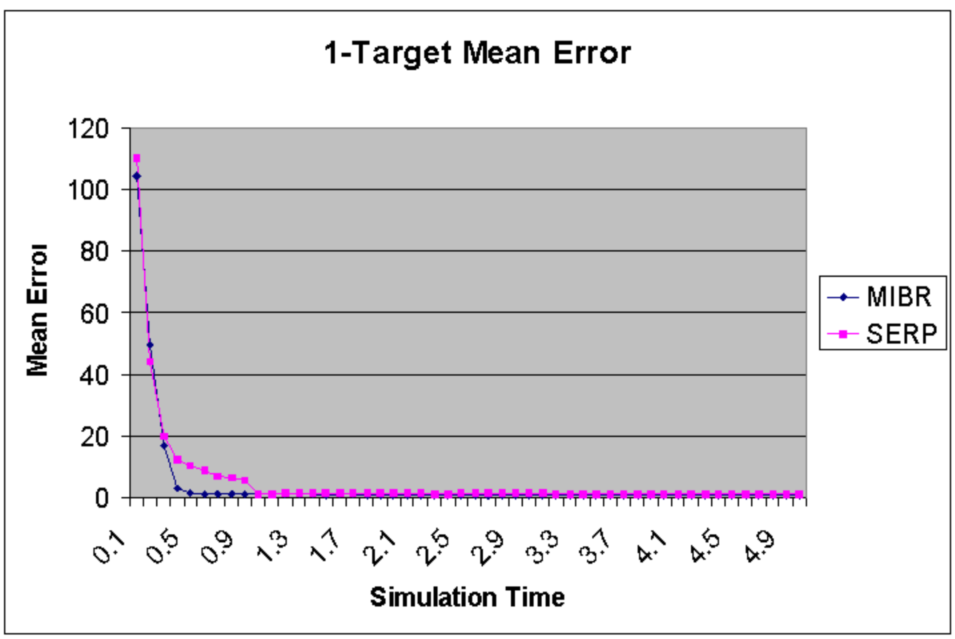

Figure 3. 1 Target Mean Error

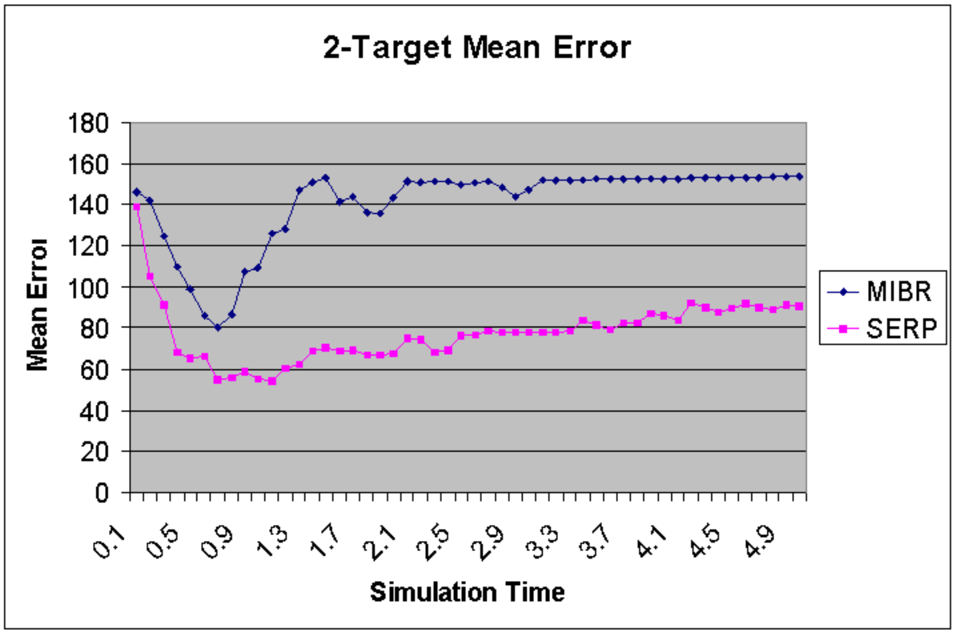

Figure 4. 2 Target Mean Error

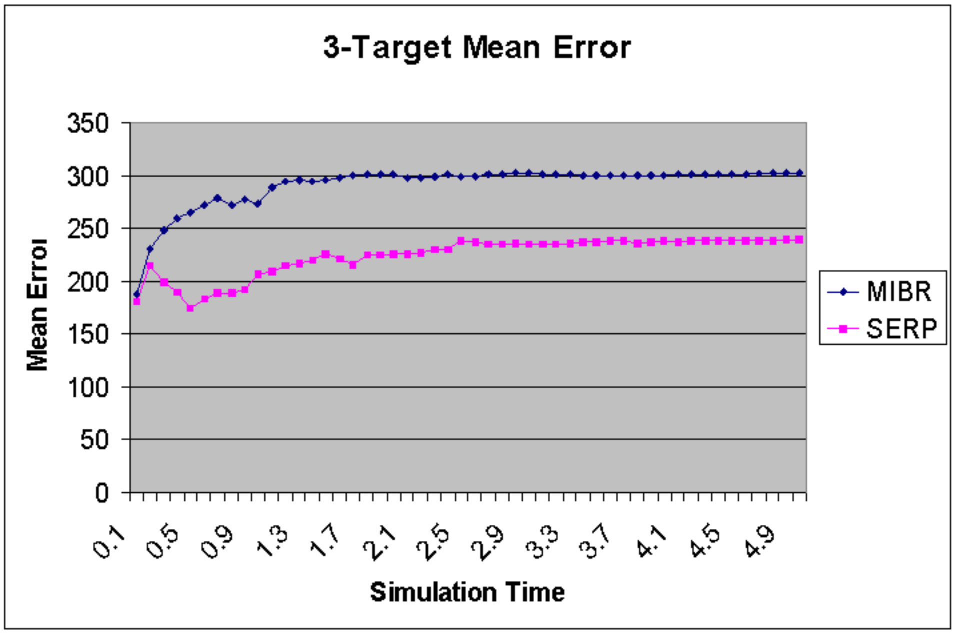

Figure 5. 3 Target Mean Error

The computation time for these simulations is not shown since processing was distributed across several architectures, making it difficult to obtain a useful benchmark. Generally, however, the MIBR method required more processing time, possibly because of extra duplications and deletions of particles as compared to SERP.

5. CONCLUSIONS

Except for early during the detection in the one target case, the SERP filter outperforms the MIBR filter under the given distance metric. The worse performance in the one target case is possibly due to the reduced initial particle count selected for the SERP filter, but this is compensated by the low error of SERP in the more important and difficult two and three target problems, in which SERP outperforms MIBR significantly. In the three target case, tracking two of the targets perfectly but not detecting the third would yield an error value of $\Upsilon = D = \sqrt{2} \cdot 192 \approx 271.5$. The SERP filter averages less than this error in this case, while MIBR averages worse than this value.

The selection of the ρ value for SERP will affect the filter performance. The SERP filter can have worse performance and could have better performance with other ρ values. With this ρ value, the SERP filter has better fidelity to the optimal filter and uses less processing time than the MIBR filter on this problem.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the support and sponsorship of Lockheed Martin Naval Electronics and Surveillance Systems, Lockheed Martin Canada, and the Natural Science and Engineering Research Council (NSERC) through the Prediction in Interacting Systems (PINTS) centre of the Mathematics of Information Technology and Complex Systems (MITACS) network of centres of excellence.

REFERENCES

[1] D. Ballantyne, H. Chan, and M. Kouritzin, "A novel branching particle method for tracking", in *Proceedings of SPIE, Signal and Data Processing of Small Targets 2000*, **4048**, pp. 277-287, 2000.

[2] D. Ballantyne, H. Chan, and M. Kouritzin, "A branching particle-based nonlinear filter for multitarget tracking", in *Proceedings of the 4th ISIF/IEEE International Conference on Information Fusion (Fusion 2001)*, Montreal, QC, August 2001.

[3] D. Ballantyne, S. Kim and M. Kouritzin, "A weighted interacting particle-based nonlinear filter", in *Signal Processing, Sensor Fusion and Target Recognition XI*, I. Kadar, ed., *Proceedings of SPIE*, **4729**, pp. 236-247, 2002.

[4] W. Bauer, S. Kim, M. Kouritzin, "Continuous and discrete space particle filters for predictions in acoustic positioning", in *Proceedings of SPIE-The International Society for Optical Engineering Image Reconstruction from Incomplete Data II*, **4792**, pp. 193-206, 2002.

[5] D. Crisan and T. Lyons, "Nonlinear filtering and measure-valued processes", *Probab. Theory. Relat. Fields* **109**, pp. 217-244, 1997.

[6] P. Del Moral, M.A. Kouritzin and L. Miclo, "On a class of discrete generation interacting particle systems", *Electronic J. Probab.* **6**, pp. 1-26, 2001.

[7] P. Del Moral and G. Salut, "Nonlinear filtering using Monte Carlo particle methods", *C.R. Acad. Sci. Paris* **320**, Serie I, pp. 1147-1152, 1995.

[8] N. Gordon, D. Salmond and A. Smith, "Novel approach to non-linear/non-Gaussian Bayesian state estimation", *Proc. Inst. Elect. Eng. F, Radar Signal Process.*, **vol 140**, no. 2, pp. 107-113, 1993.

[9] C. Hue, J.-P. Le Cadre and P. Pérez, "Sequential Monte Carlo methods for multiple target tracking and data fusion", *IEEE Transactions on Signal Processing*, **50**, pp. 309-325, 2002.

[10] T.G. Kurtz and J. Xiong, "Particle representation for a class of nonlinear SPDEs", *Stochastic Processes and their Applications*, **83**, pp. 103-126, 1999.

[11] R. Mahler and T. Zajic, "Bulk multitarget tracking using a first-order multitarget moment filter", in *Signal Processing, Sensor Fusion and Target Recognition XI*, I. Kadar, ed., *Proc. SPIE* **4729**, pp. 175-186, 2002.

[12] M. Orton and W. Fitzgerald, "A Bayesian approach to tracking multiple targets using sensor arrays and particle filters", Technical Report, Cambridge University Engineering Department, CUED/F-INFENG/TR. **403**, 2001.