A novel branching particle method for tracking

David J. Ballantyne[a], Hubert Y. Chan[a], and Michael A. Kouritzin[a]

[a]MITACS-PINTS,
Department of Mathematics at the University of Alberta,
Edmonton, Canada

ABSTRACT

Particle approximations are used to track a maneuvering signal given only a noisy, corrupted sequence of observations, as are encountered in target tracking and surveillance. The signal exhibits nonlinearities that preclude the optimal use of a Kalman filter. It obeys a stochastic differential equation (SDE) in a seven-dimensional state space, one dimension of which is a discrete maneuver type. The maneuver type switches as a Markov chain and each maneuver identifies a unique SDE for the propagation of the remaining six state parameters. Observations are constructed at discrete time intervals by projecting a polygon corresponding to the target state onto two dimensions and incorporating the noise.

A new branching particle filter is introduced and compared with two existing particle filters. The filters simulate a large number of independent particles, each of which moves with the stochastic law of the target. Particles are weighted, redistributed, or branched, depending on the method of filtering, based on their accordance with the current observation from the sequence. Each filter provides an approximated probability distribution of the target state given all back observations.

All three particle filters converge to the exact conditional distribution as the number of particles goes to infinity, but differ in how well they perform with a finite number of particles. Using the exactly known ground truth, the root-mean-squared (RMS) errors in target position of the estimated distributions from the three filters are compared. The relative tracking power of the filters is quantified for this target at varying sizes, particle counts, and levels of observation noise.

Keywords: target tracking, branching interacting particle system, nonlinear filtering

1. INTRODUCTION

1.1. Tracking Filters

We consider a single target tracking problem of the form

$$dX_t = A_{\chi_t}(X_t) + B_{\chi_t}(X_t)dB_t, \tag{1}$$

where X_t is the unobserved signal to track and χ_t is a finite-state Markov chain indicating the maneuver type, and

$$Y_{t_k} = H(X_{t_k}) + V_k, \tag{2}$$

where Y_{t_k} is a sequence of observations of the signal that are corrupted by noise given by V_k. Tracking filters are useful in a variety of problem areas, such as surveillance, aeronautics, and search-and-rescue. In the case in which A, B, and H are linear, χ_t is constant, and V_k is Gaussian, the conditional distribution can be efficiently computed by the Kalman filter. For our case, it is a assumed that A, B, and H are nonlinear and that there are no exact finite or infinite-dimensional filters applicable to the problem.

Exact filtering loosely refers to those filtering problems that degenerate into the evolution of finite-dimensional sufficient statistics or into FFT-based convolution with a known kernel. That is, a difficult, infinite dimensional

Further author information: (Send correspondence to D.J.B.)
D.J.B.: E-mail: dballant@math.ualberta.ca
H.Y.C.: E-mail: hubert@math.ualberta.ca
M.A.K.: E-mail: mkouritz@math.ualberta.ca
PINTS web page: http://www.math.ualberta.ca/pints

In *Signal and Data Processing of Small Targets 2000*, Oliver E. Drummond, Editor,
Proceedings of SPIE Vol. 4048 (2000) • 0277-786X/00/$15.00

equation degenerates into a known, readily implementable computer algorithm without need for approximation. For general nonlinear problems where there is no exact filter, the theoretical solution requires the use of Fokker-Plank density evolution combined with Bayes' rule,[1] which is computationally intractable when the dimension of the signal state space is large, and is often difficult even in small dimension spaces. In response, suboptimal methods such as the extended Kalman filter and interacting multiple models[2] have been developed.

1.2. Particle Filters

Another class of nonlinear filter techniques are the particle filters. A particle filter approximates the conditional distribution of the signal, given the observations, by a finite sum of Dirac densities. Each particle ξ^j represents a Dirac density in the space of the signal. For each new observation, all particles are evolved forward to account for the stochastic dynamics of the signal and then the set of particles is adjusted to account for the information from the observation. In this manner, the particles can function as an adaptive Monte-Carlo method for the filtering problem.

The set of particles then approximates the full data of the distribution of the signal conditioned on the set of all back observations. The approximated conditional probability that the signal lies within a given area is computed by dividing the number of particles in that area by the total number of particles.

Particle filters require an appropriate algorithm for the adjustment phase such that the filter provably converges to the exact conditional distribution as the number of particles approaches infinity. Numerous algorithms have been devised which fulfill these requirements. However, despite the theoretical convergence of each such filter, the particle filters have different performance characteristics in practical implementations with a finite number of particles. This paper compares the empirical efficacy of three particle filters applied to a simulated search-and-rescue tracking problem.

2. TRACKING PROBLEM

In this problem, we simulate a dinghy lost at sea that is observed from over the ocean surface by, for example, a helicopter using a digitized camera. The helicopter obtains a sequence of images of the ocean surface that are corrupted and distorted by spatial noise and sensor truncation effects. This noise is large enough that the position of the boat cannot be accurately estimated from a single image. However, knowledge of the stochastic law of the boat along with a sequence of observations over time enables a close tracking of the state of the target over a range of possible dinghy sizes. No image preprocessing is applied before the observations are provided to the filter algorithms.

2.1. Target Description

The lost dinghy has seven state parameters: $x_t, y_t, \theta_t, \dot{x}_t, \dot{y}_t, \dot{\theta}_t$, and χ_t. Parameters x_t and y_t indicate the x- and y-coordinates of the boat at a given time t, and θ_t indicates the orientation of the boat in the plane at time t. Parameters $\dot{x}_t, \dot{y}_t$ and $\dot{\theta}_t$ indicate the rate of change of x_t, y_t, and θ_t, respectively, at time t. The seventh component of the dinghy state is χ_t, a discrete variable that indicates the current maneuver type. We will use the notation $X_t = (x_t, y_t, \theta_t, \dot{x}_t, \dot{y}_t, \dot{\theta}_t)$. The stochastic behaviour of the dinghy is described by nonlinear systems whose parameters evolve with time according to a finite state-space Markov chain in which each possible maneuver type is a state. In this simulation, the Markov chain has three states representing the maneuvers adrift, rowing, and motoring. The dinghy switches from each state to the other two at equal rates, that is, it is equally likely to switch to either of the other states from the current one when a transition occurs. This transition rate is 0.15 per time unit for each alternate state. While rapid transitions are not usually the best strategy to use when lost, this switching does increase the difficulty of the tracking problem and highlights any differences between the three filters in their capacity to function well under such stress.

For each value of χ_t, the dinghy is simulated using Euler approximations according to the following formulae:

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

$$d \begin{bmatrix} \dot{x}_t \\ \dot{y}_t \\ \dot{\theta}_t \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.5 \end{bmatrix} dB_t^a + \begin{bmatrix} \mathcal{F}_{\dot{x}}(\dot{x}, \dot{y}, \dot{\theta}, \theta) \\ \mathcal{F}_{\dot{y}}(\dot{x}, \dot{y}, \dot{\theta}, \theta) \\ \mathcal{F}_{\dot{\theta}}(\dot{x}, \dot{y}, \dot{\theta}, \theta) \end{bmatrix} dt, \tag{3}$$

where B_t^a is three-dimensional standard Brownian motion. To make the simulation more realistic, friction $\mathcal{F}(\cdot, \cdot, \cdot, \cdot)$ is included in the equation. The calculation of this friction is identical for each motion type and is described below.

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

$$\begin{bmatrix} \dot{x}_t \\ \dot{y}_t \end{bmatrix} = \begin{bmatrix} \dot{f}_t \cos\theta_t \\ \dot{f}_t \sin\theta_t \end{bmatrix}, \tag{4}$$

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

$$d\begin{bmatrix} \dot{f}_t \\ \dot{\theta}_t \end{bmatrix} = \begin{bmatrix} (3.5 - \dot{f}_t) \\ 0 \end{bmatrix} dt + \begin{bmatrix} \sqrt{(4 - \dot{f}_t)(\dot{f}_t - 3)} & 0 \\ 0 & 0.4 \end{bmatrix} dB_t^b + \begin{bmatrix} \mathcal{F}_{\dot{f}}(\dot{x}_t, \dot{y}_t, \dot{\theta}_t, \theta_t) \\ \mathcal{F}_{\dot{\theta}}(\dot{x}_t, \dot{y}_t, \dot{\theta}_t, \theta_t) \end{bmatrix} dt, \tag{5}$$

where B_t^b is a two-dimensional standard Brownian motion. This simple Itô equation is designed to maintain the speed of the boat between three and four pixels per time unit. Of course, the dinghy could attain this speed while adrift also.

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

$$\begin{bmatrix} \dot{x}_t \\ \dot{y}_t \end{bmatrix} = \begin{bmatrix} \dot{f}_t \cos\theta_t \\ \dot{f}_t \sin\theta_t \end{bmatrix}, \tag{6}$$

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

$$d\begin{bmatrix} \dot{f}_t \\ \dot{\theta}_t \end{bmatrix} = \begin{bmatrix} (9.5 - \dot{f}_t) \\ 0 \end{bmatrix} dt + \begin{bmatrix} \sqrt{(10 - \dot{f}_t)(\dot{f}_t - 9)} & 0 \\ 0 & 0.4 \end{bmatrix} dB_t^c + \begin{bmatrix} \mathcal{F}_{\dot{f}}(\dot{x}_t, \dot{y}_t, \dot{\theta}_t, \theta_t) \\ \mathcal{F}_{\dot{\theta}}(\dot{x}_t, \dot{y}_t, \dot{\theta}_t, \theta_t) \end{bmatrix} dt, \tag{7}$$

where B_t^c is a two-dimensional standard Brownian motion.

Friction is calculated according to the following model:

$$\mathcal{F}_{\dot{x}}(\dot{x}, \dot{y}, \dot{\theta}, \theta) = \begin{cases} -\frac{\dot{x}\sqrt{\dot{x}^2+\dot{y}^2}}{\sqrt{(\dot{x}\cdot\cos\theta+\dot{y}\cdot\sin\theta)^2+\frac{1}{4}\cdot(\dot{y}\cdot\cos\theta-\dot{x}\cdot\sin\theta)^2}} \cdot f_\ell & \text{if } \dot{x} \neq 0, \dot{y} \neq 0 \\ 0 & \text{if } \dot{x} = \dot{y} = 0 \end{cases}, \tag{8}$$

$$\mathcal{F}_{\dot{y}}(\dot{x}, \dot{y}, \dot{\theta}, \theta) = \begin{cases} -\frac{\dot{y}\sqrt{\dot{x}^2+\dot{y}^2}}{\sqrt{(\dot{x}\cdot\cos\theta+\dot{y}\cdot\sin\theta)^2+\frac{1}{4}\cdot(\dot{y}\cdot\cos\theta-\dot{x}\cdot\sin\theta)^2}} \cdot f_\ell & \text{if } \dot{x} \neq 0, \dot{y} \neq 0 \\ 0 & \text{if } \dot{x} = \dot{y} = 0 \end{cases}, \tag{9}$$

$$\mathcal{F}_{\dot{f}}(\dot{x}, \dot{y}, \dot{\theta}, \theta) = \sqrt{\mathcal{F}_{\dot{x}}(\dot{x}, \dot{y}, \dot{\theta}, \theta)^2 + \mathcal{F}_{\dot{y}}(\dot{x}, \dot{y}, \dot{\theta}, \theta)^2}, \tag{10}$$

$$\text{and } \mathcal{F}_{\dot{\theta}}(\dot{x}, \dot{y}, \dot{\theta}, \theta) = -\dot{\theta} f_\theta, \tag{11}$$

where f_ℓ and f_θ are constant parameters indicating the magnitude of the planar velocity and change-in-orientation frictions, and equal to 0.6 and 0.5 respectively. These formulae have the effect of increasing the planar velocity friction when the dinghy velocity vector lies in a direction towards the sides of the boat, and decreasing this friction as the velocity vector is more aligned with or directly opposite the forward orientation of the boat. In simulation, the equations are approximated using Picard iterations to improve the accuracy. This method is much like iterated Newton or secant methods and involves recursively calculating a value and reusing it in the calculation until it converges to the extent that the change in each iteration is less than some accuracy threshold.

At the start of each simulation, the dinghy is positioned at a random location in the observation area. 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.

2.2. Observations

The observations consist of a discrete sequence Y_k of images, each of which is a two-dimensional raster of pixels in a 192 by 192 square. These images are constructed by superimposing a figure based on a projection of the dinghy state, X_{t_k}, onto the raster $R = \{(\ell, m)\}$ and adding noise by the formula

$$Y_k^{(\ell,m)} = h^{(\ell,m)}(X_{t_k}) + V_k^{(\ell,m)}, \tag{12}$$

where $V_k^{(\ell,m)}$ is pixel-by-pixel zero-mean independent Gaussian noise, the variance of which is a parameter to the simulation. Here,

$$h^{(\ell,m)}(X) = \begin{cases} 0 & \text{if } (\ell,m) \notin S_X \\ 10 \cdot (t_k - t_{k-1}) & \text{if } (\ell,m) \in S_X \end{cases}, \tag{13}$$

where S_X is the set of points contained in the polygon representation of the x, y, and θ dimensions of X. In particular, for a given size parameter s, S_X is formed as follows:

- Place a box with sides of length $2s$ perpendicular to the raster grid and centred at the point (x,y).
- Add a triangle of height s to the right side of the box so that the base of the triangle is the side of the box.
- Rotate the resulting polygon by the angle θ about (x,y).

The time period for the observations is set to a constant $(t_k - t_{k-1}) = 0.05$ time units. Each of the filter methods being compared is provided with the same noisy observation sequence for any one simulation. The observations are not preprocessed; the information from the raster pixels is used directly in the filter algorithms.

2.3. Objective

The problem is to estimate the conditional distribution of the dinghy state based on the observations, that is,

$$P(X_{t_k} \in dx, \chi_t = a \,|\, Y_i, 0 \le i \le k). \tag{14}$$

Note that the particle filters are asymptotically mean-square optimal in determining the maneuver type in addition to the position and the velocities. Since we are using approximated distributions, the intent is to provide the closest possible approximation to the optimal conditional distribution. The closeness of the approximation is meant in the loose sense that some combined root-mean-square (RMS) error between the approximated conditional mean and the signal, as well as other approximate conditional statistics to functions of the signal, is small. In this case, the optimal filter is effectively intractable, so the three particle methods are compared by calculating the mean squared error between each approximated conditional mean and the actual simulated dinghy position.

3. FILTER TECHNIQUES

To our knowledge, particle methods were first introduced by Del Moral and Salut.[3] We use an interacting particle method developed by Del Moral and Miklo,[4] a weighted particle method discussed in the work of Kurtz and Xiong,[5] and a novel branching particle method introduced by Kouritzin and developed by Kouritzin and Blount.[6]

All three methods are initialized with N particles $\{\xi_0^j\}_{j=1}^N$ uniformly distributed in the domain of X. At each observation, the three methods progress through the following stages: evolution of the particles, particle selection or weighting, and the approximation of the conditional distribution of the dinghy state.

3.1. Evolution

In all three methods the evolution stage is the same. Each of the particles is evolved independently for the time period between observations ($t_k - t_{k-1} = 0.05$ time units) according to the stochastic differential equation and Markov chain transitions of the dinghy, as described in Equations (3) to (11): $\xi_{t_{k-1}}^j \to \xi_{t_k}^j$.

3.2. Particle Adjustment

3.2.1. Interacting particle method[4]

By Bayes rule,

$$p_{X_{t_k}|Y_1,\dots,Y_k}(x|Y_1,\dots,Y_k) = \frac{p_{X_{t_k}|Y_1,\dots,Y_{k-1}}(x) \cdot p_{Y_k|X_{t_k}}(Y_k|x)}{\int p_{X_{t_k}|Y_1,\dots,Y_{k-1}}(\xi) \cdot p_{Y_k|X_{t_k}}(Y_k|\xi)d\xi}. \tag{15}$$

Given the particle approximation for $p_{X_{t_{k-1}}|Y_1,\dots,Y_{k-1}}(x)$ from the previous step, the interacting method replaces the factor $p_{X_{t_k}|Y_1,\dots,Y_{k-1}}(x)$ in approximation by the evolution of the particles $\{\xi^j\}_{j=1}^N$. Given observation Y_k consisting of values assigned to the points (ℓ, m) in the raster R, weights W_k^j are then assigned to each particle by

$$W_k^j = \frac{p_{Y_k|X_{t_k}}(Y_k|\xi_k^j)}{\sum_{i=1}^N p_{Y_k|X_{t_k}}(Y_k|\xi_k^i)} = \frac{\prod_{(\ell,m)\in R} f(Y_k^{(\ell,m)}, h^{(\ell,m)}(\xi_k^j))}{\sum_{i=1}^N p_{Y_k|X_{t_k}}(Y_k|\xi_k^i)}, \tag{16}$$

where $f(y, x)$ is the probability of obtaining an observation pixel intensity of y at a raster point given that the value of $h^{(\ell,m)}$ for the pixel (ℓ, m) and for that particle is x, that is, given that the pixel is either inside or outside of the polygon representation of the state of that particle. This probability is also determined by the noise distribution V_k as described in Sect. 2.2.

The particles ξ^j are then redistributed among the previous particle sites $\xi_{t_k^-}^j$ according to the multinomial distribution

$$P(n_1 \text{ particles at } \xi_k^1, \dots, n_N \text{ particles at } \xi_k^N) = \binom{N}{n_1 \cdots n_N} (W_k^1)^{n_1} \cdots (W_k^N)^{n_N}. \tag{17}$$

This is equivalent to each particle $\xi_{t_k}^i$ being relocated to site $\xi_{t_k^-}^j$ at time t_k with probability W_k^j independent of all other particle relocations.

3.2.2. Weighted particle method[5]

This method approximates the estimated probability density with a set of N particles ξ_t^j with associated time-varying weights M_t^j. The particle weights are governed by the stochastic equation

$$M_t^j = M_0^j + \int_{[0,t]} \sum_{(\ell,m)\in R} M_s^j \, h^{(\ell,m)}(\xi_s^j) \, \tilde{Y}^{(\ell,m)}(ds), \tag{18}$$

where $\tilde{Y}$ is the running accumulation of continuous observation data. Since the observations in this case occur at discrete times, we approximate the integral as a sum:

$$\begin{aligned} M_{t_k}^j &\approx M_0^j + \sum_{i=1}^{k} \sum_{(\ell,m)\in R} M_{t_i^-}^j \, h^{(\ell,m)}(\xi_{t_i^-}^j) \left(Y_{t_i}^{(\ell,m)} - \tilde{Y}_{t_i^-}^{(\ell,m)} \right) \\ &\approx M_{t_{k-1}}^j + \sum_{(\ell,m)\in R} M_{t_k-}^j \, h^{(\ell,m)}(\xi_{t_k^-}^j) \left(\tilde{Y}_{t_k}^{(\ell,m)} - \tilde{Y}_{t_k^-}^{(\ell,m)} \right) \\ &= M_{t_{k-1}}^j + M_{t_k^-}^j \sum_{(\ell,m)\in R} h^{(\ell,m)}(\xi_{t-}^j) \left(\tilde{Y}_{t_k}^{(\ell,m)} - \tilde{Y}_{t_k^-}^{(\ell,m)} \right) \end{aligned} \tag{19}$$

for $k \geq 1$, and use the simulation raster $Y_k = (\tilde{Y}_{t_k} - \tilde{Y}_{t_k^-})$ as the observation at time t_k as in the interacting case.

Initially $M_0^j = 1$, $1 \leq j \leq N$. The particles ξ^j are evolved independently for the time period of each observation, as for the interacting method, and the weights are then recalculated according to the above formula to account for each observation. These weights are retained to be used in the approximation of the conditional distribution and for the next time step when the subsequent observation arrives.

3.2.3. Branching particle method[6]

For this method, a varying number n_{t_k} of unweighted particles $\{\xi_{t_k}^j\}_{j=1}^{n_{t_k}}$ are used (with $n_0 = N$), and the particles are duplicated or removed based on their agreement with each observation. Given the observation, the evolved particles are each assigned a value ζ_k^j according to a calculation on Y_k and $h(\xi_{t_k}^j)$. A method is used to control the total number of current particles n_{t_k} to be close to the initial number of particles N. Then, a uniform-(0,1) random variable U_k^j is determined for each particle and

- if $(\zeta_k^j \geq U_k^j)$, a new particle $\xi_{t_k}^* = \xi_{t_k}^j$ is added,

- if $(\zeta_k^j \leq -U_k^j)$, $\xi_{t_k}^j$ is removed,
- in the most frequent case where $|\zeta_k^j| < U_k^j$, the particle is not branched and and is left in the state that it evolved to,

with the attendant renumbering, and the setting of n_k to equal the total number of current particles, after the entire process.

3.3. Estimation

By independence and the law of large numbers,

$$\frac{1}{N}\sum_{j=1}^{N}\delta_{\xi_{t_k}^j}(A) \overset{N\to\infty}{\longrightarrow} \int_A \rho^k(x)dx, \tag{20}$$

where $\rho^k(x)$ is the conditional probability density of the state of the target dinghy at time t_k given observations $Y_1, \dots, Y_k$.

For the weighted particle filter the estimation of the density is approximated according to

$$\frac{1}{\sum_{j=1}^{N} M_{t_k}^j}\sum_{j=1}^{N} M_{t_k}^j \delta_{\xi_{t_k}^j}(A) \overset{N\to\infty}{\longrightarrow} \int_A \rho^k(x)dx. \tag{21}$$

A graphical presentation of the simulations has been constructed, as depicted in Fig. 1. This is one frame of an animation sequence that follows the observation sequence. The top left panel indicates the simulated dinghy position. The upper middle panel is the observation raster, as it is presented to the filter algorithms. The bottom three panels display the current estimates of the three particle-based filter methods, the interacting method at the left, the weighted method in the middle, and the branching method at the right. In each filter panel, the box indicates the filter extimate of the dinghy state, with the centre of the box at the estimated (x, y) location and the size of the box indicating the variance in the location distribution. The arrow points in the direction of the estimated dinghy orientation θ, with a longer arrow indicating a more certain estimate.

3.4. Computation Simplifications

A number of simplifications of the calculations have been implemented in the filters such that the computational complexity is directly proportional to the number of particles and is also proportional to the pixel area of the dinghy polygon. Specifically, simplifications have been exploited so that major calculations on the entire raster are only performed once for each observation, and then each subsequent particle calculation references only those raster pixels that are within the target polygon of that particle.

For example, in the interacting method, $\prod_{(\ell,m)\in R} f(Y_k^{(\ell,m)}, h^{(\ell,m)}(\xi_k^j))$ can be calculated as

$$\prod_{(\ell,m)\in R} f(Y_k^{(\ell,m)}, \check{h}) \cdot \frac{\prod_{(\ell,m)\in S_{\xi_k^j}} f(Y_k^{(\ell,m)}, h^{(\ell,m)}(\xi_k^j))}{\prod_{(\ell,m)\in S} f(Y_k^{(\ell,m)}, \check{h})}, \tag{22}$$

where $\check{h}$ is the value of $h^{(\ell,m)}(\xi)$ when the point (ℓ, m) is not contained in the polygon representation S_ξ of the particle ξ. The product over all raster points is then calculated only once when the observation first arrives, or in the interacting case, not at all since the term cancels out in all later relative weight computations. Each particle ξ then need only compute over the observation values of those pixels within the associated polygon representation, S_ξ.

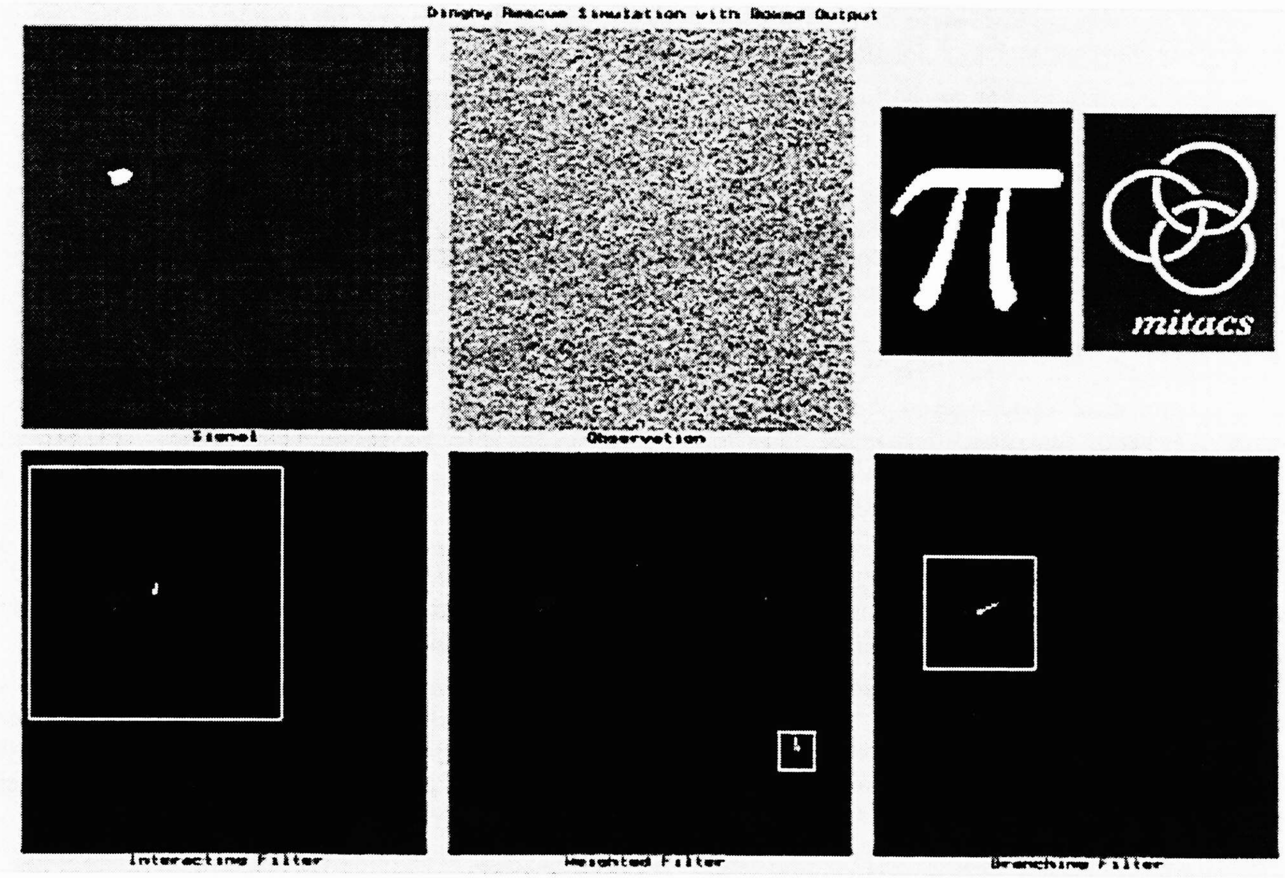

Figure 1. Sample frame from a dinghy problem animation.

4. FILTER COMPARISONS

Three dinghy tracking scenarios are simulated corresponding to three differently sized dinghy targets — a large target with $s = 5$ and an average dinghy area of 125 pixels, a medium target with $s = 3$ and an average area of 45 pixels, and a small target with $s = 1$ and an average area of 5 pixels. The s parameter is as described in Sect. 2.2. Using the situation of 125, 45, or 5 independent, noisy observations as a reference, one would estimate that the difficulty of filtering would decrease with the square of the number of these observations, that is, in the ratio $\sqrt{125} : \sqrt{45} : \sqrt{5} = 5 : 3 : 1$. While this is not the exact case here, since the filter does not know *a priori* which of the 125, 45, or 5 pixels are associated with the signal, it suffices as an approximation. Based on this, the standard deviation of the noise $V^{(\ell,m)}$, on a pixel-by-pixel basis, is set to $60/\sqrt{t_k - t_{k-1}}$, $36/\sqrt{t_k - t_{k-1}}$, and $12/\sqrt{t_k - t_{k-1}}$ in the large, medium, and small target cases. Using a signal strength of $10 \cdot (t_k - t_{k-1})$ as described in Sect 2.2, these correspond to SNR of -28.57dB, -24.14dB, and -14.59dB respectively.

Because of the computation simplifications, the algorithms execute more quickly when the assumed target size is small. This is accounted for in the simulations by allowing more particles to the smaller target cases so that the execution times are approximately equivalent. The number of particles used is $N = 10000$ for the large target, $N = 17500$ for the medium target, and $N = 30000$ for the small target.

The weighted and branching filters execute approximately three times faster than the interacting filter in our code using the same number of particles. However, the following results were obtained using the identical initial number of particles, as given above, for each method. Given the code timings and the direct proportionality of the algorithmic complexity of the filters with the number of particles, it would be fair to provide the weighted and branching filters with approximately three times the number of particles, which would tend to decrease their mean-squared errors. Though the data do not include such an adjustment, the conclusions regarding the branching method would only be strengthened if these adjustments were made.

Graphs of the average RMS error in the position estimate of the three filters over the simulation time are provided for the large target case in Fig. 2, for the medium target case in Fig. 3, and for the small target case in Fig. 4.

Note that the dinghy has an initial position that is randomly and uniformly selected from the area of the observation raster, and that all of the particles have similarly uniformly distributed initial states. Because of this, each of the particle filters begins with an estimated position of the dinghy in the centre of the observation area, and thus there is an average RMS error at the start of the simulation that all of the filters will share. However, it is possible that during the course of the simulation a poor filter will, on average, return an average RMS error in conditional mean dinghy position which is larger than this initial error. Because of this, some filters will seem to perform worse over time as they approach their long-term behaviour.

5. CONCLUSIONS

There are a number of advantages to particle-based nonlinear filters. The filters have demonstrated effectiveness and efficiency at tracking problems. It is simple to scale the power of the filter to the power of the available processor by increasing the number of particles. As well, further computational gains are easily obtained by distributing the required processing power among multiple processors.

However, there are considerable differences evident in the practical power of the various particle filter types. For this tracking problem, the weighted method has considerable difficulty, especially with smaller targets. The inability to adapt the particle placement to optimally cluster particles near suspected dinghy locations renders the weighted filter ineffective without a very large number of particles relative to the observation area.

The branching method exhibits superior performance in all three scenarios of this simulation. By neither duplicating nor removing most particles, but rather leaving them as they are, the branching method gains a speed advantage and also is more cautious in its particle adjustment. It is less likely, in its approximation of the distribution, to immediately cluster all of the particles near to suspected target positions, and thus, with a practical number of particles, is less likely to make erroneous particle adjustments that hamper the long-term adaption.

ACKNOWLEDGMENTS

The authors greatfully acknowledge the support and sponsorship of Lockheed Martin Naval Electronics and Surveillance Systems, Lockheed Martin Canada, VisionSmart of Edmonton, the Pacific Institute for the Mathematical Sciences, 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, and the University of Alberta.

REFERENCES

1. A. H. Jazwinski, *Stochastic Processes and Filtering Theory*, Academic Press, New York, 1970.
2. H. A. P. Blom and Y. Bar-Shalom, "The interacting multiple model for systems with Markovian switching coefficients," *IEEE Trans. Auto. Cont.* **33**, pp. 780–783, 1988.
3. P. Del Moral and G. Salut, "Non-linear filtering using Monte Carlo particle methods," *C.R. Acad. Sci. Paris* **320, Série I**, pp. 1147–1152, 1995.
4. P. Del Moral and L. Miclo, "Branching and interacting particle systems approximations of Feynman-Kac formulae with applications to non-linear filtering," in *Séminaire de Probabilités XXXIV*, J. Azema, M. Emery, M. Ledoux, and M. Yor, eds., *Lecture Notes in Mathematics* **1729**, pp. 1–145, 2000.
5. T. G. Kurtz and J. Xiong, "Particle representations for a class of nonlinear SPDEs," *Stochastic Processes and their Applications* **83**, pp. 103–126, 1999.
6. M. A. Kouritzin and D. Blount, "Consistency for the historical process of a novel branching particle filter." Preprint.

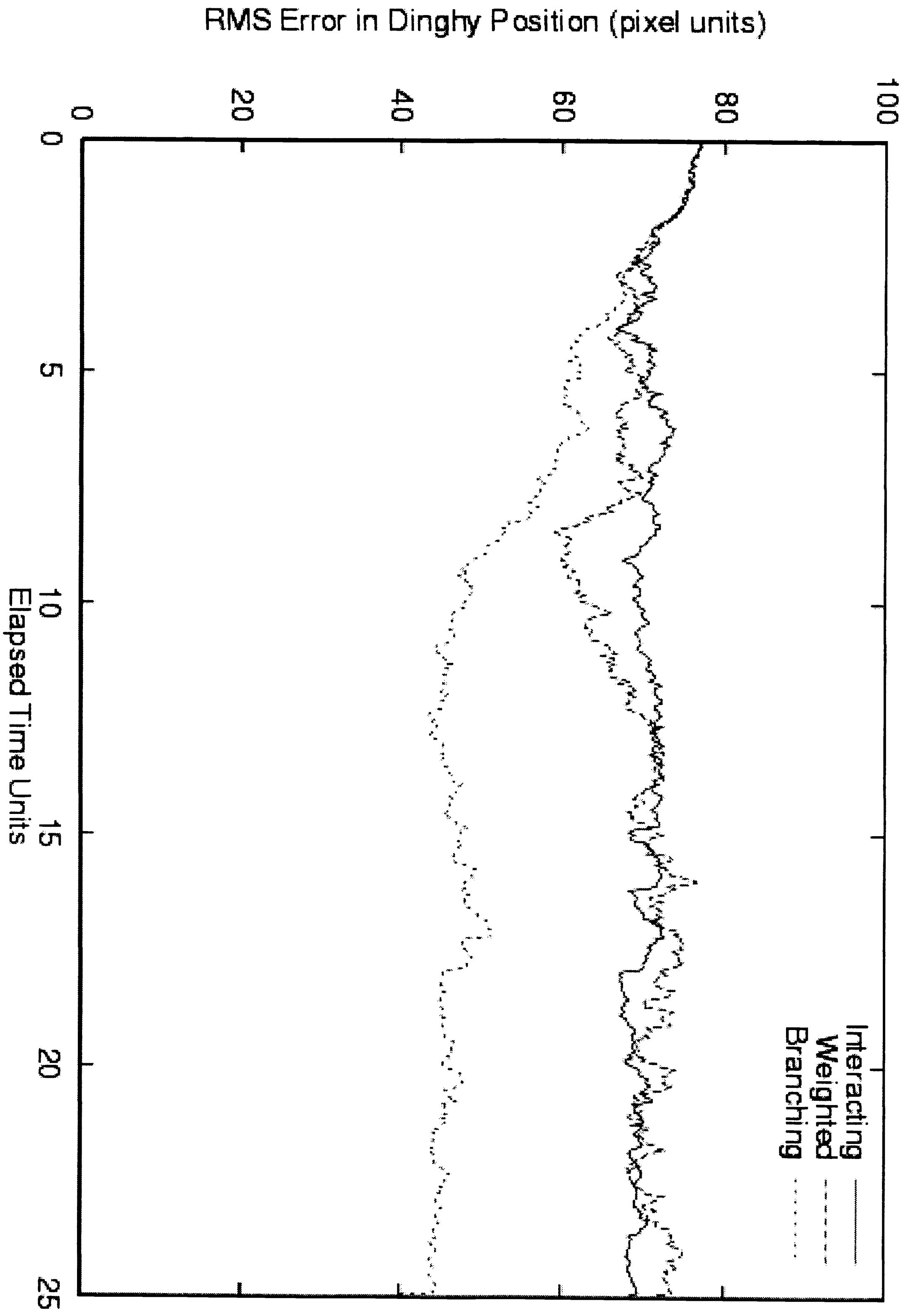

Figure 2. Error in dinghy position estimate for the LARGE dinghy scenario.

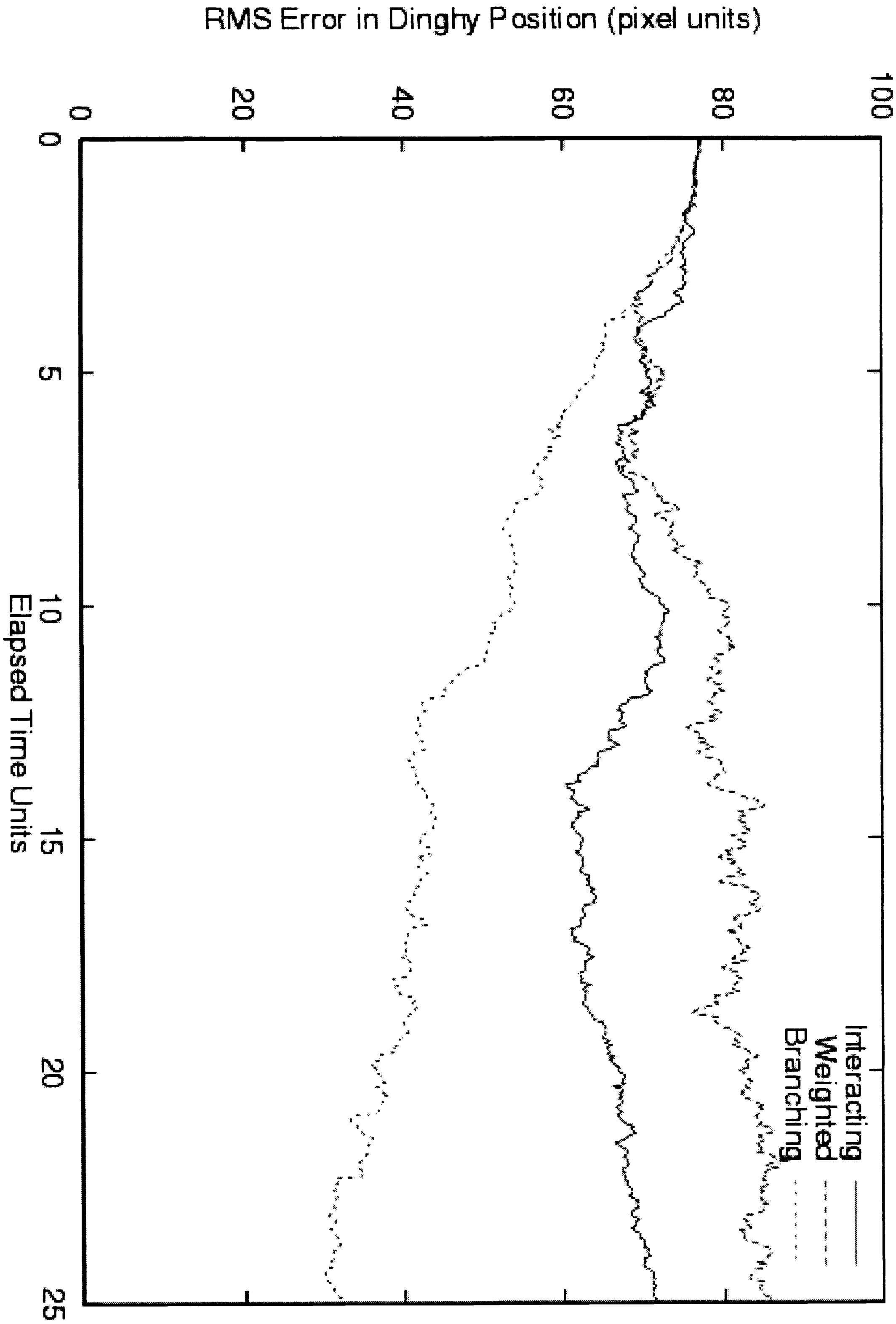

Figure 3. Error in dinghy position estimate for the MEDIUM dinghy scenario.

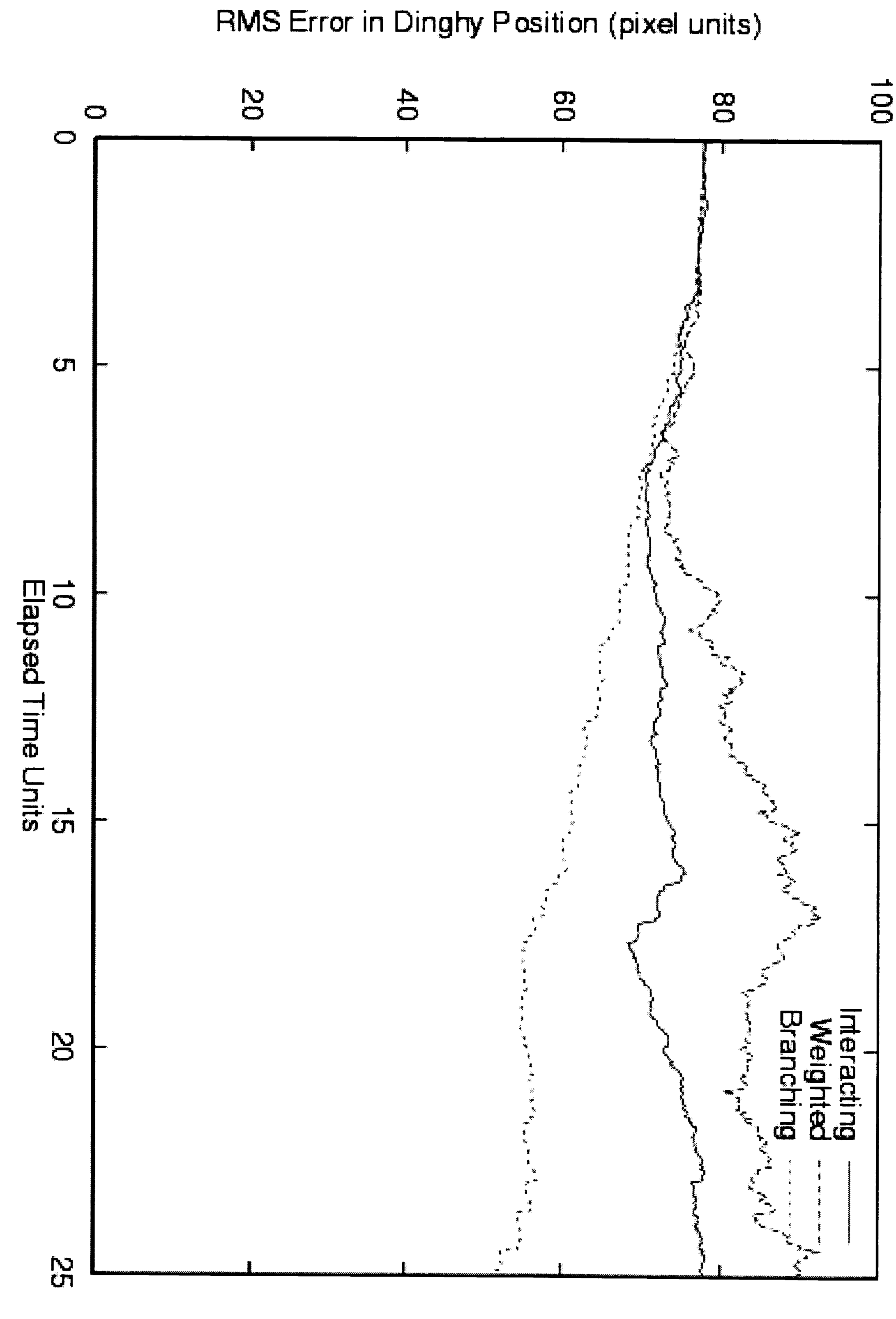

Figure 4. Error in dinghy position estimate for the SMALL dinghy scenario.