Repeated Filtering for Smoothing Particle Filters

STEPHEN L. ANDERSON
LAWRENCE D. STONE
SIMON MASKELL

This paper presents the repeated filtering method for finding a smoothed, Bayesian estimate of the path of a stochastic process over a time interval [0, T] when one has used a particle filter to estimate the state of the process. It provides good resolution over [0, T], is easy to implement, and can be used with any sequential importance resampling particle filter regardless of the probabilistic model employed by the stochastic process. Repeated filtering is general, powerful, and simple. It does not require the restrictive assumptions or complex calculations of other methods. It is suitable for real-time operational use in complex situations. We demonstrate the method on two single-target tracking examples. The second of these tracking examples is very difficult to solve by any other method known to us. We then apply repeated filtering to a standard nonlinear time series model that has been used extensively for testing numerical filtering techniques. To further illustrate the power of repeated filtering, we show how adding reflecting boundaries to this time series creates a process that is difficult to smooth with existing techniques but simple with repeated filtering.

Manuscript received October 6, 2022; revised March 22, 2023; released for publication May 29, 2023.

Associate Editor: Florian Meyer.

S. L. Anderson and L. D. Stone are with Metron, Reston VA, USA (e-mail: anderson@metsci.com; stone@metsci.com).
S. Maskell is with the Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool, UK (e-mail: s.maskell@liverpool.ac.uk).

1557-6418/23/$17.00

I. INTRODUCTION

Particle filters are powerful and general tools for performing nonlinear, non-Gaussian filtering. For target tracking, they provide an estimate of the distribution on target state at the time of the last measurement. However, it is often desirable to compute the posterior distribution on the target's path over an interval of time [0, T] given the measurements received in that interval. The process of computing this distribution is called fixed interval smoothing.

We present the repeated filtering method for smoothing in the context of surveillance and tracking, but the method is applicable to very general situations where one can use a sequential importance resampling (SIR) particle filter to estimate the history of the state of a stochastic system. To illustrate this, we apply the repeated filtering method to smooth the nonlinear time series analyzed in Example 1 of [5]. According to [5], this series has been used extensively for testing numerical filtering techniques. In the final example, we add reflecting boundaries to this time series. We show in Section IV-D that this process is difficult to smooth using the methods of [5], but simple with repeated filtering.

Repeated filtering is conceptually simple. Once one has developed a particle filter for the problem of interest, they have done the hard part. Repeated filtering proceeds as follows. Run the particle filter on the measurements received in [0, T] while preserving the path histories of the particles. Choose a smoothed path from the filtered result at the end of the time interval [0, T]. Repeat the filtering process using the same measurements as in the first run of the filter but using independent random numbers to generate the particle paths. Choose a smoothed path as before. Repeat this process to obtain M independent draws from the posterior distribution on the paths of the process, given the measurements received in [0, T]. This produces a discrete sample path approximation of the posterior distribution.

Repeated filtering is simple and general. It can be incorporated into operational systems and used by operators who are not experts in tracking or data fusion. Many operational problems require motion models that are not Markovian or do not have a closed-form transition function as required by other particle filter smoothing methods. Repeated filtering produces smoothed paths, not just smoothed marginal distributions at discrete times, as many smoothing techniques do. Moreover, repeated filtering can be applied to both discrete and continuous time motion models. For continuous-time models, the smooth paths are continuous time paths.

Despite the conceptual simplicity of repeated filtering, we have not been able to find a reference to it. The first two authors spent over a year trying unsuccessfully to solve a smoothing problem conceptually similar to the surveillance problem in Example 2. The existing methods for smoothing particle filters, which are referenced below, require assumptions that do not fit this problem. We tried a method involving Markov Chain Monte Carlo (MCMC) sampling, which was technically correct but computationally complex and delicate. In addition, it was unstable, producing qualitatively different results from different starting paths.

Perhaps by adjusting some of the tuning parameters of the MCMC, such as the acceptance probability or the proposal distribution, we could have made the MCMC method work. However, we decided that, even if we could make the method work, it would not be suitable for an operational system. By contrast, the repeated filtering method solved the problem easily and quickly and has been incorporated into an operational system. Its simplicity and robustness suggest that one might want to consider this method for some particle filter smoothing problems that can be solved by other methods.

A. Smoothing Particle Filters

In tracking situations, one is often faced with nonlinear measurements, such as lines-of-bearing and non-Gaussian motion models. The combination of nonlinear measurements and non-Gaussian motion models means that the traditional Kalman filter approach to tracking does not work well in these situations. For bearings-only tracking, particle filters have been shown [9] to outperform a Kalman filter as well as numerous nonlinear extensions of it.

In the case of a Kalman filter, there are efficient methods for smoothing, for example, the Rauch–Tung–Striebel smoother described in Section 3.2.3 of [12] or in [10]. Smoothing a particle filter is more difficult. If the particle filter preserves the full target path as the particles are split and reweighted during the resampling process, then the surviving paths and their posterior weights provide an estimate of the posterior distribution on the target paths. The difficulty with this smoother is that resampling particles usually leads to a set of surviving particles (paths) that descend from a small number of initial paths, and in some cases, only one initial path. As a result, this estimate loses resolution as one proceeds backward in time.

This generates the need for a better method of estimating the smoothed (posterior) distribution on the paths of a particle filter. Reference [10] provides a succinct review of Bayesian smoothing methods and, along with [5] and [8], provides an excellent overview of smoothing methods for particle filters.

Forward–backward smoothing, as described in Section 3.1.4 of [12] is a general solution to the smoothing problem. The difficulty with this solution is that, except in the case of Kalman filtering, one cannot evaluate the integrals involved explicitly. As a result, numerical methods are required for problems such as smoothing the output of a particle filter.

References [8], [6], and [3] present numerical methods for smoothing discrete-time particle filters that are aimed at producing marginal distributions on the state of the smoothed process at the discrete times of the process. These methods assume that the process is Markovian with an explicit functional form for the transition density.

Under these assumptions, Godsill et al. [5] developed a numerical approach to forward–backward smoothing called backward simulation. As with repeated filtering, backward simulation begins with the output from a particle filter with particles that preserve the full path of the particle. Backward smoothing produces a discrete set of independent sample paths from the posterior distribution on sample paths given the measurements received in $[0, T]$. By construction, the state of a smoothed path at time t is equal to one of the states in the particle filter approximation to the distribution at time t. References [1] and [2] remove this restriction to provide improved diversity and accuracy of the smoothing approximation.

Unfortunately, the smoothing problem that we wished to solve concerned a surveillance tracking system that used a quite natural but somewhat complex, continuous-time motion model that did not have a closed-form transition function. Although we could not use the methods referenced above, the structure of the problem allowed us to apply an MCMC method for generating the posterior distribution on target paths. In Example 2 of [11], we applied this method to a simplified version of this problem. Even for the simplified problem, the procedure was difficult and complex. Because of the nature of the motion model, a reversible-jump MCMC was required, which is even more complex than a standard MCMC. See the Appendix of [11]. However, the method obtained reasonable results on this difficult problem. As part of the further analysis and testing of this smoother, we examined the stability of the results. To do this, we ran the MCMC for 1 million iterations to estimate the posterior distribution on paths. This process took 4 h or more on a modest laptop. To test the stability of the procedure, we made a second MCMC run with 1 million iterations using the same inputs as the first run but with a different starting path for the iterations. The results were qualitatively different. The MCMC process had not converged even after 1 million iterations.

As mentioned above, we decided that the MCMC approach is too complex and delicate for an operational system. In its place, we developed the much simpler, faster, more robust, and more general repeated filtering approach presented here. Repeated filtering can be used with any stochastic process model for which one can generate independent sample paths from the process distribution. For measurements, the only requirement is that one be able to calculate likelihood functions for them. In Example IV.B, the repeated filtering method is applied to the surveillance problem mentioned above, where it performs well.

B. Outline of Paper

Section II presents a quick overview of Bayesian particle filtering to establish the notation and terminology used in this paper. Section III describes the repeated filtering method of smoothing, and Section IV presents four examples of the application of repeated filtering. Examples 1 and 2 have the same settings as Examples 1 and 2 in [11], but the results are obtained by repeated filtering. In Example 1, we apply repeated filtering to a problem that has a Kalman smoother solution and show that it produces a good approximation to that solution. Example 2 is the surveillance problem to which we applied the MCMC method in [11]. We have no yardstick by which to measure the accuracy of the solution we obtained. However, by comparing the repeated filtering solution to the actual target path, we show that this method provides a reasonable and stable solution for this problem.

Example 3 applies repeated filtering to smooth a nonlinear time series not related to tracking. This is the same problem as in Example 1 of [5], which obtained a smoothed solution to the time series using methods that require a discrete-time Markov process with a closed-form transition density. We apply repeated filtering to this problem and obtain results comparable to those in [5]. We then modified the stochastic process by adding reflecting boundaries, which produces a problem that is very difficult to solve with the methods of [5] but is simple to solve using repeated filtering.

II. BAYESIAN PARTICLE FILTERING

For this discussion, Bayesian particle filtering begins with a prior distribution on a time-varying parameter (e.g., target state) in the form of a stochastic process X on the state space S. Time is continuous, running over $[0, T]$, and the state space S can be continuous, discrete, or a combination of the two. The modifications when time is discrete will be obvious.

We approximate the prior stochastic process X (target motion model) by making a large number N of independent draws from the sample paths of the process. These sample paths form a discrete path approximation to the process. There may be times when it is more efficient to use a proposal distribution for obtaining independent sample paths from the process prior and to weight these appropriately to obtain an approximation of the stochastic process X. However, we do not consider that possibility here.

Let $\{x_n, n = 1, \ldots, N\}$ be the set of N sample paths that we have drawn from the process X. Each x_n specifies a possible target path with $x_n(t) \in S$ being the target state at time t for $t \in [0, T]$. We call x_n a *particle path* and $x_n(t)$ a *particle state* at time t. We assign probability $p(n) = 1/N$ to x_n for $n = 1, \ldots, N$ and define

$$P_N = \{(x_n, p(n)), n = 1, \ldots, N\}$$

to be the prior *particle path distribution*. This distribution is a discrete sample path approximation to the prior distribution on the process X. The distribution P_N produces a prior *particle state distribution* for each $t \in [0, T]$ by

$$P_N^t = \{(x_n(t), p(n)), n = 1, \ldots, N\}.$$

Bayesian particle filtering computes the Bayesian posterior distribution on this discrete particle state approximation at time t given the measurements received by time t.

In performing Bayesian filtering on this discrete sample path approximation, we obtain a solution to the particle filtering problem that is an approximation to the filtering problem on X. Thus, we find an exact solution to a problem that approximates the problem we wish to solve. The quality of this solution will depend on the quality of the discrete sample path approximation used to represent X.

A. Bayesian Recursion

We receive measurements at a discrete sequence of possibly random times $0 \le t_1 < t_2 \cdots < t_K \le T$. Let $L_k(y_k|\cdot)$ be the likelihood function for the measurement $Y_k = y_k$ received at t_k. Specifically,

$$L_k(y_k|s) = \Pr\{Y_k = y_k | X(t_k) = s\} \text{ for } s \in S. \quad (1)$$

Note, we use Pr to indicate probability or probability density as appropriate.

Suppose we have received the measurement $Y_1 = y_1$ at time t_1. We compute

$$p(n|y_1) = \frac{L_1(y_1|x_n(t_1))\, p(n)}{\sum_{m=1}^{N} L_1(y_1|x_m(t_1))\, p(m)} \text{ for } n = 1, \ldots, N \quad (2)$$

to obtain

$$P_N(y_1) = \{(x_n, p(n|y_1)), n = 1, \ldots, N\}, \quad (3)$$

which is the posterior particle path distribution given $Y_1 = y_1$.

Define $y_{1:k} = \{y_1, \ldots, y_k\}$ and $Y_{1:k} = \{Y_1, \ldots, Y_k\}$ for $k = 1, \ldots, K$. Suppose

$$P_N(y_{1:k-1}) = \{(x_n, p(n|y_{1:k-1})), n = 1, \ldots, N\}$$

is the posterior particle path distribution given $Y_{1:k-1} = y_{1\,k-1}$, and we receive the measurement $Y_k = y_k$ at time t_k. We compute

$$p(n|y_{1:k}) = \frac{L_k(y_k|x_n(t_1))\, p(n|y_{1:k-1})}{\sum_{m=1}^{N} L_k(y_k|x_m(t_1))\, p(m|y_{1:k-1})} \quad (4)$$

for $n = 1, \ldots, N$ to obtain

$$P_N(y_{1:k}) = \{(x_n, p(n|y_{1:k})), n = 1, \ldots, N\}, \quad (5)$$

which is the posterior particle path distribution given $Y_{1:k} = y_{1:k}$.

B. Resampling

A common problem with particle filters is that as measurements are received and processed into the filter, the posterior probability distribution tends to become concentrated on a small number of particles, causing the filter to lose resolution. This problem can be solved by resampling.

When one must resample, the filtering process becomes more complicated than described above. Instead of generating a set of N complete paths at the beginning of the filter, one must generate the paths sequentially in time so that at the time of kth measurement, one has a set of N paths (particles) over $[0, t_k]$ that provides good resolution for the distribution on system state at time t_k.

One method of resampling, described in Sections 3.3.3 and 3.3.4 of [12], splits high-probability particles into multiple (almost identical) particles and "kills off" low-probability particles in a manner that produces exactly N particles. Each child particle inherits the path history of its parent but has a slightly different state at time t_k. The resulting set of particles have probability $p(n|y_{1:l}) = 1/N$ for $n = 1, \ldots, N$.

The paths of the resampled particles are then extended to the time t_{k+1} of the next measurement to obtain $P_N^{t_{k+1}}(y_{1:k})$, the particle state distribution at time t_{k+1} given the measurements $y_{1:k}$. When $Y_{k+1} = y_{k+1}$ is received, we compute the posterior distribution on the particle paths using (4), with k replaced by $k + 1$. Alternatively, one may want to use a proposal distribution in place of $P_N^{t_{k+1}}(y_{1:k})$ to compute the posterior distribution on the particle paths.

C. The Problem With Resampling

The above procedure is a bootstrap version of the SIR particle smoother of Kitagawa [7]. This works well to provide a high-resolution estimate of the posterior distribution of the present target state at the time of the last measurement. The difficulty is that the surviving resampled particles tend to originate from a small number of the original particles, so the posterior distribution on sample paths lacks resolution as one moves from present time back to time 0. The set of particle paths obtained by time T in this fashion form an estimate of the smoothed distribution on sample paths. However, it is not a very good estimate. See [3] and [4]. Simply increasing the initial number of sample paths is not an effective solution to this problem in most cases; see [8]. Repeated filtering was developed to solve this problem without the Markov or discrete-time assumptions required by other methods.

III. REPEATED FILTERING

The increasing speed, memory capacity, and capability of present-day computers allow us to propose the following method, which would not have been practical a few years ago. The method is called *repeated filtering*. It is implemented by the following recursion, which produces M independent sample paths from the smoothed distribution on sample paths.

A. Repeated Filtering Recursion

- **Step 1**. Make an initial run of the particle filter, processing the measurements received over the time interval $[0, T]$ and resampling as necessary.
- **Step 2**. Resample the particles at time T to obtain N equal probability particle paths $\{x_n; n = 1, \ldots, N\}$. Choose one of these paths at random, with each path having probability $1/N$ of being chosen. Save the chosen sample path $\bar{x}$.
- **Step 3**. Rerun the particle filter with the same measurements as in Step 1, but drawing particles that are independent of those drawn in Step 1. This will ensure that we choose new and independent samples of the target state at time 0 and at the measurement times.
- **Step 4**. Make a random draw to choose one of the sample paths as in Step 2. Save this sample path.
- **Step 5**. Repeat Steps 3 and 4, using particles that are independent of those drawn previously, until one obtains M smoothed sample paths $\bar{x}_m$ for $m = 1, \ldots, M$. Define the particle path distribution

$$\bar{P}_N\left(y_{1:K}\right) = \left\{(\bar{x}_m, 1/M)\,;\, m = 1, \ldots, M\right\}. \quad (6)$$

Then $\bar{P}_N(y_{1:K})$ is a discrete path approximation to the posterior distribution on sample paths given the measurements received in $[0, T]$.

The solution in (6) gives each smoothed path an equal weight. We hypothesize that an alternate weighting scheme applied to the smoothed paths in (6) would produce a better solution. However, none of the methods we have tried have done this. This is an area for further investigation.

B. Marginal Distributions

For any $t \in [0, T]$, we can obtain from $\bar{P}_N(y_{1:K})$ a particle state estimate for the smoothed marginal distribution at time t as follows:

$$\bar{P}_N^t\left(y_{1:K}\right) = \left\{(\bar{x}_m(t), 1/M)\,;\, m = 1, \ldots, M\right\}. \quad (7)$$

We often provide a visual representation of such a distribution by imposing a grid of cells on the state space, summing the probability of the points in each cell, and color coding the cells to represent the probabilities in the cells.

The ability to estimate the distribution of the state of the smoothed process at a time between measurements can be particularly important in situations where there are large time gaps between measurements, as occurs in some surveillance problems. In addition, having a set of smoothed paths can be helpful in determining patterns of motion. Moreover, as noted in [5], having sample paths also allows one to explore relationships between the state of the process at different times.

C. Resolution of the Smoothed Solution

Resampling within each run of the particle filter is necessary to preserve the resolution of the estimate of the posterior distribution as the number of measurements increases. Making independent runs of the particle filter in Step 3 to obtain M posterior sample paths is necessary to preserve the resolution of the estimate of the posterior as one goes back in time toward 0. Determining the number of particle filter runs required and the number of particles for a run generally requires some experimentation.

We expect that there is some limitation on the length of the interval $[0, T]$ over which this process produces solutions with good resolution, or more likely, as the time interval gets longer, the number of particle filter runs M may need to get larger. We have not explored this question. Another possibility is to break the interval $[0, T]$ into two or more subintervals and splice the solutions from the subintervals together in some fashion. We have not explored this possibility either.

D. Computation Time

The computation time to obtain a repeated filtering solution depends on the time to perform one filter run, which depends on the complexity of the problem. Generating M independent samples from the posterior will take M times as long as a single filter run. If time becomes a problem, one can easily apply coarse grain parallel processing by allocating the repetition of Steps 3 and 4 across a number of processors.

IV. EXAMPLES

This section presents four examples of estimating the posterior distribution on sample paths in $[0, T]$ using repeated filtering. The first example compares repeated filtering to a Kalman smoother where the exact solution is known. The second example involves a simplified surveillance situation where the target is moving through an area in which it has to avoid certain regions. Even though this is a simplified situation, it is still a challenge for smoothing. We use a motion model called a generalized random tour (GRT), see [11] or Section 1.3.3 of [12], which is a special case of a variable rate particle filter. We incorporate avoidance regions to provide a more realistic and challenging motion model. The last two examples smooth a standard nonlinear time series used to test particle filters.

A. Example 1: Comparison to a Kalman Smoother

For this comparison, the motion model is the almost-constant velocity model described below, and the measurements are position measurements with additive circular normal errors. We find a repeated filtering solution for this example and compare it to the solution from the Rauch–Tung–Striebel smoother [10, p. 135].

1) Almost Constant Velocity Model: The target state is given by a position–velocity pair (x, v). The state at time 0 is

$$(x_0, v_0) \sim \eta\left(\cdot, (\bar{x}, \bar{v}), \Sigma_0\right), \tag{8}$$

where we use $\eta(\cdot, \mu, \Sigma)$ to denote a normal density function with mean μ and covariance Σ. Let Δ be a fixed time increment. There are I time increments and $T = I\Delta$. The target proceeds at velocity v_0 until time $t_1 = \Delta$ at which time a new velocity v_1 is obtained by adding a small, independent, mean-zero, Gaussian distributed variation to v_0 to obtain v_1. The target continues at this velocity until the next time increment. We may express this mathematically as follows. Let (x_i, v_i) be the target state at time $t_i = i\Delta$. Then

$$\begin{pmatrix} x_i \\ v_i \end{pmatrix} = \begin{pmatrix} x_{i-1} + \Delta v_{i-1} \\ v_{i-1} + w_i \end{pmatrix} \text{ for } i = 1, 2 \ldots, \tag{9}$$

where $\{w_i : i = 1, \ldots, I\}$ are independent, identically distributed random variables with $w_i \sim \eta(\cdot, (0, 0), Q)$ and Q is a "small" covariance matrix.

Let $\mathbf{I}_m$ be the m-dimensional identity matrix. The parameters of the motion model are $\Delta = 1$ hr,

$$\bar{x} = (0, 0)\,, \bar{v} = 7 \text{ kn} (\cos(\theta), \sin(\theta)) \text{ where } \theta = \frac{\pi}{6}$$

$$\Sigma_0 = \begin{bmatrix} \sigma_x^2 \mathbf{I}_2 & 0 \\ 0 & \sigma_v^2 \mathbf{I}_2 \end{bmatrix} \text{ where } \sigma_x = 4 \text{ nm}, \sigma_v = 1 \text{ kn}$$

$$Q = \sigma_w^2 \mathbf{I}_2, \text{ where } \sigma_w = 1 \text{ kn}. \tag{10}$$

We use the abbreviations nm for nautical miles and kn for knots. A knot equals 1 nm/h.

The target follows a slightly curved path starting at the origin, as shown by the black line in Fig. 1. The time duration is 10 h. There are eleven position measurements received at 1-h increments over the duration of the path. The 2-sigma uncertainty ellipses for the measurements are shown in black. The measurements have circular Gaussian errors with a standard deviation of 4 nm.

2) Comparison of Kalman and Repeated Filtering Smoothers: The Kalman smoother provides an analytic solution to this problem. The red ellipses are the 2-sigma ellipses from the Kalman smoother solution at equally spaced times on the path.

Repeated filtering was applied to this problem by drawing 10 000 particle paths from the almost-constant velocity motion model and performing the recursion in Section III to obtain 400 samples from the posterior distribution on target paths. The green ellipses are the 2-sigma ellipses derived from the empirical means and covariances of the path positions at the same times as the Kalman smoother ellipses. As one can see, there is good agreement between the two plots of 2-sigma ellipses. Note that agreement improves as time increases

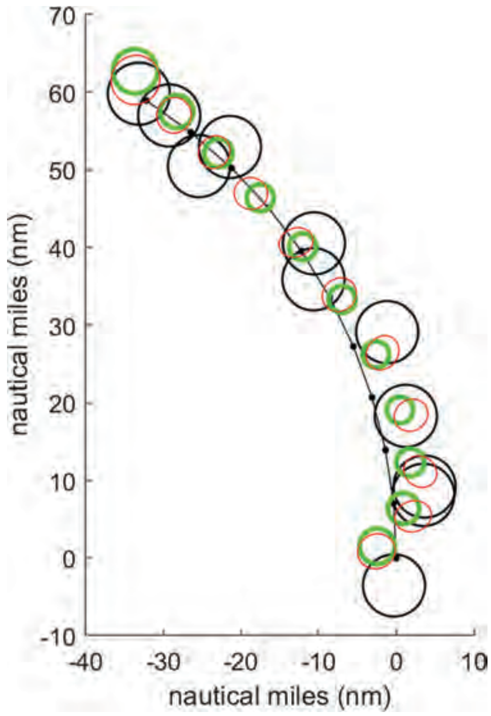


Fig. 1. Comparison of Kalman and repeated filtering smoothers. The black line is the target's path, which starts at (0,0); black circles are 2-σ uncertainty circles for measurements; red ellipses are 2-σ ellipses for the Kalman smoother solution; green ellipses are 2-σ ellipses for the repeated filtering smoother solution.

because the smoother has more history to use for the smoothing.

We made the following computation to measure how well repeated filtering approximated the optimal solution from the Kalman smoother. At each of the eleven equally spaced points on the target track in Fig. 1, we computed the mean squared distance from the point to the bivariate normal distribution at that point computed by the Kalman smoother and to the bivariate normal distribution corresponding to the $2-\sigma$ ellipse for the repeated filtering result. We averaged these results over the eleven points and took the square root of this average to obtain the square root of the average mean squared missed distance for the Kalman and repeated filtering smoothers.

The results were 3.97 nm for the Kalman smoother and 4.61 nm for repeated filtering. The repeated filtering result is only 16% larger than the Kalman result demonstrating that repeated filtering provides a good approximation to the posterior path distribution in a case where we can calculate the exact distribution. We produced only 400 smoothed paths for this example. Using more paths would improve the approximation.

B. Example 2: Surveillance Problem

As before, we use a target state space that is 2-dimensional in position and velocity and use (x, v) to represent a position and velocity pair in this space. A

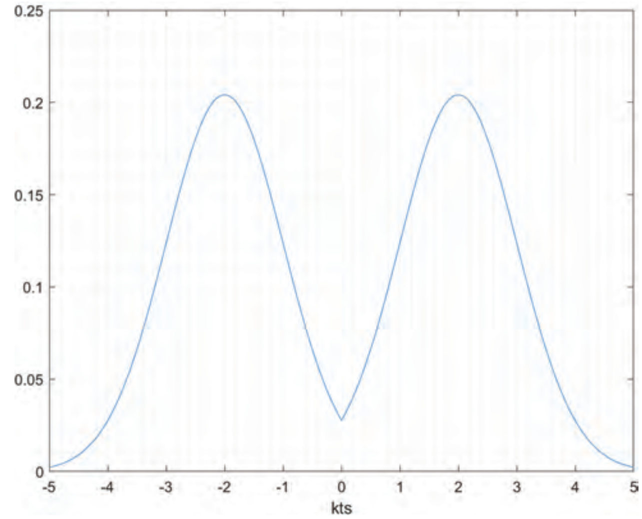


Fig. 2. Speed change distribution.

GRT motion model is specified by first specifying a probability (density) function $p_0(x, v)$ on the position and velocity (x_0, v_0) of the target at time 0. As time progress, the target changes velocity (instantaneously) at the event times of a Poisson process with rate λ.

Between velocity changes, the target follows a constant velocity path at the previously chosen velocity. When the target changes velocity, its new velocity v_i is drawn from a probability (density) function $p(\cdot|v_{i-1})$, where v_{i-1} is the velocity just prior to the change. For many tracking problems, the GRT motion model is more operationally realistic than the almost-constant velocity motion model or other often-used Gaussian-diffusion motion models.

For this example, we set $\lambda = 0.25/\text{h}$ and

$$p_0(x_0, v_0) = \eta\left(x_0, (0,0), (15\,\text{nm})^2\mathbf{I}_2\right) \times \eta\left(v_0, (0,0), (10\,\text{kn})^2\mathbf{I}_2\right).$$

When a velocity change occurs, the new velocity is chosen by making independent draws to determine the changes to the speed and course of the target. The speed change distribution is symmetric about zero. On each side of zero, the distribution is proportional to that of a truncated Gaussian whose mean has an absolute value of 2 kn and a standard deviation of 1 kn, as shown in Fig. 2. Similarly, the course change distribution is symmetric about zero, with each side being proportional to a truncated Gaussian whose mean has an absolute value of 60°and a standard deviation of 30°.

As the sample paths are generated, we ensure that they stay clear of avoidance regions, as follows: When a velocity change takes place, the time on that leg is drawn as well as the new velocity. If the resulting leg hits an avoidance region, a new velocity is drawn. This process is repeated up to a maximum of 20 times until the resulting leg does not intersect an avoidance region. If no

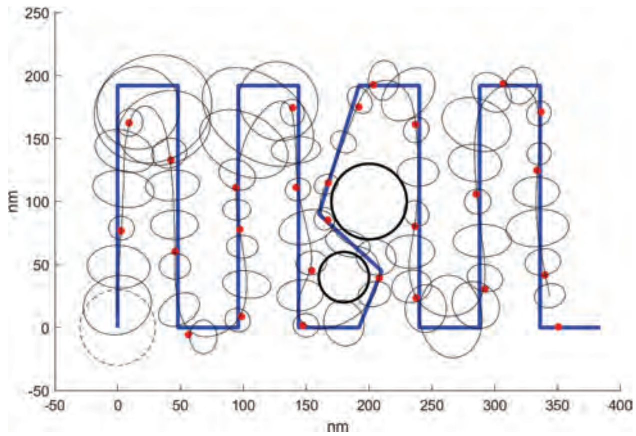


Fig. 3. Slinky plot from the first run of the repeated filtering smoother. The heavy blue line shows the target's path, which starts at (0, 0). The red dots show the measurements. The black circles show discs of the regions that the target must avoid. The red ellipses are 2-sigma ellipses generated by the repeated filtering smoother. The dashed circle shows the 2-sigma ellipse for the initial position distribution at time 0.

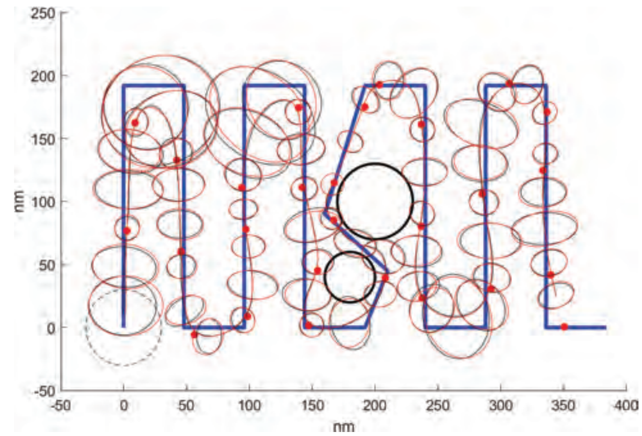


Fig. 4. Comparison of the slinky plots from two runs of the repeated filtering method using the same inputs but independent random numbers. The blue line shows the target's path.

such leg is found, then the path is discarded and a new one generated in its place.

1) Scenario Description: The actual target path, shown in blue in Fig. 3, has a fixed speed of 8 kn. It follows the ladder path with long legs of 24 h duration and short legs of 6 h duration. The black circles indicate regions that the target must avoid as it traverses its path. The target path starts near the origin. The total time is 240 h or 10 days.

The time to the first measurement is gamma-distributed with a mean of 4 h and variance of $(8/3)$ h^2. The time intervals between subsequent measurements are independent with this same gamma distribution. The measurements are of position with a circular Gaussian error distribution having a standard deviation of 10 nm. In Fig. 3, measurements are indicated by red dots, and the dashed circle shows the 2-sigma ellipse of the initial position distribution.

2) Repeated Filtering Smoother: For repeated filtering, we ran the particle filter with $N = 10\,000$ and at time T randomly chose one of the paths, with each path having an equal probability of being chosen.

We repeated Steps 3 and 4 in Section III-A to obtain $M = 400$ sample paths from the posterior (smoothed) distribution on the target paths. The 2-sigma ellipses for the position estimates were calculated every 4 h. This sequence of ellipses, called a slinky plot, is shown in Fig. 3. The ellipses represent normal approximations to the position distributions every 4 h. Thus, even though some ellipses intersect an avoidance region, the paths themselves do not.

To illustrate the stability of the repeated filtering method, we repeated this run a second time using the same measurements as in the first run but using random draws independent of those made for the first run. We overlaid the slinky plots for the two runs in Fig. 4. As the reader can see, there is little, if any, difference in the plots, which gives us confidence in the stability of the method.

A sample of the smoothed paths from repeated filtering is shown in Fig. 5. Note that none of the sample paths pass through the avoidance regions. Note also that there is more uncertainty in the distribution of the smoothed paths near (0,0), the target's starting point at time 0, than there is close to time T. It is not surprising that having past history as well as future information is helpful in estimating the target's smoothed path.

3) MCMC Smoothing: In [11], we applied an MCMC method to estimate the posterior distribution on the target paths for this example. The procedure was difficult and complex. In the hope of ensuring the stability of the results, we ran the MCMC for 1 million iterations to estimate the posterior distribution on paths. This process took 4 h or more on a modest laptop. The results looked reasonable, but to test the stability of the procedure, we made a second MCMC run with 1 million iterations

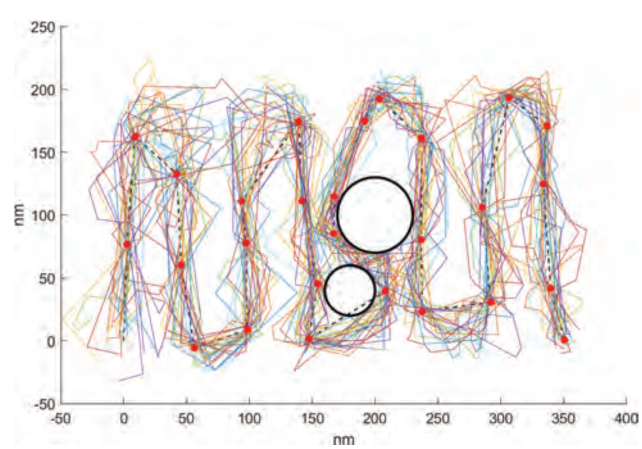


Fig. 5. Smoothed sample paths selected by random draws from the set of smoothed paths. Each path has an equal probability of being chosen. Note that none of the sample paths pass through the avoidance regions. The dashed line connects the measurements.

using the same inputs as the first run but with a different starting path for the iterations. As noted in the introduction, the results were qualitatively different. The MCMC had not converged even after 1 million iterations. At this point, we abandoned the MCMC approach and developed the repeated filtering approach.

4) Discussion: We have no analytical method with which to compare our smoothed solution in this example. The solution appears reasonable compared to the actual target path in this example, even though this is a difficult problem and there is a mismatch between the motion model and the target's motion. The smoothed paths stay out of avoidance regions, and the distribution is repeatable up to the small differences that are to be expected in Monte Carlo solutions.

We have used slinky plots to display the smoothed solution. Alternatively, one could calculate a mean smoothed path by finding the mean of the position of the paths at each time in a sequence of evenly spaced times and displaying the line connecting these means. Or, one could display both the mean path and the slinky plot.

C. Example 3: Smoothing a Nonlinear Time Series

In this example, we apply repeated filtering to smooth the output of the stochastic nonlinear time series model given in Example 1 of [5]. Reference [5] describes this model as a standard nonlinear time series model that has been used extensively for testing numerical filtering techniques.

The time series $\{X(t), t = 1, ..., 100\}$ has for its initial state $X(1) \sim \eta(\cdot, 0, 10)$ and is defined for $t \geq 2$ by

$$X(t) = \frac{X(t-1)}{2} + \frac{2.5X(t-1)}{1 + X^2(t-1)} + 8\cos(1.2t) + v(t)$$
$$v(t) \sim \eta(\cdot, 0, 10), v(t) \text{ independent of } v(s) \text{ for } s \neq t. \tag{11}$$

The measurements $\{Y(t), t = 1, ..., 100\}$ are defined by

$$Y(t) = \frac{X^2(t)}{20} + w(t)$$
$$w(t) \sim \eta(\cdot, 0, 1), \mathrm{w}(t) \text{ independent of } \mathrm{w}(s) \text{ for } s \neq t. \tag{12}$$

We cannot reproduce the results in Example 1 in [5] exactly because we do not have access to the sample path [5] of the process used for their example or the series of measurements produced. While we cannot reproduce this example exactly, we are able to produce comparable results and similar figures, which leads us to conclude that the repeated filtering method produces results comparable to the method in [5], which is limited to discrete-time Markov processes with closed-form transition functions.

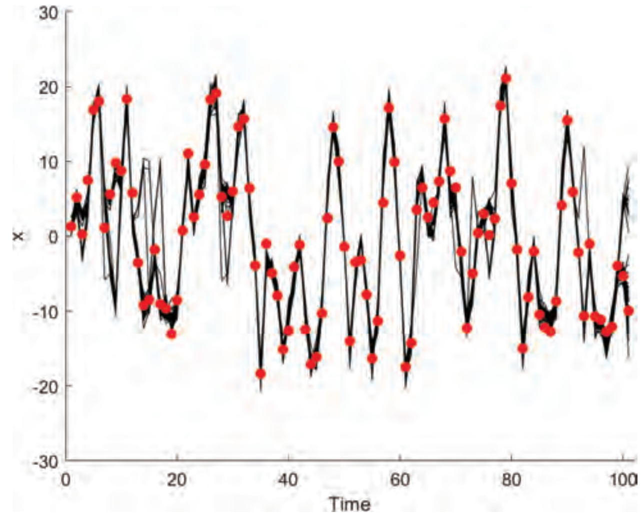


Fig. 6. Fifty smoothed paths are shown in black; the time series values are in red.

1) Repeated Filtering Approach: To apply repeated filtering, we simulated one sample path and set of measurements from the time series defined by (11) and (12). Using these as inputs, we ran a standard SIR particle filter with 1000 particles resampling as described in Section II-B. We used the stochastic process defined in (11) and (12) for our motion model for the filter. The particles were resampled at each time step so that they all had equal weight. At the conclusion of a filter run at time 100, we selected one of these paths by making a draw from this set of particles, with each particle having an equal probability of being drawn. This path was saved as a smoothed path. We repeated the filtering process 1000 times, using independent random numbers to produce the particles and drawing one of them for a smoothed path. The resulting set of 1000 independently drawn smoothed paths constitutes our estimate of the posterior distribution on the target paths given the measurements in [1,100].

Fig. 6 shows a sample of 50 smoothed paths in black and the actual values of the time series in red. Looking at the measurement equation (12), one can see that a value x of the series will produce the same measurement as $-x$. As more measurements are received, the smoother (usually) sorts out this ambiguity. This ambiguity has produced the bimodal results near time 100.

Fig. 7 shows the smoother results when restricted to the interval [0, 51]. The ambiguity in the smoothed solution near time 51 in this figure is resolved by time 100 in Fig. 6.

Figs. 8–10 below are similar to Figs. 4, 5, and 7 in [5], and the results are qualitatively similar. Since the sampled time series and measurements in our data are somewhat different from those in [5], we do not expect an exact match.

Fig. 8 shows a histogram of the smoothed distribution values at each of the 100 times. As [5] notes, one of the advantages of finding smoothed paths rather than marginal distributions at each time is the ability to analyze joint densities of values at two different times.

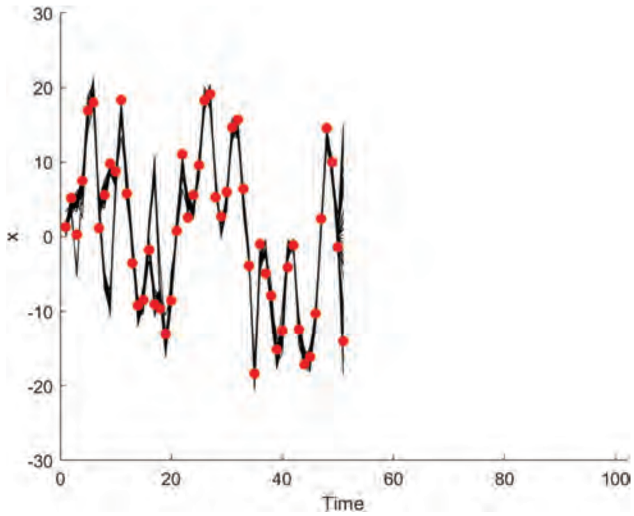


Fig. 7. Smoother results for [0,51]. Smoothed paths are in black. Time series values are in red.

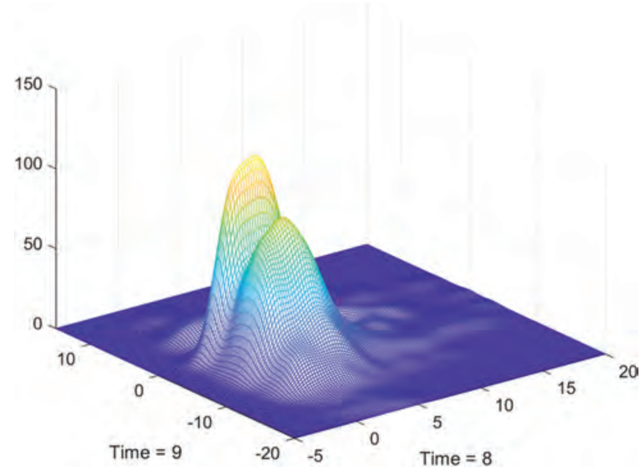


Fig. 9. Joint density plot for the values of the smoothed time series at times 8 and 9. Note the multimodal distribution.

Figures 9 and 10 show examples of these joint densities and exhibit behavior similar to that seen in [5].

D. Smothing a Nonlinear Time Series With Reflecting Boundaries

This example adds reflecting boundaries at $+15$ and -15 to the nonlinear time series example in Section IV-C. Smoothing of this process is easily performed using repeated filtering but is difficult to do using the methods of [5]. Equations (13)–(15) in the Appendix give the revision to the equation for $X(t)$ produced by the reflecting boundaries.

One can see from these equations that for each transition, one must allow for the reflection off one or more boundaries to determine the distribution of $X(t)$ given $X(t-1)$. In fact, since the term, $v(t)$ is drawn from a Gaussian distribution, the transition function involves summing an infinite number of terms to account for the number of possible reflections!

To smooth this process, we modified the particle filter in Section IV-C by adding the reflective boundaries and performed repeated filtering as above.

Fig. 11 shows 50 smoothed sample paths from this process. The smoothed paths are shown in black, and the red dots indicate the values of the process. Note the ambiguity at time 100. The value of the process is approximately -10, but the smoother shows an ambiguity about 0 because of the measurement model. This ambiguity will not be resolved until more data is received. If one truncated the time series at time 50, as in Fig. 12, then one would see a similar ambiguity that is resolved as the filter receives more data.

Fig. 13 shows the joint density of the smoothed paths at times 99 and 100. As one can see from this figure, if the series is positive (negative) at time 99, it will be positive (negative) at time 100.

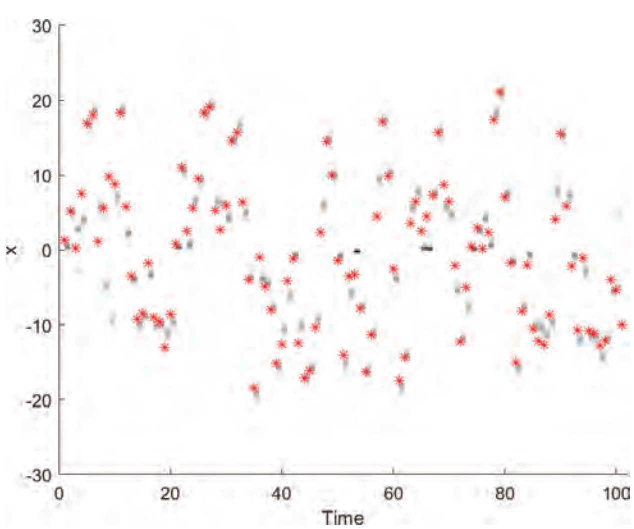


Fig. 8. Histogram of smoother values. Dark grey indicates higher-density areas. Red stars show the actual values of the time series.

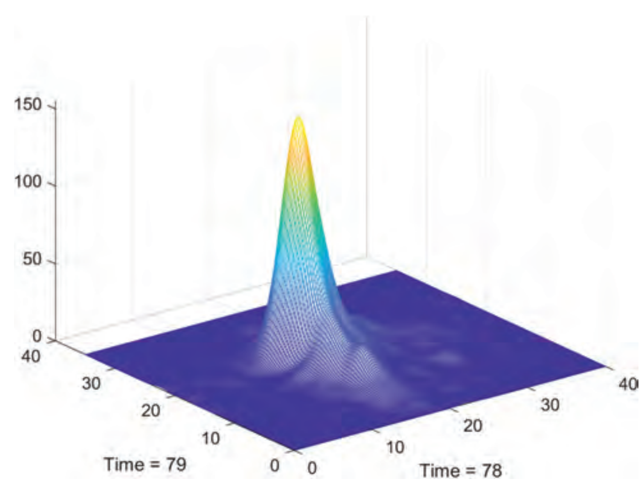


Fig. 10. Joint density plot for times 77 and 78. This density is unimodal but not Gaussian.

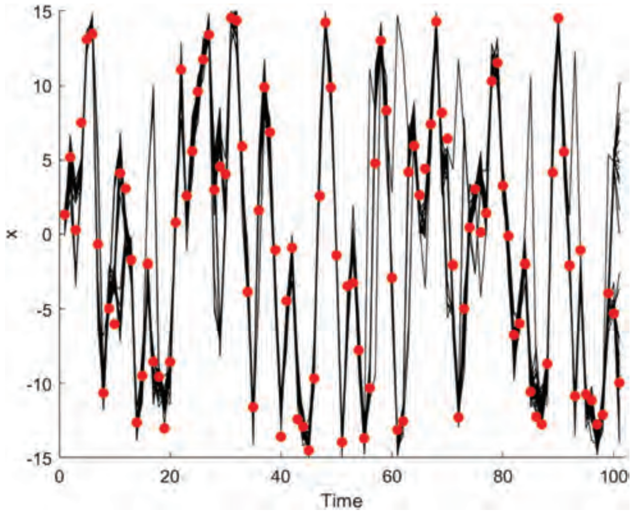


Fig. 11. Smoothed paths for time series with reflecting boundaries. Smooth paths are shown in black; times series values in red.

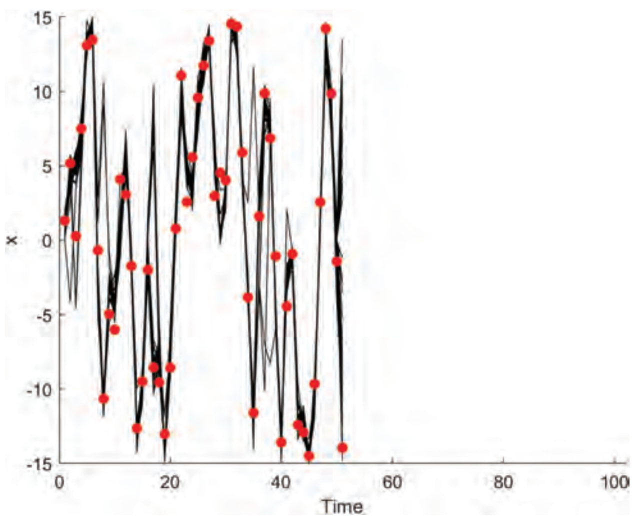


Fig. 12. Smoothed paths resulting from stopping the series at time 50.

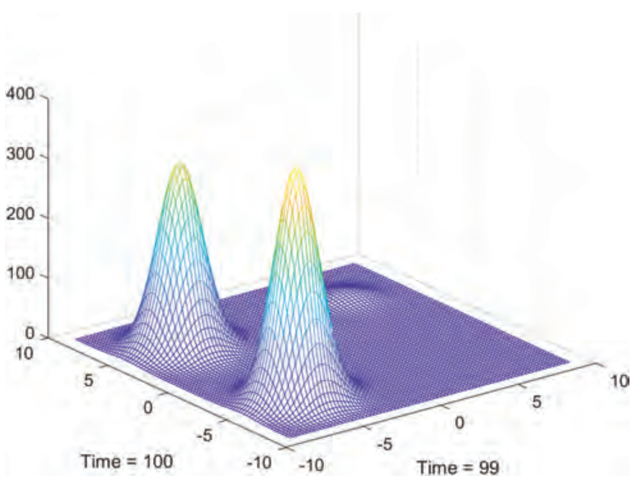


Fig. 13. Joint density at times 99 and 100.

V. CONCLUSIONS

This paper presents the repeated filtering smoother, which is a simpler method than most smoothing methods and can be applied to any SIR particle filter. Like the method in [5], the smoother produces sample paths from the smoothed distribution, allowing for more detailed analysis of path behavior than can be obtained from smoothers that produce only marginal distributions. The only restriction on the stochastic process defining the motion model used for the particle filter is that one must be able to draw independent sample paths from the process and that these paths can be produced sequentially in time. In particular, the process does not have to be Markovian in its state space.

We have demonstrated the repeated filtering method on a Kalman filter problem and shown that it produces comparable results. We demonstrated the method on a tracking problem with a motion model whose transition function does not have a closed analytical form, and which has unusual features such as avoidance areas. Next, we demonstrated the repeated filtering method on a standard nonlinear time series problem used to test many particle filters. We also performed repeated smoothing on a modification of this time series with reflecting boundaries. The smoothing was performed on this example by simply putting reflecting boundaries on the time series and applying the repeated filtering. By contrast, the method in [5] would require substantial additional effort.

In all four examples, the repeated filtering method performed well and required only modest amounts of computational effort. We stress again the simplicity and generality of this method. If one can construct a good particle filter for the process, one can easily find smoothed sample paths using repeated filtering. The computational load is easy to estimate. If you want M smoothed paths, the computational effort will be M times the effort required for a single run of the particle filter.

APPENDIX

EQUATION FOR TIME SERIES WITH REFLECTING BONDARIES

This appendix derives the modifications to (11) resulting from adding the reflecting boundaries in Example 4.

If $v(t) > 0$, let

$$Z(t) = \frac{X(t-1)}{2} + \frac{2.5X(t-1)}{1+X^2(t-1)} + 8\cos(1.2t). \quad (13)$$

If $v(t) \le 15 - Z(t)$, then

$$X(t) = Z(t) + v(t). \quad (14)$$

If $v(t) > 15 - Z(t)$, let

$$R(t) = v(t) - (15 - Z(t))$$
$$n(t) = \lfloor R(t)/30 \rfloor$$
$$f(t) = v(t) - [30n(t) + (15 - Z(t))].$$

Then

$$X(t) = (-1)^{n(t)} (15 - f(t)). \quad (15)$$

If $v(t) \leq 0$, then one obtains a similar set of equations for $X(t)$.

REFERENCES

[1] M. Briers, A. Doucet, and S. Maskell
"Smoothing algorithms for state-space models,"
Ann. Inst. Stat. Math., vol. 62, pp. 61–89, 2010.

[2] P. Bunch and S. Godsill
"Improved particle approximations to the joint smoothing distribution using Markov Chain Monte Carlo,"
IEEE Trans. Signal Process., vol. 61, no. 4, pp. 956–963, Feb. 2013.

[3] A. Doucet, J. F. G De Freitas, and N. J. Gordon
Eds.
Sequential Monte Carlo Methods in Practice. New York, NY: Springer, 2001.

[4] A. Doucet, S. J. Godsill, and C. Andrieu
"On sequential Monte Carlo sampling methods for Bayesian filtering,"
Stat. Comput., vol. 10, no. 3, pp. 197–208, 2000.

[5] S. J. Godsill, A. Doucet, and M. West
"Monte Carlo smoothing for nonlinear time series,"
J. Am. Stat. Assoc., vol. 99, no. 465, pp. 156–168, 2004.

[6] M. Hürzler and H. R. Künsch
"Monte Carlo approximations for general state space models,"
J. Comput. Graph. Stat. vol. 7, no. 2, pp. 175–193, 1998

[7] G. Kitagawa
"Monte Carlo filter and smoother for non-Gaussian, nonlinear state space models,"
J. Comput. Graph. Stat., vol. 5, no. 1, 1–25, 1996.

[8] M. Klaas, M. Briers, N. de Freitas, S. Maskell, and D. Lang
"Fast particle smoothing: If I had a million particles,"
in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 481–488.

[9] B. Ristic, S. Arulampalam, and N. Gordon
Beyond the Kalman Filter. Boston, MA: Artech House, 2004.

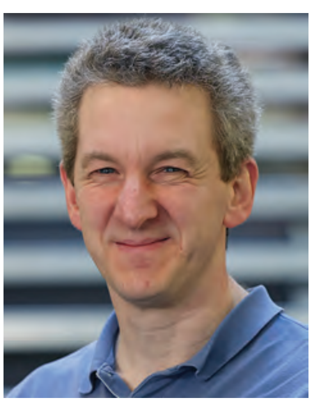
[10] S. Särkä
Bayesian Filtering and Smoothing. New York, NY: Cambridge University Press, 2013.

[11] L. D. Stone, S. L Anderson, and D. Lo
"MCMC smoothing for generalized random tour particle filters,"
in *Proc. FUSION 2018, 21st Int. Conf. Inf. Fusion*, 2018, pp. 142–150.

[12] L. D. Stone, R. L. Streit, T. L. Corwin, and K. L. Bell Bayesian
Multiple Target Tracking, 2nd ed. Boston, MA: Artech House, 2014.

Stephen L. Anderson received an B.A. (Honors) degree in mathematics from the University of Utah, Salt Lake City, UT, USA, in 1975. He earned a Ph.D. degree in mathematics from Brown University, Providence, RI, USA, in 1980. From 1980 to 1983, he was Acting Assistant Professor of Mathematics with the University of Washington, Seattle, WA, USA, and from 1983 to 1986, he was Assistant Professor of Mathematics and Computer Science at Wilkes University, Wilkes-Barre, PA, USA. From 1987 to 1993, he worked at Daniel H. Wagner, Associates, primarily on multitarget tracking applications. He joined Metron, Inc., in 1993, where he is currently Senior Research Scientist. At Metron, his work has focused primarily on target tracking and search applications.

Lawrence D. Stone is Chief Scientist at Metron Inc., Reston VA, USA. He is a member of the National Academy of Engineering and a fellow of the Institute for Operations Research and Management Science (INFORMS). He is a recipient of the Jacinto Steinhardt Award from the Military Applications Section of INFORMS in recognition of his applications of Operations Research to military problems. In 1975, the Operations Research Society of America awarded the Lanchester Prize to his text, *Theory of Optimal Search*. In 1986, he produced the probability maps used to locate the *S.S. Central America*, which sank in 1857, taking millions of dollars' worth of gold coins and bars to the ocean bottom one and one-half miles below. In 2010, he led the team that produced the probability distribution that guided the French to the location of the underwater wreckage of Air France Flight AF447. Recently, he used search theory methods to help guide the Canadian exploration company, Aurania, to the location of one of the lost Spanish gold cities in Ecuador. He coauthored the 2016 book *Optimal Search for Moving Targets*. He was one of the primary developers of the Search and Rescue Optimal Planning System, which has been used by the Coast Guard since 2007 to plan searches for people missing at sea. He continues to work on a number of detection and tracking systems for the United States Navy. He is a coauthor of the 2014 book, *Bayesian Multiple Target Tracking 2nd Ed.*



Simon Maskell (Member, IEEE) received the M.A., M.Eng., and Ph.D. degrees in engineering from the University of Cambridge, Cambridge, UK, in 1998, 1999, and 2003, respectively. Prior to 2013, he was a Technical Manager of command, control, and information systems with QinetiQ, UK. Since 2013, he has been a Professor of Autonomous Systems with the University of Liverpool, Liverpool, UK. His research interests include Bayesian inference applied to signal processing, multitarget tracking, data fusion, and decision support with particular emphasis on the application of sequential Monte Carlo methods in challenging data science contexts.