

**A GIRSANOV MONTE CARLO APPROACH TO
PARTICLE FILTERING FOR MULTI-TARGET TRACKING**

By

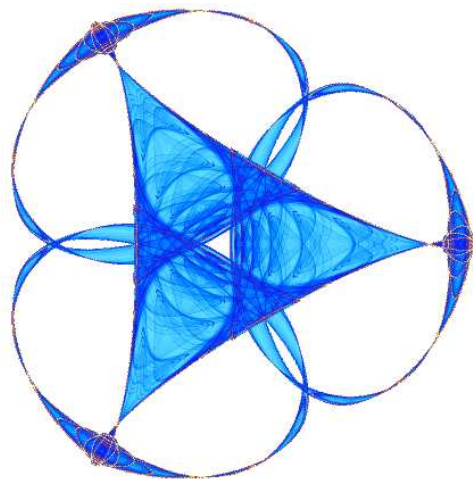
Vasileios Maroulas

and

Panos Stinis

IMA Preprint Series # 2325

(July 2010)



INSTITUTE FOR MATHEMATICS AND ITS APPLICATIONS

UNIVERSITY OF MINNESOTA
400 Lind Hall
207 Church Street S.E.
Minneapolis, Minnesota 55455-0436

Phone: 612-624-6066 Fax: 612-626-7370

URL: <http://www.ima.umn.edu>

A Girsanov Monte Carlo approach to particle filtering for multi-target tracking

Vasileios Maroulas
Institute for Mathematics and its Applications
University of Minnesota
Minneapolis, MN 55455
and
Panos Stinis
Department of Mathematics
University of Minnesota
Minneapolis, MN 55455

Abstract

We present a novel approach for improving particle filters for multi-target tracking. The suggested approach is based on Girsanov's change of measure theorem for stochastic differential equations. Girsanov's theorem is used to design a Markov Chain Monte Carlo step which is appended to the particle filter and aims to bring the particle filter samples closer to the observations. Also, we present a simple Metropolis Monte Carlo algorithm for tackling the target-observation association problem. We have used the proposed approach on the problem of multi-target tracking for both linear and nonlinear observation models. The numerical results show that the suggested approach can improve significantly the performance of a particle filter.

Introduction

Multi-target tracking is a central and difficult problem arising in many scientific and engineering applications including radar and signal processing, air traffic control and GPS navigation [10]. The tracking problem consists of computing the best estimate of the targets' trajectories based on noisy measurements (observations).

Several strategies have been developed for addressing the multi-target tracking problem, see e.g. [1, 5, 4, 13, 7, 9, 10, 11, 12, 19]. In this paper we

focus on particle filter techniques [4, 13]. The popularity of the particle filter method has increased due to its flexibility to handle cases where the dynamic and observation models are non-linear and/or non-Gaussian. The particle filter approach is an importance sampling method which approximates the target distribution by a discrete set of weighted samples (particles). The weights of the samples are updated when observations become available in order to incorporate information from the observations.

Despite the particle filter's flexibility, it is often found in practice that most samples will have a negligible weight with respect to the observation, in other words their corresponding contribution to the target distribution will be negligible. Therefore, one may resample the weights to create more copies of the samples with significant weights [7]. However, even with the resampling step, the particle filter might still need a lot of samples in order to approximate accurately the target distribution. Typically, a few samples dominate the weight distribution, while the rest of the samples are in statistically insignificant regions. Thus, some authors (see e.g. [6, 20]) have suggested the use of an extra step, after the resampling step, which can help move more samples in statistically significant regions.

The extra step for the particle filter is a problem of conditional path sampling for stochastic differential equations (SDEs). In [16], a new approach to conditional path sampling was presented. In that paper, it was also shown how the algorithm can be used to perform the extra step of a particle filter. In the current work, we have applied the conditional path sampling algorithm from [16] to perform the extra step of a particle filter for the problem of multi-target tracking.

The suggested approach is based on Girsanov's change of measure transformation [14] for the SDE system which describes the dynamics of the targets. The dynamics of an SDE are governed by a deterministic term, called the drift, and a stochastic term, called the noise. While unconditional path sampling is straightforward for SDEs, albeit expensive for high dimensional systems, conditional path sampling can be difficult even for low dimensional systems. On the other hand, it can be easier to find conditional paths for an SDE with a modified drift which is usually simpler than the drift of the original equation. Of course, these simplified paths may have a very low probability of being paths of the original SDE. One has to take this into account and assign to the paths of the simplified equation the right weight with respect to the original equation. Stochastic analysis, and in particular Girsanov's theorem (see e.g. Section 8.6 in [14]), provides an explicit formula for the calculation of the right weight. Moreover, one can view the Girsanov weight formula as a probability density, and use it to sample paths

of the modified equation that also have a high probability as paths of the original equation (see Appendix A for more details).

In addition to the extra step for the particle filter, we have designed and implemented a simple Metropolis Monte Carlo algorithm for the target-observation association problem. This algorithm effects a probabilistic search of the space of possible associations to find the best target-observation association. Of course, one can use more sophisticated association algorithms (see [13] and references therein) but the Monte Carlo algorithm performed very well in the numerical experiments.

This paper is organized as follows. Sections 1.1 and 1.2 provide a brief presentation of particle filters for single and multiple targets (more details can be found in [7, 4, 8, 13]), which will serve to highlight the versatility and drawbacks of this popular filtering method. Sections 1.3 and 1.4 demonstrate how one can use an extra step to improve the performance of particle filters for single and multiple targets. In particular, we discuss how Girsanov's theorem can be used to effect the extra step (more details in Appendix A). Section 2 describes the Monte Carlo sampling algorithm for computing the best association between observations and targets for the case of multiple targets. Section 3 presents numerical results for multi-target tracking for the cases of linear and nonlinear observation models. Finally, Section 4 contains a discussion of the results as well as directions for future work.

1 Particle filtering

Particle filters are a special case of sequential importance sampling methods. In Sections 1.1 and 1.2 we discuss the generic particle filter for a single and multiple targets respectively. In Sections 1.3 and 1.4 we discuss the addition of an extra step to the generic particle filter for the cases of a single and multiple targets respectively.

1.1 Generic particle filter for a single target

Suppose that we are given an SDE system and that we also have access to noisy observations Z_{T_1}, \dots, Z_{T_K} of the state of the system at specified instants T_1, \dots, T_K . The observations are functions of the state of the system, say given by $Z_{T_k} = G(X_{T_k}, \xi_k)$, where $\xi_k, k = 1, \dots, K$ are mutually independent random variables. For simplicity, let us assume that the distribution of the observations admits a density $g(X_{T_k}, Z_{T_k})$, i.e., $p(Z_{T_k}|X_{T_k}) \propto g(X_{T_k}, Z_{T_k})$.

The filtering problem consists of computing estimates of the conditional expectation $E[f(X_{T_k})|\{Z_{T_j}\}_{j=1}^k]$, i.e., the conditional expectation of the state of the system given the (noisy) observations. Equivalently, we are looking to compute the conditional density of the state of the system given the observations $p(X_{T_k}|\{Z_{T_j}\}_{j=1}^k)$. There are several ways to compute this conditional density and the associated conditional expectation but for practical applications they are rather expensive.

Particle filters fall in the category of importance sampling methods. Because computing averages with respect to the conditional density involves the sampling of the conditional density which can be difficult, importance sampling methods proceed by sampling a reference density $q(X_{T_k}|\{Z_{T_j}\}_{j=1}^k)$ which can be easily sampled and then compute the weighted sample mean

$$E[f(X_{T_k})|\{Z_{T_j}\}_{j=1}^k] \approx \frac{1}{N} \sum_{n=1}^N f(X_{T_k}^n) \frac{p(X_{T_k}^n|\{Z_{T_j}\}_{j=1}^k)}{q(X_{T_k}^n|\{Z_{T_j}\}_{j=1}^k)}$$

or the related estimate

$$E[f(X_{T_k})|\{Z_{T_j}\}_{j=1}^k] \approx \frac{\sum_{n=1}^N f(X_{T_k}^n) \frac{p(X_{T_k}^n|\{Z_{T_j}\}_{j=1}^k)}{q(X_{T_k}^n|\{Z_{T_j}\}_{j=1}^k)}}{\sum_{n=1}^N \frac{p(X_{T_k}^n|\{Z_{T_j}\}_{j=1}^k)}{q(X_{T_k}^n|\{Z_{T_j}\}_{j=1}^k)}}, \quad (1)$$

where N has been replaced by the approximation

$$N \approx \sum_{n=1}^N \frac{p(X_{T_k}^n|\{Z_{T_j}\}_{j=1}^k)}{q(X_{T_k}^n|\{Z_{T_j}\}_{j=1}^k)}.$$

Particle filtering is a recursive implementation of the importance sampling approach. It is based on the recursion

$$p(X_{T_k}|\{Z_{T_j}\}_{j=1}^k) \propto g(X_{T_k}, Z_{T_k})p(X_{T_k}|\{Z_{T_j}\}_{j=1}^{k-1}), \quad (2)$$

$$\text{where } p(X_{T_k}|\{Z_{T_j}\}_{j=1}^{k-1}) = \int p(X_{T_k}|X_{T_{k-1}})p(X_{T_{k-1}}|\{Z_{T_j}\}_{j=1}^{k-1})dX_{T_{k-1}}. \quad (3)$$

If we set

$$q(X_{T_k}|\{Z_{T_j}\}_{j=1}^k) = p(X_{T_k}|\{Z_{T_j}\}_{j=1}^{k-1}),$$

then from (2) we get

$$\frac{p(X_{T_k}|\{Z_{T_j}\}_{j=1}^k)}{q(X_{T_k}|\{Z_{T_j}\}_{j=1}^k)} \propto g(X_{T_k}, Z_{T_k}).$$

The approximation in expression (1) becomes

$$E[f(X_{T_i})|\{Z_{T_j}\}_{j=1}^k] \approx \frac{\sum_{n=1}^N f(X_{T_k}^n)g(X_{T_k}^n, Z_{T_k})}{\sum_{n=1}^N g(X_{T_k}^n, Z_{T_k})} \quad (4)$$

From (4) we see that if we can construct samples from the predictive distribution $p(X_{T_k}|\{Z_{T_j}\}_{j=1}^{k-1})$ then we can define the (normalized) weights $W_{T_k}^n = \frac{g(X_{T_k}^n, Z_{T_k})}{\sum_{n=1}^N g(X_{T_k}^n, Z_{T_k})}$, use them to weigh the samples and the weighted samples will be distributed according to the posterior distribution $p(X_{T_k}|\{Z_{T_j}\}_{j=1}^k)$.

In many applications, most samples will have a negligible weight with respect to the observation, so carrying them along does not contribute significantly to the conditional expectation estimate (this is the problem of degeneracy [8]). To create larger diversity one can resample the weights to create more copies of the samples with significant weights. The particle filter with resampling is summarized in the following algorithm due to Gordon *et al.* [7].

Particle filter for a single target

1. Begin with N unweighted samples $X_{T_{k-1}}^n$ from $p(X_{T_{k-1}}|\{Z_{T_j}\}_{j=1}^{k-1})$.
2. **Prediction:** Generate N samples $X_{T_k}^m$ from $p(X_{T_k}|X_{T_{k-1}})$.
3. **Update:** Evaluate the weights

$$W_{T_k}^n = \frac{g(X_{T_k}^m, Z_{T_k})}{\sum_{n=1}^N g(X_{T_k}^m, Z_{T_k})}.$$

4. **Resampling:** Generate N independent uniform random variables $\{\theta^n\}_{n=1}^N$ in $(0, 1)$. For $n = 1, \dots, N$ let $X_{T_k}^n = X_{T_k}^{l_j}$ where

$$\sum_{l=1}^{j-1} W_{T_k}^l \leq \theta^j < \sum_{l=1}^j W_{T_k}^l$$

where j can range from 1 to N .

5. Set $k = k + 1$ and proceed to Step 1.

The particle filter algorithm is easy to implement and adapt for different problems since the only part of the algorithm that depends on the

specific dynamics of the problem is the prediction step. This has led to the particle filter algorithm's increased popularity [4]. However, even with the resampling step, the particle filter can still need a lot of samples in order to describe accurately the conditional density $p(X_{T_k}|\{Z_{T_j}\}_{j=1}^k)$. Snyder *et al.* [15] have shown how the particle filter can fail in simple high dimensional problems because one sample dominates the weight distribution. The rest of the samples are not in statistically significant regions. Even worse, as we will show in the numerical results section, there are simple examples where not even one sample is in a statistically significant region. In the next subsection we present how drift relaxation can be used to push samples closer to statistically significant regions.

1.2 Generic particle filter for multiple targets

Suppose that we have $\lambda = 1, \dots, \Lambda$ targets. Also, for notational simplicity, assume that the λ th target comes from the λ th observation. Even when this is not the case, we can relabel the observations to satisfy this assumption. We will discuss in Section 2 how targets can be associated to observations. The targets are assumed to evolve independently so that the observation weight of a sample of the vector of targets is the product of the individual observation weights of the targets [13]. The same is true for the transition density of the vector of targets between observations. We denote the vector of targets at observation T_k by

$$X_{T_k} = (X_{1,T_k}, \dots, X_{\Lambda,T_k})$$

and the observation vector at T_k by

$$Z_{T_k} = (Z_{1,T_k}, \dots, Z_{\Lambda,T_k}).$$

Also, we can have different observation weight densities g_λ , $\lambda = 1, \dots, \Lambda$ for different targets. However, in the numerical examples we have chosen the same observation weight density for all targets.

Following [13] we can write the particle filter for the case of multiple targets as

Particle filter for multiple targets

1. Begin with N unweighted samples $X_{T_{k-1}}^n$ from $p(X_{T_{k-1}}|\{Z_{T_j}\}_{j=1}^{k-1}) = \prod_{\lambda=1}^{\Lambda} p(X_{\lambda,T_{k-1}}|\{Z_{\lambda,T_j}\}_{j=1}^{k-1})$.

2. **Prediction:** Generate N samples $X_{T_k}^n$ from

$$p(X_{T_k}|X_{T_{k-1}}) = \prod_{\lambda=1}^{\Lambda} p(X_{\lambda,T_k}|X_{\lambda,T_{k-1}}).$$

3. **Update:** Evaluate the weights

$$W_{T_k}^n = \frac{\prod_{\lambda=1}^{\Lambda} g_{\lambda}(X_{\lambda,T_k}^n, Z_{\lambda,T_k})}{\sum_{n=1}^N \prod_{\lambda=1}^{\Lambda} g_{\lambda}(X_{\lambda,T_k}^n, Z_{\lambda,T_k})}.$$

4. **Resampling:** Generate N independent uniform random variables $\{\theta^n\}_{n=1}^N$ in $(0, 1)$. For $n = 1, \dots, N$ let $X_{T_k}^n = X_{T_k}^{I_j}$ where

$$\sum_{l=1}^{j-1} W_{T_k}^l \leq \theta^j < \sum_{l=1}^j W_{T_k}^l$$

where j can range from 1 to N .

5. Set $k = k + 1$ and proceed to Step 1.

1.3 Particle filter with MCMC step for a single target

Several authors (see e.g. [6, 20]) have suggested the use of a MCMC step after the resampling step (Step 4) in order to move samples away from statistically insignificant regions. There are many possible ways to append an MCMC step after the resampling step in order to achieve that objective. The important point is that the MCMC step must preserve the conditional density $p(X_{T_k}|\{Z_{T_j}\}_{j=1}^k)$. In the current section we show that the MCMC step constitutes a case of conditional path sampling. In Appendix A we present an algorithm for conditional path sampling problem based on Girsanov's theorem and drift relaxation. We also discuss the necessary modifications needed in order to apply the conditional path sampling algorithm to the MCMC step of a particle filter.

We begin by noting that one can use the resampling step (Step 4) in the particle filter algorithm to create more copies not only of the good samples according to the observation, but also of the values (initial conditions) of the samples at the previous observation. These values are the ones who have evolved into good samples for the current observation (see more details in [20]). The motivation behind producing more copies of the pairs of initial and final conditions is to use the good initial conditions as starting points

to produce statistically more significant samples according to the current observation. This process can be accomplished in two steps. First, Step 4 of the particle filter algorithm is replaced by

Resampling: Generate N independent uniform random variables $\{\theta^n\}_{n=1}^N$ in $(0, 1)$. For $n = 1, \dots, N$ let $(X_{T_{k-1}}^n, X_{T_k}^n) = (X_{T_{k-1}}^{l_j}, X_{T_k}^{l_j})$ where

$$\sum_{l=1}^{j-1} W_{T_k}^l \leq \theta^j < \sum_{l=1}^j W_{T_k}^l$$

To motivate the second step we use the recursive particle filter formulas (2), (3) to write an update equation for conditional expectations. This is done to conform with the notation in Appendix A and elucidate how Girsanov's theorem and subsequently drift relaxation can be used to produce samples at the current observation which are statistically more significant. In fact, we have

$$E[f(X_{T_k})|\{Z_{T_j}\}_{j=1}^k] = \int f(X_{T_k}) \frac{g(X_{T_k}, Z_{T_k})p(X_{T_{k-1}}|\{Z_{T_j}\}_{j=1}^{k-1})dX_{T_{k-1}}dP}{\int g(X_{T_k}, Z_{T_k})p(X_{T_{k-1}}|\{Z_{T_j}\}_{j=1}^{k-1})dX_{T_{k-1}}dP}, \quad (5)$$

where P is the Wiener measure. Formula (5) can be approximated (using the initial conditions of the samples produced by the modified resampling step above) as follows:

$$E[f(X_{T_k})|\{Z_{T_j}\}_{j=1}^k] \approx \frac{1}{N} \sum_{n=1}^N \int f(X_{T_k}^n) \frac{g(X_{T_k}^n, Z_{T_k})dP}{\int g(X_{T_k}, Z_{T_k})p(X_{T_{k-1}}|\{Z_{T_j}\}_{j=1}^{k-1})dX_{T_{k-1}}dP}, \quad (6)$$

i.e., we have approximated the integration over the initial conditions $X_{T_{k-1}}$ appearing in (5) by an average over the initial conditions produced by the modified resampling step. Note that we have not approximated the integration over $X_{T_{k-1}}$ appearing in the denominator because the denominator is a normalization constant which will not be needed in the MCMC sampling performed during drift relaxation. Now, we can use Girsanov's theorem (see Appendix A) on the numerator and we have

$$E[f(X_{T_k})|\{Z_{T_j}\}_{j=1}^k] \approx \frac{1}{N} \sum_{n=1}^N \int f(Y_{T_k}^n) \frac{M_{T_k-T_{k-1}}^n g(Y_{T_k}^n, Z_{T_k})dP}{\int g(X_{T_k}, Z_{T_k})p(X_{T_{k-1}}|\{Z_{T_j}\}_{j=1}^{k-1})dX_{T_{k-1}}dP}, \quad (7)$$

where $Y_{T_k}^n$ is the value, at T_k , of the sample that starts at $X_{T_{k-1}}^n$ and evolves according to an SDE with modified drift (note that we still have not specified the modified drift). Also, $M_{T_k-T_{k-1}}^n$ is the Girsanov theorem correction to account for the modified drift.¹ The superscript n appearing in $M_{T_k-T_{k-1}}^n$ is to denote the dependence of the quantity on the initial condition at the previous observation. Also, it denotes the fact that we can use a *different* modified drift for each sample.

Equation (7) is of crucial importance. It can be interpreted in two ways. First, it means that the conditional expectation conditioned on the observations for the original SDE can be computed by constructing sample paths, from a modified SDE, starting from the *same* initial conditions at the previous observation and correcting for the drift modification by the quantity $M_{[T_{k-1}, T_k]}^n$. Second, and more importantly, it gives us a way to sample paths from a modified SDE which are statistically more significant according to the current observation *and* are also paths of the original SDE with high probability. This can be accomplished by starting from the same initial conditions at the previous observation and sampling paths according to the density

$$\frac{M_{T_k-T_{k-1}}^n g(Y_{T_k}^n, Z_{T_k})}{\int g(X_{T_k}, Z_{T_k}) p(X_{T_{k-1}} | \{Z_{T_j}\}_{j=1}^{k-1}) dX_{T_{k-1}} dP} \quad (8)$$

with respect to the Wiener measure P . This density can be sampled using the drift relaxation process described in Appendix A. The sampling of the density (8) is the second step in the process of producing samples that are statistically significant according to the current observation.

The approximation in equation (7) involves an integration over the Wiener measure. This is because for each of the initial conditions at the previous observation one can create different paths depending on the choice of the Brownian path. The integration over dP can be approximated by an average over Brownian paths and we find

$$E[f(X_{T_k}) | \{Z_{T_j}\}_{j=1}^k] \approx \frac{1}{NM} \sum_{n=1}^N \sum_{m=1}^M f(Y_{T_k}^{n,m}) \frac{M_{T_k-T_{k-1}}^{n,m} g(Y_{T_k}^{n,m}, Z_{T_k})}{\int g(X_{T_k}, Z_{T_k}) p(X_{T_{k-1}} | \{Z_{T_j}\}_{j=1}^{k-1}) dX_{T_{k-1}} dP}, \quad (9)$$

The approximation in (9) has an error of order $O(\frac{1}{\sqrt{NM}})$. This means that even if M is kept finite, the approximation still converges in the infinite

¹As is often the case, the SDE of interest is autonomous, and time can be reset to 0 at every observation and thus, Girsanov's theorem can be applied in $[0, T]$ instead of $[T_{k-1}, T_k]$. We use the more general form for the presentation for the sake of completeness.

limit for N . In the numerical implementation we pick $M = 1$ and focus our computational resources on the averaging over initial conditions.

We are now in a position to present the particle filter with MCMC step algorithm

Particle filter with MCMC step for a single target

1. Begin with N unweighted samples $X_{T_{k-1}}^n$ from $p(X_{T_{k-1}}|\{Z_{T_j}\}_{j=1}^{k-1})$.
2. **Prediction:** Generate N samples $X_{T_k}^n$ from $p(X_{T_k}|X_{T_{k-1}})$.
3. **Update:** Evaluate the weights

$$W_{T_k}^n = \frac{g(X_{T_k}^n, Z_{T_k})}{\sum_{n=1}^N g(X_{T_k}^n, Z_{T_k})}.$$

4. **Resampling:** Generate N independent uniform random variables $\{\theta^n\}_{n=1}^N$ in $(0, 1)$. For $n = 1, \dots, N$ let $(X_{T_{k-1}}^n, X_{T_k}^n) = (X_{T_{k-1}}^{l_j}, X_{T_k}^{l_j})$ where

$$\sum_{l=1}^{j-1} W_{T_k}^l \leq \theta^j < \sum_{l=1}^j W_{T_k}^l$$

where j can range from 1 to N .

5. **MCMC step:** For $n = 1, \dots, N$ choose a modified drift (possibly different for each n). Construct a path for the SDE with the modified drift starting from $X_{T_{k-1}}^n$ and ending at $X_{T_k}^n$. Construct through drift relaxation (see Appendix A) a Markov chain $\{Y_{T_k}^{n,l}\}_{l=0}^\Lambda$ with initial value $Y_{T_k}^{n,0} = X_{T_k}^n$ and stationary distribution

$$\frac{M_{[T_{k-1}, T_k]}^n g(Y, Z_{T_k})}{\int g(X_{T_k}, Z_{T_k}) p(X_{T_{k-1}}|\{Z_{T_j}\}_{j=1}^{k-1}) dX_{T_{k-1}} dP}.$$

6. Set $X_{T_k}^n = Y_{T_k}^{n,\Lambda}$.
7. Set $k = k + 1$ and proceed to Step 1.

Since the samples $X_{T_k}^n = Y_{T_k}^{n,\Lambda}$ are constructed by starting from different sample paths, they are independent. Also, note that the samples $X_{T_k}^n$ are unweighted. However, we can still measure how well these samples approximate the posterior density by comparing the effective sample sizes of the

particle filter with and without the MCMC step. For a collection of N samples the effective sample size $ess(T_k)$ is defined by

$$ess(T_k) = \frac{N}{1 + C_k^2}$$

where

$$C_k = \frac{1}{W_k} \sqrt{\frac{1}{N} \sum_{n=1}^N (g(X_{T_k}^n, Z_{T_k}) - W_k)^2} \quad \text{and} \quad W_k = \frac{1}{N} \sum_{n=1}^N g(X_{T_k}^n, Z_{T_k}).$$

The effective sample size can be interpreted as that the N weighted samples are worth of $ess(T_k) = \frac{N}{1+C_k^2}$ i.i.d. samples drawn from the target density, which in our case is the posterior density. By definition, $ess(T_k) \leq N$. If the samples have uniform weights, then $ess(T_k) = N$. On the other hand, if all samples but one have zero weights, then $ess(T_k) = 1$.

1.4 Particle filter with MCMC step for multiple targets

We discuss now the case of multiple, say Λ , targets. Instead of the observations for a single target now we have a collection of observations for all the targets $\{Z_{T_j}\}_{j=1}^k = \{(Z_{T_j}^1, \dots, Z_{T_j}^\Lambda)\}_{j=1}^k$. There are two cases which we now examine. First, the function whose conditional expectation we want to compute depends only on one of the targets. Second, the function whose conditional expectation we want to compute depends at least on two targets.

We begin with the first case. Recall that the targets evolve independently and after the appropriate association between a target and an observation, we have for the conditional expectation estimate for the λ th target ($\lambda = 1, \dots, \Lambda$)

$$\begin{aligned} & E[f(X_{\lambda, T_k}) | \{Z_{T_j}\}_{j=1}^k] \\ &= \int f(X_{\lambda, T_k}) \frac{\prod_{\lambda'=1}^{\Lambda} g_{\lambda'}(X_{\lambda', T_k}, Z_{\lambda', T_k}) p(X_{\lambda', T_{k-1}} | \{Z_{\lambda', T_j}\}_{j=1}^{k-1}) dX_{\lambda', T_{k-1}} dP_{\lambda'}}{\int \prod_{\lambda'=1}^{\Lambda} g_{\lambda'}(X_{\lambda', T_k}, Z_{\lambda', T_k}) p(X_{\lambda', T_{k-1}} | \{Z_{\lambda', T_j}\}_{j=1}^{k-1}) dX_{\lambda', T_{k-1}} dP_{\lambda'}} \\ &= \int f(X_{\lambda, T_k}) \frac{g(X_{\lambda, T_k}, Z_{\lambda, T_k}) p(X_{\lambda, T_{k-1}} | \{Z_{\lambda, T_j}\}_{j=1}^{k-1}) dX_{\lambda, T_{k-1}} dP_{\lambda}}{\int g(X_{\lambda, T_k}, Z_{\lambda, T_k}) p(X_{\lambda, T_{k-1}} | \{Z_{\lambda, T_j}\}_{j=1}^{k-1}) dX_{\lambda, T_{k-1}} dP_{\lambda}} \end{aligned} \tag{10}$$

where the second identity comes from integrating out all the targets except for the λ th target. Due to this simplification we are back at the case of

a single target as in equation (5). So, we can apply Girsanov's theorem and, subsequently, perform the Girsanov MCMC step *individually* for each target. The difference with the single target case is that here we can have a different Girsanov weight $M_{\lambda, [T_{k-1}, T_k]}^n$ for each target $\lambda = 1 \dots, \Lambda$. The particle filter with MCMC step for the case of multiple targets is

Particle filter with MCMC step for multiple targets

1. Begin with N unweighted samples $X_{T_{k-1}}^n$ from $p(X_{T_{k-1}} | \{Z_{T_j}\}_{j=1}^{k-1}) = \prod_{\lambda=1}^{\Lambda} p(X_{\lambda, T_{k-1}} | \{Z_{\lambda, T_j}\}_{j=1}^{k-1})$.
2. **Prediction:** Generate N samples $X_{T_k}^m$ from

$$p(X_{T_k} | X_{T_{k-1}}) = \prod_{\lambda=1}^{\Lambda} p(X_{\lambda, T_k} | X_{\lambda, T_{k-1}}).$$

3. **Update:** Evaluate the weights

$$W_{T_k}^n = \frac{\prod_{\lambda=1}^{\Lambda} g_{\lambda}(X_{\lambda, T_k}^m, Z_{\lambda, T_k})}{\sum_{n=1}^N \prod_{\lambda=1}^{\Lambda} g_{\lambda}(X_{\lambda, T_k}^m, Z_{\lambda, T_k})}.$$

4. **Resampling:** Generate N independent uniform random variables $\{\theta^n\}_{n=1}^N$ in $(0, 1)$. For $n = 1, \dots, N$ let $(X_{T_{k-1}}^n, X_{T_k}^n) = (X_{T_{k-1}}^{j'}, X_{T_k}^{j'})$ where

$$\sum_{l=1}^{j-1} W_{T_k}^l \leq \theta^j < \sum_{l=1}^j W_{T_k}^l$$

where j can range from 1 to N .

5. **MCMC step:** For $n = 1, \dots, N$ and $\lambda = 1, \dots, \Lambda$ choose a modified drift (possibly different for each n and each λ). Construct a path for the SDE with the modified drift starting from $X_{\lambda, T_{k-1}}^n$ and ending at X_{λ, T_k}^n . Construct through drift relaxation (see Appendix A) a Markov chain $\{Y_{\lambda, T_k}^{n, l}\}_{l=0}^L$ with initial value $Y_{\lambda, T_k}^{n, 0} = X_{\lambda, T_k}^n$ and stationary distribution

$$\frac{M_{\lambda, [T_{k-1}, T_k]}^n g(Y_{\lambda}, Z_{T_k})}{\int g(X_{\lambda, T_k}, Z_{\lambda, T_k}) p(X_{T_{k-1}} | \{Z_{\lambda, T_j}\}_{j=1}^{k-1}) dX_{\lambda, T_{k-1}} dP}. \quad (11)$$

6. Set $X_{\lambda, T_k}^n = Y_{\lambda, T_k}^{n, L}$.

7. Set $k = k + 1$ and proceed to Step 1.

For a collection of N samples the effective sample size $ess_\Lambda(T_k)$ for Λ targets is

$$ess_\Lambda(T_k) = \frac{N}{1 + C_{\Lambda,k}^2}$$

where

$$C_{\Lambda,k} = \frac{1}{W_{\Lambda,k}} \sqrt{\frac{1}{N} \sum_{n=1}^N \left(\prod_{\lambda=1}^{\Lambda} g_\lambda(X_{\lambda,T_k}^n, Z_{\lambda,T_k}) - W_{\Lambda,k} \right)^2}$$

and $W_{\Lambda,k} = \frac{1}{N} \sum_{n=1}^N \prod_{\lambda=1}^{\Lambda} g_\lambda(X_{\lambda,T_k}^n, Z_{\lambda,T_k})$.

Finally, we mention the necessary modifications to compute conditional expectation estimates of a function $h(X_{1,T_k}, \dots, X_{\Lambda,T_k}) = h(X_{T_k})$ that depends, in general, on all targets.

$$\begin{aligned} & E[h(X_{T_k}) | \{Z_{T_j}\}_{j=1}^k] \\ &= \int h(X_{T_k}) \frac{\prod_{\lambda'=1}^{\Lambda} g_{\lambda'}(X_{\lambda',T_k}, Z_{\lambda',T_k}) p(X_{\lambda',T_{k-1}} | \{Z_{\lambda',T_j}\}_{j=1}^{k-1}) dX_{\lambda',T_{k-1}} dP_{\lambda'}}{\int \prod_{\lambda'=1}^{\Lambda} g'_{\lambda'}(X_{\lambda',T_k}, Z_{\lambda',T_k}) p(X_{\lambda',T_{k-1}} | \{Z_{\lambda',T_j}\}_{j=1}^{k-1}) dX_{\lambda',T_{k-1}} dP_{\lambda'}} \\ &\approx \frac{1}{N} \sum_{n=1}^N \int h(X_{T_k}^n) \frac{\prod_{\lambda'=1}^{\Lambda} g_{\lambda'}(X_{\lambda',T_k}^n, Z_{\lambda',T_k}) dP_{\lambda'}}{\int \prod_{\lambda'=1}^{\Lambda} g'_{\lambda'}(X_{\lambda',T_k}, Z_{\lambda',T_k}) p(X_{\lambda',T_{k-1}} | \{Z_{\lambda',T_j}\}_{j=1}^{k-1}) dX_{\lambda',T_{k-1}} dP_{\lambda'}} \\ &= \frac{1}{N} \sum_{n=1}^N \int h(Y_{T_k}^n) \frac{\prod_{\lambda'=1}^{\Lambda} M_{\lambda',[T_{k-1},T_k]}^n g_{\lambda'}(Y_{\lambda',T_k}^n, Z_{\lambda',T_k}) dP_{\lambda'}}{\int \prod_{\lambda'=1}^{\Lambda} g'_{\lambda'}(X_{\lambda',T_k}, Z_{\lambda',T_k}) p(X_{\lambda',T_{k-1}} | \{Z_{\lambda',T_j}\}_{j=1}^{k-1}) dX_{\lambda',T_{k-1}} dP_{\lambda'}} \end{aligned}$$

where the last equality was obtained by applying Girsanov's theorem for each target. In order to carry out the MCMC step in this case one has to construct a Markov chain with stationary density

$$\rho_n(Y) = \frac{\prod_{\lambda'=1}^{\Lambda} M_{\lambda',[T_{k-1},T_k]}^n g_{\lambda'}(Y, Z_{\lambda',T_k})}{\int \prod_{\lambda'=1}^{\Lambda} g'_{\lambda'}(X_{\lambda',T_k}, Z_{\lambda',T_k}) p(X_{\lambda',T_{k-1}} | \{Z_{\lambda',T_j}\}_{j=1}^{k-1}) dX_{\lambda',T_{k-1}} dP_{\lambda'}} \quad (12)$$

for each sample $n = 1, \dots, N$. The subscript n in $\rho_n(Y)$ denotes, as before, the fact that we may use a *different* modified drift for each sample. The sampling can be performed as before through the process of drift relaxation

described in Appendix A. The algorithm described above needs to be modified only in the MCMC step where one needs to construct a Markov chain with stationary density given by (12).

2 Monte Carlo target-observation association algorithm

As in any other sequential Monte Carlo algorithm for multi-target tracking (see e.g. [13] and references therein) we need to associate, for each sample, the evolving targets to the observations. The association problem is sensitive to the tracking accuracy of the algorithm. If we cannot follow accurately each target and two or more targets are close, then the association algorithm can assign the wrong observations to the targets. After a few observation steps this can lead to the inability to follow a target's true track anymore. There are different ways to perform the target-observation association. The ones that we are aware of are based on various types of assignment algorithms first developed in the context of computer science [3, 2]. Here we have decided on using a different algorithm to perform the target-observation association. In particular, we have designed a simple Metropolis Monte Carlo algorithm which effects a probabilistic search of the space of possible associations to find the best target-observation association.

Suppose that at observation time t we have K_t observations which include surviving and possible newborn targets. Also, suppose that for each target we have at each step J observation values for different quantities depending on the target's state. For simplicity, we assume that we observe the same quantities for all targets. Usually, one observes position related quantities, like the x, y position or the bearing and range of the target. The observation model is given by

$$Z_{k,j,t} = g_j(\mathbf{x}_{k,t}) + \xi_{k,j}, \quad \text{for } k = 1, \dots, K_t \text{ and } j = 1, \dots, J, \quad (13)$$

where for $k = 1, \dots, K_t$ and $j = 1, \dots, J$, $\xi_{k,j}$, are i.i.d. random variables. For simplicity let us assume that the $\xi_{k,j}$ are $\sim N(0, \sigma_j^2)$. Note that the observation model (13) *assumes* that we know that the k th observation comes from the k th target. However, in reality, we do not have such information and we need to make an association between observations and targets. Define an association map A given by $A(k) = m_k$ where $k, m_k = 1, \dots, K_t$. The association map assigns to the k th observation the m_k th target. For K_t observations there are $K_t!$ different observation-target association maps.

For a specific association of each observation to some target, the likelihood function for the collection of observations for the n th sample is given by

$$\begin{aligned} \rho(\mathbf{x}_{1,t}^n, \dots, \mathbf{x}_{K_t,t}^n, Z_{1,1,t}, \dots, Z_{1,J,t}, \dots, Z_{K_t,J,t}) \\ \propto \prod_{k=1}^{K_t} \prod_{j=1}^J \exp \left[-\frac{(Z_{k,j,t} - g_j(\mathbf{x}_{A(k),t}^n))^2}{2 * \sigma_j^2} \right] \end{aligned} \quad (14)$$

where the proportionality is up to an (immaterial for our purposes) normalization constant. If we define

$$\begin{aligned} H_{obs}^A(\mathbf{x}_{1,t}^n, \dots, \mathbf{x}_{K_t,t}^n, Z_{1,1,t}, \dots, Z_{1,J,t}, \dots, Z_{K_t,J,t}) \\ = \sum_{k=1}^{K_t} \sum_{j=1}^J \frac{(Z_{k,j,t} - g_j(\mathbf{x}_{A(k),t}^n))^2}{2 * \sigma_j^2} \end{aligned}$$

we can rewrite ρ (omitting all the arguments of ρ and H_{obs}^A for simplicity) as

$$\rho \propto \exp[-H_{obs}^A] \quad (15)$$

The superscript A is to denote the dependence of the value of H_{obs}^A on the specific association map A . The best association map is the one which maximizes ρ . By definition, $H_{obs}^A \geq 0$ and thus the best association is the one which minimizes H_{obs}^A . We can find the best association map by standard Metropolis Monte Carlo sampling on the space of association maps, using the density ρ (see e.g. [8] about Metropolis Monte Carlo sampling).

As in every Metropolis Monte Carlo sampling algorithm there is arbitrariness in the new configuration (association map in our case) proposal. We have tried two different proposal schemes which performed equally well, at least for cases up to 4 targets that we examined in our numerical experiments. The first proposal scheme constructs a new observation-target association map from scratch. This means that for each observation in the observation vector we choose a new target to associate it with. Note that if there is only one target there is no need for sampling since there is only one possible observation-target association. The second proposal scheme starts with an initial association map. Then it chooses randomly a pair of observations and their associated targets and exchanges the associated targets. This allows one to build a good association map incrementally. We expect that the second proposal scheme will be advantageous in the case when the number of observations K_t at the observation instant t is large. Recall that the number of possible association maps grows as $K_t!$, and it becomes prohibitively large for an exhaustive search even for moderate values of K_t .

The numerical results we present here are for the first proposal scheme. The Metropolis sampling algorithm was run with 10000 Metropolis accept/reject steps and we kept the last accepted association map.

3 Numerical results

We present numerical results for multi-target tracking using the particle filter with an MCMC step performed by drift relaxation and hybrid Monte Carlo. We have synthesized tracks of targets moving on the xy plane using a $2D$ near constant velocity model [1]. At each time t we have a total of K_t targets and the evolution of the k th target ($k = 1, \dots, K_t$) is given by

$$\begin{aligned} \mathbf{x}_{k,t} &= \mathbf{A}\mathbf{x}_{k,t-1} + \mathbf{B}\mathbf{v}_{k,t} \\ &= [x_{k,t}, \dot{x}_{k,t}, y_{k,t}, \dot{y}_{k,t}]^T, \end{aligned} \quad (16)$$

where $(x_{k,t}, \dot{x}_{k,t})$ and $(y_{k,t}, \dot{y}_{k,t})$ are the xy position and velocity of the k th target at time t . The matrices \mathbf{A} and \mathbf{B} are given by

$$\mathbf{A} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} T^2/2 & 0 \\ T & 0 \\ 0 & T^2/2 \\ 0 & T \end{bmatrix}, \quad (17)$$

where T is the time between observations. For the experiments we have set $T = 1$, i.e., noisy observations of the model are obtained at every step of the model (16). The model noise $\mathbf{v}_{k,t}$ is a collection of independent Gaussian random variables with covariance matrix Σ_v defined as

$$\Sigma_v = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix}. \quad (18)$$

In the experiments we have $\sigma_x^2 = \sigma_y^2 = 1$. Also, we have considered two possible cases for the observation model, one linear and one nonlinear. Due to the different possible combinations of targets to observations we use a different index m to denote the observations. Since we do not assume any clutter we have $m = 1, \dots, K_t$. If the m th observation $\mathbf{z}_{m,t}$ at time t comes from the k th target we have

$$\mathbf{z}_{m,t} = \begin{bmatrix} x_{k,t} \\ y_{k,t} \end{bmatrix} + \mathbf{w}_{m,t} \quad (19)$$

for the linear observation model and

$$\mathbf{z}_{m,t} = \begin{bmatrix} \arctan(\frac{y_{k,t}}{x_{k,t}}) \\ (x_{k,t}^2 + y_{k,t}^2)^{1/2} \end{bmatrix} + \mathbf{w}_{m,t} \quad (20)$$

for the nonlinear observation model. As is usual in the literature, the nonlinear observation model consists of the bearing θ and range r of a target. The observation noise $\mathbf{w}_{m,t}$ is white and Gaussian with covariance matrix

$$\Sigma_w = \begin{bmatrix} \sigma_{obs,x}^2 & 0 \\ 0 & \sigma_{obs,y}^2 \end{bmatrix} \quad (21)$$

for the linear observation model and

$$\Sigma_w = \begin{bmatrix} \sigma_\theta^2 & 0 \\ 0 & \sigma_r^2 \end{bmatrix} \quad (22)$$

for the nonlinear observation model. For the numerical experiments with the linear observation model we chose $\sigma_{obs,x}^2 = \sigma_{obs,y}^2 = 1$. For the numerical experiments with the nonlinear observation model we chose $\sigma_\theta^2 = 10^{-4}$ and $\sigma_r^2 = 1$. These values make our example comparable in difficulty to examples appearing in the literature (see e.g. [13, 18, 19]).

The synthesized target tracks were created by specifying a certain scenario, to be detailed below, of surviving, newborn and disappearing targets. According to this scenario we evolved the appropriate number of targets according to (16) and recorded the state of each target at each step. For the surviving targets we created an observation by using the state of the target in the observation model. Thus, for the linear observation model, the observations were created directly in xy space by perturbing the xy position of the target by (19). For the nonlinear observation model, the observations were created in bearing and range space θ, r by using (20). The perturbed bearing and range were transformed to xy space by the transformation $x = r \cos \theta$, $y = r \sin \theta$ to create a position for the target in xy space.

The newborn targets for the linear model were created in xy space directly by sampling uniformly in $[-100, 100]$. Afterwards, the observations of the newborn targets were constructed by perturbing the x, y positions using (19). The newborn targets for the nonlinear model were created in xy space by sampling uniformly in $[-100, 100]$. Afterwards, we transformed the x, y positions to the bearing and range space θ, r and perturbed the bearing and range according to (20). The perturbed bearing and range were again transformed back to xy space to create the position of the newborn target. Note

that both observation models do *not* involve the velocities. The newborn target velocities were sampled uniformly in $[-1, 1]$.

The number of targets at each observation instant is: $K_0 = 2$, $K_1 = 2$, $K_2 = 1$, $K_3 = 2$, $K_4 = 3$, $K_5 = K_6 = \dots = K_{200} = 4$. So, for the majority of the steps we have 4 targets which makes the problem of tracking rather difficult.

3.1 Girsanov change of measure for near constant velocity model

In order to apply the drift relaxation process we need to find the Girsanov change of measure formula. Note that the near constant velocity model is discrete while Girsanov's theorem is applicable to continuous time Itô processes. Thus, we need to formulate the drift relaxation scheme in continuous time and then discretize. Since the targets evolve independently we will write the Girsanov formula for one target with the obvious generalization for the case of more than one targets. The $2D$ near constant velocity model (16) is a simplified discrete approximation of the formal linear SDE system

$$\begin{aligned}\ddot{x}_t &= \sigma_x \dot{u}_{x,t} \\ \ddot{y}_t &= \sigma_y \dot{u}_{y,t},\end{aligned}\tag{23}$$

where $u_{x,t}, u_{y,t}$ are independent Brownian motions and $\sigma_x \dot{u}_{x,t}, \sigma_y \dot{u}_{y,t}$ are Gaussian white noises with covariances $R_x(t) = \sigma_x^2 \delta(t)$ and $R_y(t) = \sigma_y^2 \delta(t)$ respectively and $\delta(t)$ is the delta function. We can define velocities in the x and y directions, $p_{x,t} = \dot{x}_t$ and $p_{y,t} = \dot{y}_t$ and we can rewrite (23) as

$$\begin{aligned}\dot{x}_t &= p_{x,t}, \\ \dot{p}_{x,t} &= \sigma_x \dot{u}_{x,t}, \\ \dot{y}_t &= p_{y,t}, \\ \dot{p}_{y,t} &= \sigma_y \dot{u}_{y,t}.\end{aligned}\tag{24}$$

The formal system (24) can be written rigorously

$$\begin{aligned}dx_t &= p_{x,t} dt, \\ dp_{x,t} &= \sigma_x du_{x,t}, \\ dy_t &= p_{y,t} dt, \\ dp_{y,t} &= \sigma_y du_{y,t}.\end{aligned}\tag{25}$$

The equations for the positions have zero noise and the equations for the velocities have zero drift. We can write the SDE system (25) in matrix form as

$$\begin{bmatrix} dx_t \\ dp_{x,t} \\ dy_t \\ dp_{y,t} \end{bmatrix} = \begin{bmatrix} p_{x,t} \\ 0 \\ p_{y,t} \\ 0 \end{bmatrix} dt + \begin{bmatrix} 0 & 0 \\ \sigma_x & 0 \\ 0 & 0 \\ 0 & \sigma_y \end{bmatrix} \begin{bmatrix} du_{x,t} \\ du_{y,t} \end{bmatrix}. \quad (26)$$

Define the 4×1 vectors $z_t = [z_{1,t}, \dots, z_{4,t}]^T$, $a(z_t) = [a_{1,t}, \dots, a_{4,t}]^T$, the 2×1 vector $w_t = [w_{1,t}, w_{2,t}]^T$ and a 4×2 constant matrix σ by

$$z_t = \begin{bmatrix} x_t \\ p_{x,t} \\ y_t \\ p_{y,t} \end{bmatrix}, \quad a(z_t) = \begin{bmatrix} p_{x,t} \\ 0 \\ p_{y,t} \\ 0 \end{bmatrix}, \quad w_t = \begin{bmatrix} u_{x,t} \\ u_{y,t} \end{bmatrix} \quad \text{and} \quad \sigma = \begin{bmatrix} 0 & 0 \\ \sigma_x & 0 \\ 0 & 0 \\ 0 & \sigma_y \end{bmatrix}.$$

Note that

$$\begin{bmatrix} z_{1,t} \\ z_{2,t} \\ z_{3,t} \\ z_{4,t} \end{bmatrix} = \begin{bmatrix} x_t \\ p_{x,t} \\ y_t \\ p_{y,t} \end{bmatrix} \quad \text{and} \quad a(z_t) = \begin{bmatrix} z_{2,t} \\ 0 \\ z_{4,t} \\ 0 \end{bmatrix}.$$

With these definitions we can rewrite the SDE system (26) as

$$dz_t = a(z_t)dt + \sigma dw_t \quad (27)$$

We will apply Girsanov's theorem to the SDE system (27). Consider the linear SDE system

$$dz'_t = b(z'_t)dt + \sigma dw_t \quad (28)$$

where

$$b(z'_t) = \begin{bmatrix} z'_{2,t} \\ \mu_x \\ z'_{4,t} \\ \mu_y \end{bmatrix}$$

and μ_x, μ_y are drifts that we can choose. In our numerical experiments we chose μ_x, μ_y to be constant. In order to apply Girsanov's theorem (see Theorem 8.6.8 in [14]) we need to find a 2×1 vector $\rho = [\rho_1, \rho_2]^T$ such that $\sigma\rho = b(\lambda) - a(\lambda)$ for $\lambda \in \mathbb{R}^4$. This means that we need to find ρ such that

$$\begin{bmatrix} 0 & 0 \\ \sigma_x & 0 \\ 0 & 0 \\ 0 & \sigma_y \end{bmatrix} \begin{bmatrix} \rho_1 \\ \rho_2 \end{bmatrix} = \begin{bmatrix} \lambda_2 - \lambda_2 \\ \mu_x - 0 \\ \lambda_4 - \lambda_4 \\ \mu_y - 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \mu_x \\ 0 \\ \mu_y \end{bmatrix}. \quad (29)$$

The system (29) reduces to

$$\sigma_x \rho_1 = \mu_x \quad \text{and} \quad \sigma_y \rho_2 = \mu_y,$$

which has the solution $\rho_1 = \frac{\mu_x}{\sigma_x}$ and $\rho_2 = \frac{\mu_y}{\sigma_y}$. The Girsanov change of measure transformation with respect to (28) on an interval $[0, T]$ is given by (see (46) in Appendix A)

$$\begin{aligned} M_T &= \exp\left(-\int_0^T \left[\frac{\mu_x}{\sigma_x} dw_{1,s} + \frac{\mu_y}{\sigma_y} dw_{2,s}\right] - \frac{1}{2} \int_0^T \left\{ \left[\frac{\mu_x}{\sigma_x}\right]^2 + \left[\frac{\mu_y}{\sigma_y}\right]^2 \right\} ds\right) \\ &= \exp\left(-\int_0^T \left[\frac{\mu_x}{\sigma_x} du_{x,s} + \frac{\mu_y}{\sigma_y} du_{y,s}\right] - \frac{1}{2} \int_0^T \left\{ \left[\frac{\mu_x}{\sigma_x}\right]^2 + \left[\frac{\mu_y}{\sigma_y}\right]^2 \right\} ds\right) \\ &= \exp\left(-\left[\frac{\mu_x}{\sigma_x} u_{x,T} + \frac{\mu_y}{\sigma_y} u_{y,T}\right] - \frac{T}{2} \left\{ \left[\frac{\mu_x}{\sigma_x}\right]^2 + \left[\frac{\mu_y}{\sigma_y}\right]^2 \right\}\right) \quad (30) \end{aligned}$$

since μ_x, μ_y are constant and $u_{x,t}, u_{y,t}$ are Brownian motions which are zero at $t = 0$.

For the l th level ($l = 1, \dots, L$) of the drift relaxation process we have instead of the SDE system

$$dz_t = a(z_t)dt + \sigma dw_t,$$

the SDE system

$$dz_t^l = (1 - \epsilon_l)b(z_t^l) + \epsilon_l a(z_t^l)dt + \sigma dw_t.$$

The Girsanov change of measure transformation with respect to (28) for the l th level (see (50) in Appendix A) becomes

$$M_T^{\epsilon_l} = \exp\left(-\epsilon_l \left[\frac{\mu_x}{\sigma_x} u_{x,T} + \frac{\mu_y}{\sigma_y} u_{y,T}\right] - (\epsilon_l)^2 \frac{T}{2} \left\{ \left[\frac{\mu_x}{\sigma_x}\right]^2 + \left[\frac{\mu_y}{\sigma_y}\right]^2 \right\}\right).$$

In the numerical experiments we used $L = 10$ levels with $\epsilon_l = l/10$, $l = 1, \dots, 10$.

3.1.1 The choice of the drift for the modified system

We proceed to comment on the values of μ_x, μ_y . Since the particle filter is a recursive algorithm we are interested in computing μ_x, μ_y between two consecutive observations. Thus, it is enough to show how to compute μ_x, μ_y in the interval $[0, T]$, where T is the time interval between two consecutive observations. In our experiments we have chosen $T = 1$.

We suppose that we are at observation time T and we have N samples. After the resampling step at observation time 0, we have obtained a collection of pairs (z_0^n, z_T^n) , for $n = 1, \dots, N$. When μ_x, μ_y , are constant the system (28) can be solved explicitly. In particular, we have for the position and velocity in the x direction

$$\begin{aligned} z'_{1,T} &= z'_{1,0} + z'_{2,0}T + \mu_x \frac{T^2}{2} + \int_0^T \sigma_x u_{x,s} ds \\ z'_{2,T} &= z'_{2,0} + \mu_x T + \sigma_x u_{x,T}. \end{aligned} \quad (31)$$

where we have used the fact that the Brownian motion $u_{x,t}$ is zero at $t = 0$. For the n th sample, if we use $z'_{1,0} = z_{1,0}^n$ and $z'_{2,0} = z_{2,0}^n$ we have

$$\begin{aligned} z_{1,T}^n &= z_{1,0}^n + z_{2,0}^n T + \mu_x \frac{T^2}{2} + \int_0^T \sigma_x u_{x,s}^n ds \\ z_{2,T}^n &= z_{2,0}^n + \mu_x T + \sigma_x u_{x,T}^n, \end{aligned} \quad (32)$$

where $u_{x,s}^n, s \in [0, T]$ is a *new* Brownian path for each sample. Define μ_x as

$$\mu_x = \frac{1}{N} \sum_{n=1}^N \mu_x^n \quad \text{with} \quad \mu_x^n = \frac{z_{1,T}^n - z_{1,0}^n}{T^2/2} - \frac{2z_{2,0}^n}{T} \quad (33)$$

The motivation behind this definition comes from considering the usual case where the resampling step of the particle filter has produced only copies of one sample, say the n_0 th sample. Then $\mu_x = \mu_x^{n_0} = \frac{z_{1,T}^{n_0} - z_{1,0}^{n_0}}{T^2/2} - \frac{2z_{2,0}^{n_0}}{T}$. From (32) we get

$$z_{1,T}^{n_0} = z_{1,0}^{n_0} + \int_0^T \sigma_x u_{x,s}^{n_0} ds.$$

Due to the symmetry of Brownian motion around 0, the values $z_{1,T}^{n_0}$ are symmetrically distributed around $z_{1,0}^{n_0}$. Since our objective during the MCMC step is to bring the state of the sample closer to the observation, it is advantageous to start from a symmetrically perturbed state of the sample that the generic particle filter steps have produced. Also note, that in our numerical experiments the observation functions depend only on the position, so choosing a drift that explores the area around the sample's position produced by the generic particle filter steps is again to our advantage. Also, in the numerical experiments we have made the approximation $\int_0^T \sigma_x u_{x,s}^n ds \approx T \sigma_x u_{x,T}^n$.

Similarly, for the drift μ_y in the y direction we define

$$\mu_y = \frac{1}{N} \sum_{n=1}^N \mu_y^n \quad \text{with} \quad \mu_y^n = \frac{z_{3,T}^n - z_{3,0}^n}{T^2/2} - \frac{2z_{4,0}^n}{T}. \quad (34)$$

3.1.2 Hybrid Monte Carlo formulation

We present briefly the hybrid Monte Carlo (HMC) formulation that we have used to sample the conditional density for each target. Since the targets evolve independently of one another and we are interested in computing conditional expectation estimates of functions that depend only on one target, we need to formulate HMC only for the case of a single target. We will formulate HMC for the l th level of the drift relaxation process. We have already seen that the Girsanov density at the l th level can be written as

$$M_T^{\epsilon_l} = \exp\left(-\epsilon_l \left[\frac{\mu_x}{\sigma_x} u_{x,T} + \frac{\mu_y}{\sigma_y} u_{y,T} \right] - (\epsilon_l)^2 \frac{T}{2} \left\{ \left[\frac{\mu_x}{\sigma_x} \right]^2 + \left[\frac{\mu_y}{\sigma_y} \right]^2 \right\}\right).$$

Thus, the conditional density that we have to sample for the linear observation model at the l th level is given by (see (11) in particle filter algorithm with MCMC step for multiple targets)

$$\begin{aligned} M_T^{\epsilon_l} g_x(Z_{1,T}, z_{1,T}^m) g_y(Z_{3,T}, z_{3,T}^m) \\ \propto \exp\left(-\epsilon_l \left[\frac{\mu_x}{\sigma_x} u_{x,T} + \frac{\mu_y}{\sigma_y} u_{y,T} \right] - (\epsilon_l)^2 \frac{T}{2} \left\{ \left[\frac{\mu_x}{\sigma_x} \right]^2 + \left[\frac{\mu_y}{\sigma_y} \right]^2 \right\}\right) \\ \exp\left[-\frac{(Z_{1,T} - z_{1,T}^m)^2}{2 * \sigma_{obs,x}^2}\right] \exp\left[-\frac{(Z_{3,T} - z_{3,T}^m)^2}{2 * \sigma_{obs,y}^2}\right] \end{aligned} \quad (35)$$

where $Z_{1,T}$ the observation value of the x position of the target and $Z_{3,T}$ the observation value of the y position of the target. Note that the x, y positions $z_{1,T}^m, z_{3,T}^m$ of the n th sample are determined by

$$z_{1,T}^m \approx z_{1,0}^n + z_{2,0}^n T + \mu_x \frac{T^2}{2} + T \sigma_x u_{x,T}^n \quad (36)$$

$$z_{3,T}^m \approx z_{3,0}^n + z_{4,0}^n T + \mu_y \frac{T^2}{2} + T \sigma_y u_{y,T}^n \quad (37)$$

where we have used the fact that we go in one step of size T from one observation to the next to make the approximations $\int_0^T \sigma_x u_{x,s}^n ds \approx T \sigma_x u_{x,T}^n$ and $\int_0^T \sigma_y u_{y,s}^n ds \approx T \sigma_y u_{y,T}^n$. Due to (36), (37) and since the initial conditions $z_{1,0}^n, \dots, z_{4,0}^n$ are fixed by the resampling step, the only quantities that we need to sample with the conditional density (35) are the Brownian values $u_{x,T}^n, u_{y,T}^n$. With the help of (36), (37) and after grouping the exponentials

we can approximate $M_T^{\epsilon_l} g_x(Z_{1,T}, z_{1,T}^n) g_y(Z_{3,T}, z_{3,T}^n)$ as

$$\begin{aligned}
& M_T^{\epsilon_l} g_x(Z_{1,T}, z_{1,T}^n) g_y(Z_{3,T}, z_{3,T}^n) \\
& \approx \exp \left(- \left\{ \epsilon_l \left[\frac{\mu_x}{\sigma_x} u_{x,T} + \frac{\mu_y}{\sigma_y} u_{y,T} \right] + (\epsilon_l)^2 \frac{T}{2} \left\{ \left[\frac{\mu_x}{\sigma_x} \right]^2 + \left[\frac{\mu_y}{\sigma_y} \right]^2 \right\} \right. \right. \\
& \quad + \frac{(Z_{1,T} - z_{1,0}^n - z_{2,0}^n T - \mu_x \frac{T^2}{2} - T \sigma_x u_{x,T}^n)^2}{2 * \sigma_{obs,x}^2} \\
& \quad \left. \left. + \frac{(Z_{3,T} - z_{3,0}^n - z_{4,0}^n T - \mu_y \frac{T^2}{2} - T \sigma_y u_{y,T}^n)^2}{2 * \sigma_{obs,y}^2} \right\} \right) \quad (38)
\end{aligned}$$

The approximation (38) for $M_T^{\epsilon_l} g_x(Z_{1,T}, z_{1,T}^n) g_y(Z_{3,T}, z_{3,T}^n)$ is a Gaussian density for $u_{x,T}^n, u_{y,T}^n$. We do not need HMC to sample it. We can rewrite it as the product of two Gaussian densities, one for $u_{x,T}^n$ and one for $u_{y,T}^n$ and sample it with standard methods for Gaussian densities. However, we show how HMC is implemented because for the nonlinear observation model, the density to be sampled will no longer be Gaussian.

Define the potential $V_{\epsilon_l}(u_{x,T}^n, u_{y,T}^n)$ by

$$\begin{aligned}
V_{\epsilon_l}(u_{x,T}^n, u_{y,T}^n) &= \epsilon_l \left[\frac{\mu_x}{\sigma_x} u_{x,T} + \frac{\mu_y}{\sigma_y} u_{y,T} \right] + (\epsilon_l)^2 \frac{T}{2} \left\{ \left[\frac{\mu_x}{\sigma_x} \right]^2 + \left[\frac{\mu_y}{\sigma_y} \right]^2 \right\} \\
& + \frac{(Z_{1,T} - z_{1,0}^n - z_{2,0}^n T - \mu_x \frac{T^2}{2} - T \sigma_x u_{x,T}^n)^2}{2 * \sigma_{obs,x}^2} \\
& + \frac{(Z_{3,T} - z_{3,0}^n - z_{4,0}^n T - \mu_y \frac{T^2}{2} - T \sigma_y u_{y,T}^n)^2}{2 * \sigma_{obs,y}^2} \quad (39)
\end{aligned}$$

and the density $M_T^{\epsilon_l} g_x(Z_{1,T}, z_{1,T}^n) g_y(Z_{3,T}, z_{3,T}^n)$ becomes

$$M_T^{\epsilon_l} g_x(Z_{1,T}, z_{1,T}^n) g_y(Z_{3,T}, z_{3,T}^n) \approx \exp \left(-V_{\epsilon_l}(u_{x,T}^n, u_{y,T}^n) \right).$$

Consider $u_{x,T}^n, u_{y,T}^n$ as the position variables of a Hamiltonian system. We define the $2D$ position vector $q = [q_1, q_2]^T$ with $q_1 = u_{x,T}^n$ and $q_2 = u_{y,T}^n$. To each of the position variables we associate a momentum variable and we write the Hamiltonian

$$H_{\epsilon_l}(q, p) = V_{\epsilon_l}(q) + \frac{p^T p}{2},$$

where $p = [p_1, p_2]^T$ is the momentum vector. Thus, the momenta variables are Gaussian distributed random variables with mean zero and variance 1. The equations of motion for this Hamiltonian system are given by Hamilton's equations

$$\frac{dq_i}{d\tau} = \frac{\partial H_{\epsilon_l}}{\partial p_i} \quad \text{and} \quad \frac{dp_i}{d\tau} = -\frac{\partial H_{\epsilon_l}}{\partial q_i} \quad \text{for } i = 1, \dots, 2.$$

HMC proceeds by assigning initial conditions to the momenta variables (through sampling from $\exp(-\frac{p^T p}{2})$), evolving the Hamiltonian system in fictitious time τ for a given number of steps of size $\delta\tau$ and then using the solution of the system to perform a Metropolis accept/reject step (more details in [8]). After the Metropolis step, the momenta values are discarded. The most popular method for solving the Hamiltonian system, which is the one we also used, is the Verlet leapfrog scheme. In our numerical implementation, we did not attempt to optimize the performance of the HMC algorithm. For the sampling at each level of the drift relaxation process we used 10 Metropolis accept/reject steps and 1 HMC step of size $\delta\tau = 10^{-1}$ to construct a trial path. A detailed study of the drift relaxation/HMC algorithm for conditional path sampling problems outside of the context of particle filtering will be presented in a future publication.

For the nonlinear observation model

$$\begin{aligned} & M_T^{\epsilon_l} g_\theta(Z_{\theta,T}, z_T^m) g_r(Z_{r,T}, z_T^m) \\ & \propto \exp\left(-\epsilon_l \left[\frac{\mu_x}{\sigma_x} u_{x,T} + \frac{\mu_y}{\sigma_y} u_{y,T} \right] - (\epsilon_l)^2 \frac{T}{2} \left\{ \left[\frac{\mu_x}{\sigma_x} \right]^2 + \left[\frac{\mu_y}{\sigma_y} \right]^2 \right\}\right) \\ & \quad \exp\left[-\frac{(Z_{\theta,T} - \theta(z_T^m))^2}{2 * \sigma_\theta^2}\right] \exp\left[-\frac{(Z_{r,T} - r(z_T^m))^2}{2 * \sigma_r^2}\right] \end{aligned} \quad (40)$$

where $Z_{\theta,T}, Z_{r,T}$ are the bearing and range observation values for the target and

$$\theta(z_T^m) = \arctan\left(\frac{z_{3,T}^m}{z_{1,T}^m}\right) \quad \text{and} \quad r(z_T^m) = (z_{1,T}^m{}^2 + z_{3,T}^m{}^2)^{1/2}$$

are the bearing and range values for the n th sample. We can use the same procedure as in the linear observation model to define a Hamiltonian system and its associated equations. We omit the details.

3.2 Linear observations

We start the presentation of our numerical experiments with results for the linear observation model (19). Figures 1 and 2 show the evolution in the

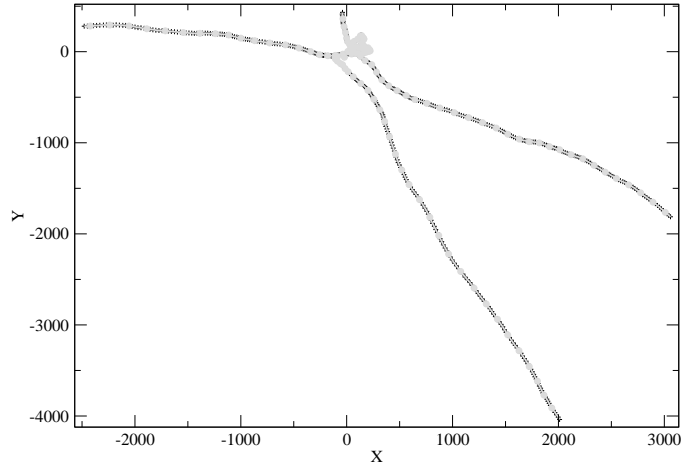


Figure 1: Linear observation model. The solid lines denote the true target tracks, the crosses denote the observations and the dots the conditional expectation estimates from the Girsanov particle filter. We have plotted the conditional expectation estimates every 5 observations to avoid cluttering in the figure.

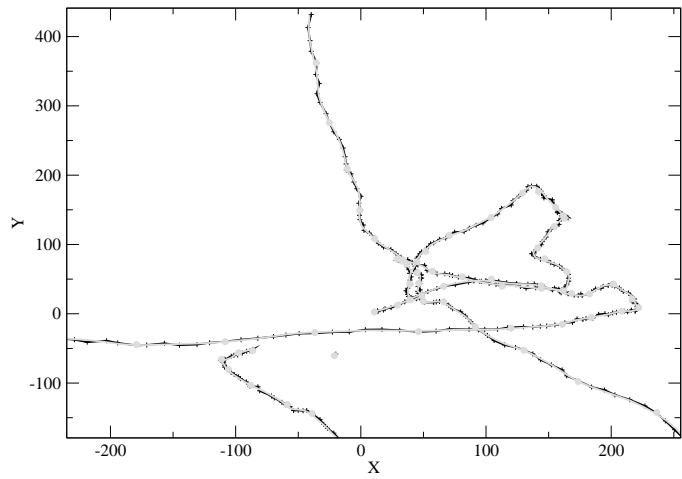


Figure 2: Linear observation model. Detail of Figure 1.

xy space of the true targets, the observations as well as the estimates of the Girsanov particle filter. It is obvious from the figures that the Girsanov particle filter follows accurately the targets and there is no ambiguity in the identification of the target tracks.

The performance of the Girsanov particle filter with 100 samples is compared to the performance of the generic particle filter with 120 samples in Figure 3 by monitoring the evolution in time of the RMS error per target. The RMS error per target (RMSE) is defined with reference to the true target tracks by the formula

$$RMSE(t) = \sqrt{\frac{1}{K_t} \sum_{k=1}^{K_t} \|\mathbf{x}_{k,t} - E[\mathbf{x}_{k,t}|Z_1, \dots, Z_t]\|^2} \quad (41)$$

where $\|\cdot\|$ is the norm of the position and velocity vector. Note that the state vector norm involves both positions and velocities even though the observations use information only from the positions of a target. $\mathbf{x}_{k,t}$ is the true state vector for target k . $E[\mathbf{x}_{k,t}|Z_1, \dots, Z_t]$ is the conditional expectation estimate calculated with the Girsanov or generic particle filter depending on whose filter's performance we want to calculate.

The Girsanov particle filter has a computational overhead of the order of a few percent compared to the generic particle filter. We have thus used the generic particle filter with more samples than the Girsanov particle filter. This additional number of samples more than accounts for the computational overhead of the Girsanov particle filter. As can be seen in Figure 3 the generic particle filter's accuracy deteriorates quickly. On the other hand, the Girsanov particle filter maintains an $O(1)$ RMS error per target for the entire tracking interval. The average value of the RMS error over the entire time interval of tracking is about 2.5 with standard deviation of about 0.5. For the generic particle filter, the average of the RMS error over the time interval of tracking is about 800 with standard deviation of about 760.

Figure 4 compares the effective sample size for the generic particle filter and the Girsanov particle filter. Because the number of samples is different for the two filters we have plotted the effective sample size as a percentage of the number of samples. We have to note that, after about 50 steps, the generic particle filter started producing observation weights (before the normalization) which were numerically zero. This makes the normalization impossible. In order to allow the generic particle filter to continue we chose at random one of the samples, since all of them are equally bad, and assigned all the weight to this sample. We did that for all the steps for which the observation weights were zero before the normalization. As a result, the

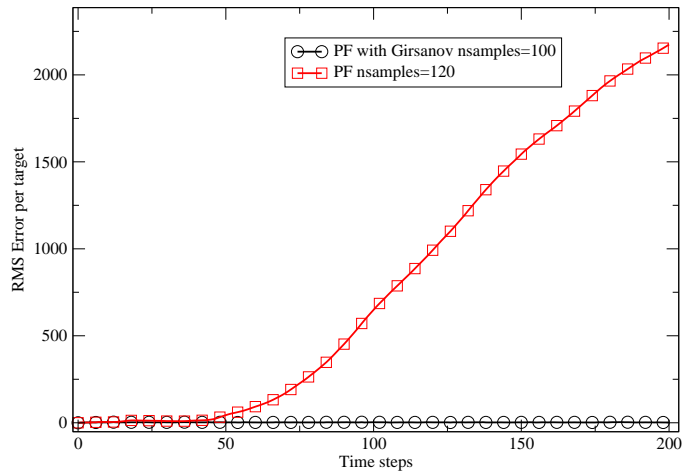


Figure 3: Linear observation model. Comparison of RMS error per target for the Girsanov particle filter and the generic particle filter.

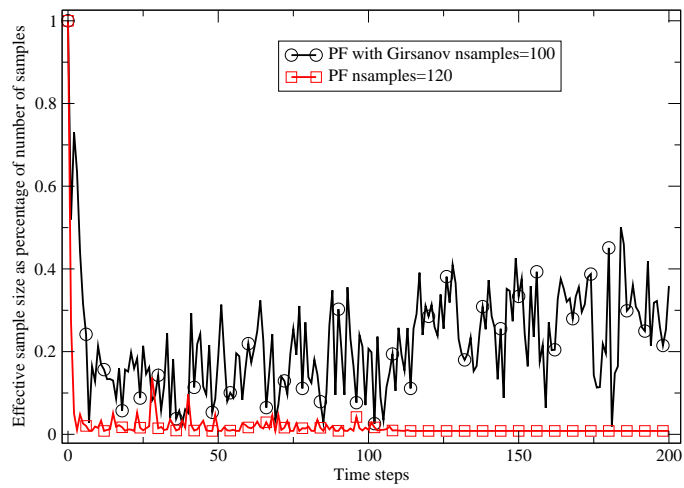


Figure 4: Linear observation model. Comparison of effective sample size for the Girsanov particle filter and the generic particle filter.

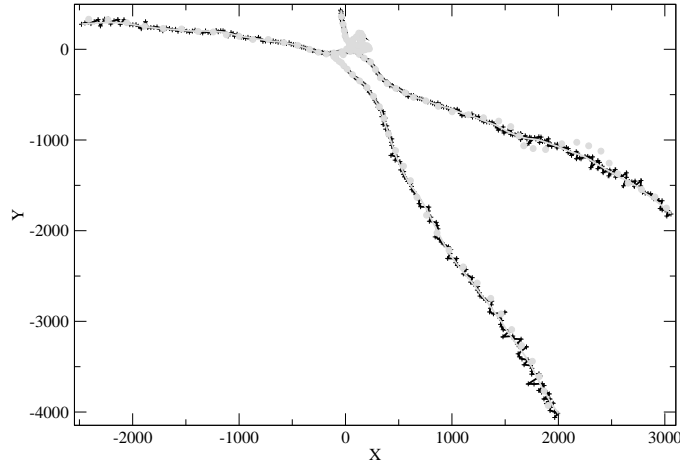


Figure 5: Nonlinear observation model. The solid lines denote the true target tracks, the crosses denote the observations and the dots the conditional expectation estimates from the Girsanov particle filter. We have plotted the conditional expectation estimates every 5 observations to avoid cluttering in the figure.

effective sample size for the generic particle filter drops down to 1 sample after about 50 steps. Once the generic particle filter deviates from the true target tracks there is no mechanism to correct it. Also, we tried assigning equal weights to all the samples when the observation weights dropped to zero. This did not improve the generic particle's performance either. On the other hand, the Girsanov particle filter maintains an effective sample size which is about 25% of the number of samples.

3.3 Nonlinear observations

We continue with results for the nonlinear observation model (20). Figures 5 and 6 show the evolution in the xy space of the true targets, the observations as well as the estimates of the Girsanov particle filter. Again, as in the case of the linear observation model, the Girsanov particle filter follows accurately the targets and there is no ambiguity in the identification of the target tracks.

The case of the nonlinear observation model is much more difficult than the case of the linear observation model. The reason is that for the nonlinear observation model, the observation errors, though constant in bearing and

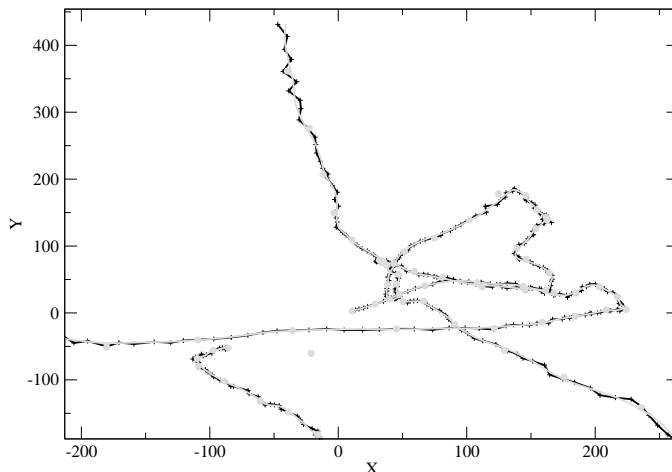


Figure 6: Nonlinear observation model. Detail of Figure 5.

range space, they become position dependent in xy space. In particular, when x and/or y are large, the observation errors can become rather large. This is easy to see by Taylor expanding the nonlinear transformation from bearing and range space to xy space around the true target values. Suppose that the true target bearing and range are θ_0, r_0 and its xy space position is $x_0 = r_0 \cos \theta_0, y_0 = r_0 \sin \theta_0$. Also, assume that the observation error in bearing and range space is, respectively, $\delta\theta$ and δr . The xy position of a target that is perturbed by $\delta\theta$ and δr in bearing and range space is (to first order)

$$\begin{aligned} x &= x_0 - y_0 \delta\theta - \delta r \cos \theta_0 \\ y &= y_0 + x_0 \delta\theta - \delta r \sin \theta_0. \end{aligned}$$

Thus, the perturbation in xy space can be significant even if $\delta\theta$ and δr are small. In our example we have $\sigma_\theta = 10^{-2}$. So, when the true target x and y values become of the order of 10^3 as happens for some of the targets, the observation value in bearing and range space can be quite misleading as far as the xy space position of the target is concerned. As a result, even if one does a good job in following the observation in bearing and range space, the conditional expectation estimate of the xy space position can be inaccurate.

With this in mind, we have used 200 samples for the Girsanov particle filter and 220 samples for the generic particle filter. Again, the extra samples used for the generic particle filter more than account for the computational

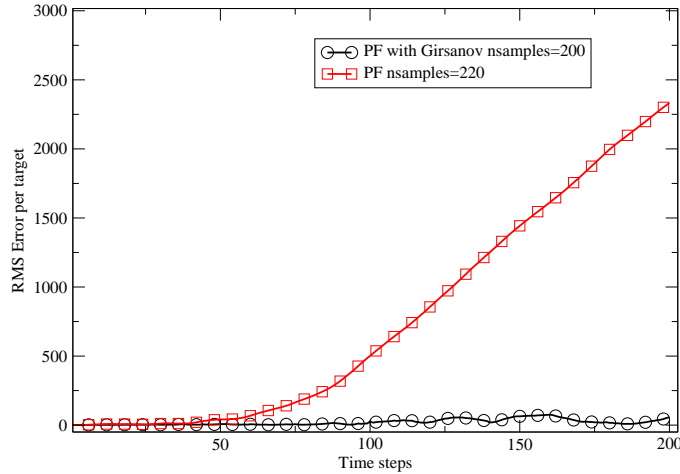


Figure 7: Nonlinear observation model. Comparison RMS error per target for the Girsanov particle filter and the generic particle filter.

overhead of the Girsanov particle filter. The performance of the Girsanov particle filter is compared to the performance of the generic particle filter in Figure 7 by monitoring the evolution in time of the RMS error per target. The generic particle filter’s accuracy again deteriorates rather quickly. The error for the Girsanov particle filter is larger than in the linear observation model but never exceeds about 80 even after 200 steps when the targets have reached large values of x and/or y . The average value of the RMS error over the entire time interval of tracking is about 22 with standard deviation of about 21. For the generic particle filter, the average of the RMS error over the time interval of tracking is about 760 with standard deviation of about 770.

Figure 8 compares the effective sample size for the generic particle filter and the Girsanov particle filter. After about 60 steps, the generic particle filter, started producing observation weights (before the normalization) which were numerically zero. This makes the normalization impossible. In order to allow the generic particle filter to continue we chose at random one of the samples, since all of them are equally bad, and assigned all the weight to this sample. We did that for all the steps for which the observation weights were zero before the normalization. As a result, the effective sample size for the generic particle filter drops down to 1 sample after about 60 steps. Once the generic particle filter deviates from the true target tracks there is

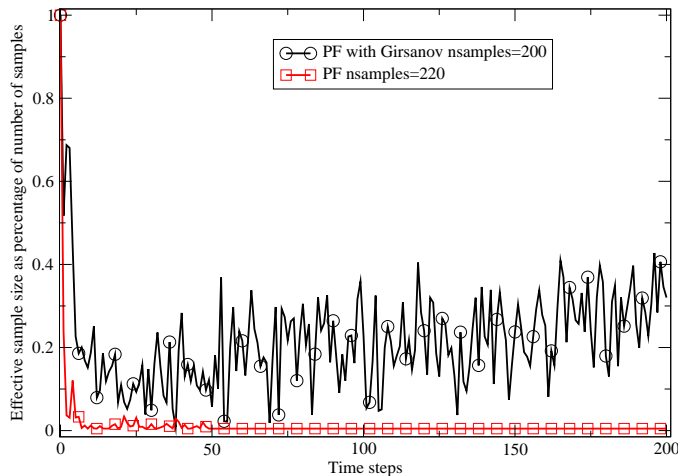


Figure 8: Nonlinear observation model. Comparison of effective sample size for the Girsanov particle filter and the generic particle filter.

no mechanism to correct it. Also, we tried assigning equal weights to all the samples when the observation weights dropped to zero. This did not improve the generic particle’s performance either. On the other hand, the Girsanov particle filter maintains an effective sample size which is about 25% of the number of samples.

4 Discussion

We have presented an algorithm for multi-target tracking which is based on Girsanov’s change of measure transformation for stochastic differential equations. The algorithm builds on the existing particle filter methodology for multi-target tracking by appending an MCMC step after the particle filter resampling step. The purpose of the addition of the MCMC step is to bring the samples closer to the observation. Even though the addition of an MCMC step for a particle filter has been proposed and used before [6], to the best of our knowledge, the use of Girsanov’s theorem to effect the MCMC step is novel (see also [16]). The use of Girsanov’s theorem leads to a Monte Carlo sampling problem which we addressed through the use of hybrid Monte Carlo and drift relaxation.

We have tested the performance of the algorithm on the problem of tracking multiple targets evolving under the near constant velocity model [1].

We have examined two cases of observation models: i) a linear observations model involving the positions of the targets and ii) a nonlinear observation model involving the bearing and range of the targets. For both cases the proposed Girsanov particle filter exhibited a significantly better performance than the generic particle filter. Since the Girsanov particle filter requires more computations than the generic particle filter it is bound to be more expensive. However, the computational overhead of the Girsanov particle filter is rather small, of the order of a few extra samples worth for the generic particle filter.

We plan to perform a detailed study of the proposed algorithm in more realistic cases involving clutter, spawning and merging of targets. Also, we want to study the behavior of the algorithm for cases with random birth and death events as well as for larger number of targets. Finally, the algorithm can be coupled to any target detection algorithm (see e.g. [13]) to perform joint detection and tracking.

After completing the current project, we realized a simplification in the formulas occurring for the case of linear dynamics which allows us to perform the MCMC step without recourse to Girsanov's theorem. This will be explored in a future publication. However, the method proposed here is more general and can be applied also for the case of nonlinear dynamics.

Acknowledgements

We are grateful to Prof. J. Weare for many discussions and moral support. Also, we would like to thank the Institute for Mathematics and its Applications in the University of Minnesota for its support.

A Conditional path sampling by drift relaxation

A.1 Conditional path sampling and Girsanov's theorem

We discuss the problem of conditional path sampling and how drift relaxation can be used to address this problem. The MCMC step of a particle filter described in Section 1.3 constitutes a case of conditional path sampling. Suppose that we are given a system of SDEs

$$dX_t = a(X_t)dt + \sigma(X_t)dB_t, \quad (42)$$

where $X_t = (X_1(t), \dots, X_N(t))$, is a N -dimensional Itô process with drift $a(X_t) = (a_1(X_t), \dots, a_N(X_t))$ and $B_t = (B_1(t), \dots, B_M(t))$ is a M -dimensional

Brownian motion. Also, $\sigma(X_t)$ is a $\mathbb{R}^N \times \mathbb{R}^M$ diffusion matrix. The problem of conditional path sampling is to generate sample paths of (42) with initial and endpoint conditions X_0 and X_T . The initial and endpoint conditions can be known exactly or can be specified through some probability distribution. In the first case,

$$X_0 = Z_0 \text{ and } X_T = Z_T. \quad (43)$$

In the second case,

$$X_0 \sim h(Z_0, X_0) \text{ and } X_T \sim g(Z_T, X_T). \quad (44)$$

The functions $h(Z_0, X_0)$ and $g(Z_T, X_T)$ are specified probability densities (with respect to Lebesgue measure) for the initial and endpoints respectively. The values Z_0, Z_T are also known parameters. In the case of filtering/smoothing Z_0, Z_T can be noisy observations of the state of the system. To motivate the algorithm, we discuss a situation which arises in many applications and which highlights why the problem of computing conditional sample paths of (42) can be complicated. Let us assume that the endpoint condition density $g(X_T)$ is sharply peaked in a region of \mathbb{R}^N that is rare for the solution to visit. This can happen whether the stochastic term $\sigma(X_t)dB_t$ is large, small or comparable to the drift term. Then most paths will result in an endpoint which has a very small probability with respect to the endpoint density $g(X_T)$. Thus, one would need to produce a very large number of sample paths in order to get one path that conforms to the endpoint condition. The drift relaxation algorithm aims to remedy this situation.

We will present the drift relaxation algorithm only for the case when the initial and endpoints are specified through given probability distributions. The algorithm relies on the use of Girsanov's theorem for SDEs (see in particular Theorem 8.6.8 in [14].) Even though there exist several different versions of Girsanov's theorem, the main result that we will be using here is that if one wants to produce sample paths of (42), one can instead produce sample paths from a different SDE system (with the same dimensionality and diffusion matrix)

$$dY_t = b(Y_t)dt + \sigma(Y_t)dB_t. \quad (45)$$

Then, one accounts (corrects) for the drift modification by weighing each sample path by an appropriate weight.

To be more precise, let us assume that we have produced sample paths from the systems (42) and (45) starting from the *same* initial condition $X_0 = Y_0 = x$. Also, suppose we can find a process $u : \mathbb{R}^N \rightarrow \mathbb{R}^M$ such that $\sigma(y)u(y) = b(y) - a(y)$, for $y \in \mathbb{R}^N$. Define the quantity

$$M_T = \exp\left(-\int_0^T \langle u(Y_s), dB_s \rangle - \frac{1}{2} \int_0^T |u(Y_s)|^2 ds\right), \quad (46)$$

where $\langle \cdot, \cdot \rangle$ stands for the inner product in \mathbb{R}^M . Let P be the Wiener measure on continuous functions on $[0, T]$ and define a new measure Q (absolutely continuous with respect to P) by $dQ = M_T dP$. Girsanov's theorem states that, under appropriate conditions on the drifts a and b and the diffusion matrix σ , the Q -law of Y_t is the same as the P -law of X_t in $[0, T]$. This path measure equivalence translates into equivalence for expectations taken with the measures Q and P . We have

$$E_P[f(X_t^x)] = E_Q[f(Y_t^x)] = E_P[M_T f(Y_t^x)], \quad (47)$$

for a function f . In relation (47), E_P denotes expectation with respect to P and E_Q expectation with respect to Q (see Section 8.6 in [14] for a lucid and detailed presentation of Girsanov's theorem).

In words, Girsanov's theorem states that if one has constructed sample paths for the process Y_t , these can be used to compute expectations for the process X_t . However, it allows us to do much more. The relation $dQ = M_T dP$ provides us with a way to quantify the probability of a path from Y_t being a path of X_t . So, one can treat M_T as a probability density and sample it through Monte Carlo, i.e., construct paths from Y_t that have high probability of being paths from X_t . Technically, M_T is the Radon-Nikodym derivative of Q with respect to P .

We return now to the conditional path sampling problem and see how it can be facilitated by the preceding discussion of Girsanov's theorem. The conditional path sampling problem is equivalent to computing the conditional expectation $E[f(X_t)|Z_0, Z_T]$. The corresponding initial and endpoint conditions for X_0 and X_T are connected to Z_0, Z_T through (44) which we repeat here

$$X_0 \sim h(Z_0, X_0) \text{ and } X_T \sim g(Z_T, X_T).$$

For the conditional expectation $E[f(X_t)|Z_0, Z_T]$ we have

$$\begin{aligned} E[f(X_t)|Z_0, Z_T] &= \int f(X_t) \frac{h(X_0, Z_0)g(X_T, Z_T)dP}{\int h(X_0, Z_0)g(X_T, Z_T)dP} \\ &= \int f(Y_t) \frac{h(Y_0, Z_0)M_T g(Y_T, Z_T)dP}{\int h(X_0, Z_0)g(X_T, Z_T)dP} \end{aligned} \quad (48)$$

where in the last equation we have used Girsanov's theorem. We have deliberately written the numerator as $h(Y_0, Z_0)M_T g(Y_T, Z_T)$ to denote the

probabilistic weight of the initial condition of the path (the term $h(Y_0, Z_0)$), the path connecting the initial and endpoints (the term M_T) and the endpoint condition (the term $g(Y_T, Z_T)$) respectively. So, we see that the conditional path sampling problem in this case amounts to sampling the density $h(Y_0, Z_0)M_Tg(Y_T, Z_T)$ with respect to the Wiener measure P . Note that in a MCMC sampling context the normalization constant $\int h(X_0, Z_0)g(X_T, Z_T)dP$ is immaterial.

A.2 Drift relaxation

We have reformulated the problem of conditional path sampling as a problem of sampling the density $h(Y_0, Z_0)M_Tg(Y_T, Z_T)$. This sampling should be done with care if we want it to be efficient. We proceed now to discuss how this sampling can be realized in an efficient way. Let us start by pointing out why sampling $h(Y_0, Z_0)M_Tg(Y_T, Z_T)$ can be difficult. This will motivate the way in which we propose to perform this sampling. If the endpoint condition is highly unlikely for paths of the original process X_t then the sampled conditional paths from Y_t will be quite different from the desired conditional paths of X_t . This difference creates two difficulties. First, the process $u(Y_s)$ which monitors the drift discrepancy between the original and modified drifts can be large. This means, that the density $h(Y_0, Z_0)M_Tg(Y_T, Z_T)$ is rather sharply peaked around a few paths of Y_t . As a result, a naive Monte Carlo sampling will result in very low acceptance rates for the new path configurations. Second, (see expression (46) for M_T) we have to sample a path for Y_t , $t \in [0, T]$, as well as the associated Brownian path B_t , $t \in [0, T]$. Inevitably, we have to discretize the integrals in M_T and sample discretized paths. A Monte Carlo sampling algorithm which proceeds incrementally (after discretization of the path) by changing one segment of the paths at a time can be slow, especially if the path discretization is dense.

We discuss now how to handle these two difficulties. We begin with the problem of the large discrepancy between the original drift a and the modified drift b . Instead of attempting to sample directly from the density $h(Y_0, Z_0)M_Tg(Y_T, Z_T)$, i.e., correct for the full discrepancy $b - a$ of the drifts, we propose the following relaxation (hence the name drift relaxation): Consider a collection of $L + 1$ modified SDE systems

$$dY_t^l = (1 - \epsilon_l)b(Y_t^l)dt + \epsilon_la(Y_t^l)dt + \sigma(Y_t^l)dB_t, \quad (49)$$

where $\epsilon_l \in [0, 1]$, $l = 0, \dots, L$, with $\epsilon_l < \epsilon_{l+1}$, $\epsilon_0 = 0$ and $\epsilon_L = 1$. The SDE system for $l = 0$ has drift b , while the SDE system for $l = L$ has the original drift a . Correspondingly, we can define a collection of L (note not $L + 1$)

sampling densities $M_T^l, l = 1, \dots, L$. Recall that the density M_T provides us with the probability of a path from one process being the path of another process. In other words, it involves two processes. One choice is to define M_T^l with reference to the process with drift b . This resembles the technique of simulated annealing in equilibrium statistical mechanics [8]. The densities M_T^l are given (again through Girsanov's theorem) by

$$M_T^l = \exp\left(-\epsilon_l \int_0^T \langle u(Y_s^0), dB_s \rangle - \frac{1}{2}(\epsilon_l)^2 \int_0^T |u(Y_s^0)|^2 ds\right), \quad l = 1, \dots, L. \quad (50)$$

Another choice is to define M_T^l with reference to the previous process (level) in the sequence. This resembles homotopy methods for nonlinear algebraic equations [17]. For this choice, the densities M_T^l are given by

$$M_T^l = \exp\left(-(\epsilon_l - \epsilon_{l-1}) \int_0^T \langle u(Y_s^{l-1}), dB_s \rangle - \frac{1}{2}(\epsilon_l - \epsilon_{l-1})^2 \int_0^T |u(Y_s^{l-1})|^2 ds\right), \quad (51)$$

for $l = 1, \dots, L$. In the current paper we use only the form of M_T^l given in (50). The form of M_T^l given in (51) will be studied in a future publication.

We start with a sample from the process with drift b and use the sample from one level as the initial condition for sampling at the next level in the sequence. The relaxation allows us to enforce the original drift gradually by making at each level relatively small changes to the conditional path. The relaxation process mitigates the large discrepancy between the drifts a and b . Note that at the $L + 1$ st level we sample from the density M_T which we wanted to sample from originally. More details are provided in the numerical results section.

The second problem of the Monte Carlo sampling is how to propose a new trial path for Y_t and B_t . As we have already pointed out, an incremental change of the path by changing one segment at a time can be rather slow. This problem can be remedied by the use of hybrid Monte Carlo (HMC) [8]. HMC proceeds by constructing a Hamiltonian system associated with the variables that we want to sample. The Hamiltonian system is evolved in fictitious (algorithmic) time and the state of the system after some predetermined fictitious time steps is used as trial configuration for the Monte Carlo accept/reject step. For our case this allows the proposal of entire paths for Y_t and B_t . This results in higher acceptance rates and faster exploration of the path space. We provide more details of the HMC implementation in the section on numerical results.

References

- [1] Bar-Shalom Y. and Fortmann T.E., Tracking and Data Association, Academic Press, 1988.
- [2] Bar-Shalom Y. and Blair W.D., Eds. Multitarget-Multisensor Tracking: Applications and Advances, vol. III, Norwood, MA, Artech House, 2000.
- [3] Blackman S. and Popoli R., Design and Analysis of Modern Tracking Systems, Norwood, MA, Artech House, 1999.
- [4] Doucet A., de Freitas N. and Gordon N. (eds.) , Sequential Monte Carlo Methods in Practice, Springer NY, 2001.
- [5] Fortmann T. E., Bar-Shalom Y. and Scheffe M., Sonar tracking of multiple targets using joint probabilistic data association, IEEE J. Ocea. Eng., vol.8 (1983) pp.173-184.
- [6] Gilks W. and Berzuini C., Following a moving target. Monte Carlo inference for dynamic Bayesian models, J. Royal Stat. Soc. B 63 (1) (1999) pp. 2124-2137.
- [7] Gordon N.J., Salmond D.J. and Smith A.F.M., Novel approach to nonlinear/non-Gaussian Bayesian state estimation, Proc. Inst. Elect. Eng. F 140(2) (1993) pp. 107-113.
- [8] Liu J.S., Monte Carlo Strategies in Scientific Computing, Springer NY, 2001.
- [9] Liu J.S. and Chen R. Sequential Monte Carlo Methods for Dynamic Systems. Journal of the American Statistical Association, vol.93 no. 443 (1993) pp. 1032-1044.
- [10] Mahler R. P., Statistical Multisource-Multitarget Information Fusion, Artech House Publishers MA, 2007.
- [11] Mahler R. P. S., Multitarget Bayes filtering via first-order multitarget moments, IEEE Trans. Aero. Elect. Sys., Vol. 39 no. 4 (2003) pp. 1152-1178.
- [12] Mahler R.P.S. and Maroulas V. Tracking Spawning Objects, (2010) *submitted*.

- [13] Ng W., Li J.F., Godsill S.J. and Vermaak J., A hybrid approach for online joint detection and tracking for multiple targets, Proc. IEEE Aerospace Conference (2005).
- [14] Oksendal B., Stochastic Differential Equations, An Introduction with Applications, Sixth Edition Springer Heidelberg 2005.
- [15] Snyder C., Bengtsson T., Bickel P. and Anderson J., Obstacles to High-dimensional Particle Filtering, Mon. Wea. Rev., Vol. 136 (2008) pp. 4629-4640.
- [16] Stinis P., Conditional path sampling for stochastic differential equations by drift relaxation, (2010), arXiv:1006.2492v1.
- [17] Stoer J, and Bulirsch R., Introduction to Numerical Analysis, Third Edition, Springer 2002.
- [18] Vermaak J., Godsill S. and Perez P., Monte Carlo filtering for multi-target tracking and data association, IEEE Trans. Aero. Elect. Sys., 41(1) (2005) pp. 309-332.
- [19] Vo B-N., Singh S. and Doucet A., Sequential Monte Carlo Methods for Multi-Target Filtering with Random Finite Sets, IEEE Trans. Aero. Elect. Sys., 41(4) (2005) pp. 1224-1245.
- [20] Weare J., Particle filtering with path sampling and an application to a bimodal ocean current model, J. Comp. Phys. 228 (2009) pp. 4312-4331.