

On the Monte Carlo Marginal MAP Estimator for General State Space Models

S. Saha ^{*†}, P. K. Mandal [‡], A. Bagchi [‡], Y. Boers [‡] and H. Driessen [‡]

1 Introduction

Consider a state-space model

$$x_k = f(x_{k-1}, w_k), \tag{1}$$

$$y_k = h(x_k, v_k), \quad w_k \perp v_k, \quad k = 1, 2, \dots \tag{2}$$

where x_k is the (unobserved) state with initial density $p(x_0)$, y_k is the measurement at time step k . w_k and v_k are the corresponding process and measurement noises. The dynamic estimation problem is concerned with estimating the unknown state x_k given the set of measurements $y_{1:n} = y_1, \dots, y_n$. The classic solution is the posterior probability density function $p(x_k|y_{1:n})$ which reflects all knowledge about the current state x_k . However, for a general nonlinear dynamic system, this posterior density is analytically intractable and therefore many approximated methods with varying success have been employed to arrive at the solution. In recent times, starting with Gordon's seminal paper ([14]), particle based sequential Monte Carlo method has been receiving increasing attention due to its capability of efficiently approximating such difficult posterior distributions. In this method, the posterior is approximated by a cloud of N weighted particles, whose empirical measure closely approximates the true posterior for large N ([10],[1],[21]). For the inference purposes, however, instead of the whole posterior, often a single point estimate is more convenient. Two such commonly used point estimates are the minimum mean square error (MMSE) estimate ($E(x_k|y_{1:n})$) and the maximum a posteriori (MAP) estimate ($x_{k|n}^{MAP} = \arg \max_{x_k} p(x_k|y_{1:n})$).

Whenever the posterior is multimodal (which, for example, often arises in target tracking due to data association problem, multiple models of target dynamics ([2],[3]) or *mixed labeling problem* in multiple target tracking ([5]), MMSE estimate may be located in a region between the modes with a very

^{*}This work is supported by a research grant from THALES Nederland BV.

[†]Department of Applied Mathematics, University of Twente, 7500 AE, Enschede, The Netherlands, Email: {s.saha, p.k.mandal, a.bagchi}@math.utwente.nl

[‡]THALES Nederland BV, Haaksbergerstraat 49, 7554 PA, Hengelo, The Netherlands, Email: {yvo.boers, hans.driessen}@nl.thalesgroup.com

low probability and thus, producing an unreasonable estimate. In such a situation, MAP estimate is more relevant. The situation is described in figure 1. Estimating the MAP involves maximization over the posterior density, which is

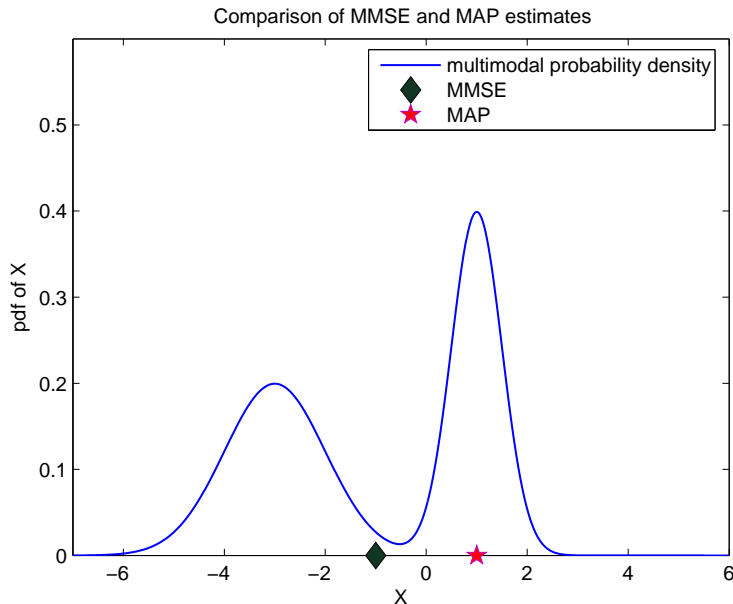


Figure 1: Comparison of MMSE and MAP estimates

not readily available either analytically or from the approximation such as particle filter. In the literature, particle with the maximum weight is often taken as the MAP estimate ([25],[7],[4]). But, recently it has been argued that the particle with the highest weight does not necessarily represent the most probable state estimate and can actually be far from true MAP ([11],[8]). Thus, this estimator is not really a fair approximation of the true MAP. Naturally, the crux of the problem lies in constructing the posterior density from the weighted cloud representation of the distribution. One classic approach is the kernel based method where a kernel is fitted around each particle to approximate the posterior density ([24]). This method requires a choice of kernel bandwidth which is not obvious and the method is computationally demanding, which restricts its use in many practical applications.

Recently, the authors in ([11],[12]) have proposed an elegant scheme for computing the MAP of the filter distribution, directly from the output of a running particle filter. This method thus avoids the need of bandwidth selection associated with the kernel based methods. In principle, this new method can evaluate the posterior density $p(x_k|y_{1:k})$ at any point. In this paper, we explore the different properties of this estimator and extend this idea to compute the smoothed marginal MAP from particle representation of smoother distribution

$p(x_k|y_{1:n})$, $k < n$. The rest of the article is organized as follows. Following a brief introduction to the particle based filter MAP estimator in section 2, we compare it with the Viterbi-Godsill method in section 3. Next, we discuss the effect of continuous optimization to extract the MAP in section 4. In section 5, the tweaking of particle population in constructing the predictive density is studied. Finally, in section 6, we extend the MAP estimator to marginal smoother, which is then applied to estimate the unknown initial state of a dynamic system.

For numerical studies, in this article, we consider the following two models:

(A) Linear model

$$x_k = \alpha x_{k-1} + w_k, \quad w_k \sim \mathcal{N}(0, 1) \quad (3)$$

$$y_k = x_k + v_k, \quad v_k \sim \mathcal{N}(0, 0.01), \quad k = 1, 2, \dots, \quad (4)$$

where the initial distribution $p(x_0) \sim \mathcal{N}(0, 2)$.

(B) Nonlinear model

$$x_k = \frac{x_{k-1}}{2} + \frac{25x_{k-1}}{1 + x_{k-1}^2} + 8 \cos(1.2k) + w_k, \quad w_k \sim \mathcal{N}(0, 10) \quad (5)$$

$$y_k = \frac{x_k^2}{20} + v_k, \quad v_k \sim \mathcal{N}(0, 1), \quad k = 1, 2, \dots, \quad (6)$$

with the initial distribution $p(x_0) \sim \mathcal{N}(0, 5)$.

2 Particle based filter MAP estimator (pf-MAP)

In this section, we describe the method of obtaining the filter MAP estimate based on the running particle filter only (without any kernel fitting). We recall that the naive approach of choosing the particle with the highest weight may not lead to the actual MAP([11],[8]). The method that we will be following here is originally described in ([11],[12]). The main advantage of this method is that it can approximate the posterior density not only at particles forming the clouds but at any point. To elucidate the method further, consider the same dynamic system as given in equations (1)-(2). The MAP estimate of the filtering density at time k is then defined as

$$x_k^{MAP} = \arg \max_{x_k} p(x_k|y_{1:k}) \quad (7)$$

where $y_{1:k} = (y_1, y_2, \dots, y_k)$ is the set of all observations up to time k . For a general nonlinear model, analytical expression for the filtering density $p(x_k|y_{1:k})$ can hardly be obtained in closed form. However, using particle filtering technique, one can approximate this posterior distribution by a cloud of N weighted particles as

$$\hat{P}(dx_k|y_{1:k}) \simeq \sum_{j=1}^N w_k^{(j)} \delta_{x_k^{(j)}}(dx_k). \quad (8)$$

Now using the Baye's rule, the posterior (filtering) density in equation (7) can be written as

$$p(x_k|y_{1:k}) = \frac{p(y_k|x_k)p(x_k|y_{1:k-1})}{p(y_k|y_{1:k-1})}. \quad (9)$$

Observing that the denominator is independent of x_k , one can write

$$p(x_k|y_{1:k}) \propto p(y_k|x_k)p(x_k|y_{1:k-1}). \quad (10)$$

The MAP estimate can thus be expressed as

$$x_k^{MAP} = \arg \max_{x_k} p(y_k|x_k)p(x_k|y_{1:k-1}). \quad (11)$$

Since the conditional likelihood $p(y_k|x_k)$ is known for each x_k , the main issue for evaluating MAP is the calculation of the predictive density $p(x_k|y_{1:k-1})$. Though not available in closed form, one can use the relation

$$p(x_k|y_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|y_{1:k-1})dx_{k-1} \quad (12)$$

and the running particle filter given by (8) to approximate $p(x_k|y_{1:k-1})$ using Monte Carlo integration as

$$p(x_k|y_{1:k-1}) \approx \sum_j p(x_k|x_{k-1}^{(j)})w_{k-1}^{(j)}. \quad (13)$$

Substituting (13) into (10) we get the posterior density and the MAP estimation is then obtained by finding the global maxima of it. In general, this maximization step is nontrivial due to the possible multi modalities arising from the non Gaussian nature of the posterior.

Here we describe a method to approximate the MAP estimator, which reduces the computational load substantially. With the view that the cloud of particles $\{x_k^{(i)}\}_{i=1}^N$ in a running particle filter constitutes an adaptive discretization of the state space at time k , one can approximately locate the MAP by first evaluating equation (10) at the predicted particle $\{x_k^{(i)}\}_{i=1}^N$ and finally selecting the particle with the highest density. This leads to the approximate particle based MAP estimate (pf-MAP) as

$$x_k^{MAP} = \arg \max_{x_k^{(i)}} p(y_k|x_k^{(i)}) \sum_j p(x_k^{(i)}|x_{k-1}^{(j)})w_{k-1}^{(j)} \quad (14)$$

One should note that for each time step, the memory requirement of this MAP estimator is $O(N)$ and the computational complexity is $O(N^2)$. For the theoretical convergence properties of this estimator, the readers are referred to [12]. The predictive density in equation (13) can be seen to be a weighted mixture of state transition densities and thus the variance of the process noise is expected to play a role in estimating the continuous posterior density. To make the point

clear, let us suppose that the variance of the process noise is very low such that the mixture components are well separated. Now a support point which lies between two such mixture components can have unreasonably low predictive density, although the true likelihood may be high. As a consequence, this will result in a very low posterior density. This situation may arise, for example, when the parameter is treated as augmented state with small artificial process noise and starting with a wide initial distribution. Here, if the particles sampled from the initial distribution are away from the true parameter, due to the small process noise variance, the particles can not be propagated (in time) to the true neighborhood of the parameter and as a result, the posterior would be very low around the true neighborhood (in the limit, as the variance of the artificial process noise goes to zero, the continuous nature of this predictive density breakdowns to the sum of Dirac-delta functions).

3 Comparing pf-MAP with EP-VGM

In the existing literature on the MAP estimation for a general state space model, we find that in the past, the focus has been on estimating the MAP sequence of the joint states over time i.e. $x_{0:T}$, instead of the marginal MAP of x_t ; see e.g.[9]. We note here that it is relatively easy to construct analytically the joint posterior of the states ($x_{0:t}$) given all the observations ($y_{1:t}$), whereas the marginal is hardly analytically tractable. In our opinion, this may have been partly the reason why the estimator for marginal MAP was not pursued. The point is further explained below:

Consider the model as given in (1) and (2). Now, using the Bayes' rule, it is easy to check that

$$\begin{aligned} p(x_{0:T}|y_{1:T}) &\propto p(x_{0:T}, y_{1:T}) \\ &\propto p(x_0) \prod_{k=1}^T p(x_k|x_{k-1})p(y_k|x_k). \end{aligned}$$

Since $p(x_0)$, $p(x_k|x_{k-1})$ and $p(y_k|x_k)$ are known, $p(x_{0:T}|y_{1:T})$ can be found easily. On the other hand, $p(x_T|y_{1:T})$ would always involve integration terms and this is difficult to obtain analytically. As a result, any corresponding marginal MAP estimator is also difficult to obtain.

In the recent past, Godsill et al.([13]) have developed a method for estimating the MAP sequence in nonlinear non-Gaussian dynamic models using Viterbi algorithm. The method uses the particle cloud representation as an adaptive discretization of the state space and then employ the Viterbi algorithm on this discretized state space to find the MAP sequence. As this MAP sequence is essentially a representation of (estimated) states over time, the last element of this sequence can be viewed as an estimate of the current state. Subsequently, we call this estimate as end point Viterbi-Godsill MAP (EP-VGM). In this section we compare the behaviors of EP-VGM and pf-MAP as an estimate of x_k . It is to be noted though that the estimate obtained using EP-VGM and the pf-MAP are

not necessarily the same. While Viterbi-Godsill algorithm aims at maximizing $p(x_{1:k}|y_{1:k})$, pf-MAP aims at maximizing $p(x_k|y_{1:k})$. The two estimates would be, at least in principle, same when the system dynamics are linear-Gaussian, as $p(x_{1:k}|y_{1:k})$ is then multivariate Gaussian. However, when there is significant nonlinearity and/or non-Gaussianity in the model, one expects that the difference would pop up for the above two estimators. Subsequently, we investigate their behaviors through numerical simulations. First, we study the linear Gaussian model (A) with $\alpha = 1$ for comparing these two estimators. Here, the true MAP for the current state can be extracted using Kalman filter. The root mean square error (RMSE) estimate with respect to true MAP over 20 Monte Carlo runs with 200 time steps along with the average CPU time (in second) are shown in the following table :

	pf-MAP	EP-VGM	pf-MAP	EP-VGM
Nr. sample	RMSE	RMSE	avg CPU	avg CPU
100	0.007459	0.007430	0.0085	0.4765
200	0.006604	0.006653	0.0265	1.7461
400	0.006202	0.006224	0.1008	7.8359
1000	0.005948	0.006034	0.4890	49.9367

The average CPU time reported here does not include the cost of computation common for both the methods as we have used the same running particle filter for each Monte Carlo run. It is evident that for the same number of particles, RMSE's for both the methods behave similarly while the average CPU time for pf-MAP is substantially less.

Next we consider the nonlinear model (B). Since for this example, the true MAP can not be obtained analytically, we compare them as an estimator of the current state. We use three different starting seeds for estimating the RMSE. In each case, the RMSE estimate is done with respect to true (synthetic) state over 20 Monte Carlo runs with 200 time steps. The result is shown in the following table :

		pf-MAP	EP-VGM	pf-MAP	EP-VGM
Nr. sample		RMSE	RMSE	avg CPU	avg CPU
100	<i>Seed1</i>	5.2964	5.5667	3.515e-05	0.0033
	<i>Seed2</i>	5.3732	5.4908	3.125e-05	0.0030
	<i>Seed3</i>	5.4251	5.5875	3.126e-05	0.0031
250	<i>Seed1</i>	4.9707	5.2567	1.836e-04	0.0194
	<i>Seed2</i>	5.4035	5.5508	1.914e-04	0.0191
	<i>Seed3</i>	4.7676	5.2368	2.578e-04	0.0195
500	<i>Seed1</i>	4.8530	5.4184	6.054e-04	0.0851
	<i>Seed2</i>	4.8282	5.3885	5.664e-04	0.0852
	<i>Seed3</i>	4.8990	5.2521	5.742e-04	0.0851
1000	<i>Seed1</i>	4.4659	5.2433	0.0022	0.3457
	<i>Seed2</i>	5.2127	5.6355	0.0020	0.3393
	<i>Seed3</i>	4.8509	5.4876	0.0022	0.3429

We observe that in terms of RMSE estimate, pf-MAP performs better as an estimate of current state while it is also computationally much cheaper.

4 Improvement over the estimated pf-MAP

In section 2, the pf-MAP is estimated by comparing the density values at the current particles and choosing the one with the highest density. We note, however that the predictive density $p(x_k|y_{1:k-1})$ can be approximated, as given by (13), at any point x_k . The same is then true for the posterior density $p(x_k|y_{1:k})$ by virtue of (10). Consequently, we can use different optimization methods such as Genetic Algorithm or any gradient based continuous optimization method to maximize the posterior density. For subsequent numerical experiments in this section, we use the unconstrained optimization function 'fmincon' of MATLAB with default parameter setting. Here, the pf-MAP as obtained in section 2, is taken as initial starting point for the MAP estimate using gradient based optimization technique (subsequently referred to as 'grad-MAP').

We consider the nonlinear model as given in equations(5)-(6). For the running particle filter, we use $N = 500$ particle, $T = 200$ and state transition density as proposal. For a typical run, the results are shown in the following figures (figure2 to figure4).

We observe that the gradient-MAP obtained with the pf-MAP as a starting

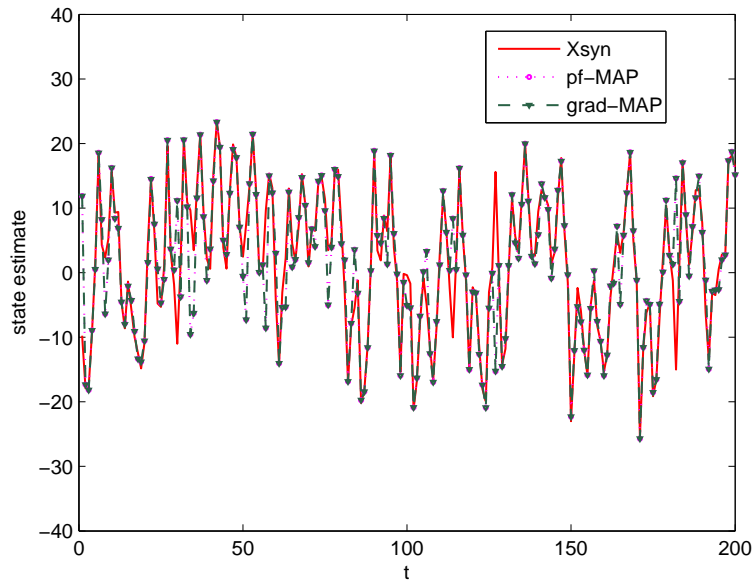


Figure 2:

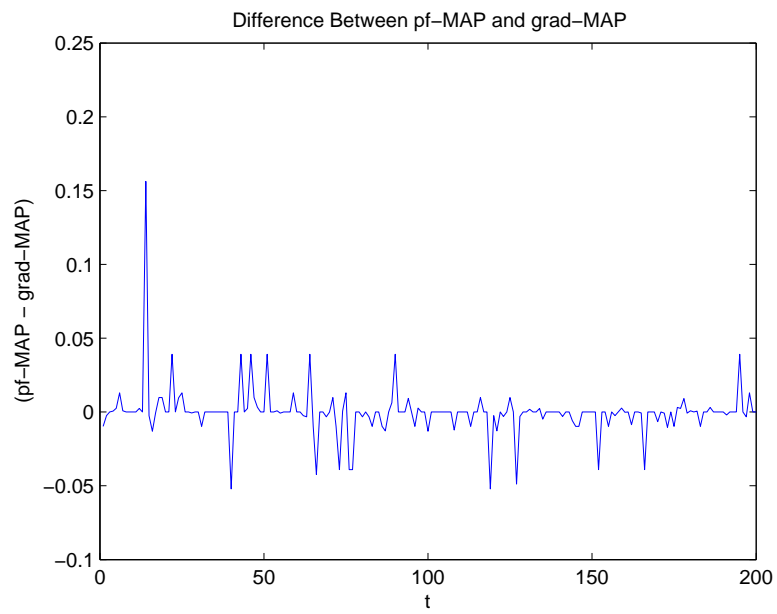


Figure 3:

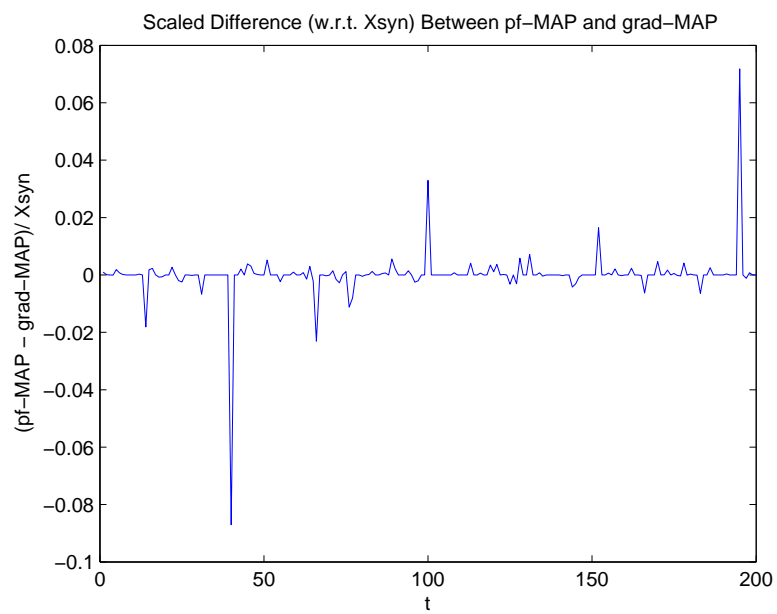


Figure 4:

point has slight marginal edge over the pf-MAP in terms of estimation efficiency while, it is computationally heavier due to the further overload of continuous optimization method.

Other suggestion to improve the overall performance would be to use only a subset of particles in the maximization step of finding pf-MAP of section 2 and with this (crude) estimate as starting point, obtain the grad-MAP. This may result in a high reduction of computational load, with the level of estimation efficiency expected to remain similar. Furthermore, this approach can be combined with reducing the number of particles in constructing the predictive density (as discussed in the next section). However, these suggestions are not pursued here further.

5 Particle tweaking effect on predictive density

While evaluating the pf-MAP, we have additional flexibility to explore the effect of tweaking the number of particles used in constructing the predictive density. The predictive density is approximately obtained from the running particle filter using the Monte Carlo integration as given in equation (13). For this Monte Carlo integration, instead of using the whole cloud, one can use a subset of $\{x_{k-1}^{(j)}, w_{k-1}^{(j)}\}_{j=1}^N$ to reduce the computational cost. The reduced set of particle (N_R) for example, can be obtained from the original cloud through a resampling step. We subsequently compare the tweaking effect numerically in terms of average CPU time (in second) and RMSE estimate over 20 Monte Carlo runs with 200 time steps. For the running particle filter, the state transition density is chosen as the proposal. For the linear model (with $\alpha = 0.5$), the RMSE is estimated with respect to the true MAP extracted from Kalman filter (with $p(x_0) \sim \mathcal{N}(0, 2)$) and the result is shown in the table below :

Original size (N)	Tweaking size (N_R)	RMSE	Avg. CPU
500	500	0.0216	15.48
	250	0.0216	11.49
	125	0.0216	09.55
<hr/>			
1000	1000	0.0153	66.47
	500	0.0153	45.13
	250	0.0153	37.82

For this example, we observe that the tweaking leads to huge computational savings while it does not affect the accuracy in terms of RMSE estimate. To investigate the effect further, we next consider the nonlinear model. Since, in this case, the true MAP is not known analytically, the RMSE is estimated with respect to true (synthetic) state over 20 Monte Carlo runs with 200 time steps. The result is shown in the following table :

Original size (N)	Tweaking size (N_R)	RMSE	Avg. CPU
1000	1000	5.1922	78.10
	500	5.1608	54.26
	250	5.2115	46.39
	100	5.3503	39.81

Here also, we observe that by reducing the number of particles in constructing predictive density, the average CPU time falls drastically while the RMSE does not change so much.

In conclusion, it seems that the particle tweaking in predictive density evaluation can lead to a substantial improvement in computational saving at a little compromise on RMSE value.

6 Particle based marginal smoother MAP estimator (ps-MAP)

So far we have considered the different aspects of the filter MAP estimation using the weighted particles. In this section, our main objective is to extend the MAP estimation idea to the marginal smoothing. We again consider the same dynamic system as given by (1) and (2). Now, the problem can be mathematically posed as finding

$$x_{t|T}^{MAP} = \arg \max_{x_t} p(x_t | y_{1:T}) \quad (15)$$

where $t < T$.

Our starting point is that there already exists a (weighted) particle cloud for the marginal smoother. Based on these weighted cloud representation, we calculate the marginal smoothing density using similar idea as in pf-MAP. From the literature, we find that the "forward-backward smoothing" is the conventional approach to generate the particle clouds of marginal smoother. Subsequently, we derive the marginal smoother MAP from it. As an application we use this marginal MAP to find the initial condition. Recently "generalized two-filter smoothing" has been used to generate the particle cloud for the marginal smoother. In this case as well, we explain how to extract the marginal smoother MAP from the particle clouds.

6.1 Forward-Backward Smoothing

The marginal smoother can be obtained using forward- backward smoother ([18]) as

$$p(x_t | y_{1:T}) = p(x_t | y_{1:t}) \int \frac{p(x_{t+1} | y_{1:T}) p(x_{t+1} | x_t)}{p(x_{t+1} | y_{1:t})} dx_{t+1}, \quad (16)$$

where, $p(x_t|y_{1:t})$ and $p(x_{t+1}|y_{1:t})$ are the filtering density and one step ahead predictive density respectively, at time t . Thus, starting with $p(x_T|y_{1:T})$, $p(x_t|y_{1:T})$ can be recursively obtained from $p(x_{t+1}|y_{1:T})$. Using the above recursion, the marginal smoothing distribution can now be approximated by the weighted particle cloud as described in ([6],[16]). Here, one starts with the forward filtering pass for computing the filtered distribution at each time step using particle filter as

$$\hat{P}(dx_t|y_{1:t}) = \sum_{i=1}^N \omega_t^{(i)} \delta_{x_t^{(i)}}(dx_t). \quad (17)$$

Then one performs the backward smoothing pass as given by (16) to approximate the smoothing distribution.

$$\hat{P}(dx_t|y_{1:T}) = \sum_{i=1}^N \omega_{t|T}^{(i)} \delta_{x_t^{(i)}}(dx_t), \quad (18)$$

where the smoothing weights are obtained through the following backward recursion:

$$\omega_{t|T}^{(i)} = \omega_t^{(i)} \sum_{j=1}^N \left[\omega_{t+1|T}^{(j)} \frac{p(x_{t+1}^{(j)}|x_t^{(i)})}{\sum_{k=1}^N p(x_{t+1}^{(j)}|x_t^{(k)}) \omega_t^{(k)}} \right] \quad (19)$$

with $\omega_{T|T}^{(i)} = \omega_T^{(i)}$. It is important to note that the forward- backward Smoother keeps the same particle support as used in filtering step and re-weights the particles to obtain the approximated particle based smoothed distribution. Thus, success of this method crucially hinges on the filtered distribution having supports where the smoothed distribution is significant.

To obtain the marginal MAP smoother, one needs the posterior density $p(x_t|y_{1:T})$ from the above cloud representation. Here, we proceed as follows. Using the Bayes' rule, one can write the one step ahead predictive density in equation (16) as

$$p(x_{t+1}|y_{1:t}) = \frac{p(x_{t+1}|y_{1:t+1})p(y_{t+1}|y_{1:t})}{p(y_{t+1}|x_{t+1})}. \quad (20)$$

Then equation (16) becomes

$$\begin{aligned} p(x_t|y_{1:T}) &= p(x_t|y_{1:t}) \int \frac{p(x_{t+1}|y_{1:T})p(x_{t+1}|x_t)p(y_{t+1}|x_{t+1})}{p(x_{t+1}|y_{1:t+1})p(y_{t+1}|y_{1:t})} dx_{t+1} \\ &= \frac{p(x_t|y_{1:t})}{p(y_{t+1}|y_{1:t})} \int \left[\frac{p(x_{t+1}|x_t)p(y_{t+1}|x_{t+1})}{p(x_{t+1}|y_{1:t+1})} \right] p(x_{t+1}|y_{1:T}) dx_{t+1} \\ &= \frac{p(x_t|y_{1:t})}{p(y_{t+1}|y_{1:t})} \int \left[\frac{p(x_{t+1}|x_t)p(y_{t+1}|x_{t+1})}{p(x_{t+1}|y_{1:t+1})} \right] \hat{P}(dx_{t+1}|y_{1:T}). \end{aligned}$$

Approximating the above integration by Monte Carlo integration method, one obtains

$$p(x_t|y_{1:T}) \approx \frac{p(x_t|y_{1:t})}{p(y_{t+1}|y_{1:t})} \sum_{j=1}^N \left[\frac{p(x_{t+1}^{(j)}|x_t)p(y_{t+1}|x_{t+1}^{(j)})}{p(x_{t+1}^{(j)}|y_{1:t+1})} \right] \omega_{t+1|T}^{(j)}. \quad (21)$$

Further approximating the filtered density $p(x_{t+1}|y_{1:t+1})$ from the running particle filter ([11]) as

$$p(x_{t+1}|y_{1:t+1}) \approx \frac{p(y_{t+1}|x_{t+1}) \sum_k p(x_{t+1}|x_t^{(k)}) \omega_t^{(k)}}{p(y_{t+1}|y_{1:t})} \quad (22)$$

we can rewrite equation (21) as

$$p(x_t|y_{1:T}) \approx p(x_t|y_{1:t}) \sum_{j=1}^N \left[\frac{p(x_{t+1}^{(j)}|x_t)}{\sum_{k=1}^N p(x_{t+1}^{(j)}|x_t^{(k)}) \omega_t^{(k)}} \right] \omega_{t+1|T}^{(j)}. \quad (23)$$

The MAP estimate of the marginal smoothing density, $p(x_t|y_{1:T})$ can then be obtained by finding the location of its global maxima. This maximization can be performed using different optimization methods. As in the case of pf-MAP, here, as well, we maximize along the particles. This leads to the approximate particle based MAP estimate as

$$x_{t|T}^{MAP} \approx \arg \max_{x_t^{(i)}} p(x_t^{(i)}|y_{1:t}) \sum_{j=1}^N \left[\frac{p(x_{t+1}^{(j)}|x_t^{(i)})}{\sum_{k=1}^N p(x_{t+1}^{(j)}|x_t^{(k)}) \omega_t^{(k)}} \right] \omega_{t+1|T}^{(j)}, \quad (24)$$

where $i = 1, \dots, N$ and N is the number of particles in the cloud at each time step. By using equation (19), the estimator can be further simplified to

$$x_{t|T}^{MAP} = \arg \max_{x_t^{(i)}} p(x_t^{(i)}|y_{1:t}) \frac{\omega_{t|T}^{(i)}}{\omega_t^{(i)}}, \quad (25)$$

where the filtered density $p(x_t|y_{1:t})$ at the particle cloud $\{x_t^{(i)}\}_{i=1}^N$ can be evaluated during the forward filtering step ([11]) as

$$p(x_t^{(i)}|y_{1:t}) \approx \frac{p(y_t|x_t^{(i)}) \sum_j p(x_t^{(i)}|x_{t-1}^{(j)}) w_{t-1}^{(j)}}{p(y_t|y_{1:t-1})}. \quad (26)$$

Since $p(y_t|y_{1:t-1})$ in equation (26) is independent of $x_t^{(i)}$, to obtain $x_{t|T}^{MAP}$, one can replace $p(x_t^{(i)}|y_{1:t})$ in equation (25) by the un-normalized filtered density

$$q(x_t^{(i)}|y_{1:t}) = p(y_t|x_t^{(i)}) \sum_j p(x_t^{(i)}|x_{t-1}^{(j)}) w_{t-1}^{(j)}. \quad (27)$$

We note here that a numerical problem may arise in evaluating equation (25) if the filtered weights attached to some particles are very small. This may happen when the "particle degeneracy" occurs. This problem can be effectively addressed using a combination of efficient importance proposal ([10],[15],[23]) with resampling steps. The algorithm for the marginal smoother MAP proposed above is given below.

Algorithm :

- Given observation $y_{1:T}$,
For $i = 1, \dots, N$, where N is the number of particles

Forward Filtering step

- Assume $p(x_0)$, draw $x_0^{(i)}$ from $p(x_0)$, set $\omega_0^{(i)} = \frac{1}{N}$.
- Run Particle Filter to generate and store $\{x_t^{(i)}, \omega_t^{(i)}\}$ for $t = 0, \dots, T$
- Evaluate (un-normalized) filtered pdf for $t = 1, \dots, T$, at cloud points i

$$q(x_t^{(i)} | y_{1:t}) = p(y_t | x_t^{(i)}) \sum_j p(x_t^{(i)} | x_{t-1}^{(j)}) \omega_{t-1}^{(j)}$$

starting with $q(x_0^{(i)}) = p(x_0^{(i)})$ and store

Backward Smoothing step

- Set $\omega_{T|T}^{(i)} = \omega_T^{(i)}$
- For $t = T - 1, \dots, 0$ evaluate the smoother importance weights as

$$\omega_{t|T}^{(i)} = \omega_t^{(i)} \sum_{j=1}^N [\omega_{t+1|T}^{(j)} \frac{p(x_{t+1}^{(j)} | x_t^{(i)})}{\sum_{k=1}^N p(x_{t+1}^{(j)} | x_t^{(k)}) \omega_t^{(k)}}]$$

- Evaluate the approximate smoother MAP as

$$x_{t|T}^{MAP} = \arg \max_{x_t^{(i)}} q(x_t^{(i)} | y_{1:t}) \frac{\omega_{t|T}^{(i)}}{\omega_t^{(i)}}$$

The memory requirement of this marginal MAP smoother is $O(N)$ and the computational complexity is $O(N^2)$. This complexity may possibly be reduced using the method suggested by Klass *et al* ([20]). We do not discuss this any further here.

6.1.1 Estimation of (unknown) initial condition

We first consider the linear Gaussian model as given by (3)-(4) with $\alpha = 0.8$, but, the initial state x_0 is assumed to be unknown (constant). The synthetic data $\{x_k, y_k\}_{k=0:500}$ is generated starting with $x_0^* = 10$. To estimate the unknown initial state x_0 , we start with initial prior $p(x_0) \sim U[0, 20]$ where $U[a, b]$ denotes uniform probability density function with lower bound a and upper bound b respectively. We use "efficient proposal" as given in ([10]) in the forward filtering step with particle sample size $N = 500$. The estimate of the initial unknown state is given by the particle based MAP of $p(x_0|y_{0:T})$. We repeat this MAP state estimate for 30 Monte Carlo runs. The mean and variance of the estimator are shown in Table 1. The result shows that the smoothed initial density peaks around the true initial state, even though we have started with a pretty wide uniform initial prior. We also plot for a particular realization, the (backward) evolution of the marginal smoother estimates (i.e. mean and the MAP) for the first 10 time steps and the un-normalized filtered and smoothed probability density functions (pdfs) of x_0 in figure 5 and figure 6 respectively. As expected, the mean and MAP are almost similar and the smoothed density is more concentrated than the filtered density around the true value 10. Next, we

$Mean(x_{0 500}^{MAP})$	$Var(x_{0 500}^{MAP})$
9.9726	0.0915

Table 1: Mean and Variance of estimated initial state

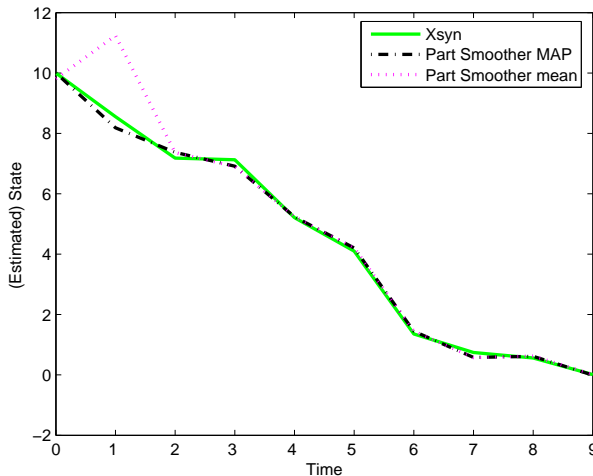


Figure 5: MAP and mean of the marginal smoothing posterior for the first 10 time steps

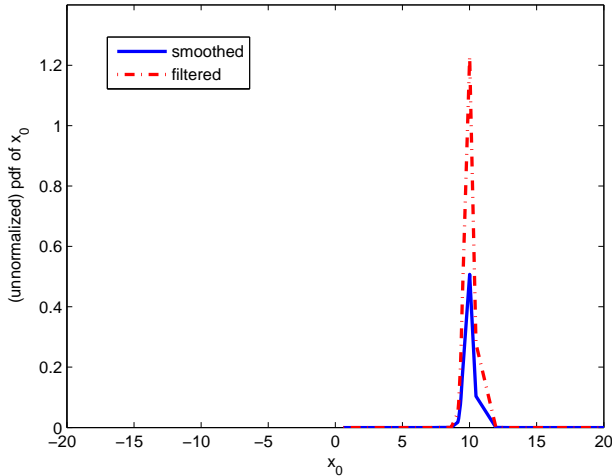


Figure 6: Filtered and smoothed probability density functions for the initial state x_0

consider the nonlinear time series model as given by (5) and (6). The synthetic data $\{x_k, y_k\}_{k=0:500}$ is generated starting with $x_0^* = 10$. As in the previous case, we start with initial prior $p(x_0) \sim U[0, 20]$. For this nonlinear problem, we use the "Exact Moment matching (EMM) proposal" as given in ([22]) during forward filtering step with particle sample size $N = 500$. The estimate of the initial unknown state is given by the particle based MAP of $p(x_0|y_{0:T})$. We repeat this MAP state estimate for 30 Monte Carlo runs. The mean and variance of the estimator are shown in Table 2. The result in Table 2 is really remarkable as we can see by comparing with Table 1. Even for highly nonlinear model as considered above and with wide uniform initial prior, the result is almost as good as in linear case. Of course the variance is somewhat larger, but that is to be expected given the highly nonlinear nature of the problem.

It is also interesting to study the behavior of the smoother when the initial

$Mean(x_{0 500}^{MAP})$	$Var(x_{0 500}^{MAP})$
9.7165	0.9236

Table 2: Mean and Variance of estimated initial state

distribution is supported on a larger interval. Starting with $p(x_0) \sim U[-40, 40]$, the (backward) evolution of the marginal smoother estimates (i.e. mean and the MAP) for the first 10 time steps for a particular realization are shown in figure 7 while the corresponding un-normalized filtered and smoothed pdfs for x_0 are shown in figure 8. It is interesting to note that the smoothed pdf of the initial state is bimodal (the smaller peak is near -10). Although the dominant

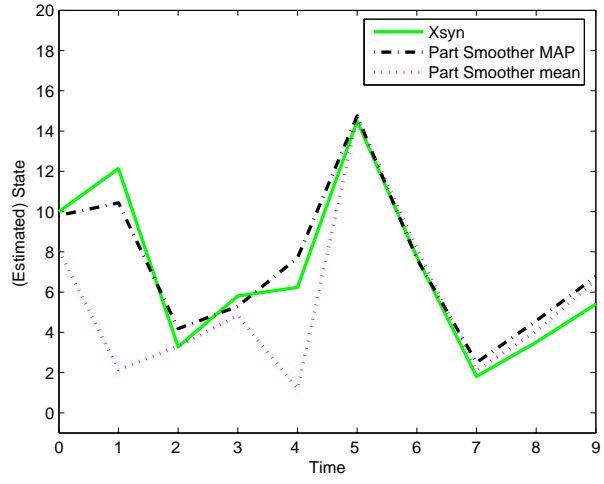


Figure 7: MAP and mean of the marginal smoothed posterior for the first 10 time steps

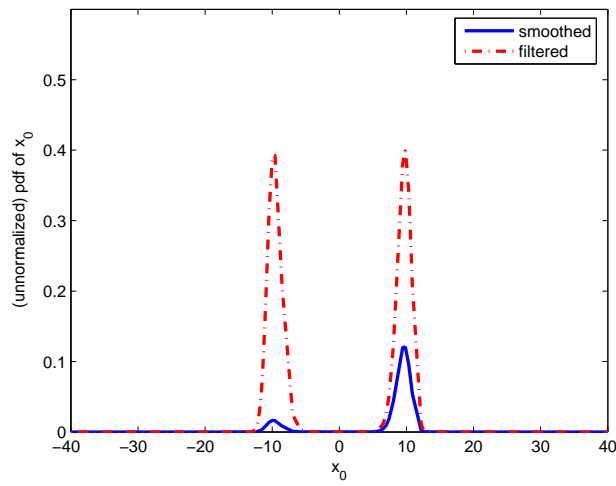


Figure 8: Filtered and smoothed probability density functions for the initial state x_0

mode is very close to the true initial state, $x_0^* = 10$, the contribution from the weaker mode, shifts the smoothed mean away from x_0^* (as seen from figure 7, the smoothed mean is near 8 here). This further strengthens the justification of using MAP in such scenario.

6.2 Two-Filter Smoothing

One of the drawbacks of the forward-backward smoother is its reliance on the support generated during the forward filtering pass. To circumvent this problem, two-filter smoother has been envisaged in the literature ([6],[19],[17]), where one combines samples from particle filter in the forward direction with those from a so called "backward information filter" to produce the (weighted) cloud representation of $p(x_t|y_{1:T})$. We describe, in this section, how the marginal MAP smoother can be obtained from the particle cloud generated by the generalized two-filter smoother.

In the two-filter smoother framework, the backward information filter is used to calculate $p(y_{t:T}|x_t)$ sequentially from $p(y_{t+1:T}|x_{t+1})$ as

$$p(y_{t:T}|x_t) = p(y_t|x_t) \int p(x_{t+1}|x_t)p(y_{t+1:T}|x_{t+1})dx_{t+1}. \quad (28)$$

As noted in [6], $p(y_{t:T}|x_t)$ is not a probability density function in x_t and actually, its integral over x_t may not even be finite. The smoothing algorithm in ([19],[17]) assumes implicitly that $\int p(y_{t:T}|x_t)dx_t < \infty$. However, if this assumption does not hold, SMC based methods, which can only approximate finite measures, will not work anymore. To avoid this, "generalized two-filter smoothing" has been proposed in [6], where the smoothing distributions are computed through a combination of forward filter and an auxiliary probability distribution $\tilde{p}(x_t|y_{t:T})$ in argument x_t . This auxiliary density is defined through a sequence of artificial distributions $\gamma_t(x_t)$ as

$$\tilde{p}(x_t|y_{t:T}) \propto \gamma_t(x_t)p(y_{t:T}|x_t).$$

It then follows from (28) that

$$\tilde{p}(x_t|y_{t:T}) \propto \gamma_t(x_t)p(y_t|x_t) \int p(x_{t+1}|x_t) \frac{\tilde{p}(x_{t+1}|y_{t+1:T})}{\gamma_{t+1}(x_{t+1})} dx_{t+1}, \quad (29)$$

which is used ([6]) to generate recursively the weighted particle representation of the backward information filter

$$\tilde{p}(x_t|y_{t:T}) \simeq \sum_{k=1}^N \delta(x_t - \tilde{x}_t^{(k)}) \tilde{\omega}_t^{(k)}. \quad (30)$$

The marginal smoother $p(x_t|y_{1:T})$ is then computed by combining the outputs of the forward filter and the backward information filter ([6]) as

$$\begin{aligned} p(x_t|y_{1:T}) &\propto p(x_t|y_{1:t-1})p(y_{t:T}|x_t) \\ &= \left(\int p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})dx_{t-1} \right) \left(\frac{\tilde{p}(x_t|y_{t:T})}{\gamma_t(x_t)} \right). \end{aligned} \quad (31)$$

Evaluating the integral in (31) by Monte Carlo integration using the forward filter cloud $(x_{t-1}^{(j)}, \omega_{t-1}^{(j)})$ one obtains

$$p(x_t|y_{1:T}) \propto \left(\sum_{j=1}^N p(x_t|x_{t-1}^{(j)})\omega_{t-1}^{(j)} \right) \left(\frac{\tilde{p}(x_t|y_{t:T})}{\gamma_t(x_t)} \right). \quad (32)$$

Finally, the particle cloud representation is obtained using the cloud $(\tilde{x}_t^{(k)}, \tilde{\omega}_t^{(k)})$ from the backward filter:

$$p(x_t|y_{1:T}) \simeq \sum_{k=1}^N \delta(x_t - \tilde{x}_t^{(k)}) \tilde{\omega}_{t|T}^{(k)} \quad (33)$$

where

$$\tilde{\omega}_{t|T}^{(k)} \propto \frac{\tilde{\omega}_t^{(k)}}{\gamma_t(\tilde{x}_t^{(k)})} \sum_{j=1}^N p(\tilde{x}_t^{(k)}|x_{t-1}^{(j)})\omega_{t-1}^{(j)}. \quad (34)$$

Thus, in essence the particles from the forward filter are used to re-weight those from the backward filter so that they represent the marginal smoother distribution. We refer the readers to the original article [6] for more details.

To derive the smoothing density from the particle smoother obtained as above we note that using (30) one can rewrite equation (29) as

$$\tilde{p}(x_t|y_{t:T}) \propto \gamma_t(x_t)p(y_t|x_t) \sum_{k=1}^N \frac{p(\tilde{x}_{t+1}^{(k)}|x_t)}{\gamma_{t+1}(\tilde{x}_{t+1}^{(k)})} \tilde{\omega}_{t+1}^{(k)}. \quad (35)$$

It then follows from (32) that

$$p(x_t|y_{1:T}) \propto \left(\sum_{j=1}^N p(x_t|x_{t-1}^{(j)})\omega_{t-1}^{(j)} \right) \left(p(y_t|x_t) \sum_{k=1}^N \frac{p(\tilde{x}_{t+1}^{(k)}|x_t)}{\gamma_{t+1}(\tilde{x}_{t+1}^{(k)})} \tilde{\omega}_{t+1}^{(k)} \right). \quad (36)$$

The required marginal smoother MAP can now be obtained by maximizing the unnormalized smoothing density, given by the right hand side of equation (36). Furthermore, when this maximization is done along the particles $\tilde{x}_t^{(i)}$, we have

$$\begin{aligned} & p(\tilde{x}_t^{(i)}|y_{1:T}) \\ & \propto \left(\sum_{j=1}^N p(\tilde{x}_t^{(i)}|x_{t-1}^{(j)})\omega_{t-1}^{(j)} \right) \left(p(y_t|\tilde{x}_t^{(i)}) \sum_{k=1}^N \frac{p(\tilde{x}_{t+1}^{(k)}|\tilde{x}_t^{(i)})}{\gamma_{t+1}(\tilde{x}_{t+1}^{(k)})} \tilde{\omega}_{t+1}^{(k)} \right) \\ & = \left(\frac{1}{\gamma_t(\tilde{x}_t^{(i)})} \sum_{j=1}^N p(\tilde{x}_t^{(i)}|x_{t-1}^{(j)})\omega_{t-1}^{(j)} \right) \left(\gamma_t(\tilde{x}_t^{(i)})p(y_t|\tilde{x}_t^{(i)}) \sum_{k=1}^N \frac{p(\tilde{x}_{t+1}^{(k)}|\tilde{x}_t^{(i)})}{\gamma_{t+1}(\tilde{x}_{t+1}^{(k)})} \tilde{\omega}_{t+1}^{(k)} \right). \end{aligned}$$

From equations (34) and (35) this reduces to

$$p(\tilde{x}_t^{(i)}|y_{1:T}) \propto \left(\frac{\tilde{\omega}_{t|T}^{(i)}}{\tilde{\omega}_t^{(i)}} \right) \left(\tilde{p}(\tilde{x}_t^{(i)}|y_{t:T}) \right). \quad (37)$$

Hence, the required MAP can be obtained as

$$x_{t|T}^{MAP} = \arg \max_{\tilde{x}_t^{(i)}} \tilde{p}(\tilde{x}_t^{(i)} | y_{t:T}) \frac{\tilde{\omega}_{t|T}^{(i)}}{\tilde{\omega}_t^{(i)}}, \quad (38)$$

where $\tilde{p}(\tilde{x}_t^{(i)} | y_{1:T})$ is evaluated using equation (35).

7 Conclusion

In this article, we study the particle filter based MAP estimates corresponding to a general state space model. A comparison of the existing filter MAP estimate ([11]) with the so called Viterbi-Godsill algorithm shows the superiority of the former. Exploiting the fact that the existing method provides the posterior density $p(x_t | y_{1:t})$ at any support point x_t , we have employed gradient based optimization method, which does not restrict itself to the particles when finding the MAP. We have, however, found that the gradient based method demands more computational load while estimation efficiency remaining almost the same. Some suggestions were made for improvement, though they have not been pursued. It has been shown numerically that by tweaking the number of particle during the evaluation of predictive density, one can reduce the computational load substantially without compromising much on RMSE value.

Finally, we have extended the idea of filter MAP to develop marginal smoother MAP based on a particle representation of the smoother distribution $p(x_t | y_{1:T})$. We derive a very simple formula for the smoother MAP when either forward-backward smoother or generalized two-filter smoother is used to generate the particle cloud. The smoother MAP is applied to estimate the unknown initial condition of a dynamic system. We observe that the estimation works quite well even in nonlinear setting.

References

- [1] S. Arulampalam, S. Maskell, N. Gordon and T. Clapp, "A Tutorial on Particle Filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transaction on Signal Processing*, vol. 50(2), pp. 174–188, Feb. 2002.
- [2] Y. Bar-Shalom and X. Li, *Multitarget-Multisensor Tracking: Principles and Techniques*, Academic Press, New York, 1995.
- [3] H. A. P. Blom, E. A. Bloem, Y. Boers and J. N. Driessen, "Tracking closely spaced targets: Bayes outperformed by an approximation?," in *Proc. of the 11th International Conference on Information Fusion*, Cologne, Germany, 2008.
- [4] Y. Boers and J. N. Driessen, "Particle filter based detection schemes," *Signal and data Processing of Small Targets, SPIE*, vol. 4728, pp. 128–137, 2007.

- [5] Y. Boers and J. N. Driessen, "The Mixed Labelling Problem in Multi-target Particle Filter," in *Proc. of the 10th International Conference on Information Fusion*, Quebec, Canada, 2007.
- [6] M. Briers, A. Doucet and S. R. Maskell, "Smoothing algorithm for state-space models," *Tech. Report CUED/F-INFENG/TR.498*, Cambridge University Engineering Department, Aug. 2004.
- [7] J. V. Candy, "Bootstrap particle filtering," *IEEE Signal Processing Magazine*, vol. 24(4), pp. 73–85, 2007.
- [8] O. Cappe, S. J. Godsill and E. Moulines, "An overview of existing methods and recent advances in sequential Monte Carlo," *IEEE Proceedings*, vol. 95(5), pp. 899–924, 2007.
- [9] H. Cox, "On the estimation of state variables and parameters for noisy dynamic systems," *IEEE Transaction on Automatic Control*, vol. 9(1), pp. 5–12, Jan. 1964.
- [10] A. Doucet, S. Godsill and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, pp. 197–208, 2000.
- [11] J. N. Driessen and Y. Boers, "Particle filter MAP estimation in dynamical systems," in *The IET Seminar on Target Tracking and Data Fusion: Algorithms and Applications*, Birmingham, UK, April. 2008.
- [12] J. N. Driessen and Y. Boers, "MAP Estimation in Nonlinear Dynamic Systems," submitted to *IEEE Transactions on Signal Processing*, 2008.
- [13] S. Godsill, A. Doucet and M. West, "Maximum a posteriori sequence estimation using Monte Carlo particle filters," *Annals of the Institute of Statistical Mathematics*, vol. 53, pp. 82–96, 2001.
- [14] N. J. Gordon, D. J. Salmond and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proc. F - Radar and Signal Processing*, vol. 140(2), pp. 107–113, 1993.
- [15] D. Guo, X. Wang and R. Chen, "New sequential Monte Carlo methods for nonlinear dynamic systems," *Statistics and Computing*, vol. 15(2), pp. 135–147, Apr. 2005.
- [16] M. Hurzeler and H. R. Kunsch, "Monte Carlo Approximations for General State-Space Models," *Journal of Computational and Graphical Statistics*, vol. 7(2), pp. 175–193, June. 1998.
- [17] M. Isard and A. Blake, "A smoothing filter for condensation," in *Proc. of 5th European Conference on Computer Vision*, pp. 767–781, 1998.

- [18] G. Kitagawa, "Non-Gaussian state-space modeling of nonstationary time series," *Journal of the American Statistical Association*, vol. 82(400), pp. 1032–1063, 1987.
- [19] G. Kitagawa, "Monte Carlo filter and smoother for non-Gaussian non-linear state space models," *Journal of Computational and Graphical Statistics*, vol. 5(1), pp. 1–25, 1996.
- [20] M. Klaas, M. Briers, N. de Freitas, S. Maskell and D. Lang, "Fast Particle smoothing: if I had a million particles," in *Proc. of the 23rd International Conference on Machine Learning*, Pittsburgh, Pennsylvania, 2006, pp. 481–488.
- [21] J. S. Liu and R. Chen, "Sequential Monte Carlo Methods for Dynamic Systems," *Journal of the American Statistical Association*, vol. 93(443), pp. 1032–1044, 1998.
- [22] S. Saha, P. K. Mandal, Y. Boers and H. Driessen, "Exact moment matching for efficient importance functions in SMC methods," in *Proc. of NSSPW 2006*, Cambridge, UK, 2006.
- [23] S. Saha, P. K. Mandal, Y. Boers, H. Driessen and A. Bagchi, "Gaussian proposal density using moment matching in SMC methods," *Statistics and Computing*, DOI 10.1007/s11222-008-9084-9.
- [24] B. Silverman, *Density Estimation for Statistics and Data Analysis*. Chapman and Hall/CRC, 1986.
- [25] S. K. Zhou, R. Chellappa and B. Moghaddam, "Visual Tracking and Recognition using appearance -adaptive Models in particle filters," *IEEE Transaction on Image Processing*, vol. 13(11), pp. 1491–1506, Nov. 2004.