

Particle Based MAP State Estimation: A Comparison

S. Saha¹, Y. Boers², H. Driessen², P.K. Mandal¹ and A. Bagchi¹

¹Department Of Applied Mathematics
University of Twente
The Netherlands

Email: s.saha, p.k.mandal, a.bagchi@ewi.utwente.nl

²THALES Nederland BV

Email: yvo.boers, hans.driessen@nl.thalesgroup.com

Abstract – MAP estimation is a good alternative to MMSE for certain applications involving nonlinear non Gaussian systems. Recently a new particle filter based MAP estimator has been derived. This new method extracts the MAP directly from the output of a running particle filter. In the recent past, a Viterbi algorithm based MAP sequence estimator has been developed. In this paper, we compare these two methods for estimating the current state and the numerical results show that the former performs better.

Keywords: filter MAP, Bayesian point estimation, sequential Monte Carlo, particle filter.

1 Introduction

The dynamic estimation problem concerned with estimating the unknown state given a set of measurements is a cornerstone for many practical problems. Applications include, inter alia, target tracking, process control and financial market analysis. The Bayesian approach to this problem is given by computing the posterior probability density function based on the observed measurements. 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 ([15]), particle based sequential Monte Carlo (SMC) 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 ([11],[1],[17]). For the inference purposes, however, instead of the whole posterior, often a single point estimate is more convenient. One such commonly used point estimate is the minimum mean square error (MMSE) estimate, which happens to be the mean of the posterior density. In particle filter set up, this can be easily obtained by the sample mean of the weighted particles.

Nonetheless, the MMSE estimate has some apparent disadvantages. For example, when the posterior is multimodal, the MMSE estimate may be located in a region (between the modes) where the posterior can have a very small value and thus, producing an unreasonable estimate. In such a scenario, another point estimate, namely, the maximum a posteriori (MAP) estimate, which picks the state that maximizes the posterior, is more relevant. The situation is described in (Figure 1).

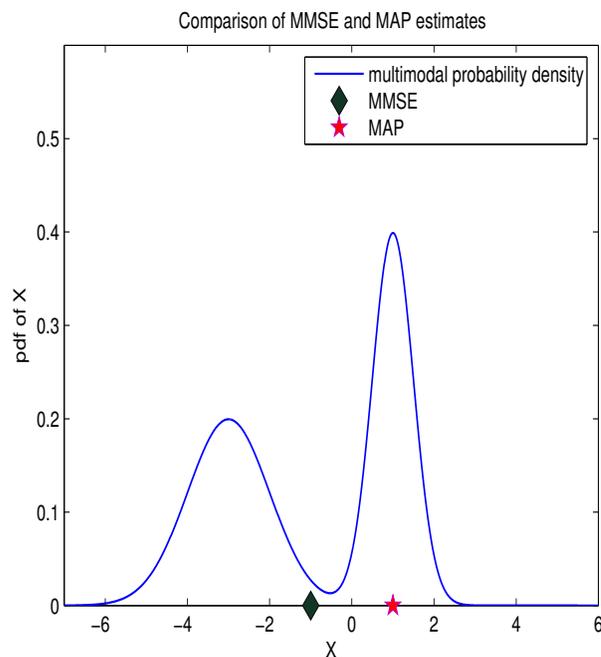


Figure 1: Comparison of MMSE and MAP estimates

In many real life applications multi modality will exhibit itself in a natural manner. Terrain aided navigation is one such example. In this application the *a posteriori* filtering distribution is defined in terms of

the position and velocity of an airplane. The measurements are obtained by means of an altimeter and a digital terrain map. In principle this information makes it possible to estimate the airplane's horizontal position and velocity, see e.g. ([8]) and references therein for more details. A snapshot of a particle filter based terrain navigation application in action is provided in (Figure 2). Similar terrain patterns (in altitude) lead to multi modality of the particle cloud. Also in target

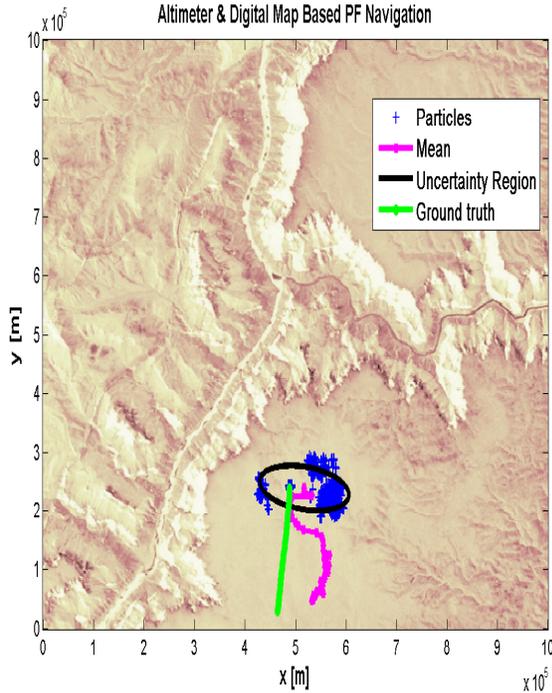


Figure 2: Terrain navigation example

tracking this multi modality is very common, e.g. due to data association problem, multiple models of target dynamics ([2],[3]) and mixed labeling problem in multiple target tracking, see ([5]) and ([6]). For such applications, extracting the exact object location is very crucial for subsequent strategic decisions and inference purposes. However, the popular MMSE based estimate may potentially lead to a situation where the estimates are far from the actual target positions. This problem is especially apparent in joint multi-target particle filters, see e.g. ([6]). An illustration of this is also provided in the following snapshot (Figure 3) showing two targets in a binary sensor network, see ([7]) for more modeling details.

In the example in (Figure 3) the two targets initially started out well separated, then they move toward each other, remain close to each other for quite some time, before separating again. Moreover, while together, they are also close in terms of the related sensor resolutions. We see that after separation, they cannot be distin-

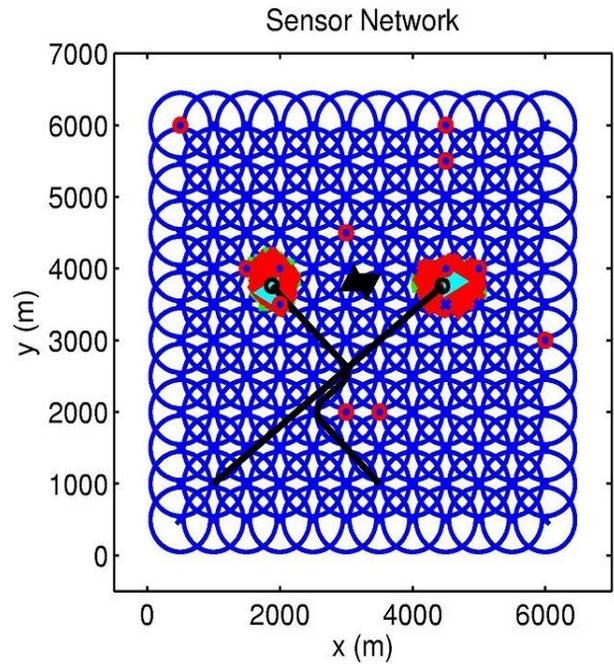


Figure 3: Two targets in binary sensor network example

guished by the filter. In this situation, the *a posteriori* probability density is called completely mixed, see ([6]). The results shown are the particle clouds for the two targets (in red and green), the corresponding MMSE estimates (the black triangles) and the true positions of both the targets (the black circles), which are well separated after passing each other as well as the MAP estimates (light blue triangles). Here, the MMSE estimates, which end up somewhere between the particle clouds, are obviously misleading as both estimates are far away from the actual target positions. The MAP estimates are obviously much closer to the true target positions. The reason for this is that the MAP estimator is not so badly influenced by this mixing and at least the state estimates obtained through the MAP estimator are quite good.

The examples presented in this section are a motivation for using the MAP estimate instead of the MMSE estimate, at least in certain applications. In the remainder of the paper we will describe two approaches to calculate the MAP estimates, based on a running particle filter. We will compare these two MAP estimates both in terms of computational load and performance.

2 Problem Description

Consider a state-space model

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

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

where x_k is the 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 problem here is to estimate sequentially the current state x_k given all the observations $y_{1:k} \equiv (y_1, y_2, \dots, y_k)$. In this paper, we will be concerned with the MAP estimate or equivalently, to estimate the value of x_k that maximizes the filtering density $p(x_k|y_{1:k})$. This can be stated mathematically as

$$x_{k|k}^{MAP} = \arg \max_{x_k} p(x_k|y_{1:k}). \quad (3)$$

3 Filter MAP estimate for a general nonlinear dynamic system

Estimating the MAP involves maximization over the posterior density, which is 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 ([19],[9],[4]). But, recently it has been shown that the particle with the highest weight does not necessarily represent the most probable state estimate and can actually be far from true MAP ([12],[10]). 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 ([18]). 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 ([12],[13]) have proposed a new 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 method. Furthermore, the approximate MAP estimates obtained through this method are shown to converge to the true MAP estimates. The method is briefly explained in section 3.1. In section 3.2, we describe another method which also provides an estimate of x_t (current state) using MAP.

3.1 Particle based filter MAP estimate (pf-MAP)

The method has originally been described in ([12],[13]). 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, consider the same dynamic system as given in equations (1)-(2). The MAP estimate of the filtering density at time k is then defined by equation (3).

For a general nonlinear model, analytical expression for the filtering density $p(x_k|y_{1:k})$ can hardly be ob-

tained in closed form. However, using particle filtering technique, one can approximate this posterior distribution by a cloud of N weighted particles as ([11])

$$\hat{P}(dx_k|y_{1:k}) \simeq \sum_{j=1}^N w_k^{(j)} \delta_{x_k^{(j)}}(dx_k). \quad (4)$$

Now using the Baye's rule, the posterior (filtering) density in equation (3) can be written as

$$p(x_k|y_{1:k}) = \frac{p(y_k|x_k)p(x_k|y_{1:k-1})}{p(y_k|y_{1:k-1})}. \quad (5)$$

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

$$p(x_k|y_{1:k}) \propto p(y_k|x_k)p(x_k|y_{1:k-1}). \quad (6)$$

The MAP estimate can thus be expressed as

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

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

$$p(x_k|y_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|y_{1:k-1})dx_{k-1} \quad (8)$$

and the running particle filter given by (4) to approximate $p(x_k|y_{1:k-1})$ using Monte Carlo integration as

$$p(x_k|y_{1:k-1}) \approx \sum_j p(x_k|x_{k-1}^{(j)})w_{k-1}^{(j)}. \quad (9)$$

Substituting (9) into (6) we get the posterior density and the MAP estimation is then obtained by finding the global maxima of it. At this point, we can in principle, employ any standard optimization technique to arrive at the MAP estimate. In general, however, this maximization step is nontrivial due to the possible multimodalities arising from the non Gaussian nature of the posterior. An approximate method for the MAP estimate, proposed in the original article, is described next.

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 (6) 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

$$x_k^{MAP} = \arg \max_{x_k^{(i)}} p(y_k|x_k^{(i)}) \sum_j p(x_k^{(i)}|x_{k-1}^{(j)})w_{k-1}^{(j)} \quad (10)$$

For the theoretical convergence properties of this estimator, the reader is referred to ([13]). One should

note that for each time step, the memory requirement of this MAP estimator is $O(N)$ and the computational complexity is $O(N^2)$. This complexity may possibly be reduced using the method suggested by Klaas *et al* ([16]). We do not discuss this any further in this paper.

3.2 End Point Viterbi-Godsill MAP (EP-VGM)

In the recent past, Godsill et al.([14]) have developed a method for estimating the MAP sequence in nonlinear non-Gaussian dynamic models using Viterbi algorithm. At each time t , 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 estimate of the sequence $x_{0:t}$. In target tracking problem, this corresponds to providing (new) MAP estimate of the whole trajectory of the target at every time point t (using the new observation). Clearly, 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) estimate of x_t . Actually in this method, the last element is obtained first and then using backward recursion, the other elements of the MAP sequence are obtained. For details, see ([14]).

4 Comparing pf-MAP with 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. In this context, we emphasize that the calculation of EP-VGM does not require the backtracking step of the Viterbi algorithm as we do not seek the whole sequence. Consequently, the CPU time for EP-VGM does not involve any backtracking step.

For our numerical experiments, we first start with the following linear Gaussian model:

$$x_k = x_{k-1} + w_k, \quad (11)$$

$$y_k = x_k + v_k, \quad k = 1, 2, \dots, \quad (12)$$

where the initial distribution $p(x_0) \sim \mathcal{N}(0, 2)$, $w_k \sim \mathcal{N}(0, 1)$ and $v_k \sim \mathcal{N}(0, 0.01)$. 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 Table 1 and Table 2 :

	pf-MAP	EP-VGM
Sample	RMSE	RMSE
100	0.007459	0.007430
200	0.006604	0.006653
400	0.006202	0.006224
1000	0.005948	0.006034

Table 1: Linear Gaussian model: RMSE

	pf-MAP	EP-VGM
Sample	avg. CPU	avg. CPU
100	0.0085	0.4765
200	0.0265	1.7461
400	0.1008	7.8359
1000	0.4890	49.9367

Table 2: Linear Gaussian model: average CPU time

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 following nonlinear model:

$$x_k = \frac{x_{k-1}}{2} + \frac{25x_{k-1}}{1+x_{k-1}^2} + 8\cos(1.2k) + w_k, \quad (13)$$

$$y_k = \frac{x_k^2}{20} + v_k, \quad k = 1, 2, \dots, \quad (14)$$

with the initial distribution $p(x_0) \sim \mathcal{N}(0, 5)$, $w_k \sim \mathcal{N}(0, 10)$ and $v_k \sim \mathcal{N}(0, 1)$. Since for this example, the true MAP can not be obtained analytically, we compare them as an estimator of the current state. The RMSE estimate is done with respect to true (synthetic) state over 20 Monte Carlo runs with 200 time steps. The results are shown in Table 3 and Table 4:

	pf-MAP	EP-VGM
Sample	RMSE	RMSE
100	5.2964	5.5667
250	4.9707	5.2567
500	4.8530	5.4184
1000	4.4659	5.2433

Table 3: Nonlinear model: RMSE

	pf-MAP	EP-VGM
Sample	avg. CPU	avg. CPU
100	3.515e-05	0.0033
250	1.836e-04	0.0194
500	6.054e-04	0.0851
1000	0.0022	0.3457

Table 4: Nonlinear model: average CPU time

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.

5 Conclusion

The posterior of a nonlinear non Gaussian dynamic system can be successfully constructed using particle filtering. The MMSE is a popular point estimate and can easily be obtained from this posterior. However, for certain practical applications like target tracking, where multi modality of posterior is very common, MMSE does not always represent a reasonable estimate. In such situations, MAP estimator can serve as a good alternative to MMSE. Recently a new particle based MAP estimator has been derived. In this paper, we compare this estimator with the so called Viterbi-Godsill MAP sequence estimator for estimating the current state given all available observations. Numerical experiments suggest the superiority of the former.

6 Acknowledgement

S. Saha's research is financially supported by a research grant from THALES Nederland B.V.

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] Y. Boers, E. Sviestins and J.N. Driessen. Mixed Labelling in Multi Target Particle Filtering. Accepted for publication in *IEEE Transactions on Aerospace and Electronic Systems*, 2008
- [7] Y. Boers, J. N. Driessen and L. Schipper, "Particle Filter Based Sensor Selection in Binary Sensor Networks," in *Proc. of the 11th International Conference on Information Fusion*, Cologne, Germany, July 2008.
- [8] F. Gustafsson. Particle Filter Theory and Practice, Provisionally accepted for *IEEE Aerospace and Electronic Systems*, Tutorial, 2009.
- [9] J. V. Candy, "Bootstrap particle filtering," *IEEE Signal Processing Magazine*, vol. 24(4), pp. 73–85, 2007.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] J. N. Driessen and Y. Boers, "MAP Estimation in Nonlinear Dynamic Systems," submitted, 2008.
- [14] 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.
- [15] 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.
- [16] 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.

- [17] 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.
- [18] B. Silverman, *Density Estimation for Statistics and Data Analysis*. Chapman and Hall/CRC, 1986.
- [19] 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.