A Bayesian filtering algorithm in Jump Markov Systems with application to Track-Before-Detect

Noémie Bardel*, Noufel Abbassi*, François Desbouvries*, Wojciech Pieczynski* and Frédéric Barbaresco†
*CITI department and CNRS UMR 5157
TELECOM SudParis
91011 Evry, France
Email: {Noemie.Bardel, Noufel.Abbassi, Francois.Desbouvries, Wojciech.Pieczynski}@it-sudparis.eu
†THALES AIR SYSTEMS
Surface Radar Business Line
Strategy, Technology & Innovation Department
Hameau de Roussigny, 91470 Limours, France
Email: frederic.barbaresco@thalesgroup.com

***Abstract*— Track-before-detect (TBD) aims at tracking trajectories of a target prior to detection by integrating raw measurements over time. Many TBD algorithms have been developed in the literature, based on the Hough Transform, Dynamic Programming or Maximum Likelihood estimation. However these methods fail in the case of maneuvering targets and/or non straight-line motion, or become very computationally expensive when the SNR gets low. Other techniques are based on the so-called switching or jump-Markov state-space system (JMSS) model. However, a drawback of JMSS is that it is not possible to perform exact Bayesian restoration. As a consequence, one has to resort to approximations such as particle filtering (PF). In this paper we propose an alternative method to approximate the optimal filter, which does not make use of Monte Carlo approximation. Our method is validated by computer simulations.**

I. Introduction

Tracking weak targets in a noisy or clutter environment is an important and non-trivial task in defence radar systems.

In classical radar target tracking, detection and tracking are performed successively. The radar measurements are thresholded in each frame, yielding so-called plots, and a hard decision is made instantaneously, i.e. without using other measurements from the near past. The subset of measurements where a detection has been done are then used for extracting tracks, i.e. for estimating the target kinematic state properties (position, velocity, acceleration).

Since the raw data is thresholded there is inevitably a loss of information. In particular, for targets with a weak signal-to-noise ratio (SNR), the threshold must be very low to enable an acceptable detection rate. However such a low threshold will also generate a high false alarm rate and then a lot of false tracks.

By contrast, in TBD algorithms detection and tracking are processed simultaneously. This approach uses the whole set of raw measured data (i.e., reflected power) and detection is performed only at the end of the processing chain, when all the information has been treated and integrated over time. Since the pre-thresholding step is avoided, this approach facilitates the detection of low-SNR targets.

Many TBD algorithms have been developed in the literature. In [1], a feature detector based on the Hough Transform is used to detect the tracks from the measured data. This method gives good results for slow maneuvering targets with straight-line motion but fails when those conditions are not satisfied. This issue is avoided by the use of a Dynamic Programming (DP) algorithm [2], [3], [4] which extracts tracks by performing an efficient exhaustive search over all possible trajectories. However the method becomes computationally very expensive when the SNR gets too low. In [5], a maximum likelihood approach is applied to track targets, but only with a constant velocity.

Another classical TBD approach consists in modelling the radar problem by a JMSS model [6], [8], [9], [10], [12]. However, even in the simplest (i.e. linear and Gaussian) case, exact Bayesian restoration is not feasible since the posterior density is a Gaussian mixture in which the number of components grows exponentially with time index [11], [12], [13]. Hence a practical filter in such a model has to be suboptimal and based on some approximations. An intuitive solution would be to limit the number of mixture components in some way (e.g. by pruning or merging) as done in the popular Interacting Multiple Model (IMM) algorithm [14]. Even if these methods are computationally cheap they can fail in difficult situations. More recently, an alternate class of algorithms based on sequential Monte Carlo approximations has been developed and widely disseminated in the tracking field.

The aim of this paper is to introduce an alternative approximate solution to the radar TBD problem in the JMSS model, which is not based on a Monte Carlo approximation. More precisely, we modify the relationship between the random variables of the model to perform an exact calculation of some probability density function (pdf) of interest.

The paper is organized as follows. In section II we recall the JMSS model and the associated particle filtering (PF) methodology. In section III we present our alternate Bayesian filtering algorithm, which performances as compared to existing techniques are demonstrated by simulations in section IV.

II. Particle filtering algorithms in JMSS

JMSS are generalizations of the classical state-space system in which a so-called "jump" variable is introduced in order to monitor the abrupt change of channel or kinematic characteristics and/or the presence (or absence) of a target in a tracking context.

Let us consider the following general JMSS model :

$$x_n = f(r_n, x_{n-1}, w_n) \quad (1)$$
$$y_n = h(r_n, x_n, v_n) \quad (2)$$

in which n denotes the discrete time index, x_n is the state vector, y_n the observation, w_n the process noise, v_n the measurement noise, $r_n \in \{1, ..., S\}$ is the discrete jump variable which classically is modeled by an exogeneous Markov chain, and f and h are possibly nonlinear functions (in our tracking application it will be assumed that the state equation (1) is linear with additional noise (see section 4), however for the moment we do not need to make this assumption and we thus directly deal with the general model (1)-(2)). Furthermore we assume that $\{w_n\}$ and $\{v_n\}$ are sequences of identically distributed random variables from known probability density functions (pdf) which are independent, mutually independent and independent of x_0, the pdf of which is also known.

This model has been widely used for single target tracking in radar environment (see [6], [8], [9], [15] or [17] for example). The jump variable r_n can classically model the absence or presence of the target in the measurement y_n, or the changes of the kinematic characteristics of the state-space model in the case of a maneuvering target.

A classical Bayesian filtering problem consists in propagating in time the joint posterior pdf $p(x_{0:n}, r_{1:n}|y_{1:n})$. This problem cannot be solved exactly, even in the linear and Gaussian case [12], [13], so one has to resort to suboptimal solutions. In particular, PF based methods propagate in time a discrete random approximation of $p(x_{0:n}, r_{1:n}|y_{1:n})$ (or a marginal thereof).

More precisely, these methods can be subdivided into two groups: those in which PF is used for estimating both $x_{0:n}$ and $r_{1:n}$, and those in which PF is used for estimating $p(r_{1:n}|y_{1:n})$ only.

Let us briefly describe the first class of methods (see e.g. [16], [17], [18]), that is a Sequential Importance Sampling and Resampling (SISR) procedure applied to the augmented state (x_n, r_n). Assume that at time $n-1$, we have N_p weighted trajectories $\{x^i_{0:n-1}, r^i_{1:n-1}\}_{i=1}^{N_p}$ sampled from an importance distribution $q(x_{0:n-1}, r_{1:n-1}|y_{1:n-1})$ and associated with the importance weights $\{w^i_{n-1}\}_{i=1}^{N_p}$, where:

$$w^i_{n-1} = \frac{p(x^i_{0:n-1}, r^i_{1:n-1}|y_{1:n-1})}{q(x^i_{0:n-1}, r^i_{1:n-1}|y_{1:n-1})}$$

Then it is possible to approximate $p(x_{0:n-1}, r_{1:n-1}|y_{1:n-1})$ by the weighted random measure:

$$\sum_{i=1}^{N_p} w^i_{n-1}\delta_{(x^i_{0:n-1}, r^i_{1:n-1})}(x_{0:n-1}, r_{1:n-1}) \quad (3)$$

At time n, we extend each trajectory $(x^i_{0:n-1}, r^i_{1:n-1})$ by sampling (x^i_n, r^i_n) from the conditional importance density (CID) $q(x_n, r_n|x^i_{0:n-1}, r^i_{1:n-1}, y_{1:n})$ and updating the importance weights as:

$$w^i_n \propto \frac{p(y_n|x^i_n, r^i_n)p(x^i_n|x^i_{n-1}, r^i_n)p(r^i_n|r^i_{n-1})}{q(x^i_n, r^i_n|x^i_{1:n-1}, r^i_{1:n-1}, y_{1:n})} w^i_{n-1}$$

Then $\{(x^i_{0:n}, r^i_{1:n}) = (x^i_{0:n-1}, x^i_n, r^i_{1:n-1}, r^i_n), w^i_n\}_{i=1}^{N_p}$ yields an approximation of $p(x_{0:n}, r_{1:n}|y_{1:n})$.

The performance of the algorithm depends in particular on the choice of the CID. The one that minimizes the variance of the weights conditionally to $y_{1:n}$ is $p(x_n, r_n|x_{n-1}, r_{n-1}, y_n) = p(r_n|r_{n-1})p(x_n|x_{n-1}, r_n, y_n)$ and is commonly called optimal CID. However it is often not computable so that a common choice is the transitional prior $p(x_n, r_n|x_{n-1}, r_{n-1}) = p(r_n|r_{n-1})p(x_n|x_{n-1}, r_n)$.

Let us now describe methods in which PF is used only for estimating $p(r_{1:n}|y_{1:n})$. This Rao-Blackwellised algorithm has been first proposed in [10] for linear JMSS models, and further extended to nonlinear systems (see e.g. [15]). Let us start from the following factorization of the joint posterior pdf:

$$p(x_n, r_{1:n}|y_{1:n}) = p(x_n|r_{1:n}, y_{1:n})p(r_{1:n}|y_{1:n}) \quad (4)$$

Given $r_{1:n}$, the first term in the right side of (4) is the familiar filtering pdf in a general state-space system and can thus be approximated by a nonlinear filtering method, such as the Extended Kalman Filter (EKF) or the more efficient Unscented Kalman Filter (UKF) [25], [26], which we briefly present now.

The principle of UKF is as follows: instead of linearizing the equations of the JMSS model, UKF approximates the first and second order statistics of the conditional distribution of the state random variable $p(x_n|y_{1:n})$ with a minimal set of deterministically chosen sample points and propagates them through the true nonlinear equations of the model. More precisely, assume that a set of L points $\{\bar{x}^l_{n-1|n-1}\}_{l=1}^L$ called "sigma points" possesses the correct conditional mean $m_{n-1|n-1}$ and covariance $P_{n-1|n-1}$. Then the set $\{\bar{x}^l_{n|n-1} = f(\bar{x}^l_{n-1|n-1})\}_{l=1}^L$ obtained by propagating the sigma points through the state equation has a mean and a covariance reasonably close from the true $m_{n|n-1}$ and $P_{n|n-1}$. Then the computation of $\{\bar{y}^l_{n|n-1} = h(\bar{x}^l_{n|n-1})\}_{l=1}^L$ leads to a reasonable approximation of $E[y_n|y_{1:n-1}]$, $Cov(y_n|y_{1:n-1})$ and $Cov(x_n, y_n|y_{1:n-1})$. From that it is possible to compute $m_{n|n}$ and $P_{n|n}$ by a Kalman-like procedure.

Therefore the second term $p(r_{1:n}|y_{1:n})$ in (4) can be estimated by a PF with optimal CID $p(r_n|r^i_{1:n-1}, y_{1:n}) \propto p(y_n|r^i_{1:n-1}, r_n, y_{1:n-1})p(r_n|r^i_{n-1})$, where $p(y_n|r^i_{1:n-1}, r_n, y_{1:n-1})$ is the value at point y_n of a Gaussian distribution with parameters calculated via UKF. The associated weights are thus given by

$$w^i_k \propto p(y_n|r^i_{1:n-1}, y_{1:n-1}) = \sum_{j=1}^{S} p(y_n|r^i_{1:n-1}, r_n = j, y_{1:n-1})$$

and require the execution of S prediction steps of UKF. Then an estimate of the conditional expectation of x_n is

$$\hat{E}(x_n|y_{1:n}) = \sum_{i=1}^{N_p} w^i_n E(x_n|y_{1:n}, r^i_{1:n}).$$

At each time the algorithm requires then to run SN_p UKF in parallel.

The complexity of this algorithm can be reduced by using the prior distribution $p(r_n|r^i_{n-1})$ and the associated weights $w^i_n \propto p(y_n|y_{1:n-1}, r^i_{1:n})$, which are evaluated by only one prediction step of UKF. At the end the estimate of the conditional expectation of x_n requires to run N_p UKF at each time.

III. An alternative Bayesian filtering algorithm

Let us consider again the nonlinear model (1)-(2). Instead of approximating (totally or partially) a posterior pdf by PF, and then taking an expectation of interest, we directly focus on the computation of $E[x_n|y_{1:n}]$ itself, which is our ultimate goal for tracking problems. So let us start from:

$$E[x_n|y_{1:n}] = \sum_{r_n} E[x_n|r_n, y_{1:n}]p(r_n|y_{1:n}) \qquad (5)$$

We see that $E[x_n|y_{1:n}]$ can be computed if we can calculate (or estimate) $E[x_n|r_n, y_{1:n}]$ and $p(r_n|y_{1:n})$ for each value of $r_n \in \{1, ..., S\}$. Let us consider these two subproblems one after the other.

First, when the value of r_n is fixed, the problem is equivalent to the computation of a conditional expectation in a nonlinear state-space model $(x_{1:N}, y_{1:N})$, which can be done by some nonlinear filtering technique. In particular $E[x_n|r_n, y_{1:n}]$ can be computed at time n by using UKF . As a result, if $p(r_n|y_{1:n})$ can be evaluated, the computation of (5) requires to run S UKF in parallel. Compared to the algorithm presented in section II, this is a major improvement in term of computational complexity since $N \gg S$ in general.

Let us now focus on the computation of $p(r_n|y_{1:n})$. Since the exact computation of this quantity is an NP-hard problem we need to use some approximation.

In order to avoid the estimation of r_n by PF, we make the assumption that $(r_{1:N}, y_{1:N})$ is a stochastic model in which the computation of $p(r_n|y_{1:n})$ is feasible.

Let us thus assume that $(r_{1:N}, y_{1:N})$ is a Partially Pairwise Markov Chain (PPMC) [23], whose joint distribution satisfies the following equation:

$$p(r_{1:N}, y_{1:N}) = p(r_1, y_1) \prod_{n=1}^{N-1} p(r_{n+1}, y_{n+1}|r_n, y_{1:n}) \qquad (6)$$

and such that $p(y|r)$ are Gaussian and $p(r_{n+1}|r_n, y_{1:n}) = p(r_{n+1}|r_n)$ (which implies that $r_{1:N}$ is a Markov Chain) [22].

The interests of this assumption are multiple. First it takes into accounts long memory observations; next the densities $p(r_n|y_{1:n})$ can be computed exactly from the estimated parameters by a forward procedure. Finally the parameters $p(r_1, y_1)$, $p(r_{n+1}|r_n)$ and $p(y_{n+1}|r_{n+1}, r_n, y_{1:n})$ of the PPMC which is closest to the true model $(r_{1:N}, y_{1:N})$ can be estimated by an Iterative Conditional Estimation (ICE) method [22]. This implies in particular that the prior distribution $p(r_{n+1}|r_n)$ of the Markov Chain can be estimated. Hence it has not to be necessarily known and partially unsupervised estimation is feasible.

To sum up, rather than using PF for approximating $p(r_{1:n}|y_{1:n})$ in (4) (as in the methods described at the end of section II), we approximate the stochastic model which describes the relations between $r_{1:N}$ and $y_{1:N}$, but we compute $p(r_n|y_{1:n})$ exactly.

At this point of the discussion it is not clear to see which method is the most efficient. However simulations show that our alternative filtering algorithm indeed outperforms the PF method presented at the end of the section II, see section IV.

IV. Simulations

A. Specification of the model

For the TBD application we use a model similar to [7]. Let us consider the following dynamic model:

$$x_n = Ax_{n-1} + Gw_n$$

in which the state vector $x_n = [\rho_n \quad d_n]^T$ is composed by the range and the doppler speed of the target at time n. If T denotes the sampling interval, we will take

$$A = \begin{pmatrix} 1 & T \\ 0 & 1 \end{pmatrix}, G = \sigma \begin{pmatrix} \sigma_\rho & 0 \\ 0 & \sigma_d \end{pmatrix}, w_n \sim \mathcal{N}(0,1).$$

To modelise the fact that the target can be either present or absent from the field-of-view, we introduce a two-state stationary Markov chain $\{r_{1:N}\}$ which takes its values in $\{0,1\}$: $r_n = 0$ (resp. 1) if the target is absent (resp. present) at time n. This chain is characterized by its initial distribution μ_1 and its transition probabilities matrix

$$\Pi = \begin{pmatrix} 1 - P_{01} & P_{01} \\ P_{10} & 1 - P_{10} \end{pmatrix}.$$

At each time the observation y_n is a vector composed by $N_\rho \times N_d$ measurements y_n^{ij}, where N_ρ and N_d are the number of range and doppler cells. The measurement equation is the following:

$$\begin{aligned} y_n &= h_n(x_n) + v_n & \quad if \quad r_n = 1 \\ y_n &= v_n & \quad if \quad r_n = 0 \end{aligned}$$

When the target is present, we have:

$$h^{ij}(x_n) = P - \frac{(\rho_i - \rho_n)^2}{K_\rho} L_\rho - \frac{(d_i - d_n)^2}{K_d} L_d$$

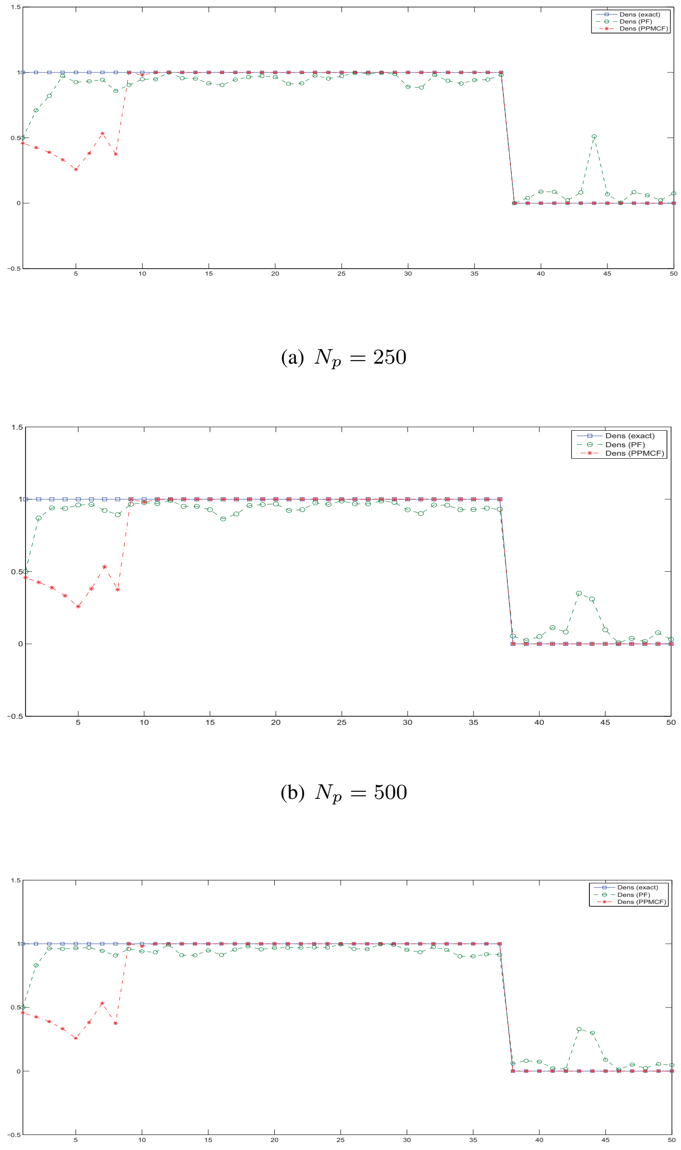
in which K_ρ and K_d are constants related to the size of the cells, L_ρ and L_d are somme constants of loss, and ρ_n and d_n are the range and doppler of the target.

Furthermore the additive noise v_n is assumed to follow an exponential distribution.

B. Simulations results

We compare the UKF/PF algorithm presented in the end of section II and the UKF/PPMC algorithm presented in section III by simulations. Since ICE is an iterative procedure the accuracy of the algorithm depends on the number of iterations. In practice, we see that only a few are sufficient to ensure good estimation results. Hence the procedure is very fast and requires a low computational cost.

The performance of the PF depends on the number of particles and on the choice of the importance density. To limit the computational complexity we use the prior distribution $p(r_n|r_{n-1})$. We have run the algorithms for different numbers of particles (see figure 1 and table I), and found that a good compromise between a reasonable computational cost and good estimation results is $N_p = 500$.



(a) $N_p = 250$

(b) $N_p = 500$

(c) $N_p = 750$

Fig. 1. Estimates of $p(r_n = 1|y_{1:n})$ for different values of N_p - The true density is in squared line, the UKF/PF estimated density in circled line and the UKF/PPMC estimated density in crossed line

	UKF/PF	UKF/PPMC
$N_p = 250$	0.2803	0.2902
$N_p = 500$	0.2766	0.2902
$N_p = 700$	0.2744	0.2902

TABLE I

MSE of the state variable estimates $\hat{x}_n$ for different values of N_p over 50 time steps

In figure 2 and table II we compare the estimates of the jump variables r_n given by the UKF/PF and UKF/PPMC algorithms to the true values of r_n and give the associated mean square errors (MSE), for different values of the probability of transition P_{01} (which we take equal to P_{10}). We can remark that the UKF/PPMC algorithm needs about 10 time steps to become accurate, and then provides an estimate very close to the true value of the jump variable. If the performances of the two filters are quite similar for small values of P_{01}, it has to be noticed that for a larger probability of transition, that corresponds to a more stealthy target, the UKF/PPMC algorithm gives much more accurate estimates of r_n than the UKF/PF algorithm.

In tables I and II it has to be noticed that the calculated mean square errors take into account all the estimates $\hat{x}_n$ up to time $n = 50$. Since the UKF/PPMC algorithm needs a few iterations to become efficient, these quantities are not always lower than those associated to the UKF/PF algorithm. However after some time steps the estimates $\hat{x}_n$ themselves are more accurate for the UKF/PPMC algorithm, as a consequence of a better estimation of the jump variable.

	UKF/PF	UKF/PPMC
$P_{01} = 0.02$	0.0229	0.0178
$P_{01} = 0.1$	0.0352	0.0297
$P_{01} = 0.3$	0.0631	0.0309

TABLE II

MSE of the jump variable estimates $\hat{r}_n$ for different values of P_{01} over 50 time steps

Remark: Our algorithm is based on model (6), which enables the exact computation of $p(r_n|y_{1:n})$. Note that this is not the only model in which $p(r_n|y_{1:n})$ can be computed exactly. For instance we could assume the classical Hidden Markov Chain model with independent noises defined by:

$$p(r_{1:N}, y_{1:N}) = p(r_1) \prod_{k=2}^{N} p(r_k|r_{k-1}) \prod_{k=1}^{N} p(y_k|r_k)$$

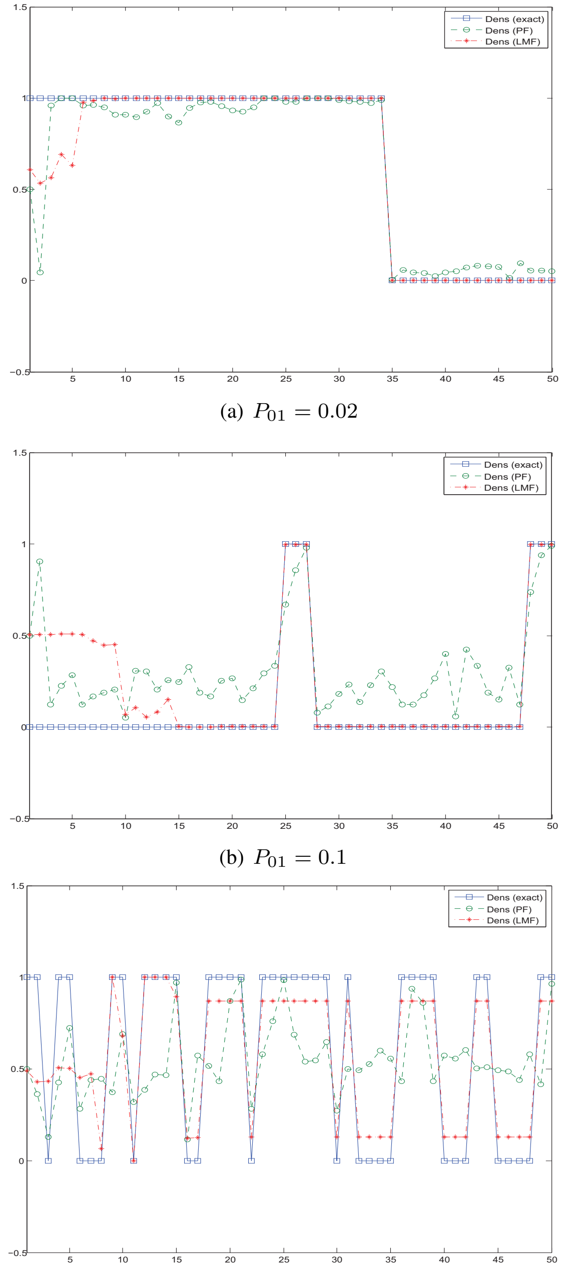
and use the Forward-Backward (or Baum-Welch) algorithm [27], [28], [29], or a Pairwise Markov Chain Model:

$$p(r_{1:N}, y_{1:N}) = p(r_1, y_1) \prod_{k=2}^{N} p(r_k, y_k|r_{k-1}, y_{k-1})$$

and use the algorithm presented in [23]. We implemented these different algorithms; the best results are available when $(r_{1:N}, y_{1:N})$ is assumed to follow model (6), probably because this model takes into account long dependencies, see [22] for details.

V. Conclusion

In this paper we addressed a Bayesian restoration algorithm in nonlinear JMSS model for Track-Before-Detect. We proposed a new solution in which a Monte Carlo approximation is replaced by an exact computation in an alternate model which takes into account long-range dependencies. Simulations showed that this algorithm is competitive in term of efficiency and performance.



(a) $P_{01} = 0.02$

(b) $P_{01} = 0.1$

(c) $P_{01} = 0.3$

Fig. 2. Estimates of $p(r_n = 1|y_{1:n})$ for different values of P_{01} - The true density is in squared line, the UKF/PF estimated density in circled line and the UKF/PPMC estimated density in crossed line

Acknowledgment

The authors would like to thank the French MOD DGA / MRIS for financial support of the PhD of N. Bardel.

REFERENCES

[1] B. D. Carlson, E. D. Evans and S. L. Wilson, "Search radar detection and track with the Hough transform", *IEEE Transactions on Aerospace and Electronic Systems*, vol. 30, pp. 102-108, January 1994.

[2] J. Arnold, S. Shaw and H. Pasternack, "Efficient target tracking using dynamic programming", *IEEE Transactions on Aerospace and Electronic Systems*, vol. 29, pp. 44-56, January 1993.

[3] S. M. Tonissen and R. J. Evans, "Performance of dynamic programming techniques for track-before-detect", *IEEE Transactions on Aerospace and Electronic Systems*, vol. 32, pp. 1440-1451, October 1996.

[4] L. A. Johnston and V. Krishnamurthy, "Performance analysis of a dynamic programming track before detect algorithm", *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, pp. 228-242, January 2002.

[5] S. M. Tonissen and Y. Bar-Shalom, "Maximum likelihood track-before detect with fluctuating target amplitude", *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, pp. 796-808, July 1998.

[6] D. J. Salmond and H. Birch, "A particle filter for track-before-detect", in *Proceedings of the American Control Conference*, Arlington, VA, pp. 3755-3760, June 2001.

[7] Y. Boers and J. N. Driessen, "Particle filter based detection for tracking", *Proceedings of the American Control Conference*, Arlington, VA, pp. 3755-3760, June 2001.

[8] Y. Boers and J. N. Driessen, "Multitarget particle filter track before detect application", *IEE Proc. Radar Sonar Navig.*, Vol 151, N. 6, December 2004.

[9] Y. Boers and J. N. Driessen, "A Track Before Detect Approach for Extended Objects", *9th International Conference on Information Fusion*, pp. 1-7, July 2006.

[10] A. Doucet, N. J. Gordon and V. Krishnamurthy, "Particle filters for state estimation of Jump Markov Linear Systems", *IEEE Transactions on Signal Processing*, vol. 49, 613-624, March 2001.

[11] G.A. Ackerson and K.S. Fu, "On state estimation in switching environments", *IEEE Transactions on Automatic Control*, vol. AC-15, N.1, pp. 10-17, February 1970.

[12] J. K. Tugnait, "Adaptive estimation and identification for discrete systems with Markov jump parameters", *IEEE Transactions on Automatic Control* 25, 1054-1065, 1982.

[13] Y. Bar-Shalom, X. R. Li and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*, New York: John Wiley & Sons, 2001.

[14] H.A.P. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with Markovian switching coefficients", *IEEE Transactions on Automatic Control*, vol.33, pp. 780-783, 1988.

[15] B. Ristic, S. Arulampalam and N. Gordon, *Beyond the Kalman Filter - Particle filters for tracking applications*, Artech House, Boston, MA. 2004.

[16] C. Musso, N. Oudjane and F. LeGland, "Improving regularised particle filters", in *Sequential Monte Carlo Methods in Practice* (A. Doucet, N. De Freitas, and N.J. Gordon, eds.), New York: Springer, 2001.

[17] S. McGinnity and G. W. Irwin, "Multiple model bootstrap filter for maneuvring target tracking", *IEEE Transactions Aerospace and Electronic Systems*, vol. 36, no.3, pp. 1006-1012, 2000.

[18] O. Cappé, E. Moulines and T. Ryden, *Inference in hidden Markov models*, Springer. 2005.

[19] O. Zoeter and T. Heskes, "Deterministic approximate inference techniques for conditionally Gaussian state space models", *Statistical Computation*, 16, 279-292, 2006.

[20] P. Giordani, R. Kohn and D. Van Dijk, "A unified approach to nonlinearity, structural change, and outliers", *Journal of Econometrics*, 137, 112-133, 2007.

[21] W. Pieczynski, and F. Desbouvries, "On triplet Markov chains", *International Symposium on Applied Stochastic Models and Data Analysis (ASMDA)*, Brest, France, 2005.

[22] P. Lanchantin, J. Lapuyade-Lahorgue and W. Pieczynski, "Unsupervised segmentation of triplet Markov chains with long-memory noise", *Signal Processing*, Vol.8, No.5, pp. 1134-1151, May 2008.

[23] S. Derrode and W. Pieczynski, "Signal and Image Segmentation using Pairwise Markov Chains", *IEEE Trans. On Signal Processing*, Vol. 52, No. 9, pp. 2477-2489, 2004.

[24] N. Abbassi and W. Pieczynski, "Filtrage exact partiellement non supervisé dans les modèles cachés à sauts markoviens", *GRETSI 2009*, Dijon, 8-11 septembre 2009.

[25] S. J. Julier and J. K. Uhlmann, "A new extension of the Kalman Filter to nonlinear systems", *Proc. of AeroSense: The 11th International Symposium on Aerospace/Defence Sensing, Simulation and Controls*, Orlando, Florida., Vol. Multi Sensor Fusion, Tracking and Resource Management II, 1997.

[26] S. J. Julier and J. K. Uhlmann, "Unscented Filtering and Nonlinear Estimation", *Proc. IEEE*, Vol. 92, pp. 401-422, Mar. 2004.

[27] L. R. Bahl, J. Cocke, F. Jelinek and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate", *IEEE Transactions on Information Theory*, Vol. 20, no. 2, pp. 284-287, Mar. 1974.

[28] A. P. Dempster, N. M. Laird and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm", *Journal of the Royal Statistical Society*, Series B, Vol. 39, no. 1, pp. 1-38, 1977.

[29] G. J. McLachlan and T. Krishnan, *EM algorithm and extensions*, Wiley, 1997.