

Tracking Multiple Objects with Particle Filtering

C. HUE

J-P. LE CADRE, Member, IEEE
IRISA/Université de Rennes1, IRISA/CNRS
France

P. PÉREZ

Microsoft Research

We address the problem of multitarget tracking (MTT) encountered in many situations in signal or image processing. We consider stochastic dynamic systems detected by observation processes. The difficulty lies in the fact that the estimation of the states requires the assignment of the observations to the multiple targets. We propose an extension of the classical particle filter where the stochastic vector of assignment is estimated by a Gibbs sampler. This algorithm is used to estimate the trajectories of multiple targets from their noisy bearings, thus showing its ability to solve the data association problem. Moreover this algorithm is easily extended to multireceiver observations where the receivers can produce measurements of various nature with different frequencies.

Manuscript received December 21, 2000; revised January 24, 2002; released for publication February 25, 2002.

IEEE Log No. T-AES/38/3/11413.

Refereeing of this contribution was handled by P. K. Willett.

Authors' addresses: C. Hue and J-P. Le Cadre, IRISA, Campus de Beaulieu, 35042 Rennes Cedex, France, E-mail: ({chue,lecadre}@irisa.fr); P. Pérez, 7 J.J. Thomson Avenue, Cambridge, CB3 0FB, UK, E-mail: (pperez@microsoft.com).

0018-9251/02/\$17.00 © 2002 IEEE

I. INTRODUCTION

Multitarget tracking (MTT) deals with the state estimation of an unknown number of moving targets. Available measurements may both arise from the targets if they are detected, and from clutter. Clutter is generally considered as a model describing false alarms. Its (spatio-temporal) statistical properties are quite different from those of the target, which makes the extraction of target tracks from clutter possible. To perform MTT the observer can rely on a huge amount of data, possibly collected on multiple receivers. Elementary measurements are receiver outputs, e.g., bearings, ranges, time-delays, Dopplers, etc. The main difficulty, however, comes from the assignment of a given measurement to a target model. These assignments are generally unknown, as are the true target models. This is a neat departure from classical estimation problems. Thus, two distinct problems have to be solved *jointly*: the data association and the estimation.

The simplest approach is probably the nearest neighbor approach. Using only the observation the closest to the predicted state, this algorithm is not robust enough in many situations. As long as the association is considered in a deterministic way, the possible associations must be exhaustively enumerated. This leads to an NP-hard problem because the number of possible associations increases exponentially with time, as in the multiple hypotheses tracker (MHT) [1]. To cope with this problem, pruning and gating eliminate the less likely hypotheses but can unfortunately eliminate good ones as well. In the joint probabilistic data association filter (JPDAF) [2], the association variables are considered as stochastic variables and one needs only to evaluate the validated association probabilities at each time step. However, the dependence assumption on the associations implies the exhaustive enumeration of all possible associations at the current time step. When the association variables are instead supposed statistically independent like in the probabilistic MHT (PMHT [3, 4]), the complexity is reduced. For instance in [3, 4], the algorithm is presented as an incomplete data problem solved by an EM algorithm. There is no measurement gating as in the JPDAF and all the associations are considered. The results are then satisfactory when the measurement equation is linear and when the trajectories are deterministic. In [5] the algorithm is extended to the tracking of maneuvering targets with a hidden "model-switch" process controlled by a Markov probability structure. A comparison of the PMHT with the JPDAF is described in a practical two-target scenario in [6], focusing on the mean-square estimation errors and the percentage of lost tracks. Unfortunately, the above algorithms do not cope with nonlinear models and non-Gaussian noises.

Under such assumptions (stochastic state equation and nonlinear state or measurement equation, non-Gaussian noises), sequential Monte Carlo methods, also called particle filtering methods, are particularly appealing. They mainly consist of propagating, in a possibly nonlinear way, a weighted set of particles which approximates the probability density of the state conditioned on the observations according to Monte Carlo integration principles. The weights of the particles are updated using Bayes's formula. Particle filtering can be applied under very general hypotheses, is able to cope with heavy clutter, and is very easy to implement. Such filters have been used in very different areas for Bayesian filtering, under different names: the bootstrap filter for target tracking in [7] and the Condensation algorithm in computer vision [8] are two examples among others. In earliest studies, the algorithm was only composed of two periods: the particles were predicted according to the state equation during the prediction step; then their weights were calculated with the likelihood of the new observation combined with the former weights. A resampling step has rapidly been added to dismiss the particles with lower weights and avoid the degeneracy of the particle set into a unique particle of high weight [7]. Many ways have been developed to accomplish this resampling whose final goal is to enforce particles in areas of high likelihood. The frequency of this resampling has also been studied. Also the use of kernel filters [9] has been introduced to regularize the sum of Dirac densities associated to the particles when the dynamic noise of the state equation was too low [10]. Despite this long history of studies, in which the ability of particle filter to track multiple posterior modes is claimed, the extension of the particle filter to multiple target tracking has progressively received attention only in the last five years. Such extensions have first been claimed theoretically feasible in [11, 12] but the examples chosen only dealt with one single target. In computer vision a probabilistic exclusion principle has been developed in [13] to track multiple objects but the algorithm is very dependent of the observation model and is only applied for two objects. In the same context, a Bayesian multiple-blob tracker called BraMBLe [14] has just been proposed. It deals with a varying number of objects which are depth-ordered thanks to a 3-D state space. Lately, in mobile robotic [15], a set of particle filters for each target connected by a statistical data association has been proposed. We propose here a general algorithm for MTT in the passive sonar context.

This work is organized as follows. In Section II, we describe the basic particle filter for a single target with two versions for the resampling step, which can be systematic or adaptive. The superiority of adaptive resampling towards systematic resampling is clearly demonstrated in a passive sonar application in

Section III. Section IV, the central part of this work, deals with an extension of the basic filter to multiple objects. The new algorithm combines the two major steps of prediction and weighting of the classical particle filter with a Gibbs sampler computing an estimation of the vector of the assignment probabilities. We introduce two different versions of this Gibbs sampler which take into account the past information in a different way. An extension to multireceiver data in the context of multiple targets ends this section and highlights the versatility of our approach. Finally, Section V is devoted to an application to bearings-only MTT which enables us to compare the results obtained with these two versions. In each case, the data association problem is overcome and the first version is clearly superior.

As far as the notational conventions are concerned, we always use the index i to refer to one among the M tracked objects. The index j designates one of the m_t observations obtained at instant t . The index n is reserved for the N particles denoted by s . The index τ is used for indexing the iterations in the Gibbs Sampler and r is used for the different receivers. Finally, the probability densities are denoted by p if they are continuous and by \mathbb{P} if they are discrete.

II. THE BASIC PARTICLE FILTER

For the sake of completeness, the basic particle filter is now briefly reviewed. The general principle of sampling it relies on is used throughout the paper. We consider a dynamic system represented by the stochastic process $(X_t) \in \mathbb{R}^{n_x}$ whose temporal evolution is given by the state equation:

$$X_t = F_t(X_{t-1}, V_t). \quad (1)$$

We want to estimate the state vector (X_t) at discrete times with the help of system's observations which are realizations of the stochastic process $(Y_t) \in \mathbb{R}^{n_y}$ governed by the measurement equation:

$$Y_t = H_t(X_t, W_t). \quad (2)$$

The two processes $(V_t) \in \mathbb{R}^{n_v}$ and $(W_t) \in \mathbb{R}^{n_w}$ are only supposed to be independent white noises. Note that the functions F_t and H_t are not assumed linear.

We will denote by $Y_{0:t}$ the sequence of the random variables (Y_0, \dots, Y_t) and by $y_{0:t}$ one realization of this sequence. Note that throughout the paper, the first subscript of any vector always refers to the time.

Our problem consists in computing at each time t the conditional density $L_t = p(X_t | Y_0 = y_0, \dots, Y_t = y_t)$ of the state X_t given all the observations accumulated up to t , and also in estimating any functional $g(X_t)$ of the state via the expectation $\mathbb{E}(g(X_t) | Y_{0:t})$. The recursive Bayesian filter, also named optimal filter, resolves exactly this problem in two steps at each time t .

- Initialization: $\begin{cases} s_0^n \sim p(X_0) \\ q_0^n = 1/N \end{cases} \quad n = 1, \dots, N.$
- For $t = 1, \dots, T$:
 - Proposal: sample s_t^n from $f(X_t | X_{t-1} = s_{t-1}^n, Y_t = y_t)$ for $n = 1, \dots, N$.
 - Weighting: $\begin{cases} \text{Compute un-normalized weights: } \tilde{q}_t^n = q_{t-1}^n \frac{p(s_t^n | s_{t-1}^n) l_t(y_t; s_t^n)}{f(s_t^n | s_{t-1}^n, y_t)} \text{ for } n = 1, \dots, N. \\ \text{Normalize weights: } q_t^n = \frac{\tilde{q}_t^n}{\sum_{n=1}^N \tilde{q}_t^n} \text{ for } n = 1, \dots, N. \end{cases}$
 - Return $\hat{\mathbb{E}}g(X_t) = \sum_{n=1}^N q_t^n g(s_t^n)$.

Fig. 1. Basic particle filter without resampling.

Suppose we know L_{t-1} . The *prediction step* is done according to the following equation:

$$\begin{aligned} p(X_t = x_t | Y_{0:t-1} = y_{0:t-1}) \\ = \int_{\mathbb{R}^{n_x}} p(X_t = x_t | X_{t-1} = x) L_{t-1}(x) dx. \end{aligned} \quad (3)$$

Using (1), we can calculate $p(X_t = x_t | X_{t-1} = x)$:

$$\begin{aligned} p(X_t = x_t | X_{t-1} = x) = \int_{\mathbb{R}^{n_v}} p(X_t = x_t | X_{t-1} = x, V_t = v) \\ \times p(V_t = v | X_{t-1} = x) dv. \end{aligned} \quad (4)$$

The observation y_t enables us to correct this prediction using Bayes's rule:

$$L_t(x_t) = \frac{p(Y_t = y_t | X_t = x_t) p(X_t = x_t | Y_{0:t-1} = y_{0:t-1})}{\int_{\mathbb{R}^{n_x}} p(Y_t = y_t | X_t = x) p(X_t = x | Y_{0:t-1} = y_{0:t-1}) dx}. \quad (6)$$

Under the specific assumptions of Gaussian noises V_t and W_t and linear functions F_t and H_t , these equations lead to the Kalman filter's equations. Unfortunately this modeling is not appropriate in many problems in signal and image processing, which makes the calculation of the integrals in (3) and (6) infeasible (no closed-form).

The original particle filter, which is called the bootstrap filter [7], proposes to approximate the densities $(L_t)_t$ by a finite weighted sum of N Dirac densities centered on elements of \mathbb{R}^{n_x} , which are called particles. The application of the bootstrap filter requires that one knows how to do the following:

- sample from initial prior marginal $p(X_0)$;
- sample from $p(V_t)$ for all t ;
- compute $p(Y_t = y_t | X_t = x_t)$ for all t through a known function l_t such that $l_t(y; x) \propto p(Y_t = y | X_t = x)$ where missing normalization must not depend on x .

The first particle set S_0 is created by drawing N independent realizations from $p(X_0)$ and assigning uniform weight $1/N$ to each of them. Then, suppose we have at our disposal at time $t-1$ the weighted particle set $S_{t-1} = (s_{t-1}^n, q_{t-1}^n)_{n=1, \dots, N}$ where $\sum_{n=1}^N q_{t-1}^n$

$= 1$. The *a posteriori* marginal L_{t-1} is then estimated by the probability density $L_{S_{t-1}} = \sum_{n=1}^N q_{t-1}^n \delta_{s_{t-1}^n}$. The *prediction step* consists of propagating each particle of S_{t-1} according to the evolution equation (1).

The weight of each particle is updated during the *correction step*. Up to a constant, (6) comes down to adjust the weight of predictions by multiplying it by the likelihood $p(y_t | x_t)$. In the most general setting of sequential Monte Carlo methods [16], the displacement of particles is obtained by sampling from an appropriate density f which might depend on the data as well. The general algorithm is summarized in Fig. 1. The density L_{S_t} is often multimodal as several hypotheses about the position of the object can be made at one time. It is for instance the case when one object is tracked in the presence of significant clutter. Several hypotheses about the object position can then be kept if the set of particles splits into several subsets. This is where the great strength of this filter lies. In [17] for instance, the measurement vector Y_t consists of a set of detected features along line measurements. The assumed underlying generative model affects each feature either to the target boundary, or to its interior or to the background. The likelihood function is built from this generative model and then takes into account the clutter model. An extension of the algorithm in Fig. 1, called the hybrid bootstrap filter, has also been proposed to deal with significant clutter and spurious objects for target tracking [12] and guidance [18]. The weighted sum of Dirac laws is then approximated by a Gaussian mixture obtained by a clustering method [19].

The particle sets enable one to estimate any functional of X_t in particular the two first moments with $g(x) = x$ and $g(x) = x^2$, respectively. The mean can be used to estimate the position of one object but it can be a bad estimator if the posterior is highly multimodal. In such cases the ideal would be to calculate the mean only over the particles that contribute to the principal mode but such an estimator has not been developed for the moment.

In practice, the number of particles is finite and the major drawback of this algorithm is the degeneracy of

- Initialization: $\begin{cases} s_0^n \sim p(X_0) \\ q_0^n = 1/N \end{cases} \quad n = 1, \dots, N.$
- For $t = 1, \dots, T$:
 - Proposal: sample \tilde{s}_t^n from $f(X_t|X_{t-1} = s_{t-1}^n, Y_t = y_t)$ for $n = 1, \dots, N.$
 - Weighting: $\begin{cases} \text{Compute un-normalized weights: } \tilde{q}_t^n = q_{t-1}^n \frac{p(\tilde{s}_t^n | s_{t-1}^n) l_t(y_t; \tilde{s}_t^n)}{f(\tilde{s}_t^n | s_{t-1}^n, y_t)} \text{ for } n = 1, \dots, N. \\ \text{Normalize weights: } q_t^n = \frac{\tilde{q}_t^n}{\sum_{n=1}^N \tilde{q}_t^n} \text{ for } n = 1, \dots, N. \end{cases}$
 - Return $\hat{\mathbb{E}}g(X_t) = \sum_{n=1}^N q_t^n g(\tilde{s}_t^n).$
 - Resampling: $\begin{cases} \text{Draw } s_t^n \text{ from } \sum_{k=1}^N q_t^k \delta_{\tilde{s}_t^k} \\ \text{Set } q_t^n = 1/N \end{cases} \quad n = 1, \dots, N$

Fig. 2. Basic particle filter with systematic resampling.

- Initialization: $\begin{cases} s_0^n \sim p(X_0) \\ q_0^n = 1/N \end{cases} \quad n = 1, \dots, N.$
- For $t = 1, \dots, T$:
 - Proposal: sample \tilde{s}_t^n from $f(X_t|X_{t-1} = s_{t-1}^n, Y_t = y_t)$ for $n = 1, \dots, N.$
 - Weighting: $\begin{cases} \text{Compute un-normalized weights: } \tilde{q}_t^n = q_{t-1}^n \frac{p(\tilde{s}_t^n | s_{t-1}^n) l_t(y_t; \tilde{s}_t^n)}{f(\tilde{s}_t^n | s_{t-1}^n, y_t)} \text{ for } n = 1, \dots, N. \\ \text{Normalize weights: } q_t^n = \frac{\tilde{q}_t^n}{\sum_{n=1}^N \tilde{q}_t^n} \text{ for } n = 1, \dots, N. \end{cases}$
 - Return $\hat{\mathbb{E}}g(X_t) = \sum_{n=1}^N q_t^n g(\tilde{s}_t^n).$
 - Calculate $\hat{N}_{eff} = \frac{1}{\sum_{n=1}^N (q_t^n)^2}.$
 - Resampling: if $\hat{N}_{eff} < N_{threshold}$: $\begin{cases} s_t^n \sim \sum_{k=1}^N q_t^k \delta_{\tilde{s}_t^k} \\ q_t^n = 1/N \end{cases} \quad n = 1, \dots, N, \text{ else } s_t^n = \tilde{s}_t^n \text{ for } n = 1, \dots, N.$

Fig. 3. Basic particle filter with adaptive resampling.

the particle set: only few particles keep high weights and the others have very small ones. The former carry the information, whereas the latter are mostly useless. The resampling is a good way to remedy this drawback because it eliminates the particles of smallest weight. The stochastic resampling consists of sampling N particles with replacement in the particle set with the probability q^n to draw s^n . The new particles have uniform weights equal to $1/N$. A first solution, adopted in [7] for example, consists of applying the resampling step at each time period. The corresponding algorithm of particle filter with systematic resampling is described in Fig. 2.

To measure the degeneracy of the algorithm, the effective sample size N_{eff} has been introduced in [20, 21]. We can estimate this quantity by $\hat{N}_{\text{eff}} = 1/\sum_{n=1}^N (q_t^n)^2$ which measures the number of meaningful particles. As advocated in [16], the resampling can be done only if $\hat{N}_{\text{eff}} < N_{\text{threshold}}$. This enables the particle set to better learn the process and to keep its memory during the interval where no resampling occurs. The algorithm of the basic particle filter with adaptive resampling is

described in Fig. 3. Details can be found in [16, 20, 21].

Some convergence results of the empirical distributions to the posterior distribution on the path space have been proved when the number N of particles tends towards infinity [22, 23]. In the path space $(\mathbb{R}^{n_x})^{t+1}$, each particle s_t^n at time t can be considered to be a discrete path of length $t + 1$. Compared with the particle filter presented in Fig. 1, particle filtering in the space of paths consists of incrementing the particle state space at each time step and representing each particle by the concatenation of the new position at time t and the set of previous positions between times 0 and $t - 1$. In [22], the fluctuations on path space of the so-called interacting particle systems are studied. In the context of sequential Monte Carlo methods [23], which cover most of the particle filtering methods proposed in the last years, the convergence and the rate of convergence of order $1/N$ of the average mean square error is proved. Under more restrictive hypotheses, the almost sure convergence is proved as well [23].

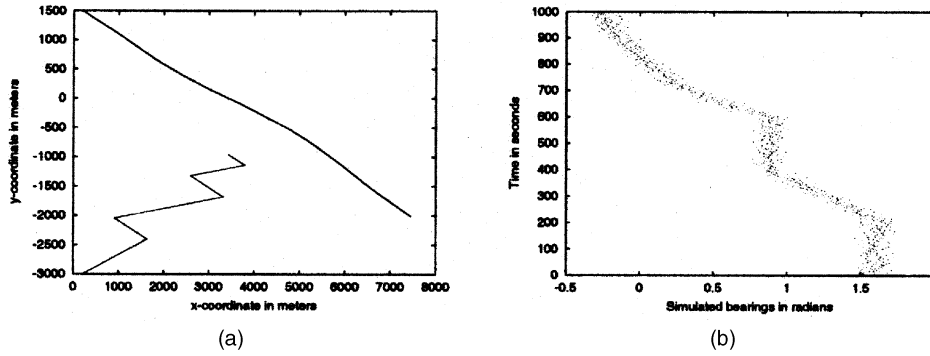


Fig. 4. Single target bearings-only experiment of Section III (a) Target and observer trajectories. (b) Bearings simulated at each time period.

III. APPLICATION TO BEARINGS-ONLY PROBLEMS

To illustrate the previous algorithms, we first deal with classical bearings-only problems. The object is then a “point-object” in the x - y plane. In the context of a slowly maneuvering target, we have chosen a nearly constant-velocity model (see [24] for a review of the principal dynamical models used in this domain).

A. The Model

The state vector X_t represents the coordinates and the velocities in the x - y plane: $X_t = (x_t, y_t, v_x, v_y)$. The discretized state equation associated with time period Δt is

$$X_{t+\Delta t} = \begin{pmatrix} I_{2 \times 2} & \Delta t I_{2 \times 2} \\ 0 & I_{2 \times 2} \end{pmatrix} X_t + \begin{pmatrix} \frac{\Delta t^2}{2} I_{2 \times 2} \\ \Delta t I_{2 \times 2} \end{pmatrix} V_t \quad (7)$$

where $I_{2 \times 2}$ is the identity matrix in dimension 2 and V_t is a Gaussian zero-mean vector of covariance matrix

$$\Sigma_V = \begin{pmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{pmatrix}.$$

Let \hat{X}_t be the estimate of X_t computed by the particle filters with $g(x) = x$, i.e., $\hat{X}_t = \sum_{n=1}^N q_t^n \tilde{s}_t^n$. The posterior covariance matrix of \hat{X}_t is

$$P_t = \sum_{n=1}^N q_t^n (\tilde{s}_t^n - \hat{X}_t)(\tilde{s}_t^n - \hat{X}_t)^T \quad (8)$$

where T denotes the transposition.

The observations are available at discrete times according to

$$Y_t = \arctan \left(\frac{y_t - y_t^{\text{obs}}}{x_t - x_t^{\text{obs}}} \right) + W_t \quad (9)$$

where W_t is a zero-mean Gaussian noise of covariance σ_w^2 independent of V_t , and x_{obs} and y_{obs} are the Cartesian coordinates of the observer, which are known.

B. Results of Particle Filters with Systematic and Adaptive Resampling, Respectively

The initial position of the target and of the observer are

$$X_0 = \begin{pmatrix} 200 \text{ m} \\ 1500 \text{ m} \\ 1.0 \text{ ms}^{-1} \\ -0.5 \text{ ms}^{-1} \end{pmatrix} \quad X_0^{\text{obs}} = \begin{pmatrix} 200 \text{ m} \\ -3000 \text{ m} \\ 1.2 \text{ ms}^{-1} \\ +0.5 \text{ ms}^{-1} \end{pmatrix}.$$

The dynamic noise is a normal zero-mean Gaussian vector with $\sigma_x = \sigma_y = 0.001 \text{ ms}^{-2}$. The observer is following a leg-by-leg trajectory. Its velocity vector is constant on each leg and modified at the following instants:

$$\begin{aligned} \begin{pmatrix} v_x^{\text{obs}}_{200} \\ v_y^{\text{obs}}_{200} \end{pmatrix} &= \begin{pmatrix} -0.6 \\ 0.3 \end{pmatrix} & \begin{pmatrix} v_x^{\text{obs}}_{400} \\ v_y^{\text{obs}}_{400} \end{pmatrix} &= \begin{pmatrix} 2.0 \\ 0.3 \end{pmatrix} \\ \begin{pmatrix} v_x^{\text{obs}}_{600} \\ v_y^{\text{obs}}_{600} \end{pmatrix} &= \begin{pmatrix} -0.6 \\ 0.3 \end{pmatrix} & \begin{pmatrix} v_x^{\text{obs}}_{800} \\ v_y^{\text{obs}}_{800} \end{pmatrix} &= \begin{pmatrix} 2.0 \\ 0.3 \end{pmatrix} \\ \begin{pmatrix} v_x^{\text{obs}}_{900} \\ v_y^{\text{obs}}_{900} \end{pmatrix} &= \begin{pmatrix} -0.6 \\ 0.3 \end{pmatrix}. \end{aligned}$$

The bearings measurements are simulated with a Gaussian noise of standard deviation $\sigma_w = 0.05 \text{ rad}$ (about 3 deg) every time period, i.e., every 6 s. The measurements set used and the trajectories of the observer and of the target are presented in Fig. 4.

First, we have studied the impact of adaptive or systematic resampling on the estimate posterior covariance defined by (8). We have used bootstrap filters, i.e., the importance function f is in fact the prior law $p(X_t | X_{t-1})$. The initialization of the filters has been done in both cases according to a Gaussian law whose mean vector is the true vector X_0 and covariance matrix X_{cov} is

$$X_{\text{cov}} = \begin{pmatrix} 20.0^2 & 0 & 0 & 0 \\ 0 & 100^2 & 0 & 0 \\ 0 & 0 & 0.05^2 & 0 \\ 0 & 0 & 0 & 0.05^2 \end{pmatrix}.$$

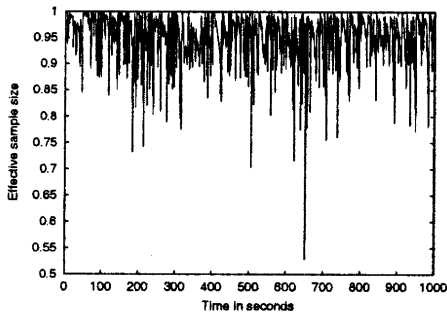


Fig. 5. Values of effective sample size \hat{N}_{eff} obtained with 1000 particles and adaptive resampling.

For the filter with adaptive resampling, the threshold $N_{\text{threshold}}$ is fixed to 0.9. The average frequency of resampling obtained for such a threshold is 0.132. The values of \hat{N}_{eff} are presented in Fig. 5 for one particular run.

The results are plotted in Fig. 6. The observer trajectory, the target trajectory, and the estimate obtained with adaptive and systematic resampling are plotted with the 2σ confidence ellipses on position and the line of sight of the observer every hundred times for 1000, 5000, and 10000 particles. The ellipses represent the regions containing 95% of the particles assuming they are Gaussian distributed. At each time, it is interesting to check that the ellipses are oriented according to the line of sight. Moreover with only 1000 particles, the advantage of using adaptive resampling is emphasized: resampling impoverishes the particle set and when applied systematically, the particle set can become so reduced that the ellipse does not contain the true trajectory any more. It is the case from instant 400 in Fig. 6(e) whereas with adaptive resampling in Fig. 6(f) the confidence ellipses still contain the true trajectory.

The estimates obtained with basic particle filters using adaptive or systematic resampling are optimal for a infinite particle number. In practice, the particle number is of course finite, and the estimation is determined by the realizations of the random processes used in the proposal. To further assess the accuracy of the two filters, we have performed 100 different runs of the particle filters with systematic resampling on the one hand and adaptive resampling on the other hand, for a given realization of the measurement process.¹ From these different runs, we have computed the averaged estimate and the 2σ confidence ellipses on position containing 95% of the estimates. They are represented in Fig. 7

¹It is also interesting to compute the particle filter estimation for different realizations of the measurement process, as it is often done to evaluate the performance of deterministic algorithms such as the Kalman filter or the extended Kalman filter. It is done in Section III E.

TABLE I

Computational Cost for One Iteration on 863 MHz Pentium III

Particle Number	1000	5000	10000
Systematic Resampling	7.9 ms	41.9 ms	88.5 ms
Resampling Procedure	4 ms	28 ms	64 ms

for $N = 1000, 5000, 10000$ particles and adaptive or systematic resampling. The averaged estimates are similar but the confidence ellipses are globally twice as small with adaptive than with systematic resampling whatever the particle number. This indicates that adaptive resampling significantly reduces the standard deviation on the particle filter estimates.

As for the computational cost, Table I contains the cost for one time iteration with systematic resampling and the time spent on the resampling procedure according to the particle number on a 863 Mhz Pentium III.

C. Overestimated State Noise

It is to be noted that in the case of real data, the variance of the state noise is not always known precisely. Some simulations have then been performed with an overestimated state noise. Intuitively, the particles are thus predicted in a larger area but are also more prone to be eliminated in the resampling step. In practice this is a great strength of particle filtering. Fig. 8 shows the averaged estimates and the 2σ confidence ellipses obtained with adaptive resampling, and with 1000 and 5000 particles. The estimation is less smooth than with a correct state noise. This jittering increases the bias and the standard deviation and more particles are needed for results comparable with those obtained with a correct state noise: with 5000 particles the standard deviation between estimates is drastically lower than with 1000 particles.

D. Shifted Initialization

We have also performed particle filters runs with a shifted initialization: the initialization is done according to a Gaussian law whose mean vector X_{mean} and covariance matrix X_{cov} are

$$X_{\text{mean}} = X_0 + \begin{pmatrix} 50.0 \text{ m} \\ 500 \text{ m} \\ 0.0 \text{ ms}^{-1} \\ -0.0 \text{ ms}^{-1} \end{pmatrix} \quad (10)$$

$$X_{\text{cov}} = \begin{pmatrix} 40.0^2 & 0 & 0 & 0 \\ 0 & 400.0^2 & 0 & 0 \\ 0 & 0 & 0.05^2 & 0 \\ 0 & 0 & 0 & 0.05^2 \end{pmatrix}.$$

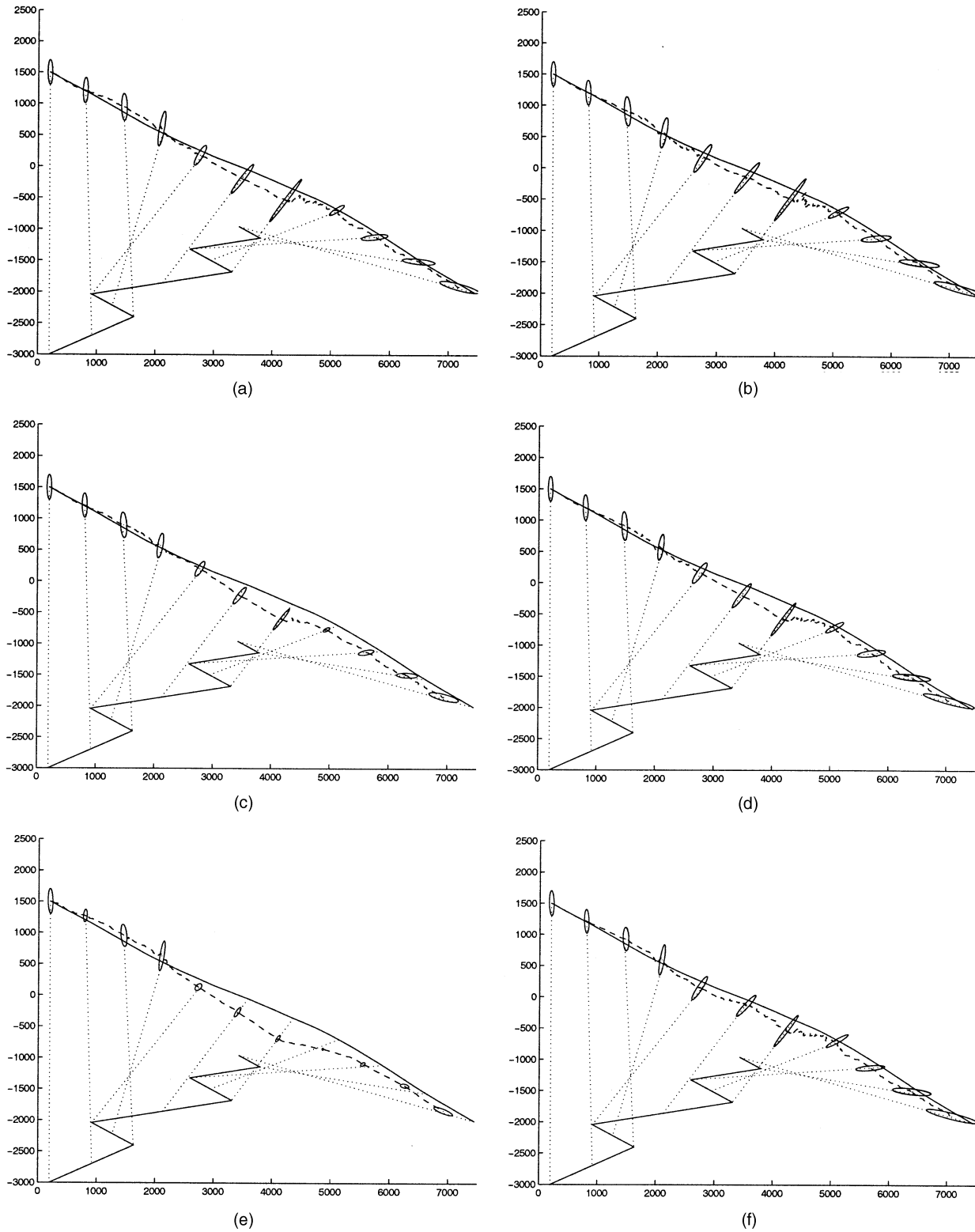


Fig. 6. Estimates for one particular run (dashed lines) and 2σ confidence ellipses obtained with bootstrap filter. Solid lines stand for target and observer trajectories. Dotted lines indicate line of sight of observer every hundred times. Top: 10000 particles, middle: 5000 particles, bottom: 1000 particles. Left column: systematic resampling; right column: adaptive resampling.

After around 200 time periods, the particles have recovered from their shifted initialization and provide satisfactory estimates as shown in

Fig. 9. The ellipses size is considerably reduced with 5000 particles compared to only 1000 particles.

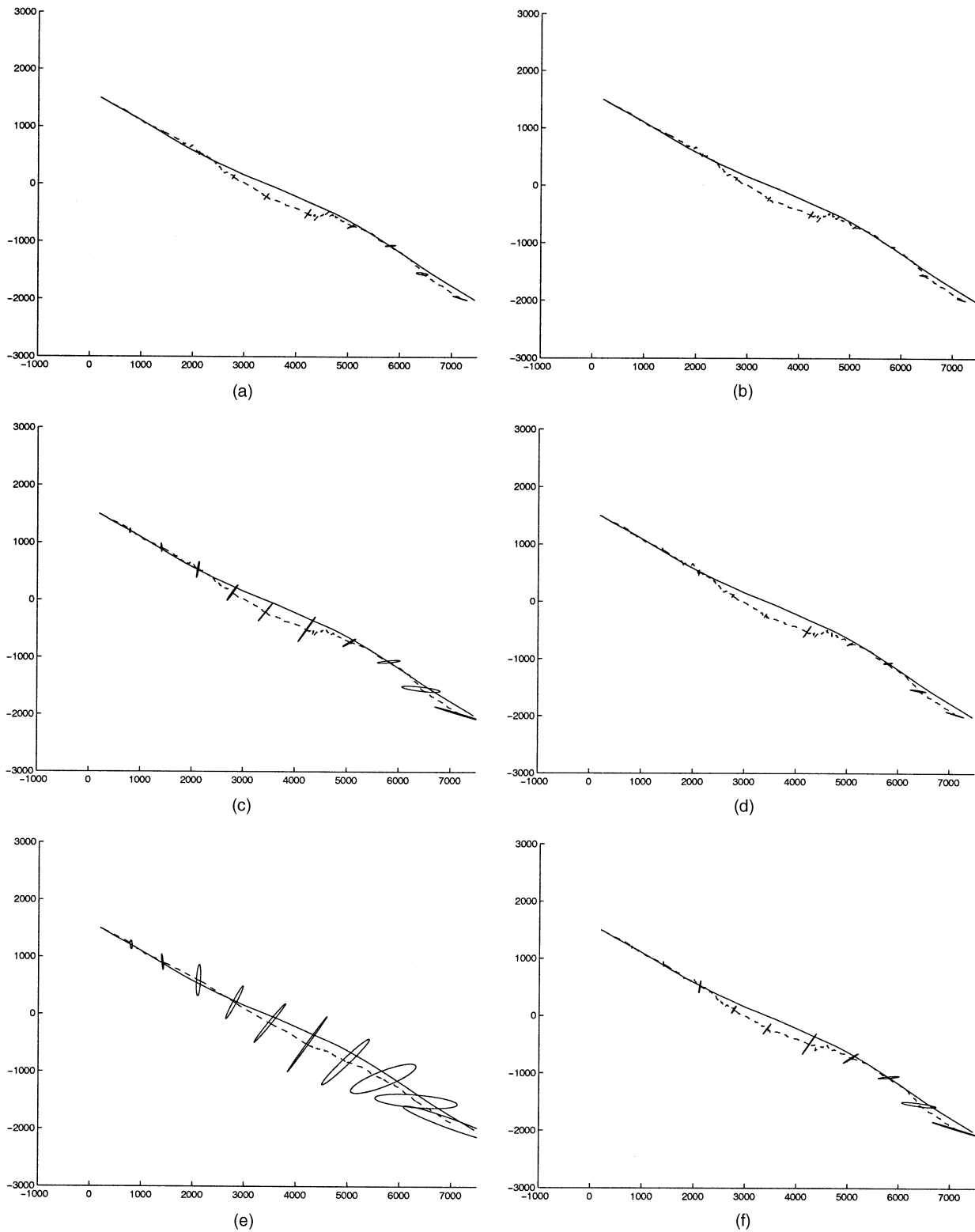


Fig. 7. Averaged estimates (dashed lines) and 2σ confidence ellipses obtained with bootstrap filter. Top: 10000 particles, middle: 5000 particles, bottom: 1000 particles. Left column: systematic resampling; right column: adaptive resampling.

E. Comparison with a Modified Polar Coordinate Extended Kalman Filter

We have finally compared particle filtering methods with a linearized filter, the modified

polar coordinate extended Kalman filter (MP-EKF). Let us denote by X_t^C , $X_t^{C,Rel}$ and X_t^{MP} the state vector in the fixed Cartesian system, in the Cartesian system relative to the observer, and in the modified polar system, respectively

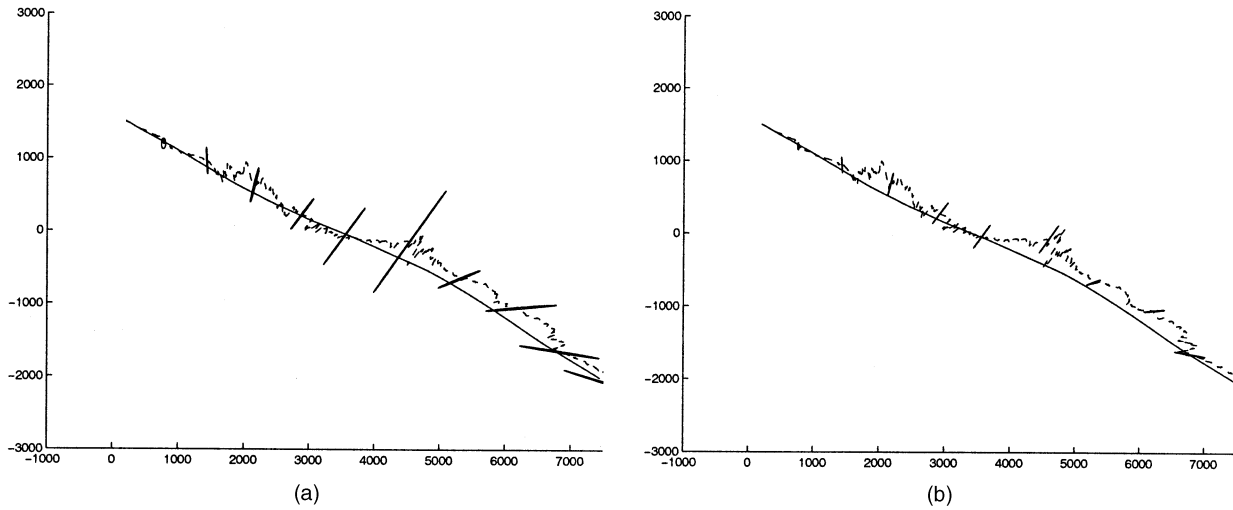


Fig. 8. Averaged estimates (dashed lines) and 2σ confidence ellipses obtained with bootstrap filter with an overestimated state noise. (a) 1000 particles. (b) 5000 particles.

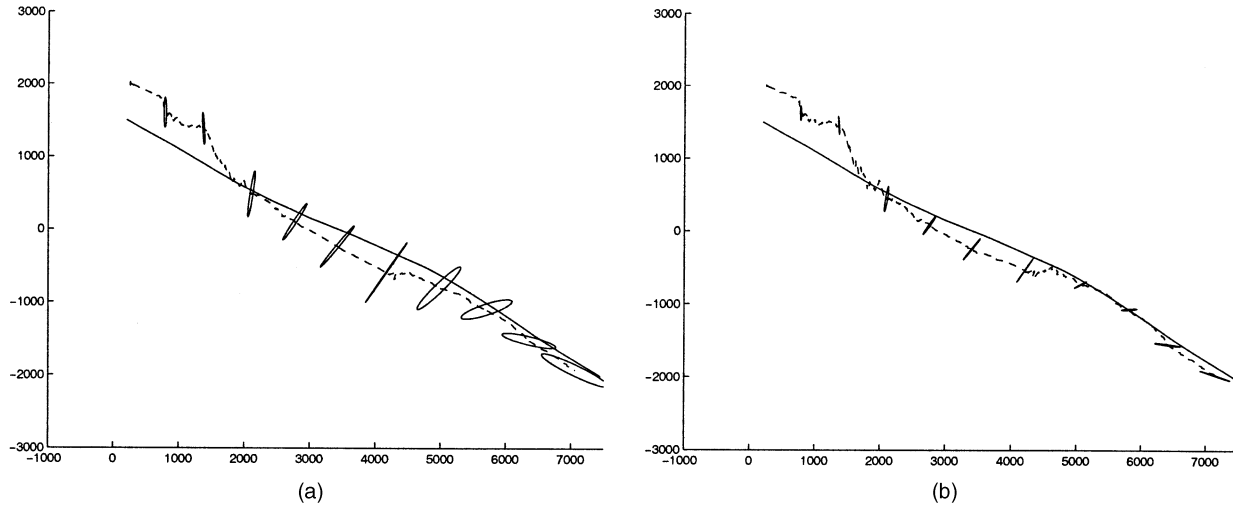


Fig. 9. Averaged estimates (dashed lines) and 2σ confidence ellipses obtained with bootstrap filter with shifted initialization. (a) 1000 particles. (b) 5000 particles.

$$X_t^C = (x_t, y_t, \dot{x}_t, \dot{y}_t),$$

$$X_t^{C,Rel} = (x_t - x_t^{obs}, y_t - y_t^{obs}, \dot{x}_t - \dot{x}_t^{obs}, \dot{y}_t - \dot{y}_t^{obs})$$

and

$$\begin{aligned} X_t^{MP} &= \left(\beta, \frac{1}{R}, \dot{\beta}, \frac{\dot{R}}{R} \right)^T \\ &= \left(\arctan \left(\frac{x_t^{C,Rel}}{y_t^{C,Rel}} \right), \frac{1}{\sqrt{x_t^{C,Rel 2} + y_t^{C,Rel 2}}}, \right. \\ &\quad \left. \frac{y_t^{C,Rel} \dot{x}_t^{C,Rel} - x_t^{C,Rel} \dot{y}_t^{C,Rel}}{x_t^{C,Rel 2} + y_t^{C,Rel 2}}, \right. \\ &\quad \left. \frac{x_t^{C,Rel} \dot{x}_t^{C,Rel} + y_t^{C,Rel} \dot{y}_t^{C,Rel}}{x_t^{C,Rel 2} + y_t^{C,Rel 2}} \right)^T. \end{aligned}$$

Let f_C^{MP} and f_{MP}^C be the functions such that:

$$X_t^{MP} = f_C^{MP}(X_t^{C,Rel}) \quad \text{and} \quad X_t^{C,Rel} = f_{MP}^C(X_t^{MP}). \quad (11)$$

It must be noted that these two functions are nonlinear. The dynamic model in the Cartesian system is defined by

$$X_{t+\Delta_t}^C = F X_t^C + V_t \quad (12)$$

where

$$F = \begin{pmatrix} I_{2 \times 2} & \Delta t I_{2 \times 2} \\ 0 & I_{2 \times 2} \end{pmatrix}$$

and V_t is a Gaussian noise of constant covariance matrix Σ_V . In the MP system it reads:

$$X_{t+\Delta_t}^{MP} = f_C^{MP}(F f_{MP}^C(X_t^{MP}) - U_{t+\Delta_t}^{obs} + V_t) \quad (13)$$

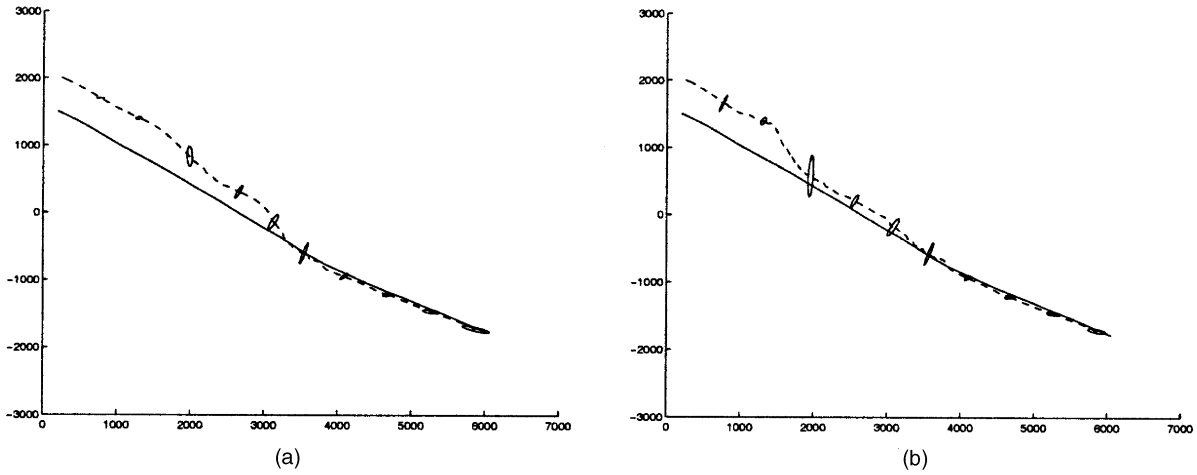


Fig. 10. Mean estimates (dotted lines) obtained over 10 realizations of measurement process and 2σ confidence ellipses with shifted initialization. (a) Estimates obtained with MP-EKF. (b) Estimates obtained with particle filters with adaptive resampling and 5000 particles.

where $U_{t+\Delta_t}^{\text{obs}} = (x_{t+\Delta_t}^{\text{obs}} - x_t^{\text{obs}} - \Delta_t \dot{x}_t^{\text{obs}}, y_{t+\Delta_t}^{\text{obs}} - y_t^{\text{obs}} - \Delta_t \dot{y}_t^{\text{obs}}, \dot{x}_{t+\Delta_t}^{\text{obs}} - \dot{x}_t^{\text{obs}}, \dot{y}_{t+\Delta_t}^{\text{obs}} - \dot{y}_t^{\text{obs}})^T$. If the observer follows a leg-by-leg trajectory, the two first components of U_t^{obs} are always null. The two last components are null except if t is a maneuver time.

In the Cartesian system, the measurement equation is defined by

$$Y_t = HX_t^C + W_t \quad (14)$$

where W_t is a Gaussian noise of covariance σ_w^2 . In the MP system, it reads

$$Y_t = QX_t^{\text{MP}} + W_t \quad (15)$$

where $Q = (1 \ 0 \ 0 \ 0)$.

Let us now describe the propagation and updating steps of the MP-EKF used. The predicted state is given by

$$\tilde{X}_{t+\Delta_t}^{\text{MP}} = f_C^{\text{MP}}(F f_{\text{MP}}^C(\hat{X}_t^{\text{MP}}) - U_t^{\text{obs}}). \quad (16)$$

The predicted covariance matrix in the Cartesian system is

$$\tilde{P}_{t+\Delta_t}^C = F \hat{P}_t^C F^T + \Sigma_V \quad (17)$$

and in the MP system it is

$$\tilde{P}_{t+\Delta_t}^{\text{MP}} = J_C^{\text{MP}} \tilde{P}_t^C (J_C^{\text{MP}})^T \quad (18)$$

where J_C^{MP} is the Jacobian matrix of f_C^{MP} taken around the predicted state.

As the measurement equation is linear in the MP system, the updating step is the same as in the classical Kalman filter:

$$K = \tilde{P}_{t+\Delta_t}^{\text{MP}} Q^T (Q \tilde{P}_{t+\Delta_t}^{\text{MP}} Q^T + \sigma_w^2)^{-1} \quad (19)$$

$$\hat{X}_{t+\Delta_t}^{\text{MP}} = \tilde{X}_{t+\Delta_t}^{\text{MP}} + K(Y_{t+\Delta_t} - Q \tilde{X}_{t+\Delta_t}^{\text{MP}}) \quad (20)$$

$$\begin{aligned} \hat{P}_{t+\Delta_t}^{\text{MP}} &= (I - KQ) \tilde{P}_{t+\Delta_t}^{\text{MP}} (I - KQ)^T + K \sigma_w^2 K^T \\ &= \tilde{P}_{t+\Delta_t}^{\text{MP}} - KQ \tilde{P}_{t+\Delta_t}^{\text{MP}} \end{aligned} \quad (21)$$

$$\hat{P}_{t+\Delta_t}^C = J_{\text{MP}}^C \hat{P}_{t+\Delta_t}^{\text{MP}} (J_{\text{MP}}^C)^T \quad (22)$$

where J_{MP}^C is the Jacobian matrix of f_{MP}^C taken around the estimated state.

More details on the MP-EKF can be found in [25–27]. As the MP-EKF is a deterministic algorithm, we have computed the estimates obtained for several realizations of the measurement process. For a given realization, the MP-EKF has to be compared with the mean of the particle filter estimates over several runs. We have compared the performance in the case of a shifted initialization like in (10) in the previous section over 10 realizations of the measurement process. For each of them, 20 runs of the particle filter with adaptive resampling and 5000 particles are computed. The final bias and standard deviation are similar but the particle filter recovers faster from the shifted initialization than the MP-EKF (see Fig. 10).

IV. MULTITARGET PARTICLE FILTERS

The purpose of the two precedent sections was to provide a general representation of particle filtering. We can now turn to its extension to MTT.

A. MTT Problem and Its Classical Treatment

Let M be the number of targets to track. This number is assumed to be known and fixed for the moment (the case of a varying unknown number will be addressed in another work). The index i designates one among the M targets and is always used as first superscript. Multitarget tracking consists in estimating the state vector made by concatenating the state vectors of all targets. It is generally assumed that the targets are moving according to independent Markovian dynamics. At time t , $X_t = (X_t^1, \dots, X_t^M)$ follows the state equation (1) decomposed in M partial equations:

$$X_t^i = F_t^i(X_{t-1}^i, V_t^i) \quad \forall \quad i = 1, \dots, M. \quad (23)$$

The noises (V_t^i) and $(V_t^{i'})$ are supposed only to be white both temporally and spatially, and independent for $i \neq i'$.

The observation vector collected at time t is denoted by $y_t = (y_t^1, \dots, y_t^{m_t})$. The index j is used as first superscript to refer to one of the m_t measurements. The vector y_t is composed of detection measurements and clutter measurements. The false alarms are assumed to be uniformly distributed in the observation area. Their number is assumed to arise from a Poisson density of parameter λV where V is the volume of the observation area and λ the number of false alarms per unit volume. As we do not know the origin of each measurement, one has to introduce the vector K_t to describe the associations between the measurements and the targets. Each component K_t^j is a random variable that takes its values among $\{0, \dots, M\}$. Thus, $K_t^j = i$ indicates that y_t^j is associated with the i th target. In this case, y_t^j is a realization of the stochastic process:

$$Y_t^j = H_t^i(X_t^i, W_t^j) \quad \text{if } K_t^j = i. \quad (24)$$

Again, the noises (W_t^j) and $(W_t^{j'})$ are supposed only to be white noises, independent for $j \neq j'$. We assume that the functions H_t^i are such that they can be associated to functional forms l_t^i such that $l_t^i(y; x) \propto p(Y_t^j = y \mid K_t^j = i, X_t^i = x)$.

We dedicate the model 0 to false alarms. Thus, if $K_t^j = 0$, the j th measurement is associated to the clutter, but we do not associate any kinematic model to false alarms.

As the indexing of the measurements is arbitrary, all the measurements have the same *a priori* probability to be associated with a given model i . At time t , these association probabilities define the vector $\pi_t = (\pi_t^0, \pi_t^1, \dots, \pi_t^M) \in [0, 1]^{M+1}$. Thus, for $i = 1, \dots, M$, $\pi_t^i \triangleq \mathbb{P}(K_t^j = i)$ for all $j = 1, \dots, m_t$ is the discrete probability that any measurement is associated with the i th target.

To solve the data association some assumptions are commonly made [28].

A1. One measurement can originate from one target or from the clutter.

A2. One target can produce zero or one measurement at one time.

The assumption (A1) expresses that the association is exclusive and exhaustive. Consequently, $\sum_{i=0}^M \pi_t^i = 1$. The assumption (A2) implies that m_t may differ from M and above all that the association variables K_t^j for $j = 1, \dots, m_t$ are dependent.

Under these assumptions, the MHT algorithm [1] builds recursively the association hypotheses. One advantage of this algorithm is that the appearance of a new target is hypothesized at each time step. However, the complexity of the algorithm increases

exponentially with time. Some pruning solutions must be found to eliminate some of the associations.

The JPDAF begins with a gating of the measurements. Only the measurements which are inside an ellipse around the predicted state are kept. The gating assumes that the measurements are distributed according to a Gaussian law centred on the predicted state. Then, the probabilities of each association $K_t^j = i$ are estimated. As the variables K_t^j are assumed dependent by (A2), this computation requires the exhaustive enumeration of all the possible associations K_t^l for $l \neq j$.

The novelty in the PMHT algorithm [3–5] consists of replacing the assumption (A2) by (A3).

A3. One target can produce zero or several measurements at one time.

This assumption is often criticized because it does not match the physical reality. However, from a mathematical point of view it ensures the stochastic independence of the variables K_t^j and it drastically reduces the complexity of the π_t vector estimation.

The assumptions (A1) and (A3) are kept in the joint filters presented later. Let us present now the existing works solving MTT with particle filtering methods.

B. Related Work: MTT with Particle Filters

In the context of MTT, particles filters are appealing: as the association needs only to be considered at a given time iteration, the complexity of data association is reduced. First, two extensions of the bootstrap filter have been considered. In [11], a bootstrap-type algorithm is proposed in which the sample state space is a “(multitarget) state space.” However, nothing is said about the association problem that needs to be solved to evaluate the sample weights. It is in fact the ability of the particle filtering to deal with multimodality due to (high) clutter that is pointed out compared with deterministic algorithms like the nearest neighbor filter or the PDA filter. No examples with multiple targets are presented: the simulations only deal with a single target in clutter with a linear observation model. In [12], a hybrid bootstrap filter is presented where the particles evolve in a single-object state space. Each particle gives a hypothesis on the state of one object. Thus, the *a posteriori* law of the targets given the measurements is represented by a Gaussian mixture. Each mode of this law then corresponds to one of the objects. However, as pointed out in [12], the likelihood evaluation is possible only under the availability of the “prior probabilities of all possible associations between” the measurements and the targets. Even if the likelihood could be evaluated, the way to represent the *a posteriori* law by a mixture can lead to the loss

of one of the targets during occlusions. The particles tracking an occluded target get very small weights and are therefore discarded during the resampling step. This fact has been pointed out in [15].

In image analysis, the condensation algorithm has been extended to the case of multiple objects as well: in [13], the case of two objects is considered. The hidden state is the concatenation of the two single-object states and of a binary variable indicating which object is closer to the camera. This latter variable solves the association during occlusion because the measurements are affected to the foreground object. Moreover, a probabilistic exclusion principle is integrated to the likelihood measurement to penalize the hypotheses with the two objects overlapping. In [14], the state is composed of an integer equal to the number of objects and of a concatenation of the individual states. A 3-D representation of the objects gives access to their depth ordering, thus solving the association issue during occlusions. Three other works combine particle filtering with the JPDA. In mobile robotic [15], a particle filter is used for each object tracked. The likelihood of the measurements is written like in a JPDAF. Thus, the assignment probabilities are evaluated according to the probabilities of each possible association. Given these assignment probabilities, the particle weights can be evaluated. The particle filters are then dependent through the evaluation of the assignment probabilities. The algorithms presented in [29, 30] are both applied to target tracking. The state space of each particle is the concatenation of the state space of each target and the likelihood of the measurements given a particle is derived according to the JPDAF. Independently of the latter works [14, 15, 29, 30], we have developed the joint filters where the data association is approached in the same probabilistic spirit as the basic PMHT [3, 4].

First, to estimate the density $L_t = p(X_t = (X_t^1, \dots, X_t^M) | Y_{0:t} = y_{0:t})$ with particle filtering methods we must choose the state space for the particles. We could use particles of X_t^i 's dimension and distribute them among the objects (providing that the objects have the same dimension which might not be the case for complex objects in image processing). Each of them would then be known to represent a given object. The weight of particle n is then:

$$q_t^n \propto p(Y_t = (y_t^1, \dots, y_t^{m_t}) | X_t^i = s_t^n). \quad (25)$$

As the observations cannot be supposed independent conditioned on only one of the M objects, the likelihood cannot be decomposed in the product of likelihoods of each observation. One solution could be to use Bayes' rule and to compute $p(X_t^i = s_t^n | Y_t = (y_t^1, \dots, y_t^{m_t}))$, but this requires that every object

produces at least one measurement (to be able to use the Total Probability Theorem), which seems too restrictive. Moreover, as mentioned before, a unique particle filter with a single-target state space seems inappropriate as the particles tracking an occluded object would be quickly discarded. We have considered using one particle filter per object but without finding a consistent way to make them dependent. The stochastic association vector K_t introduced in Section IVA could also be considered as an additional particle component. However, as the ordering of the measurements is arbitrary, it would not be possible to devise a dynamic prior on it. Moreover, the state space would increase further making particle filter less effective. Finally, we have chosen to use particles whose dimension is the sum of those of the individual state spaces corresponding to each target, as in [13, 14]. Each of these concatenated vectors then gives jointly a representation of all targets. We will call the filters associated with this representation the joint filters.

C. MTPF Algorithms

Let us first study the joint density $p(X_t, Y_t, \Pi_t, K_t)$ (implicitly conditioned on X_{t-1}) where Π_t is the stochastic variable associated with the probability π_t .

We drop the subscript t for a few lines

$$p(X, Y, \Pi, K) = p(Y | K, \Pi, X) p(K | \Pi, X) p(\Pi, X) \quad (26)$$

$$= p(Y | K, X) p(K | \Pi) p(\Pi) p(X) \quad (27)$$

where the last equality is based on the following independence assumptions.

- 1) Y and Π are independent given K and X .
- 2) the association vector K is independent from the state vector X given Π .

The factors in (27) write:

3)

$$p(Y | K, X) = \prod_j p(Y^j | K^j, X) \\ = \prod_j \begin{cases} l^i(y^j | x^i) & \text{with } i = K^j \in \{1 \dots M\} \\ 1/V & \text{if } K^j = 0. \end{cases}$$

4) To simulate according to $p(K | \Pi)$ it is sufficient to generate $N(K)$ where $N^i(K) \triangleq \#\{j : K^j = i\}$. The vector $(N^0(K), \dots, N^M(K))$ follows a multinomial law of size m_t and of parameters (Π^0, \dots, Π^M) .

5) $p(\Pi) = p(\Pi^0) p(\Pi^1, \dots, \Pi^M | \Pi^0)$ where $p(\Pi^1, \dots, \Pi^M | \Pi^0)$ is uniform on the $[0, 1 - \pi^0]^M$'s hyperplane $\sum_{i=1}^M \pi_i = 1 - \pi^0$. Moreover, π^0 is a constant that can be computed:

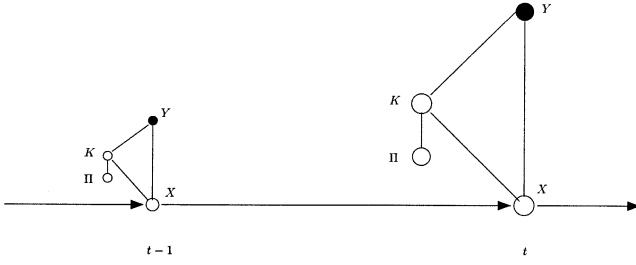


Fig. 11. Independence graph of stochastic variables X, Y, K, Π .

$$\pi^0 = \mathbb{P}(K_t^j = 0) \quad (28)$$

$$= \sum_{l=0}^{m_t} \mathbb{P}(K_t^j = 0 \mid N_t^0 = l) \mathbb{P}(N_t^0 = l) \quad (29)$$

$$= \sum_{l=0}^{m_t} \frac{l}{m_t} \exp(-\lambda V) \frac{(\lambda V)^l}{l!}. \quad (30)$$

Assuming that there are l measurements due to clutter among the m_t measurements collected at time t , the *a priori* probability that any measurement comes from the clutter is equal to l/m_t hence the equality $\mathbb{P}(K_t^j = 0 \mid N_t^0 = l) = l/m_t$ used to derive (30) from (29).

6) $p(X)$, i.e., $p(X_t \mid X_{t-1})$ conditional on X_{t-1} .

This independence structure of this distribution is illustrated in Fig. 11.

The initial particle set $S_0 = (s_0^n, 1/N)_{n=1, \dots, N}$ is such that each component $s_0^{n,i}$ for $i = 1, \dots, M$ is sampled from $p(X_0^i)$ independently from the others. Assume we have obtained $S_{t-1} = (s_{t-1}^n, q_{t-1}^n)_{n=1, \dots, N}$ with $\sum_{n=1}^N q_{t-1}^n = 1$. Each particle is a vector of dimension $\sum_{i=1}^M n_x^i$ where we denote by $s_{t-1}^{n,i}$ the i th component of s_{t-1}^n and where n_x^i designates the dimension of object i .

The prediction can be done according to the following equation

$$\tilde{s}_t^n = \begin{pmatrix} F_t^1(s_{t-1}^{n,1}, v_t^{n,1}) \\ \vdots \\ F_t^M(s_{t-1}^{n,M}, v_t^{n,M}) \end{pmatrix} \quad \text{for } n = 1, \dots, N. \quad (31)$$

Examine now the computation of the likelihood of the observations conditioned on the n th particle. We can write for all $n = 1, \dots, N$:

$$\begin{aligned} p(Y_t = (y_t^1, \dots, y_t^{m_t}) \mid X_t = \tilde{s}_t^n) \\ = \prod_{j=1}^{m_t} p(y_t^j \mid \tilde{s}_t^n) \end{aligned} \quad (32)$$

$$\propto \prod_{j=1}^{m_t} \left[\frac{\pi_t^0}{V} + \sum_{i=1}^M l_t^i(y_t^j; \tilde{s}_t^{n,i}) \pi_t^i \right]. \quad (33)$$

It must be noted that (32) is true only under the assumption of conditional independence of the measurements, which we will make. Moreover, the normalization factors between l_t^i and $p(Y_t^j = y \mid K_t^j = i, X_t^i = x)$ must be the same for all i to write the last equality (33).

We still need to estimate at each time step the association probabilities $(\pi_t^i)_{i=1, \dots, M}$, which can be seen as the stochastic coefficients of the M -component mixture. The simplest way would consist in estimating the π vector from the problem parameters like the number of targets M , the obtained number of measurements m_t and the clutter density λ . Assuming, the π_t^i are equal for all $i \neq 0$, we can empirically estimate $\pi_t^i = (m_t - \lambda V)/M$. However, these estimates do not take into account the current measurements. In particular, if a target is not detected during a time period, we would like the associated π component to be smaller than the others. For that we must incorporate information about the adequation between the measurements and the estimated targets in the π estimation procedure. Two main ways have been found in the literature to estimate the parameters of such a mixture: the EM method (and its stochastic version, the SEM algorithm [31]) and the Data Augmentation method. The second one amounts in fact to a Gibbs sampler.

In [3–5] the EM algorithm is extended and applied to multitarget tracking. This method implies that the vectors π_t and X_t are considered as parameters to be estimated. The maximization step can be easily conducted in the case of deterministic trajectories. In case of nondeterministic trajectories a maximum a priori (MAP) estimation is required to complete the M step. Yet, the nonlinearity of the state and observation functions makes this step very difficult. Finally, the estimation is done iteratively in a batch approach which we would like to avoid. For these reasons, we have not chosen an EM algorithm to estimate the association probabilities.

The Data Augmentation algorithm is quite different in its principle. The vectors X_t, K_t , and π_t are considered to be random variables with prior densities. Samples are then obtained iteratively from their joint posterior using a proper MCMC technique, namely the Gibbs sampler. This method has been studied in [32–36] for instance. It can be run sequentially at each time period. Gibbs sampler is a special case of the Metropolis-Hasting algorithm with the proposal densities being the conditional distributions, and the acceptance probability being consequently always equal to one. The interested reader can refer to [37] for an introduction to Markov chain Monte Carlo simulation methods and also for a presentation of the EM algorithm.

For $\theta_t = (X_t, K_t, \Pi_t)$, the method consists in generating a Markov chain that converges to the

stationary distribution $p(\theta_t | Y_{0:t})$ which cannot be sampled directly. For that, we must be able to get a partition $\theta_t^1, \dots, \theta_t^P$ of θ_t , and to sample alternatively from the conditional posterior distribution of each component of the partition. Let the index τ denote the iterations in the Gibbs sampler. The second subscript of the vectors refers to the iteration counter. Assume the $\tau + 1$ first elements of the Markov chain $(\theta_{t,0}, \dots, \theta_{t,\tau})$ have been drawn. We sample the P components of $\theta_{t,\tau+1}$ as follows:

$$\begin{aligned} \text{Draw } \theta_{\tau+1}^1 & \text{ from } p(\theta^1 | Y_{0:t}, \theta_{\tau}^2, \dots, \theta_{\tau}^P) \\ \text{Draw } \theta_{\tau+1}^2 & \text{ from } p(\theta^2 | Y_{0:t}, \theta_{\tau+1}^1, \theta_{\tau}^3, \dots, \theta_{\tau}^P) \\ & \vdots \\ \text{Draw } \theta_{\tau+1}^P & \text{ from } p(\theta^P | Y_{0:t}, \theta_{\tau+1}^1, \dots, \theta_{\tau+1}^{P-1}). \end{aligned}$$

The proof of the convergence of the Markov chain $(\theta_{\tau})_{\tau}$ is outlined in the Appendix. We propose to use it in the particle filter extended to multiple targets.

In our case, at a given instant t , we follow this approach with

$$\begin{cases} \theta^j = K_t^j & \text{for } j = 1, \dots, m_t \\ \theta^{m_t+i} = \pi_t^i & \text{for } i = 1, \dots, M \\ \theta^{m_t+M+i} = X_t^i & \text{for } i = 1, \dots, M. \end{cases} \quad (34)$$

The initialization of the Gibbs sampler consists of assigning uniform association probabilities, i.e., $\pi_{t,0}^i = (1 - \pi_t^0)/M$ for all $i = 1, \dots, M$, and taking $X_{t,0} = \sum_{n=1}^N q_{t-1}^n \tilde{s}_t^n$, i.e., the centroid of the predicted particle set. The K_t variables do not need initializing because at the first time step of the Gibbs sampler they will be sampled conditioned on $\pi_{t,0}^i$, $i = 1, \dots, M$ and $X_{t,0}$. Then, suppose that at instant t we have already simulated $(\theta_{t,1}, \dots, \theta_{t,\tau})$. The $\tau + 1$ th iteration is handled as follows.

+ 1) As the $(K_t^j)_{j=1, \dots, m_t}$ are supposed to be independent, their individual conditional density reads

$$p(K_t^j | Y_{0:t}, X_t, (K_t^i)_{i \neq j}, \Pi_t) = p(K_t^j | Y_t^j, X_t, \Pi_t). \quad (35)$$

(K_t^j) are discrete variables and we can write:²

$$\begin{aligned} \mathbb{P}(K_t^j = i | Y_t^j = y_t^j, X_t, \Pi_t) \\ = \frac{p(Y_t^j = y_t^j | K_t^j = i, X_t, \Pi_t) \mathbb{P}(K_t^j = i | X_t, \Pi_t)}{p(Y_t^j = y_t^j | X_t, \Pi_t)} \end{aligned} \quad (36)$$

$$\propto \begin{cases} \pi_t^i l_t^i(y_t^j; x_t^i) & \text{if } i = 1, \dots, M \\ \pi_t^0/V & \text{if } i = 0. \end{cases} \quad (37)$$

The realizations $k_{t,\tau+1}^j$ of the vector $K_{t,\tau+1}$ are then sampled according to the weights $p_{t,\tau+1}^{j,0} = \pi_t^0/V$, $p_{t,\tau+1}^{j,i} = \pi_{t,\tau}^i l_t^i(y_t^j; x_{t,\tau}^i)$ for $i = 1, \dots, M$.

²Using Bayes's rule, $p(a | b, c) = p(b | a, c)p(a | c)/p(b | c)$.

2) Mixture proportion vector $\Pi_{t,\tau+1}^{1:M}$ is drawn from the conditional density:

$$\begin{aligned} p(\Pi_t^{1:M} | K_{t,\tau+1}, X_{t,\tau}, Y_{0:t}) \\ = p(\Pi_t^1, \dots, \Pi_t^M | K_{t,\tau+1}^1, \dots, K_{t,\tau+1}^M, X_{t,\tau}, Y_{0:t}) \end{aligned} \quad (38)$$

$$= p(\Pi_t^1, \dots, \Pi_t^M | K_{t,\tau+1}^1, \dots, K_{t,\tau+1}^M) \quad (39)$$

$$\propto \text{Dirichlet}(\Pi_t^{1:M} | \{1 + N^i(K_{t,\tau+1})\}_{i=1, \dots, M}) \quad (40)$$

where we denote by $N^i(K)$ the number of k^j equal to i and where $\text{Dirichlet}(\delta_1, \dots, \delta_M)$ denotes the Dirichlet distribution on the simplex $\{(\pi_1^1, \dots, \pi_1^{M-1}, 1 - \pi_1^1 - \dots - \pi_1^{M-1}) : \pi_1^1 + \dots + \pi_1^{M-1} \leq 1\}$ with density proportional to $\pi_1^{1(\delta_1-1)} \times \dots \times \pi_1^{M-1(\delta_{M-1}-1)} \times (1 - \pi_1^1 - \dots - \pi_1^{M-1})^{\delta_M-1}$. The vector $\Pi_{t,\tau+1}^{1:M}$ is first drawn according to (40) and then normalized to restore the sum $\sum_{i=1}^M \pi_{t,\tau+1}^i$ to $1 - \pi_t^0$.

3) $X_{t,\tau+1}$ has to be sampled according to the density

$$p(X_t | Y_{0:t}, K_{t,\tau+1}, \Pi_{t,\tau+1}) = \prod_{i=1}^M p(X_t^i | Y_{0:t}, K_{t,\tau+1}, \Pi_{t,\tau+1}). \quad (41)$$

The values of $K_{t,\tau+1}$ can imply that one object is associated with zero or several measurements. Let us define $\mathcal{M}_{t,\tau+1}^i = \{j \in [1 \dots m_t] : K_{t,\tau+1}^j = i\}$. Hence we decompose the preceding product in two products:

$$\begin{aligned} \prod_{i: \mathcal{M}_{t,\tau+1}^i \neq \emptyset} p(X_t^i | Y_{0:t-1}, y_t^{\mathcal{M}_{t,\tau+1}^i}, \Pi_{t,\tau+1}) \\ \times \prod_{i: \mathcal{M}_{t,\tau+1}^i = \emptyset} p(X_t^i | Y_{0:t-1}, \Pi_{t,\tau+1}) \end{aligned} \quad (42)$$

where $y_t^{\mathcal{M}_{t,\tau+1}^i} = \{y_t^j, j \in \mathcal{M}_{t,\tau+1}^i\}$. The first product contains the targets that are associated to at least one measurement under the association $K_{t,\tau+1}$. In this case, the measurements are denoted by $y_t^{\mathcal{M}_{t,\tau+1}^i}$. The second product contains the targets that are associated with no measurements under $K_{t,\tau+1}$.

Let i be an integer in the first product. We propose two approaches to sample $X_{t,\tau+1}$

1) Without making any additional assumption we can write

$$\begin{aligned} p(X_t^i | Y_{0:t-1}, y_t^{\mathcal{M}_{t,\tau+1}^i}, \Pi_{t,\tau+1}) \\ = \frac{p(y_t^{\mathcal{M}_{t,\tau+1}^i} | X_t^i) p(X_t^i | Y_{0:t-1})}{p(y_t^{\mathcal{M}_{t,\tau+1}^i} | Y_{0:t-1})}. \end{aligned} \quad (43)$$

We are not able to sample directly from the density

$$\frac{p(y_t^{\mathcal{M}_{t,\tau+1}^i} | X_t^i) p(X_t^i | Y_{0:t-1})}{p(y_t^{\mathcal{M}_{t,\tau+1}^i} | Y_{0:t-1})}$$

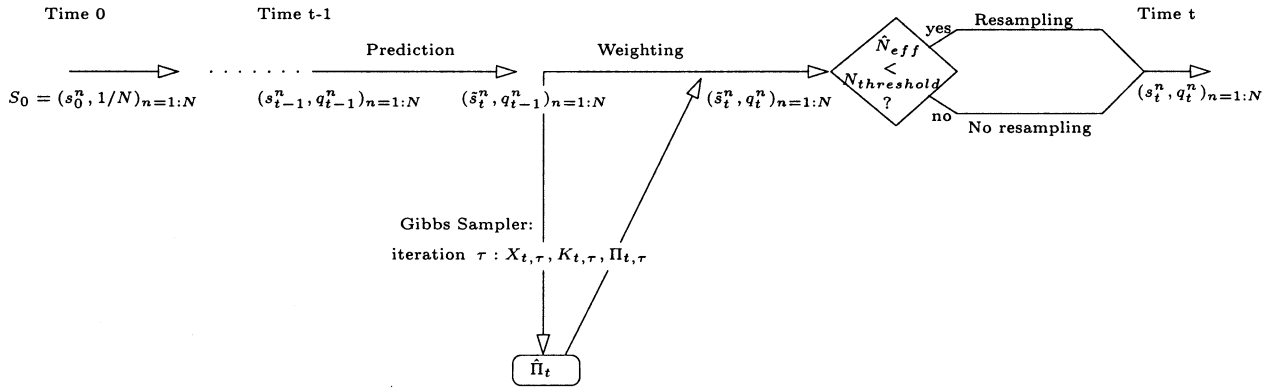


Fig. 12. Particle filter for multiple objects with adaptive resampling.

for the same reasons as those exposed in Section II to justify the use of the particle filter (intractability of the integrals). A first solution consists in building the particle set $\Sigma_{\tau+1} = (\sigma_{\tau+1}^n, \chi_{\tau+1}^n)_{n=1,\dots,N}$ whose weights $\chi_{\tau+1}^n$ measure the likelihood of the observations affected by $K_{t,\tau+1}$ to object X_t^i . More precisely, we let

$$\begin{cases} \sigma_{\tau+1}^n = \tilde{s}_t^{n,i} \\ \chi_{\tau+1}^n = \frac{p(y_t^{M_{t,\tau+1}^i} | X_t^i = \sigma_{\tau+1}^n) q_{t-1}^n}{\sum_{n=1}^N p(y_t^{M_{t,\tau+1}^i} | X_t^i = \sigma_{\tau+1}^n) q_{t-1}^n} \end{cases} \quad (44)$$

As the predicted empirical law $L_{\tilde{s}_t} = \sum_{n=1}^N q_{t-1}^n \tilde{s}_t^{n,i}$ is “close” from the predicted law $p(X_t^i | Y_{0:t-1})$, we expect the empirical distribution $\Lambda_{\tau+1} = \sum_{n=1}^N \chi_{\tau+1}^n \delta_{\sigma_{\tau+1}^n}$ to be close to $p(X_t^i | y_t^{M_{t,\tau+1}^i}, Y_{0:t-1})$. However the weak convergence of $\Lambda_{\tau+1}$ to $p(X_t^i | y_t^{M_{t,\tau+1}^i}, Y_{0:t-1})$ when N tends towards infinity remains to be proved. Not being able to sample from this last density, $X_{t,\tau+1}^i$ is drawn as a realization from $\Lambda_{\tau+1}$.

2) The second solution assumes the measurement equation enables us to sample from the density $P(X_t = x | Y_t = y)$ and to forget the observations from the past. The likelihood $p(X_t^i | Y_{0:t-1}, y_t^{M_{t,\tau+1}^i}, \Pi_{t,\tau+1})$ is then reduced to $p(X_t^i | y_t^{M_{t,\tau+1}^i})$. We do not assume that the observations $y_t^{M_{t,\tau+1}^i}$ are independent but we use their centroid \bar{y} and replace $p(X_t^i = x | y_t^{M_{t,\tau+1}^i}, K_t^{j^1} = i, \dots, K_t^{j^l} = i)$ by $p(X_t^i = x | \bar{y})$.

As far as the complexity of these two solutions is concerned, it is to be noticed that the first one depends linearly on the total number of particles whereas the second is independent of it. On the other hand, the second solution requires the ability to sample from $p(X_t | Y_t)$.

Now let i be an integer in the second product. As we do not have any measurement to correct the predicted particles we draw a realization from the density $\sum_{n=1}^N q_{t-1}^n \delta_{\tilde{s}_t^{n,i}}$ for $X_{t,\tau+1}^i$. After a finite number of iterations, we estimate the vector π_t by the average of its last realizations:

$$\hat{\pi}_t^i = \frac{1}{\tau_{\text{beg}} - \tau_{\text{end}}} \sum_{\tau=\tau_{\text{beg}}}^{\tau_{\text{end}}} \pi_{t,\tau}^i. \quad (45)$$

Finally the weights can be computed according to (33) using the estimate $\hat{\pi}_t^i$ of π_t^i . By construction, $\hat{\pi}_t^i$ follows the law $p(\Pi_t^i | Y_{0:t})$. Thus, the use of a Gibbs sampler enables to take into account the current measurements. Consequently the estimates measure in a way the *a posteriori* detection probability of each target. It improves the estimation of the targets because the measurements contribute to the estimation proportionally to these probabilities $\hat{\pi}_t$. Moreover, the *a priori* probability of detecting a target, which is usually denoted by P_d is not needed in the MTPF. This probability is needed when the associations are considered in classical algorithms like the PMHT or the JPDAF.

The resampling step is performed in an adaptive way when the estimated effective sample size \hat{N}_{eff} is under the threshold $N_{\text{threshold}}$.

Due to the estimation of the π_t vector needed for the computation of the particles likelihood, the convergence of the MTPF could be affected. It could be interesting to evaluate the error on the estimate of X_t implied by the error made on the estimate of π_t . This is not addressed in this work. Figs. 12 and 13 summarize the whole algorithm.

Before presenting the results of some simulations, let us detail the approach used in [29, 30] which is the closest to our approach.

D. SIR-JPDA Algorithm

The initialization of the particle set is done in the same way as for the joint filters and the prediction step as well, according to (31). The likelihood is then computed in a different way:

$$\begin{aligned} p(Y_t = (y_t^1, \dots, y_t^{m_t}) | X_t = \tilde{s}_t^n) \\ = \sum_{\text{all associations}} p(Y_t = (y_t^1, \dots, y_t^{m_t}) | X_t = \tilde{s}_t^n, K_t^u) p(K_t^u) \end{aligned} \quad (46)$$

- Initialization: $\begin{cases} s_0^n \sim p(X_0) \\ q_0^n = 1/N \end{cases} \quad n = 1, \dots, N.$
- For $t = 1, \dots, T$:
 - Prediction: $\begin{cases} v_t^n \sim p(V_t) \\ \tilde{s}_t^n = F_t(s_{t-1}^n, v_t^n) \end{cases} \quad n = 1, \dots, N.$
 - Weighting
 1. Initialization of the Gibbs sampler: $\begin{cases} \pi_{t,0}^i = \frac{1-\pi_0^i}{M} & i = 1, \dots, M; \\ X_{t,0}^i = \sum_{n=1}^N q_{t-1}^n \tilde{s}_t^{n,i} & i = 1, \dots, M. \end{cases}$
 2. For $\tau = 0, \dots, \tau_{end}$:
 - a. $K_{t,\tau+1}^j \sim p(K_{t,\tau+1}^j = i) \propto \begin{cases} \pi_{t,\tau}^i l_t^i(y_t^j; x_{t,\tau}^i) & \text{if } i = 1, \dots, M; \\ \pi_{t,\tau}^0/V & \text{if } i = 0. \end{cases}$
 - b. $\pi_{t,\tau+1}^{1:M} \sim \mathcal{D}((1 + N^i(K_{t,\tau+1}))_{i=1,\dots,M}), N^i(K) \triangleq \#\{j : K^j = i\}.$
 - c. For each i such that $\exists j^1, \dots, j^i / K_{t,\tau+1}^j = i$,
 First solution:

$$\blacktriangleright \begin{cases} \sigma_{\tau+1}^n = \tilde{s}_t^{n,i} \\ \chi_{\tau+1} = \frac{p(y_t^1, \dots, y_t^i | X_t^i = \sigma_{\tau+1}^n) q_{t-1}^n}{\sum_{n=1}^N p(y_t^1, \dots, y_t^i | X_t^i = \sigma_{\tau+1}^n) q_{t-1}^n} \quad n = 1, \dots, N. \end{cases}$$

$$\blacktriangleright X_{t,\tau+1}^i \sim \sum_{n=1}^N \chi_{\tau+1}^n \delta_{\sigma_{\tau+1}^n}$$
 Second solution:

$$\blacktriangleleft \bar{y} = \text{centroid of } y_t^{j^1}, \dots, y_t^{j^i}.$$

$$\blacktriangleleft X_{t,\tau+1}^i \sim p(X_t | \bar{y}).$$
 - d. For each i such that $\#\{j / K_t^j = i\} = i$, $X_{t,\tau+1}^i \sim \sum_{n=1}^N q_{t-1}^n \delta_{\tilde{s}_t^{n,i}}.$
- 3. $\hat{\pi}_t^i = \frac{1}{\tau_{beg} - \tau_{end}} \sum_{\tau=\tau_{beg}}^{\tau_{end}} \pi_{t,\tau}^i \quad i = 1, \dots, M.$
- 4. $l_t(Y_t = (y_t^1, \dots, y_t^{m_t}) | \tilde{s}_t^n) = \prod_{j=1}^{m_t} \left[\frac{\pi_t^0}{V} + \sum_{i=1}^M l_t^i(y_t^j, \tilde{s}_t^n) \hat{\pi}_t^i \right] \quad n = 1, \dots, N.$
- 5. $q_t^n \propto q_{t-1}^n \frac{p(\tilde{s}_t^n | s_{t-1}^n) l_t(Y_t | \tilde{s}_t^n)}{f(\tilde{s}_t^n | s_{t-1}^n, Y_t)} \quad n = 1, \dots, N.$
- Return $\hat{\mathbb{E}}g(X_t) = \sum_{n=1}^N q_t^n g(\tilde{s}_t^n).$
- $\hat{N}_{eff} = \frac{1}{\sum_{n=1}^N (q_t^n)^2}.$
- Resampling: if $\hat{N}_{eff} < N_{threshold}$: $\begin{cases} s_t^n \sim \sum_{k=1}^N q_t^k \delta_{\tilde{s}_t^k} \\ q_t^n = 1/N \end{cases} \quad n = 1, \dots, N.$

Fig. 13. Particle filter algorithm for multiple objects with adaptive resampling.

$$= \sum_{\text{all associations } j=1}^{m_t} \prod_{j=1}^{m_t} p(y_t^j | \tilde{s}_t^n, K_t^u) p(K_t^u) \quad (47)$$

where K_t^u denotes the u th association hypothesis at time t . The associations between the measurements and the targets are established under the assumptions (A1)–(A2) exposed in Section IVA. The *a priori* probability of this association is:

$$p(K_t^u) = \frac{\Phi^u!}{m_t!} p_F(\Phi^u) \prod_{i=1}^M P_d^{D^u(i)} \prod_{i=1}^M (1 - P_d)^{1-D^u(i)} \quad (48)$$

where Φ^u is the number of false alarms in K_t^u , $p_F(\Phi^u)$ the probability to have Φ^u false alarms, P_d the probability for a target to be detected, and $D^u(i) = \{1, 0\}$ depending on whether target i is detected or not.

E. Extension to Multireceiver Data

A natural extension of the MTPF is to consider that observations can be collected by multiple

receivers. Let R be their number. We can easily adapt the particle filter to this situation. We always consider that the M targets (their number is fixed again) obey the state equation (23). Some useful notations must be added to modify the measurement equations. The observation vector at time t will be denoted by $y_t = (y_{t,r^1}^1, \dots, y_{t,r^{m_t}}^{m_t})$ where r^j refers to the receiver which received the j th measurement. This measurement is then a realization of the stochastic process:

$$Y_{t,r^j}^j = H_{t,r^j}^i(X_t^i, W_t^j) \quad \text{if } K_t^j = i. \quad (49)$$

We assume the independence of the observations collected by the different receivers. We denote by $l_{t,r^j}^i(y; x)$ the known functions which are proportional to $p(Y_{t,r^j}^j = y | K_t^j = i, X_t^i = x)$. The likelihood of the observations conditioned on the n th particle is readily obtained:

$$\begin{aligned} p(Y_t = (y_{t,r^1}^1, \dots, y_{t,r^{m_t}}^{m_t}) | X_t = s_t^n) \\ = \prod_{j=1}^{m_t} p(y_{t,r^j}^j | s_t^n) \end{aligned} \quad (50)$$

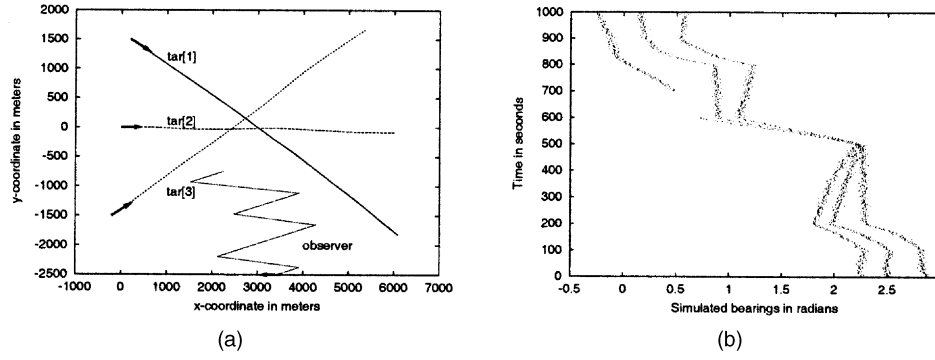


Fig. 14. Three targets bearings-only experiment in Section V (a) Trajectories of three targets and of observer. (b) Simulated bearings at each time with detection hole for one target between times 600 and 700.

$$\propto \prod_{j=1}^{m_t} \left[\frac{\pi_t^0}{V} + \sum_{i=1}^M l_{t,r,j}^i(y_{t,r,j}^j; s_t^{n,i}) \pi_t^i \right]. \quad (51)$$

There is no strong limitation on the use of the particle filter for multireceiver and MTT. The MRMTPF is obtained from the MTPF by replacing the likelihood functions $l_t^i(y; x)$ by the functions $l_{t,r,j}^i(y; x)$. Moreover it can deal with measurements of varied periodicities.

V. APPLICATION TO BEARINGS-ONLY PROBLEMS

The following multitarget scenario has been considered for illustrating our algorithm. Three targets follow a near-constant-velocity model defined by (7) with $\sigma_x = \sigma_y = 0.0005 \text{ ms}^{-2}$. The initial positions and velocities of the targets and of the observer are the following:

$$X_0^1 = \begin{pmatrix} 200 \text{ m} \\ 1500 \text{ m} \\ 1 \text{ ms}^{-1} \\ -0.5 \text{ ms}^{-1} \end{pmatrix} \quad X_0^2 = \begin{pmatrix} 0 \text{ m} \\ 0 \text{ m} \\ 1 \text{ ms}^{-1} \\ 0 \text{ ms}^{-1} \end{pmatrix}$$

$$X_0^3 = \begin{pmatrix} -200 \text{ m} \\ -1500 \text{ m} \\ 1 \text{ ms}^{-1} \\ 0.5 \text{ ms}^{-1} \end{pmatrix} \quad X_0^{\text{obs}} = \begin{pmatrix} 3500 \text{ m} \\ -2500 \text{ m} \\ -0.5 \text{ ms}^{-1} \\ 0 \text{ ms}^{-1} \end{pmatrix}.$$

The observer is following a leg by leg trajectory. Its velocity vector is constant on each leg and modified as follows:

$$\begin{pmatrix} vx_{30}^{\text{obs}} \\ vy_{30}^{\text{obs}} \end{pmatrix} = \begin{pmatrix} 1.2 \\ 0.3 \end{pmatrix} \quad \begin{pmatrix} vx_{100}^{\text{obs}} \\ vy_{100}^{\text{obs}} \end{pmatrix} = \begin{pmatrix} -3.0 \\ 0.3 \end{pmatrix}$$

$$\begin{pmatrix} vx_{200}^{\text{obs}} \\ vy_{200}^{\text{obs}} \end{pmatrix} = \begin{pmatrix} 1.2 \\ 0.3 \end{pmatrix} \quad \begin{pmatrix} vx_{500}^{\text{obs}} \\ vy_{500}^{\text{obs}} \end{pmatrix} = \begin{pmatrix} -3.0 \\ 0.3 \end{pmatrix} \quad (52)$$

$$\begin{pmatrix} vx_{600}^{\text{obs}} \\ vy_{600}^{\text{obs}} \end{pmatrix} = \begin{pmatrix} 1.2 \\ 0.3 \end{pmatrix} \quad \begin{pmatrix} vx_{800}^{\text{obs}} \\ vy_{800}^{\text{obs}} \end{pmatrix} = \begin{pmatrix} -4.0 \\ 0.3 \end{pmatrix}$$

$$\begin{pmatrix} vx_{900}^{\text{obs}} \\ vy_{900}^{\text{obs}} \end{pmatrix} = \begin{pmatrix} 1.2 \\ 0.3 \end{pmatrix}.$$

The trajectories of the three targets and of the observer are plotted in Fig. 14(a). The targets produce one measurement at each time period according to (9) with $\sigma_w = 0.02 \text{ rad}$ except during the time interval [600 700] where the first object does not produce any measurement. The simulated bearings are plotted in Fig. 14(b). As soon as the difference between two bearings issued from two different targets is lower than the standard deviation of the observation noise, the two measurements cannot be distinguished, which makes the data association of this scenario very difficult. This difficulty is increased by the detection hole for the first object. We compare the estimated trajectories when the assignment probabilities are, respectively, estimated by the two versions of the Gibbs sampler in Section IVC. In both cases, the particle filters have been initialized according to a Gaussian law whose mean vector X_{mean} and covariance matrix X_{cov} are

$$X_{\text{mean}}^1 = X_0^1 + \begin{pmatrix} 200 \text{ m} \\ 200 \text{ m} \\ 0 \text{ ms}^{-1} \\ 0 \text{ ms}^{-1} \end{pmatrix}$$

$$X_{\text{mean}}^2 = X_0^2 + \begin{pmatrix} -200 \text{ m} \\ -200 \text{ m} \\ 0 \text{ ms}^{-1} \\ 0 \text{ ms}^{-1} \end{pmatrix} \quad (53)$$

$$X_{\text{mean}}^3 = X_0^3 + \begin{pmatrix} -100 \text{ m} \\ 200 \text{ m} \\ 0 \text{ ms}^{-1} \\ 0 \text{ ms}^{-1} \end{pmatrix}$$

$$X_{\text{cov}}^i = \begin{pmatrix} 200^2 & 0 & 0 & 0 \\ 0 & 200^2 & 0 & 0 \\ 0 & 0 & 0.05^2 & 0 \\ 0 & 0 & 0 & 0.05^2 \end{pmatrix}$$

for $i = 1, 2, 3$. (54)

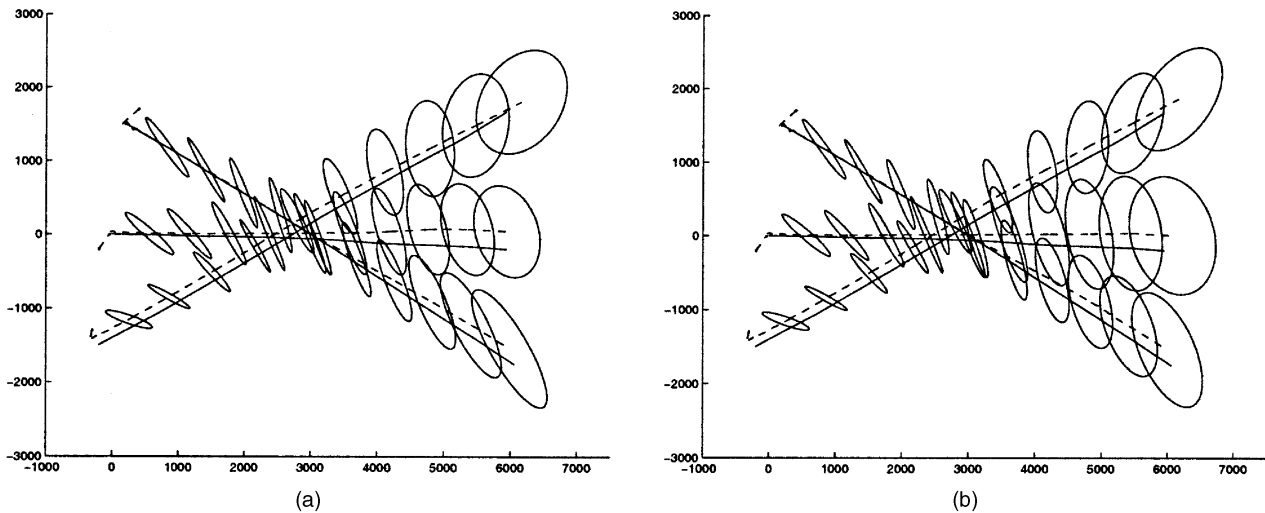


Fig. 15. Averaged estimates (dotted lines) and 2σ confidence ellipses obtained with 1000 particles. (a) Estimation of π_t with first version of Gibbs sampler. (b) Estimation of π_t with second version of Gibbs sampler.

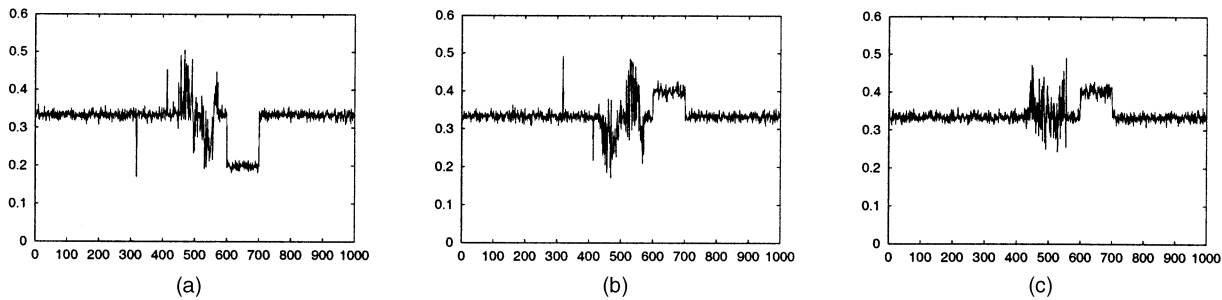


Fig. 16. Estimated components of vector π_t obtained with first version of Gibbs sampler and 1000 particles. (a) $\hat{\pi}_t^1$. (b) $\hat{\pi}_t^2$. (c) $\hat{\pi}_t^3$.

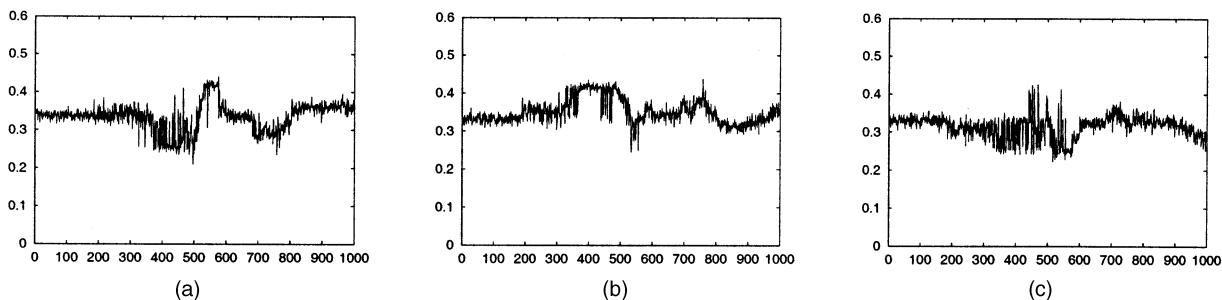


Fig. 17. Estimated components of vector π_t obtained with second version of Gibbs sampler and 1000 particles. (a) $\hat{\pi}_t^1$. (b) $\hat{\pi}_t^2$. (c) $\hat{\pi}_t^3$.

The burn-in period of the Gibbs sampler has been fixed to $N_{\text{beg}} = 100$ and the total amount of iterations to $N_{\text{end}} = 500$.

The averaged estimates with the associated 2σ confidence ellipses obtained with 1000 particles and adaptive resampling are presented in Figs. 15(a) and 15(b) using, respectively, the first and the second version of the Gibbs sampler to estimate the vector π_t . The two plots show that the data association is overcome in the two versions of the algorithm. There is no trajectory reversal and the estimates are quite satisfactory for the two versions. The confidence ellipses increase when the simulated bearings are very close, i.e., especially between times 450 and

600. After this time period, their size stabilizes. The obtained estimates are very similar for both versions of the algorithm. However, the estimation of the π_t vector is much more satisfactory for the first version. Figs. 16 and 17 show the results of the estimation of the three components of π_t , respectively, with the first and the second versions of the algorithm. Fig. 18 represents the average of each component π_t^i over successive intervals of 100 time steps and over the 20 trials. At time $100u$, the average over the time interval $[100u, 100(u+1)]$ and over the 20 trials is plotted for each component. When there is an ambiguity about the origin of the measurements (i.e., when the difference between the bearings is

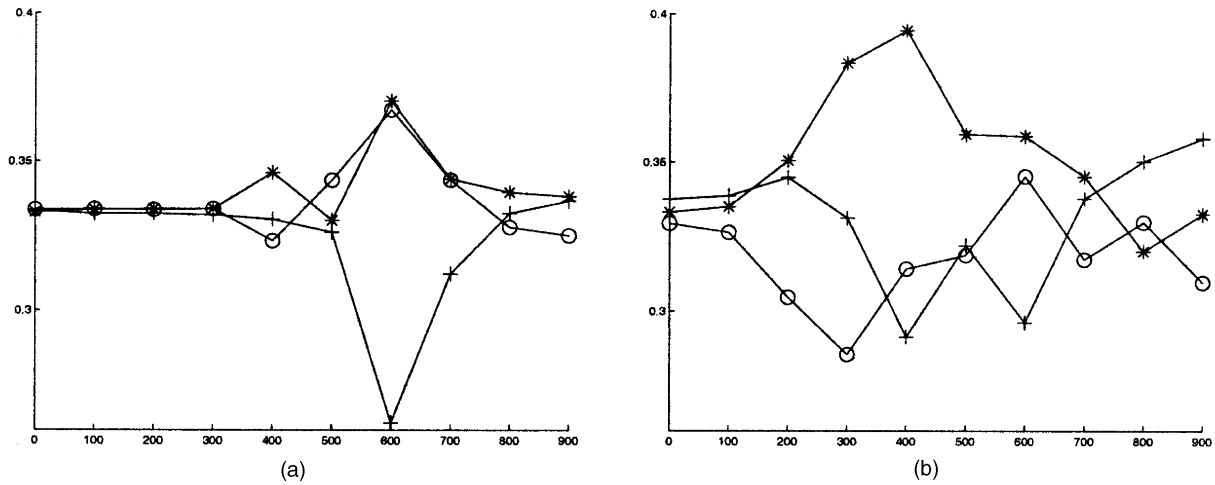


Fig. 18. Average of estimated components of vector π_t over consecutive ten time intervals of length 100 and over 20 trials. (a) First version. (b) Second version of Gibbs sampler. Averaged estimates marked with “+” for π^1 , “o” for π^2 , and “*” for π^3 .

lower than the standard deviation noise), the first version makes π vary in average around $1/3$ for $M = 3$ objects and produces uniform estimation ($1/3$ for $M = 3$ objects) when the ambiguity disappears (before time 400 or after time 700 for instance). Second, the momentary measurement gap for the first object is correctly handled: the first component π_t^1 is instantaneously estimated as 0.26 in average from instant 600 to 700 and the second and third components as 0.37.

The obtained estimates with the second version of Gibbs sampler are much less satisfactory. First, the measurement gap for the first object is not detected by this version as the π_t^1 estimate does not decrease during the time interval [600 700]. Second, as shown in Fig. 18(b), the averaged estimates diverge from instant 100 from the expected value $1/3$, whereas all the targets are detected at that time.

The expected superiority of the first algorithm seems to hold in practice, with a better estimation of π_t . However, one can be surprised that the bad estimation of π_t obtained with the second algorithm has no impact on the estimation of the targets. On this subject, it must be noticed that during the hundred first instants, the vector π_t is correctly handled by both versions. Moreover the particles recover very quickly from their shifted initialization. Thus, from time 100, as the dynamic noise on the targets is low enough, the predicted particles are all “good.” Moreover, during the time interval [400 700], the targets produce very close bearings. Consequently, the likelihood $l_t^i(y_t^j; x_t^j)$ of a measurement given a target is almost independent from the target index: $l_t^i(y_t^j; x_t^j) \simeq l$ for all i and then $\sum_{i=0}^M l_t^i(y_t^j; x_t^j) \hat{\pi}_t^i \simeq l \sum_{i=0}^M \hat{\pi}_t^i = l$ whatever the $\hat{\pi}_t^i$ estimates. The estimation quality of π_t has then little impact on the likelihood of the particles during this time interval. As for the computational cost, it

takes almost 450 ms per iteration with $N = 1000$ particles, $N_{\text{beg}} = 100$ and $N_{\text{end}} = 500$ on a 863 Mhz Pentium III.

VI. CONCLUSION

The MTT has been investigated in the framework of particle filtering and Gibbs sampling. Target state vectors and association probabilities are estimated jointly without enumeration, pruning or gating, by means of particle sets representing the joint *a posteriori* law of the target states. Two versions, derived from a common formalism, have been considered. This framework is sufficiently versatile to handle a wide variety of situations like MTT for multireceivers, including nonlinear models.

APPENDIX. GIBBS SAMPLER

By construction, $(\theta_\tau)_{\tau \in \mathbb{N}}$ is a Markov chain. Let us prove it admits the stationary distribution $\lambda = p(\theta | Y_{0:t})$. For that, let us prove that if θ_τ is distributed according to λ then $\theta_{\tau+1}$ is also distributed according to λ .

Assume $\theta_\tau = (\theta_\tau^1, \dots, \theta_\tau^P) \sim \lambda$. Then, for all (z^1, \dots, z^P) :

$$\begin{aligned}
 p(\theta_{\tau+1}^1 = z^1, \theta_{\tau+1}^2 = z^2, \dots, \theta_{\tau+1}^P = z^P) \\
 &= p(\theta_{\tau+1}^1 = z^1 | \theta_\tau^2 = z^2, \dots, \theta_\tau^P = z^P) \\
 &\quad \times p(\theta_\tau^2 = z^2, \dots, \theta_\tau^P = z^P) \\
 &= p(\theta^1 = z^1 | Y_{0:t}, \theta^2 = z^2, \dots, \theta^P = z^P) \\
 &\quad \times p(\theta^2 = z^2, \dots, \theta^P = z^P | Y_{0:t}) \\
 &= p(\theta^1 = z^1, \dots, \theta^P = z^P | Y_{0:t}). \tag{55}
 \end{aligned}$$

The second equality is obtained from the first in (55) as, by construction, $\theta_{\tau+1}^1 \sim p(\theta^1 | \theta_\tau^{2:P}, Y_{0:t})$. $(\theta_{\tau+1}^1, \theta_\tau^2, \dots, \theta_\tau^P)$ is then distributed according to the law $p(\theta^1, \theta^2, \dots, \theta^P | Y_{0:t})$. In the same way we can

show that $(\theta_{\tau+1}^1, \theta_{\tau+1}^2, \dots, \theta_{\tau+1}^P), \dots, (\theta_{\tau+1}^1, \dots, \theta_{\tau+1}^P)$ are distributed according to $p(\theta^1, \dots, \theta^P | Y_{0:t})$.

Moreover, provided that the conditional distributions $p(\theta^1 | x, \theta^2, \dots, \theta^P), \dots, p(\theta^P | x, \theta^1, \dots, \theta^{P-1})$ are strictly positive, θ_{τ} is irreducible. These two conditions imply the convergence for π -almost all θ_0 of $(\theta_{\tau})_{\tau \in \mathbb{N}}$ to λ .

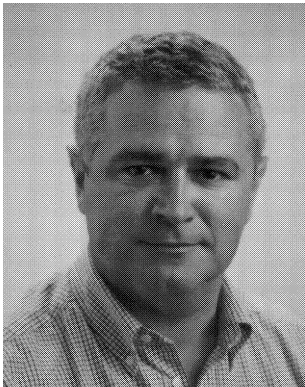
REFERENCES

- [1] Reid, D. (1979)
An algorithm for tracking multiple targets.
IEEE Transactions on Automation and Control, **24**, 6 (1979), 84–90.
- [2] Fortmann, T. E., Bar-Shalom, Y., and Scheffe, M. (1983)
Sonar tracking of multiple targets using joint probabilistic data association.
IEEE Journal of Oceanic Engineering, **8** (July 1983), 173–184.
- [3] Gauvrit, H., Le Cadre, J.-P., and Jauffret, C. (1997)
A formulation of multitarget tracking as an incomplete data problem.
IEEE Transactions on Aerospace and Electronic Systems, **33**, 4 (Oct. 1997), 1242–1257.
- [4] Streit, R. L., and Luginbuhl, T. E. (1994)
Maximum likelihood method for probabilistic multi-hypothesis tracking.
In Proceedings of SPIE International Symposium, Signal and Data Processing of Small Targets 1994, **2235**, Orlando, FL, Apr. 1994.
- [5] Willett, P., Ruan, Y., and Streit, R. (1998)
The PMHT for maneuvering targets.
In Proceedings of the Conference on Signal and Data Processing of Small Targets, **3373** (SPIE Annual International Symposium on Aerosense), Orlando, FL, Apr. 1998.
- [6] Rago, C., Willett, P., and Streit, R. (1995)
A comparison of the JPDA and PMHT tracking algorithms.
In Proceedings of the International Conference on Acoustics, Speech, Signal Processing, May 1995, 3571–3575.
- [7] Gordon, N., Salmond, D., and Smith, A. (1993)
Novel approach to nonlinear/non-Gaussian Bayesian state estimation.
IEE Proceedings, Pt. F, Radar and Signal Processing, **140**, 2 (Apr. 1993), 107–113.
- [8] Isard, M., and Blake, A. (1998)
CONDENSATION—Conditional density propagation for visual tracking.
International Journal of Computer Vision, **29**, 1 (1998), 5–28.
- [9] Hürzeler, M., and Künsch, H. R. (1998)
Monte Carlo approximations for general state space models.
Journal of Computational and Graphical Statistics, **7** (1998), 175–193.
- [10] Musso, C., and Oudjane, N. (1999)
Particle methods for multimodal filtering.
Presented at the 2nd International Conference on Information Fusion IEEE, Silicon Valley, CA, July 6–8, 1999.
- [11] Avitzour, D. (1995)
Stochastic simulation Bayesian approach to multitarget approach.
IEE Proceedings, Radar, Sonar Navigation, **142**, 2 (Apr. 1995), 41–44.
- [12] Gordon, N. (1997)
A hybrid bootstrap filter for target tracking in clutter.
IEEE Transactions on Aerospace and Electronic Systems, **33**, 1 (1997), 353–358.
- [13] MacCormick, J., and Blake, A. (1999)
A probabilistic exclusion principle for tracking multiple objects.
In Proceedings of the 7th International Conference on Computer Vision, Kerkyra, Greece, Sept. 20–27, 1999, 572–580.
- [14] MacCormick, J., and Israd, M. (2001)
Bramble: A Bayesian multiple-blob tracker.
In Proceedings of the 8th International Conference on Computer Vision, **2**, Vancouver, Canada, July 9–12, 2001, 34–41.
- [15] Schulz, D., Burgard, W., Fox, D., and Cremers, A. B. (2001)
Tracking multiple moving targets with a mobile robot using particle filters and statistical data association.
In Proceedings of IEEE International Conference on Robotics and Automation, Seoul, Korea, May 21–26, 2001, 1665–1670.
- [16] Doucet, A. (1998)
On sequential simulation-based methods for Bayesian filtering.
Technical report, CUED/F-INFENG/TR 310, Signal Processing Group, Department of Engineering, University of Cambridge, 1998.
- [17] MacCormick, J. (2000)
Probabilistic Modelling and Stochastic Algorithms for Visual Localisation and Tracking.
Ph.D. dissertation, University of Oxford, Jan. 2000.
- [18] Doucet, A., De Freitas, N., and Gordon, N. (Eds.) (2001)
Sequential Monte Carlo Methods in Practice.
New York: Springer, 2001, 525–532.
- [19] Salmond, D. J. (1990)
Mixture reduction algorithms for target tracking in clutter.
SPIE Signal and Data Processing of Small Targets, **1305** (1990), 434–445.
- [20] Kong, A., Liu, J. S., and Wong, W. H. (1994)
Sequential imputation method and Bayesian missing data problems.
Journal of American Statistical Association, **89** (1994), 278–288.
- [21] Liu, J. S. (1996)
Metropolized independent sampling with comparison to rejection sampling and importance sampling.
Statistics and Computing, **6** (1996), 113–119.
- [22] Del Moral, P., and Guionnet, A. (1999)
Central limit theorem for nonlinear filtering and interacting particle systems.
The Annals of Applied Probability, **9**, 2 (1999), 275–297.
- [23] Doucet, A. (2000)
Convergence of sequential Monte Carlo methods.
Technical report, CUED/F-INFENG/TR 381, Signal Processing Group, Department of Engineering, University of Cambridge, 2000.
- [24] Li, X., and Jilkov, V. (2000)
A survey of maneuvering target tracking: Dynamic models.
Presented at the Conference on Signal and Data Processing of Small Targets, Orlando, FL, Apr. 2000.
- [25] Hassab, J. C. (1989)
Underwater Signal and Data Processing.
Boca Raton, FL: CRC Press, 1989, 221–243.
- [26] Peach, N. (1995)
Bearings-only tracking using a set of range parameterised extended Kalman filters.
IEE Proceedings—Control Theory Appl., **142**, 1 (Jan. 1995).

- [27] Grossman, W. (1994)
Bearings-only tracking: A hybrid coordinate system approach.
Journal of Guidance, Controls, and Dynamics, **17**, 3 (May–June 1994), 451–457.
- [28] Bar-Shalom, Y., and Fortmann, T. E. (1988)
Tracking and Data Association.
New York: Academic Press, 1988.
- [29] Karlsson, R., and Gustafsson, F. (2001)
Monte Carlo data association for multiple target tracking.
In *Proceedings of IEE Seminar—Target Tracking: Algorithms and Applications* (Oct. 2001), 13/1–13/5.
- [30] Orton, M., and Marris, A. (2001)
A Bayesian approach to multi-target tracking and data fusion with out-of-sequence measurements.
In *Proceedings of IEE Seminar—Target Tracking: Algorithms and Applications* (Oct. 2001), 15/1–15/5.
- [31] Celeux, G., and Diebolt, J. (1992)
A stochastic approximation type EM algorithm for the mixture problem.
Stochastic and Stochastic Reports, **41** (1992), 119–134.
- [32] Gelfand, A. E., and Smith, A. F. M. (1990)
Sampling-based approaches to calculating marginal densities.
Journal of the American Statistical Association, **85** (1990), 398–409.
- [33] Casella, G., and George, E. I. (1992)
Explaining the Gibbs sampler.
The American Statistician, **46**, 3 (1992), 167–174.
- [34] Smith, A., and Roberts, G. (1993)
Bayesian computation via the Gibbs Sampler and related Markov chain Monte Carlo methods.
Journal of the Royal Statistical Society, Series B, **55**, 1 (1993), 3–24.
- [35] Diebolt, J., and Robert, C. P. (1994)
Estimation of finite mixture distributions through Bayesian sampling.
Journal of the Royal Statistical Society, Series B, **56** (1994), 363–375.
- [36] Stephens, M. (1997)
Bayesian methods for mixtures of normal distributions.
Ph.D. dissertation, Magdalen College, Oxford, 1997.
- [37] de Freitas, J. F. G. (1999)
Bayesian methods for neural networks.
Ph.D. dissertation, Trinity College, University of Cambridge, 1999.



Carine Hue was born in 1977. She received the M.Sc. degree in mathematics and computer science in 1999 from the University of Rennes, France. Since 1999, she has been pursuing the Ph.D. degree with IRISA, Rennes. She works on particle filtering methods for tracking in signal processing and image analysis.



Jean-Pierre Le Cadre (M'93) received the M.S. degree in mathematics in 1977, the "Doctorat de 3^{-eme} cycle" in 1982, and the "Doctorat d'Etat" in 1987, both from INPG, Grenoble.

From 1980 to 1989, he worked at the GERDSM (Groupe d'Etudes et de Recherche en Detection Sous-Marine), a laboratory of the DCN (Direction des Constructions Navales), mainly on array processing. Since 1989, he is with IRISA/CNRS, where he is "Directeur de Recherche" at CNRS. His interests are now topics like system analysis, detection, multitarget tracking, data association, and operations research.

Dr. Le Cadre has received (with O. Zugmeyer) the Eurasip Signal Processing best paper award (1993).



Patrick Pérez was born in 1968. He graduated from Ecole Centrale Paris, France, in 1990. He received the Ph.D. degree in signal processing and telecom. From the University of Rennes, France, in 1993.

After one year as an Inria post-doctoral fellow at Brown University, Providence, RI (Department of Applied Mathematics), he was appointed as a full-time Inria researcher. In 2000 he joined Microsoft Research Cambridge, UK. His research interests include probabilistic models for image understanding, high dimensional inverse problems in image analysis, analysis of motion, and tracking in image sequences.