10.2478/v10170-010-0031-6

Archives of Control Sciences
Volume 21(LVII), 2011
No. 1, *pages 37–60*

Nonlinear multiple model particle filters algorithm for tracking multiple targets

ABDENNOUR SEBBAGH and HICHAM TEBBIKH

The paper addresses multiple targets tracking problem encountered in number of situations in signal and image processing. In this paper, we present an efficient filtering algorithm to perform accurate estimation in jump Markov nonlinear systems, which we aim to contribute in solving the problem of multiple targets tracking using bearings-only measurements. The idea of this algorithm consists of the combination between the multiple model approach and particle filtering methods, which give a nonlinear multiple model particle filters algorithm. This algorithm is used to estimate the trajectories of multiple targets assumed to be nonlinear, from their noisy bearings.

Key words: estimation, particle filter, multiple targets tracking, multiple model approach

1. Introduction

Multiple targets tracking (MTT) problem deals with correctly tracking several targets given noisy sensor measurement at every instant. To perform MTT an observer can rely on a huge amount of data, possibly collected from different sensors. The main difficulty, however, comes from the assignment of a given measurement to a target model. These assignments are generally unknown, because the accurate target models are unknown as well. This is a neat departure from classical estimation problems. Thus, two distinct problems have to be solved jointly: the data association and the estimation.

Estimation and filtering are two of the most pervasive tools of engineering if the state of a system must be estimated from the noisy sensor information. Some kind of state can be employed to fuse the data from different sensors to produce an accurate estimate of the system state. When the system dynamics and observation models are linear, the minimum mean squared error (MMSE) estimate may be computed using the Kalman filter. However, in most applications the system dynamics and observation equations are nonlinear and suitable extensions of the Kalman filter have been sought. It is well-known that the optimal solution to the nonlinear filtering problem requires that a complete de-

---

The Authors are with Laboratoire d'Automatique and Informatique de Guelma (LAIG), Université 08 Mai 1945, Guelma, BP: 401, 24000 Guelma, Algeria. E-mails: abdennour.sebbagh@hotmail.com, tebbikh@yahoo.com

Received 23.11.2010.

scription of the conditional probability density is maintained [10]. Unfortunately this exact description requires a potentially unbounded number of parameters then a number of suboptimal approximations have been proposed [9].

Probably the most widely used estimator for nonlinear systems is the extended Kalman filter (EKF) [13]. The EKF applies the Kalman filter to nonlinear systems by simply linearizing all the nonlinear models so that the traditional linear Kalman filter equations can be applied. However, in practice, the use of the EKF has two well-known drawbacks:

1. Linearization can produce highly unstable filter if the assumption of local linearity is violated.

2. The derivations of the Jacobian matrices are nontrivial in most applications and often lead to significant implementation difficulties.

Under the assumptions of stochastic state equation, nonlinear state and/or measurement models and non-Gaussian noises, we present another filter to track a nonlinear targets. This filter is called sequential Monte Carlo methods or particle filtering methods. 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 formula. Particle filtering can be applied under very general hypotheses, and is very simple to implement. Such filters have been used in very different areas for Bayesian filtering, under different names: the bootstrap filter for target tracking in [8] and the condensation algorithm in computer vision [12] 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 [8]. Many ways have been developed to accomplish this resampling whose final goal is to enforce particles in areas of high likelihood.

In literature, several algorithms are developed to track multiple targets, among them we find the joint probabilistic data association filter (JPDAF), the multiple hypotheses tracker (MHT) and the probabilistic multiple hypotheses tracker (PMHT) algorithms. These later yield good performances with efficient computation especially when the measurement and state models are linear. However, if they are nonlinear, these algorithms break down. We propose in this paper another algorithm to track multiple targets with bearing only measurement under the assumptions of stochastic state equation, nonlinear state or/and measurement models and non-Gaussian noises. This algorithm is called nonlinear multiple model particle filter (NMMPF). The basic idea is to combine the multiple model approach with a particle filtering methods. A bank of filters results from this combination. Each filter is used to estimate one target model. The output of the algorithm at every instant for one measurement is the combination of all the outputs of each filter weighted by the probabilities of each filter.

The paper is organized as follows. In section 2, the mathematical formulation of air traffic control (ATC) motion models in the (X, Y) plane is presented. In section 3 and 4, we describe the particle filtering methods with adaptive resampling and its application to single target tracking with bearing-only measurement. Section 5, the central part of this work, deals with multiple targets tracking. New algorithm (NMMPF) combines the multiple model approach with particle filtering to obtain a bank of filters which estimate respective models. Finally, section 6 is devoted to application of the NMMPF to track multiple targets with bearing-only measurements and discuss the results of simulations.

2. ATC motion models

In air traffic control (ATC), civilian aircraft have two basic modes of flight [21]:

- Uniform motion (UM): the straight and level flight with a constant speed and course.
- Maneuver: turning or climbing/descending.

The horizontal and vertical motion models can be, typically, assumed to be decoupled [11]. There are many estimators discussed horizontal motion in literature. The flight modes in horizontal plane can be modeled by:

- A nearly constant velocity model for the uniform motion, implemented as WNA (white noise acceleration, or second-order kinematic) model with low-level process noise.
- A maneuvering model, which can be implemented as a WNA model with significant process noise, commensurate with the expected maneuvers.
- A nearly *coordinated turn* model.

The nearly constant model is given by

$$X_{k+1} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} X_k + \begin{bmatrix} 0.5T^2 & 0 \\ T & 0 \\ 0 & 0.5T^2 \\ 0 & T \end{bmatrix} V_k \tag{1}$$

where T is the sampling interval, X is the state of the aircraft, defined as

$$X = [\xi \quad \dot{\xi} \quad \eta \quad \dot{\eta}]' \tag{2}$$

with ξ and η denoting the Cartesian coordinates of the horizontal plane. V_k is a zero-mean Gaussian white noise used to model (*cover*) small accelerations, the turbulence, wind change, and so on, with an appropriate covariance

$$Q = \begin{bmatrix} \sigma_X^2 & 0 \\ 0 & \sigma_Y^2 \end{bmatrix},$$

which is a design parameter.

The turn of a civilian aircraft usually follows a pattern known as *coordinated turn* (CT), characterized by constant turn rate and constant speed. Although the actual turns are not exactly 'coordinated' since the ground speed is the airspeed plus the wind speed, it can be suitably described by the 'coordinated turn' model plus a fairly small noise representing the modeling error, resulting in the nearly coordinated turn model. The CT model is necessarily a nonlinear one if the turn rate is unknown constant. Augmenting the state vector in equation (2) by one more component, the turn rate ω, that is,

$$X = [\xi \quad \dot{\xi} \quad \eta \quad \dot{\eta} \quad \omega]' \tag{3}$$

makes the nearly coordinated turn model in the following form

$$X_{k+1} = \begin{bmatrix} 1 & \dfrac{\sin\omega_k T}{\omega_k} & 0 & -\dfrac{1-\cos\omega_k T}{\omega_k} & 0 \\ 0 & \cos\omega_k T & 0 & -\sin\omega_k T & 0 \\ 0 & \dfrac{1-\cos\omega_k T}{\omega_k} & 1 & \dfrac{\sin\omega_k T}{\omega_k} & 0 \\ 0 & \sin\omega_k T & 0 & \cos\omega_k T & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} X_k + \begin{bmatrix} 0.5T^2 & 0 & 0 \\ T & 0 & 0 \\ 0 & 0.5T^2 & 0 \\ 0 & T & 0 \\ 0 & 0 & T \end{bmatrix} V_k. \tag{4}$$

Note that the process noise V_k in equation (1) has different dimension then the one in equation (4).

The observations are available at discrete times according to

$$Y_k = \arctan\left(\frac{\eta = \Upsilon}{\xi = \mathrm{X}}\right) + W_k \tag{5}$$

where W_k is zero-mean Gaussian noise of covariance σ_w^2 independent of V_k, and (X , Y) is the position of the target in the horizontal plane.

3. Particle filtering methods

We consider a dynamic system represented by the stochastic process $(X_k) \in R^{n_x}$ whose evolution is given by the state equation [3, 19]:

$$X_k = F_k(X_{k-1}, V_k). \tag{6}$$

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

$$Y_k = H_k(X_k, W_k). \tag{7}$$

The two processes $(V_k) \in R^{n_v}$ and $(W_k) \in R^{n_w}$ are only supposed to be independent white noises. Note that the functions F_k and H_k are not assumed linear.

We will denote by $Y_{0:k}$ the sequence of the random variables $(Y_0, \dots, Y_k)$ and by $y_{0:k}$ one realization of this sequence. Note that throughout the paper, the first subscript of any vector always refers to the discrete time. Our problem consists in computing at each discrete time k the conditional density $L_k = p(X_k | Y_0 = y_0, \dots, Y_k = y_k)$ of the state X_k given all the observations accumulated up to k, and also in estimating any functional $g(X_k)$ of the state via the expectation $\mathrm{E}(g(X_k) | Y_{0:k})$ [2]. The recursive Bayesian filter, also named optimal filter, resolves exactly this problem in two steps at each time k as follows:

- Initialization: $\left\{ \begin{array}{l} s_0^n \sim p(X_0) \\ q_0^n = \dfrac{1}{N} \end{array} \right., n = 1, \dots, N$
- for $k = 1, \dots, T_{end}$:
  - Proposal: sample s_k^n from $f(X_k | X_{k-1} = s_{k-1}^n, Y_k = y_k)$ for $n = 1, \dots, N$
  - Weighting: $\left\{ \begin{array}{l} \text{compute un} - \text{normalized weight :} \\ \qquad \widetilde{q}_k^n = q_{k-1}^n \dfrac{p(s_k^n | s_{k-1}^n) l_k(y_k; s_k^n)}{f(s_k^n | s_{k-1}^n, y_k)} \quad \text{for } n = 1, \dots\dots, N \\ \text{normalize weights :} \\ \qquad q_k^n = \dfrac{\widetilde{q}_k^n}{\sum_{n=1}^N \widetilde{q}_k^n} \quad \text{for } n = 1, \dots., N \end{array} \right.$
  - Return $\widehat{\mathrm{E}}g(X_k) = \sum_{n=1}^N q_k^n g(s_k^n)$.

Figure 1. Particle filter without resampling.

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

$$p(X_k = x_k | Y_{0:k-1} = y_{0:k-1}) = \int_{R^{n_x}} p(X_k = x_k | X_{k-1} = x) L_{k-1}(x) dx. \tag{8}$$

Using equation (6), we can calculate $p(X_k = x_k | X_{k-1} = x)$:

$$p(X_k = x_k | X_{k-1} = x) = \int_{R^{n_x}} p(X_k = x_k | X_{k-1} = x,\ V_k = v) p(V_k = v | X_{k-1} = x) dv. \tag{9}$$

The observation y_k enables us to correct this prediction using Bayes rule:

$$L_k(x_k) = \frac{p(Y_k = y_k | X_k = x_k) p(X_k = x_k | Y_{0:k-1} = y_{0:k-1})}{\int_{R^{n_x}} p(Y_k = y_k | X_k = x_k) p(X_k = x_k | Y_{0:k-1} = y_{0:k-1})\ dx}. \tag{10}$$

Under the specific assumptions of Gaussian noises V_k and W_k and linear functions F_k and H_k, 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 equation (8) and (10) infeasible (no closed-form). The original particle filter, which is called the bootstrap filter [8, 5, 19], proposes to approximate the densities $(L_k)_k$ by a finite weighted sum of N Dirac densities centered on elements of R^{n_x}, which are called particles. The application of the bootstrap filter requires that one knows how to do the following [2, 3]:

- sample from initial prior marginal $p(X_0)$;
- sample from $p(V_k)$ for all k;
- compute $p(Y_k = y_k | X_k = x_k)$ for all k through a known function l_k such that $l_k(y;x) \propto p(Y_k = y | X_k = x)$

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 $k-1$ the weighted particle set $S_{k-1} = (s^n_{k-1}, q^n_{k-1})_{n=1,\ldots\ldots,N}$ where the *a posteriori* marginal L_{k-1} is then estimated by the probability density $L_{S_{k-1}} = \sum_{n=1}^{N} q^n_{k-1} \delta_{s^n_{k-1}}$.

The *prediction step* consists of propagating each particle of S_{k-1} according to the evolution of equation (6). The weight of each particle is updated during the *correction step*. Up to a constant, equation (10) comes down to adjust the weight of predictions by multiplying it by the likelihood $p(y_k | x_k)$. In the most general setting of sequential Monte Carlo methods [4, 7], 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_k} 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 [16] for instance, the measurement vector Y_k 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 [7] and guidance [6]. The weighted sum of Dirac laws is then approximated by a Gaussian mixture obtained by a clustering method [20]. The particle sets enable one to estimate any functional of X_k 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 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 weights. The stochastic resampling consists of sampling N particles

- Initialization: $\begin{cases} s_0^n \sim p(X_0) \\ q_0^n = \dfrac{1}{N} \end{cases}, n = 1,\ldots,N$
- for $k = 1,\ldots,T_{end}$:
  - Proposal: sample $\widetilde{s}_k^n$ from $f(X_k | X_{k-1} = s_{k-1}^n, Y_k = y_k)$ for $n = 1,\ldots,N$
  - Weighting: $\begin{cases} \text{compute un} - \text{normalized weight :} \\ \qquad \widetilde{q}_k^n = q_{k-1}^n \dfrac{p(\widetilde{s}_k^n | s_{k-1}^n) l_k(y_k; \widetilde{s}_k^n)}{f(\widetilde{s}_k^n | s_{k-1}^n, y_k)} \quad \text{for } n = 1,\ldots\ldots,N \\ \text{normalize weights :} \\ \qquad q_k^n = \dfrac{\widetilde{q}_k^n}{\sum_{n=1}^N \widetilde{q}_k^n} \quad \text{for } n = 1,\ldots.,N \end{cases}$
  - Return $\widehat{\mathrm{E}}g(X_k) = \sum_{n=1}^N q_k^n g(\widetilde{s}_k^n)$.
  - Calculate $\hat{N}_{eff} = \frac{1}{\sum_{n=1}^N (q_k^n)^2}$.
  - Resampling: if $N_{eff} < N_{threshold}$:

$$\begin{cases} s_k^n \sim \sum_{d=1}^N q_k^d \delta_{\widetilde{s}_k^d} \\ q_k^n = 1/N \end{cases} n = 1,\ldots,N \text{ else } s_k^n = \widetilde{s}_k^n \text{ for } n = 1,\ldots.N.$$

Figure 2. Particle filter with adaptive resampling.

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 [8] for example, consists of applying the resampling step at each time period. To measure the degeneracy of the algorithm, the effective sample size N_{eff} has been introduced in [14, 15]. We can estimate this quantity by $\hat{N}_{eff} = 1/\sum_{n=1}^N (q_k^n)^2$ which measures the number of meaningful particles. As advocated in [4], the resampling can be done only if $\hat{N}_{eff} < N_{threshold}$.

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 particle filter with adaptive resampling is described in Fig. 2. details can be found in [4, 14, 15].

4. Single target tracking: application to aircraft tracking with bearings-only measurements

In this section we perform some simulations to evaluate the particle filtering algorithm [1]. To illustrate this algorithm, we deal with bearings-only problems. The target is then a 'point-target in the X-Y plane. Two kinematics models were used to track this target: A constant velocity model for rectilinear motion and a constant speed turn model for curvilinear motion.

a. Rectilinear motion

For rectilinear motion, the initial position of target is $X_0 = [1500\ 10\ 1500\ 5]'$. The dynamic noise is a normal zero-mean Gaussian vector with $\sigma_x = \sigma_y = 0.005\text{ms}^{-2}$. The bearings measurements are simulated with Gaussian noises of standard deviation $\sigma_w = 0.02\text{rad}$ (about 1.5 deg), every discrete time period, i.e. every 5s. The measurements set used of the target are presented in Fig. 3. We have used the bootstrap filters, i.e. the importance function f is in fact the prior law $p(X_k|X_{k-1})$, with adaptive reasampling. The initialization of the filter has been done according to a Gaussian law whose mean vector and covariance matrix are $X_0 = [1600\ 11\ 1200\ 3\ ]'$ and

$$X_{\text{cov}} = \begin{pmatrix} 5.10^{+2} & 0 & 0 & 0 \\ 0 & 10^{-3} & 0 & 0 \\ 0 & 0 & 5.10^{+2} & 0 \\ 0 & 0 & 0 & 10^{-3} \end{pmatrix}$$

The obtained results by the application of particle filtering algorithm with 500 particles are plotted in Fig. 4, 5 and 6. In this algorithm, we have used the Root Mean Square Error (RMSE) as the performance measurement of this algorithm.

b. Curvilinear motion

The initial position of target is $X_0 = [2000\ 10\ 1500\ 5\ 0.00223]'$. The dynamic noise of this motion is the same of the first, the bearings measurements are also simulated with Gaussian noises of standard deviation $\sigma_w = 0.02\text{rad}$. The measurements set used of the target are presented in Fig. 7.

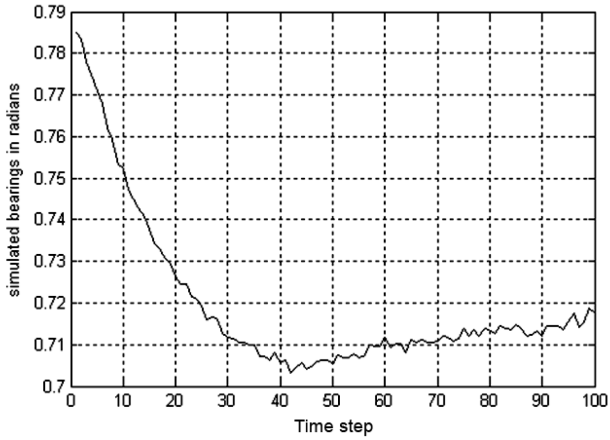

Figure 3. Simulated bearings.

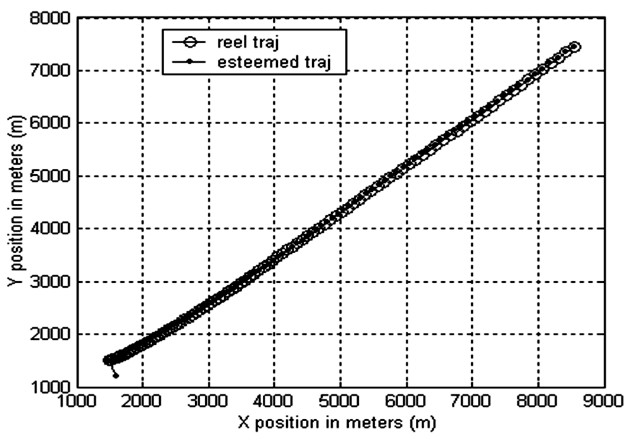

Figure 4. Real and estimated trajectory.

The mean vector and covariance matrix initialization are

$$X_0 = \begin{pmatrix} 2100 \\ 10 \\ 1400 \\ 5 \\ 0.0523 \end{pmatrix} \quad X_{\text{cov}} = \begin{pmatrix} 2.10^{+2} & 0 & 0 & 0 & 0 \\ 0 & 5.10^{-4} & 0 & 0 & 0 \\ 0 & 0 & 2.10^{+2} & 0 & 0 \\ 0 & 0 & 0 & 5.10^{-4} & 0 \\ 0 & 0 & 0 & 0 & 5.10^{-4} \end{pmatrix}$$

Fig. 4. and 8 show that the estimated and the real trajectories are superposable and almost identical. Fig. 5, 6, 9, 10, and 11, present the evolution over time of the root

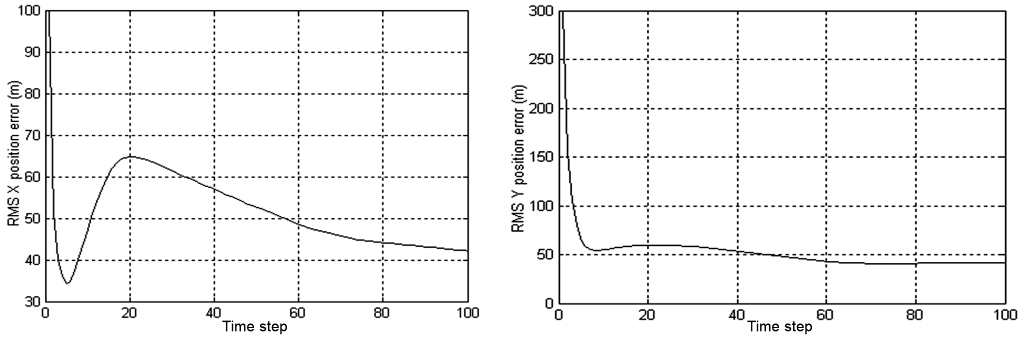

Figure 5. RMS position error (X and Y coordinates).

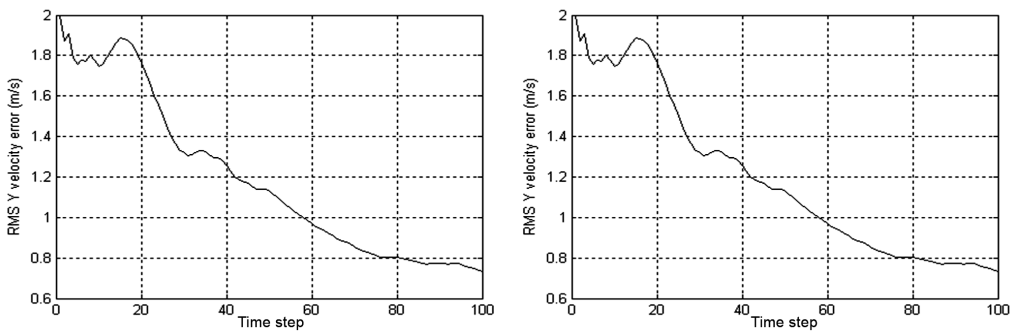

Figure 6. RMS velocity error (X and Y coordinates).

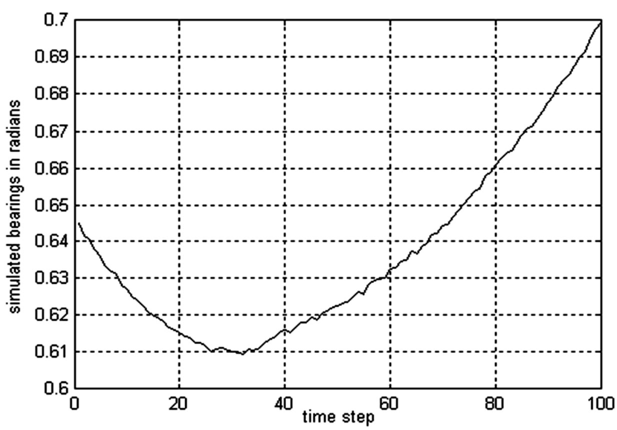

Figure 7. Simulated bearings.

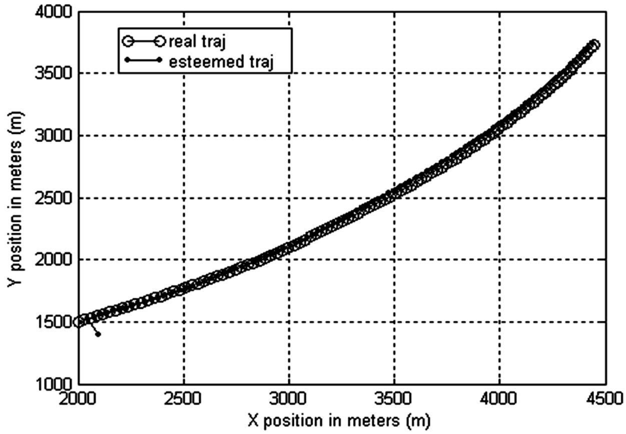

Figure 8. Real and estimated trajectory.

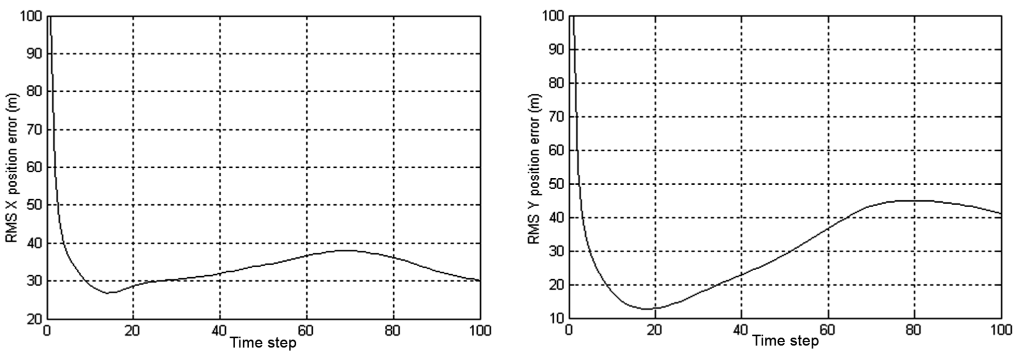

Figure 9. RMS position error (X and Y coordinates).

mean square position and velocity errors of rectilinear and curvilinear motions, and like it could be seen the algorithm track accurately the targets dynamic and converge to the right trajectory. From this, we can say that the tracker track accurately targets whose state and/or measurement are nonlinear.

5. Multiple targets tracking

a. Notations

Let r be the number of targets model. The state vector that we have to estimate is made by concatenating the state vector of each target model, at time k, $X_k = \left[X_k^1, \ldots, X_k^r\right]$

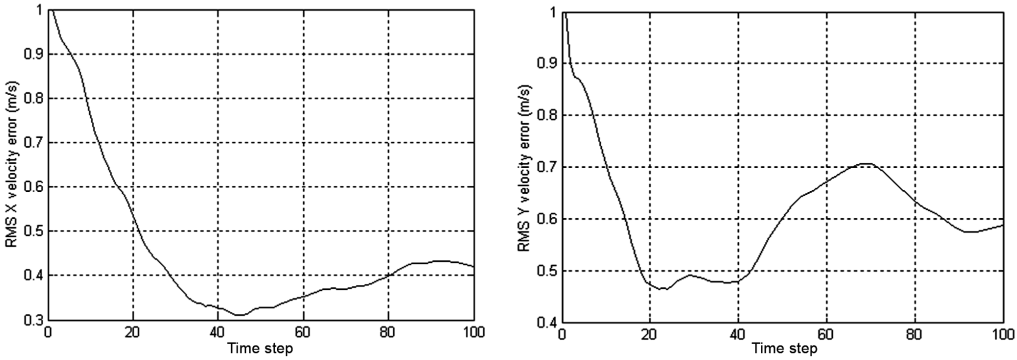

Figure 10. RMS velocity error (X and Y coordinates).

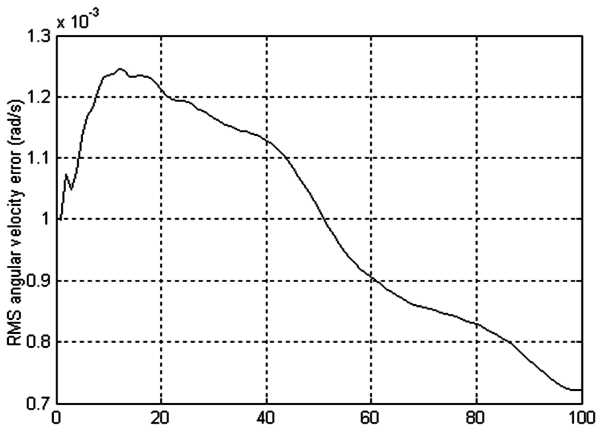

Figure 11. RMS angular velocity error (rad/s).

follows the state equation (1) decomposed in r partial equations [2, 3]:

$$X_k^i = F_k^i \left(X_{k-1}^i, V_k^i \right) \qquad \forall i = 1, \ldots, r. \tag{11}$$

The noise V_k^i and $V_k^{i'}$ are only supposed to be white both temporally and spatially, independent for $i \neq i'$. The observation vector at time k is denoted by $y_k = (y_k^1, \ldots, y_k^{m_k})$, where y_k^j is a realization of the stochastic process:

$$Y_k^j = H_k^i(X_k^i, W_k^j). \tag{12}$$

With m_k a number of the available measurements at one discrete time k, this later can be different than r (number of targets model). Again, the noises W_k^j and $W_k^{j'}$ are only supposed to be white noises, independent for $j \neq j'$.

b. Multiple model approach

In the multiple model approach (MM) [21, 17] it is assumed that the system obeys one of a finite number of models. A Bayesian framework is used: starting with prior probabilities of each model being correct (i.e., the system is in a particular mode), the corresponding posterior probabilities are obtained.

First, it will be assumed that the model that the system obeys is fixed, i.e. no switching from one mode to another occurs during the estimation process (time-invariant mode). The model assumed, is one of r possible models [18] (the system is in one of r modes).

$$M \in \{M_i\}_{i=1}^{r}$$

The prior probability that M_i is correct (the system is in mode i) is

$$P\{M_i|Y_0\} = \mu_i(0) \qquad i = 1,\ldots,r \tag{13}$$

where Y_0 is the prior information and $\sum_{i=1}^{r} \mu_i(0) = 1$ since the correct model is among the assumed r possible models.

1. Calculation of model probabilities

The event M_i is defined to represent the condition that dynamic model i is in force. No time argument is required, as the model is assumed not to switch with time. The posterior probability that model i is in force conditioned on the measurement history up to k is represented by:

$$\mu_i(k) \stackrel{\Delta}{=} P\{M_i|Y_k\}. \tag{14}$$

Expanding Y^k in equation (14) onto the combination of the previous measurement history Y^{k-1} combined with the current $y(k)$, and then using Bayes formula in both $y(k)$ and M_i yields [21]:

$$\mu_i(k) = P\{M_i|Y_k, y(k)\} = \frac{p\{M_i, y(k)|Y_{k-1}\}}{p\{y(k)|Y_{k-1}\}} = \frac{p\{y(k)|M_i, Y_{k-1}\}\ P\{M_i|Y_{k-1}\}}{p\{y(k)|Y_{k-1}\}}. \tag{15}$$

Denominator in equation (15) can be expanded using the total probability expansion over all models:

$$p\{y(k)|Y_{k-1}\} = \sum_{i=1}^{r} p\{y(k)|M_i, Y_{k-1}\} P\{M_i|Y_{k-1}\}$$

where r is the number of hypothesized models (and thus the number of elementary filters in the structure). This gives the following recursive equation for the model probabilities $\mu_i(k)$

$$\mu_i(k) = \frac{p\{y(k)|M_i, Y_{k-1}\}\ \mu_i(k-1)}{\sum_{i=1}^{r} p\{y(k)|M_i, Y_{k-1}\}\ \mu_i(k-1)} \qquad i = 1,\ldots,r \tag{16}$$

starting with the given prior probabilities in equation (13).

The first term on the right hand side above is the likelihood function of mode i at time k, which, under Gaussian assumptions, is given by the expression [17]:

$$\Lambda_i(k) \stackrel{\Delta}{=} p\left[y(k)\,|Y_{k-1}, M_i\right] = p\left[\nu_i(k)\right] = \mathrm{N}\left[\nu_i(k);\, 0;\, S_i(k)\right] \tag{17}$$

where ν_i and S_i are the innovation and its covariance from the mode matched filter corresponding to mode i.

2. Calculation of combined estimate

The output of each mode-matched filter is the mode-conditioned state estimate $\hat{X}^i$, the associated covariance P^i and the mode likelihood function Λ_i. After the filters are initialized, they run recursively on their own estimate. Their likelihood functions are used to update the mode probabilities. The central conditional mean estimate is formed as a weighted average of the elemental filter estimates using the model probabilities $\mu_i(k)$ as the weights:

$$\hat{X}(k|k) = \sum_{i=1}^{r} \mu_i(k)\hat{X}^i(k|k) \tag{18}$$

Though generally not required, the covariance of this estimate can also be formed using a weighted average, but adding the correction term which takes into account the spreading introduced by different estimates:

$$P(k|k) = \sum_{i=1}^{r} \mu_i(k)\left\{P^i(k|k) + \left[\hat{X}^i(k|k) - \hat{X}(k|k)\right]\left[\hat{X}^i(k|k) - \hat{X}(k|k)\right]'\right\} \tag{19}$$

The multiple model approach with a bank of filters is given in Fig. 12.

c. Nonlinear multiple model particle filters (NMMPF) algorithm

To avoid the problem of multiple targets tracking we present our algorithm called Nonlinear Multiple Model Particle Filters (NMMPF), the basic idea of this algorithm is to combine the multiple model approach with particle filtering. It consist of using a bank of filters based on different motion models covering the range of possible expected observed motions, and to some how combine the estimates from these filters based on the expectation of each model being the correct descriptor of the object motion. It assumes that the update system state is a linear combination of each filter in the filters bank weighted by a probability factor. A general description of NMMPF algorithm is presented in Fig. 13.

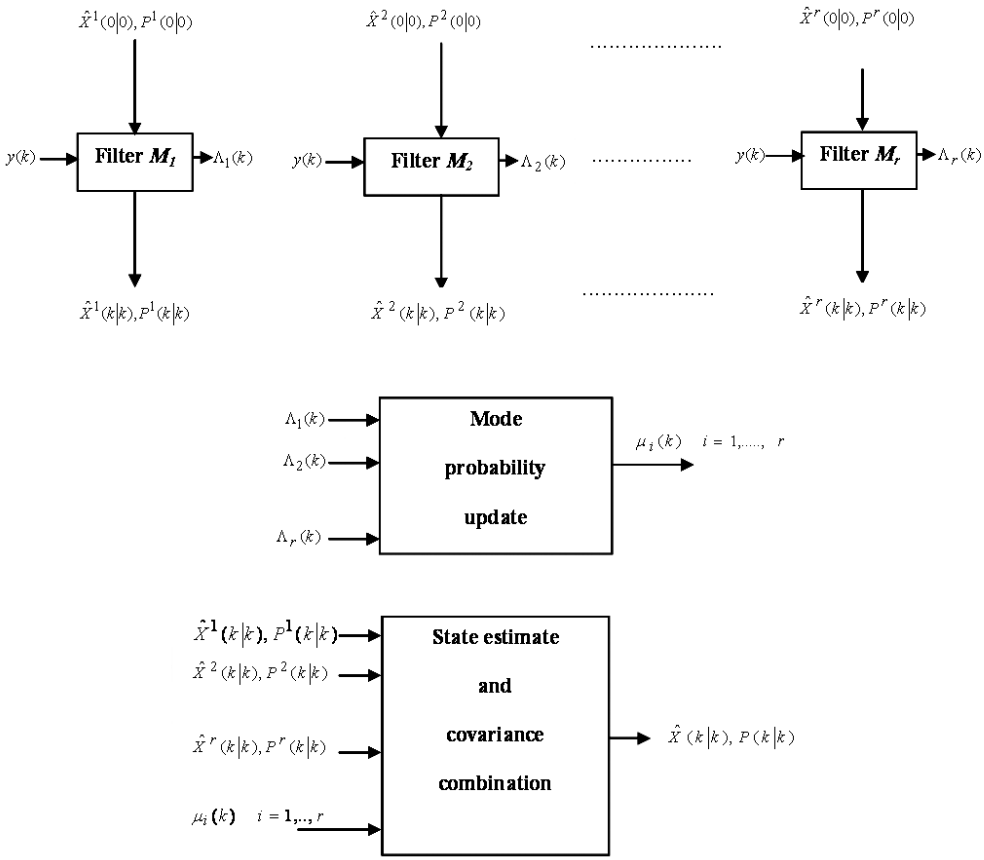

Figure 12. The multiple model approach for r fixed models.

6. Simulation and results: Application to bearings-only problems

To illustrate and explore the capability of our NMMPF algorithm to track multiple targets, we consider the following multiple targets scenario. Three targets follow:

1. A nearly constant velocity model defined in equation (1) (without acceleration noise).
2. A nearly constant velocity model defined in equation (1) (with acceleration noise).
3. A nearly coordinated turn model defined in equation (4).

◇ Models state and covariance initialization: X_0^i, P_0^i, $i = 1, \ldots, r$
◇ Initialization of models probability: $\mu_i(0) = \frac{1}{r}, \quad i = 1, \ldots, r$
◇ For $k = 1, \ldots, end$

▷ Model-matched filtering (using a bank of particle filtering)
For $i = 1, \ldots, r$

∘ Initialization: $\begin{cases} s_0^{n,i} = p(X_0^i) \\ q_0^n = 1/N \end{cases} \quad n = 1, \ldots, N$

∘ For $j = 1, \ldots, m_k$

- Proposal: sample
$(s_k^{n,\,i})$ from $f(X_k^i | X_{k-1}^i = s_{k-1}^{n,i}, Y_k^j = y_k^j)$, $n = 1, \ldots, N$
- Weighting:

$$\begin{cases} \text{compute un normalized weights :} \\ (\widetilde{q}_k^n)_j = (q_{k-1}^n)_j \dfrac{p(s_k^{n,i} | s_{k-1}^{n,i}) l_k(y_k^j; s_k^{n,i})}{f(s_k^{n,i} | s_{k-1}^{n,i}, y_k^j)} \quad \text{for } n = 1, \ldots, N \\ \text{normalize weights :} \\ (q_k^n)_j = \dfrac{(\widetilde{q}_k^n)_j}{\sum_{n=1}^N (\widetilde{q}_k^n)_j} \quad \text{for } n = 1, \ldots., N \end{cases}$$

- Return $\hat{E}g(X_k^i)_j = \sum_{n=1}^N (q_k^n)_j g((s_k^{n,i})_j)$
- Calculate $N_{eff}^j = 1 / \sum\limits_{n=1}^{N} (q_k^n)_j{}^2$
- Resampling if $N_{eff}^j < N_{threshold}$

▷ Model probability update

∘ For $j = 1, \ldots, m_k$

$$\Lambda_i^j(k) \overset{\Delta}{=} p\left[y^j(k) \middle| Y_{k-1}^j, M_i\right] = p\left[\nu_i^j(k)\right] = \mathrm{N}\left[\nu_i^j(k);\, 0;\, S_i^j(k)\right] \quad i = 1, \ldots., r$$

$$(\mu_i(k))_j = \frac{\Lambda_i^j(\mu_i(k-1))_j}{\sum\limits_{i=1}^r \Lambda_i^j(\mu_i(k-1))_j} \qquad i = 1, \ldots\ldots., r$$

▷ Estimate and covariance combination

∘ For $j = 1, \ldots, m_k$

$$\hat{X}_j(k|k) = \sum_{i=1}^r (\mu_i(k))_j (\hat{X}^i(k|k))_j$$

$$P_j(k|k) = \sum_{i=1}^r (\mu_i(k))_j \left\{ (P^i(k|k))_j + \left[(\hat{X}^i(k|k))_j - \hat{X}_j(k|k) \right] \right.$$

$$\left. \cdot \left[(\hat{X}^i(k|k))_j - \hat{X}_j(k|k) \right]' \right\}$$

Figure 13. NMMPF: nonlinear multiple model particle filters algorithm with adaptive resampling.

With $\sigma_X = \sigma_Y = 0.05\text{m/s}^2$ and the initial positions and velocities of the targets are the following:

$$X_0^1 = \begin{bmatrix} 3000 \\ 10 \\ 500 \\ 5 \end{bmatrix}, \quad X_0^2 = \begin{bmatrix} 500 \\ 10 \\ 3000 \\ 5 \end{bmatrix}, \quad X_0^3 = \begin{bmatrix} 2000 \\ 10 \\ 1500 \\ 5 \\ 0.00223 \end{bmatrix}.$$

The trajectories of the three targets are plotted in Fig. 14. We assume that each target produce one measurement at each time period $T = 4\text{s}$ according to equation (5) with $\sigma_w = 0.002\text{rad}$. The simulated bearings are plotted in Fig. 15.

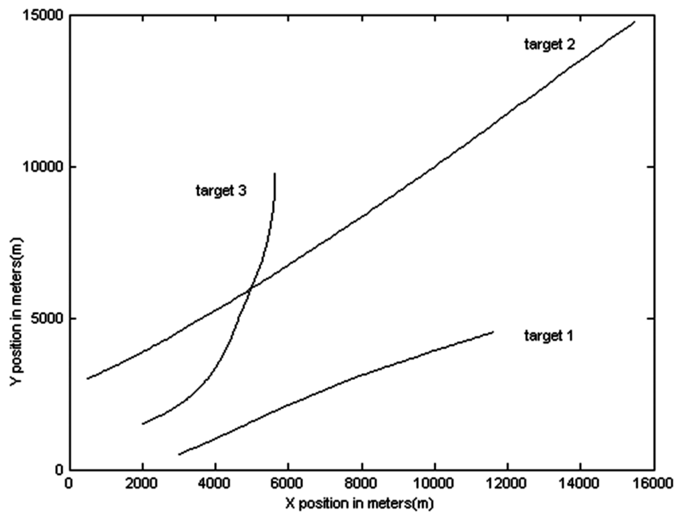

Figure 14. Targets trajectories.

So, in this case we need three filters bank. The initialization of these filters has been done according to a Gaussian law whose mean vectors and covariance matrixes are:

$$(X_0^1)_{mean} = \begin{bmatrix} 2800 \\ 11 \\ 600 \\ 4 \end{bmatrix}, \quad (X_0^2)_{mean} = \begin{bmatrix} 450 \\ 9 \\ 3200 \\ 4.5 \end{bmatrix}, \quad (X_0^3)_{mean} = \begin{bmatrix} 2100 \\ 11 \\ 1400 \\ 4 \\ 0.00323 \end{bmatrix},$$

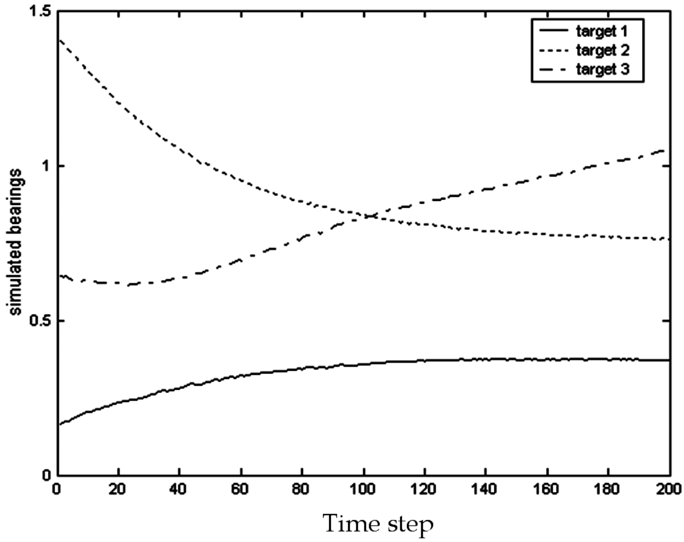

Figure 15. Simulated bearings in radian.

$$P_0^1 = P_0^2 = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0.01 \end{bmatrix}, \quad P_0^3 = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0 & 0.0001 \end{bmatrix}.$$

The initial probabilities of the three target models for each measurement are:

$$\mu_{1,i} = 1/3 \qquad \mu_{2,i} = 1/3 \qquad \mu_{3,i} = 1/3 \qquad i = 1,\ 2,\ 3.$$

It follows from the above, that at the start all models have the same chance to be selected for each measurement.

To evaluate the performance of the algorithm, we have performed 100 different Monte Carlo runs of the NMMPF with $N = 500$ particles and adaptive resampling. The resampling threshold in the particle filtering has been fixed to $N_{threshold} = 0.8$. We have chosen the root mean square error (RMSE) as the measure of the performance of this algorithm.

Fig. 16. shows that the estimated and the real trajectories are superposable and almost identical. Fig. 17, 18 and 19. show that the NMMPF algorithm needs not more than one sample to affect each measurement to the correct model. Fig. 20, 21 and 22 present the evolution over time of the root mean square errors of positions (X and Y coordinates), velocities (X and Y coordinates) and angular velocity. It can be seen that the algorithm tracks accurately the targets dynamics and confirms the result plotted in fig 16. From this we can conclude that the tracker converges and affects each measurement to the correct

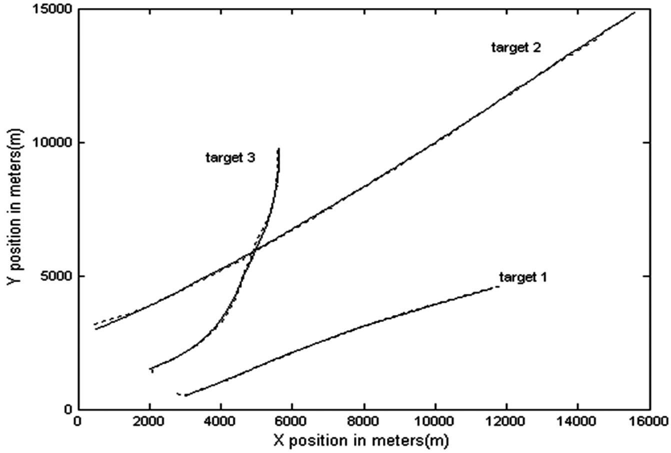

Figure 16. Real and estimated trajectories.

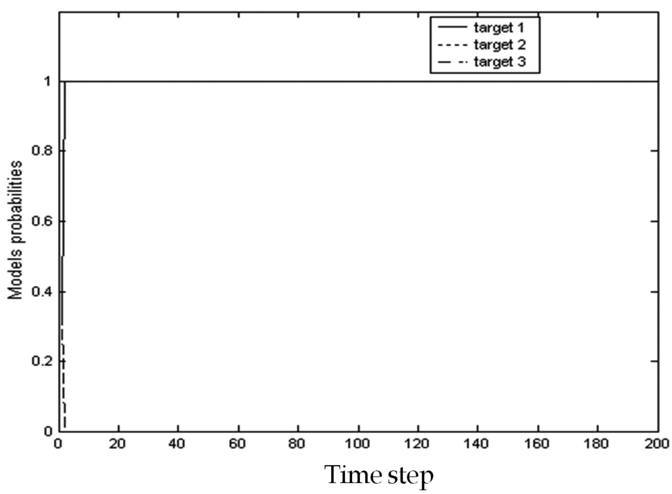

Figure 17. Models probabilities for the first measurement.

model while maintaining good tracking performance and that the NMMPF is a pertinent solution to the multiple targets tracking problem whose state and/or measurement models are nonlinear and non Gaussian noised.

When we execute the NMMPF algorithm with a Pentium IV, 3.40 GHz, $N = 500$ particles, it takes around 150 ms per time step to compute the NMMPF estimate of the three targets with bearings-only measurements.

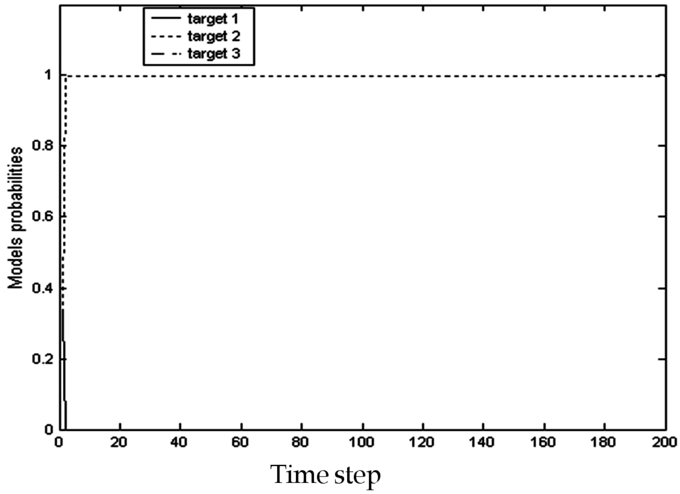

Figure 18. Models probabilities for the second measurement.

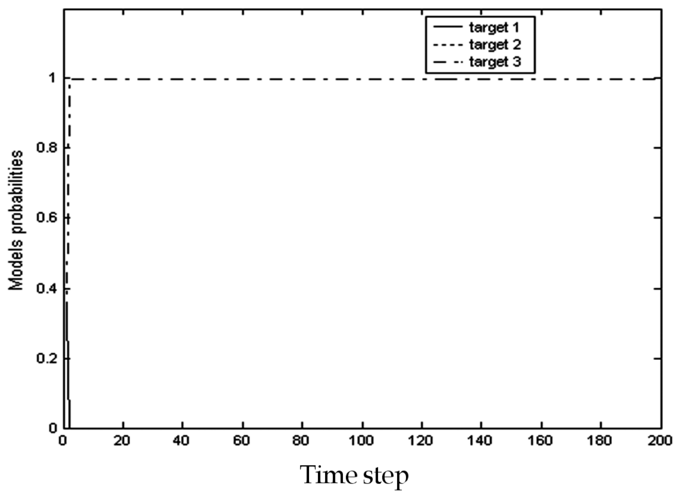

Figure 19. Models probabilities for the third measurement.

7. Conclusion

The multiple targets tracking problem is still an open problem to which we try to provide a contribution. In this paper we presented a nonlinear algorithm (NMMPF) in the framework of multiple model approach and particle filtering methods, which attempts to track efficiently a multiple targets under the assumptions of nonlinear state and/or measurement models and non Gaussian noises. Target state vectors are estimated with

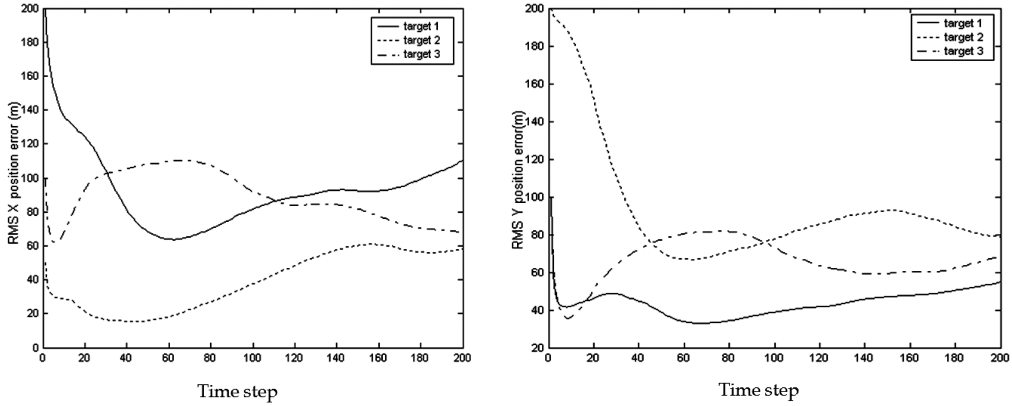

Figure 20. RMS positions errors (X and Y coordinates).

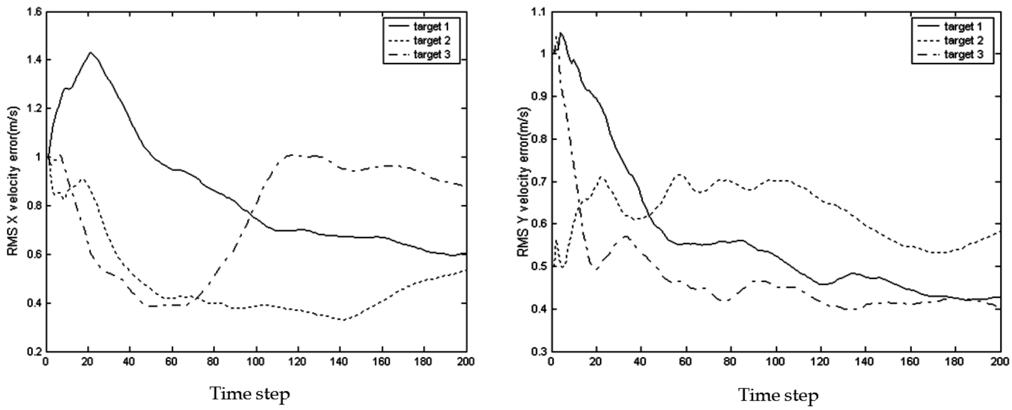

Figure 21. RMS velocities errors (X and Y coordinates).

particle filtering and the measurements affectation is done by the multiple model approach.

Among the multiple targets tracking algorithms, the joint probabilistic data association filter (JPDAF) and the probabilistic multiple hypotheses tracker (PMHT) intend to be the leaders, when the process and measurement models are linear, due to its optimality. Kalman filter is usually used in these cases. However, if the process and/or measurement models are nonlinear, the extended Kalman filter (nonlinear Kalman filter version) is no longer optimal and presents several drawbacks, among them, the linearization of the Jacobian matrix which can lead to unstable filter. To overcome these limitations we substitute the extended Kalman filter by the particle filter and we combine it with the multiple model approach. Monte Carlo simulations over the used scenario show that not

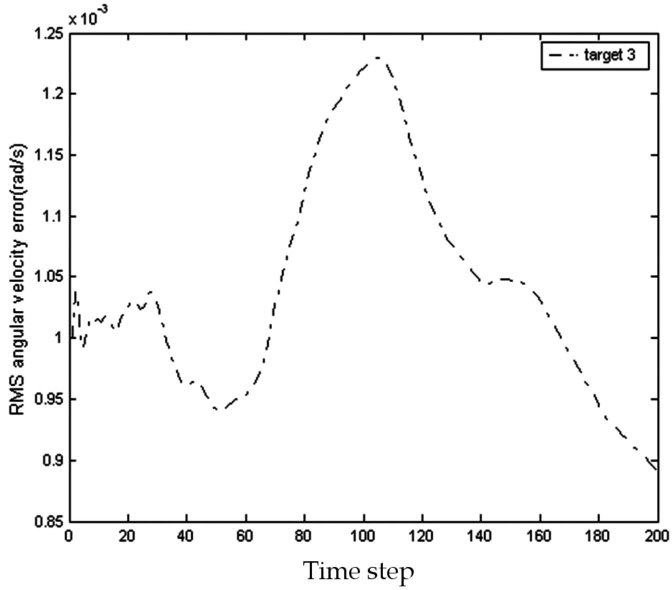

Figure 22. RMS angular velocity error (rad/s).

only our algorithm converges to the rights trajectories but also affects each measurement to the right target.

References

[1] A. SEBBAGH and H. TEBBIKH: Particle filtering for air craft tracking with bearings-only measurement. *Int. Conf. on Systems and Information Processing*, ICSIP'09, (2009), Algeria.

[2] C. HUE, J.P. LE CADRE and P. PÉREZ: Tracking multiple objects with particle filtering. *IEEE Trans. on Aerospace and Electronic Systems*, **38**(3), (2002), 791-810.

[3] C. HUE, J-P. LE CADRE and P. PÉREZ: Sequential Monte Carlo methods for multiple target tracking and data fusion. *IEEE Trans. on Signal Processing*, **50**(2), (2002), 309-325.

[4] D. ARNAUD: 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.

[5] D. ARNAUD: Convergence of sequential Monte Carlo methods. Technical report, CUED/F-INFENG/TR 381, Signal processing group, Department of Engineering, University of Cambridge, 2000.

[6] D. ARNAUD, N. DE FREITAS and N. GORDON: Sequential Monte Carlo Methods in Practice. New York: Springer, 2001, 525-532.

[7] N. GORDON: A hybrid bootstrap filter for target tracking in clutter. *IEEE Trans. on Aerospace and Electronic Systems*, **33**(1), (1997), 353-358.

[8] N. GORDON, D. SALMOND and A. SMITH: Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings, Pt. F, Radar and Signal Processing*, **140**(2), (1993), 107-113.

[9] H.J. KUSHNER: Approximations to optimal nonlinear filtering. *IEEE Trans. on Automatic Control*, **AC-12**(5), (1967), 546-556.

[10] H.J. KUSHNER: Dynamical equations for optimum nonlinear filtering. *J. of Differential Equations*, **3** (1967), 179-190.

[11] H. WANG, T. KIRUBARAJAN and Y. BAR-SHALOM: Precision large scale air traffic surveillance using an IMM estimator with assignment. *IEEE Trans. on Aerospace and Electronic Systems*, **AES-35**(1), (1999), 255-266.

[12] M. ISARD and A. BLAKE: CONDENSATION-Conditional density propagation for visual tracking. *Int. J. of Computer Vision*, **29**(1), (1998), 5-28.

[13] J.K. UHHMANN: Algorithms for multiple target tracking. *American Scientist*, **80**(2), (1992), 128-141.

[14] A. KONG, J.S. LIU and H.W. WONG: Sequential imputation method and Bayesian missing data problems. *J. of American Statistical Association*, **89** (1994), 278-288.

[15] J.S. LIU: Metropolized independent sampling with comparison to rejection sampling and importance sampling. *Statistics and Computing*, **6** (1996), 113-119.

[16] J. MACCORMICK: Probabilistic modelling and stochastic algorithms for visual localisation and tracking. Ph.D. dissertation, University of Oxford, Jan., 2000.

[17] M.S. DJOUADI, A. SEBBAGH and D. BERKANI: IMM based UKF and IMM base EKF algorithms for tracking highly manoeuverable target. *Archives of Control Sciences*, **15**(1), (2005), 19-51.

[18] M.S. DJOUADI, A. SEBBAGH and D. BERKANI: A nonlinear algorithm for maneuvering target visual-based tracking. *IEEE Proc. of the Second Int. Conf. on Intelligent Sensing and Information Proceeding*, Chennai, India, (2005), 61-66.

[19] R. VAN DER MERWE, A. DOUCET, N. DE FREITAS and E. WAN: The unscented particle filter. Technical report, CUED/F-INFENG/TR 380, Cambridge University Engineering Department, 2000.

[20] D.J. SALMOND: Mixture reduction algorithms for target tracking in clutter. *SPIE Signal and Data Processing of Small Targets*, **1305** (1990), 434-445.

[21] Y. BAR-SHALOM, X.R. LI and T. KIRUBARAJAN: Estimation with applications to tracking and navigation. USA, 2001.