Parallel Particle Filters for Multiple Target Tracking

Sebbagh Abdennour and Tebbikh Hicham
Departement of Electrитеchnic and Automatic, May 08 1945 Guelma University, Algeria

Abstract: The Multiple Targets Tracking (MTT) problem is addressed in signal and image processing. When the state and measurement models are linear, we can find several algorithms that yield good performances in MTT problem, among them, the Multiple Hypotheses Tracker (MHT) and the Joint Probabilistic Data Association Filter (JPDAF). However, if the state and measurement models are nonlinear, these algorithms break down. In this paper we propose a method based on particle filters bank, where the objective is to make a contribution for estimating the trajectories of several targets using only bearings measurements. The main idea of this algorithm is to combine the Multiple Model approach (MM) with Sequential Monte Carlo methods (SMC). The result from this combination is a Nonlinear Multiple Model Particle Filters algorithm (NMMPF) able to estimate the trajectories of multiple targets.

Keywords: MM approach, MTT, particle filtering.

1. Introduction

Multiple Targets Tracking (MTT) problem has received a wide attention in literature [3, 5, 6]. It is a special kind of dynamic state estimation problem. To perform MTT the observer has at his disposal some measurements collected from different sensors. In fact, the estimation of the states in MTT problem requires the assignment of the measurements to multiple targets. Generally, MTT presents two basic problems: the estimation and data association.

Many problems in science require estimation of the state of a system that changes over time using a sequence of noisy measurements made on the system. The most widely used estimator for nonlinear systems is the Extended Kalman Filter (EKF). However, in the case of a highly nonlinearities, EKF suffers from some limitations that make it inapplicable.

In order to accurately estimate states or measurements models of nonlinear targets with non-Gaussian noises, it is necessary to substitute the EKF by a nonlinear one. This nonlinear filter is called Sequential Monte Carlo (SMC) methods or particle filtering methods. The basic idea of this filter is to propagate a weighted set of particles to estimate the Probability Density Function (PDF) of the state conditioned on the observations, and updating particles weights using Bayes formula. More details are found in [19, 21]. SMC methods have been used in very different areas for Bayesian filtering and can be applied under very general hypotheses, under different names: the bootstrap filter for target tracking [12] and the condensation algorithm in computer vision [15] are two examples among others.

In the case of linear state and measurement models, we can find several algorithms that yield good performances in MTT problem, among them, the Multiple Hypotheses Tracker (MHT) [5] and the Joint Probabilistic Data Association Filter (JPDAF) [1, 6]. However, if the state and measurement models are nonlinear, these algorithms break down. In this paper, another algorithm based on particle filter bank for MTT is proposed and is called Nonlinear Multiple Model Particle Filters algorithm (NMMPF) [19]. The main idea of this algorithm is based on running several parallel filter bank at the same time and is obtained by the combination between Multiple Model (MM) approach and SMC methods. It results from this a bank of particle filters and each filter is designed to track one target.

This work is organized as follows. In section 2, we present the mathematic formulation of an aircraft motion models in horizontal plan. In section 3, we describe the particle filter bank: Starting by SMC methods with adaptive resampling and finishing by the combination with MM approach. It results from this: A NMMPF algorithm for tracking multiple targets. Finally, in section 4 we present the simulation results obtained by the application of NMMPF algorithm in MTT problem to track three targets.

2. Problem Formulation

In Air Traffic Control (ATC), civilian aircraft has two basic modes of flight: Uniform motion and maneuver. More details can be found in [2, 4, 19, 22].

The aircraft flight in (X, Y) plane can be modeled by:

1. A constant velocity model for the uniform motion In discrete time, the constant velocity model with noise is given by [17]:

\[ \begin{align*}
\mathbf{x}_k &= \mathbf{x}_{k-1} + \mathbf{w}_k \\
\mathbf{z}_k &= \mathbf{Hx}_k + \mathbf{v}_k
\end{align*} \]
\[
X_{i+1} = \begin{pmatrix}
1 & T & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & T \\
0 & 0 & 0 & 1
\end{pmatrix} X_i + \begin{pmatrix}
0.5T^2 & 0 \\
T & 0 \\
0 & 0.5T^2 \\
0 & T
\end{pmatrix} V_i
\]

Where \( T \): Is the sampling time, \( X_i \): Is the state vector including the position and velocity of the aircraft, defined as:
\[
X = [\xi \; \dot{\xi} \; \eta \; \dot{\eta}]^T
\]

With \( \xi \) and \( \eta \) denoting the orthogonal coordinates of the horizontal plane, and \( V_i \) is a zero-mean gaussian white noise representing the accelerations, with an appropriate covariance \( Q = \begin{pmatrix}
\sigma^2_\xi & 0 \\
0 & \sigma^2_\eta
\end{pmatrix} \).

2. A nearly constant-speed turn model for a maneuver
The turn of a civilian aircraft is characterized by the nearly constant-speed turn [2, 17]. The CT model is a nonlinear one if the turn rate is not a known constant. By adding the turn rate \( \Omega \) to the state vector \( X = [\xi \; \dot{\xi} \; \eta \; \dot{\eta} \; \Omega]^T \), results from this a new state vector defined as follows:
\[
X = [\xi \; \dot{\xi} \; \eta \; \dot{\eta} \; \Omega]^T
\]
The nearly constant-speed turn is defined as follows [2, 4, 17]:
\[
x_{i+1} = \begin{pmatrix}
\frac{\sin \Omega T}{\Omega} & 0 & \frac{1-\cos \Omega T}{\Omega} & 0 & 0 \\
0 & \cos \Omega T & 0 & -\sin \Omega T & 0 \\
0 & 1-\cos \Omega T & \sin \Omega T & 0 & 0 \\
\sin \Omega T & 0 & \cos \Omega T & 0 & 0 \\
0 & 0 & 0 & 0 & 1
\end{pmatrix} x_i + \begin{pmatrix}
0.5T^2 & 0 \\
T & 0 \\
0 & 0.5T^2 \\
0 & T \\
0 & 0
\end{pmatrix} v_i
\]

Note that, \( V_k \) in Equation 1 and \( V_k \) in Equation 4 has not the same dimension. The equation of observations is given according to:
\[
Y_k = \arctan \left( \frac{\eta}{\xi} \right) + W_k
\]

Where \((X, Y)\): Is the position of the target and \( W_k \) is gaussian noise with covariance \( \sigma^2_y \).

3. Particle Filters Bank
3.1. General Framework
We suppose that \( r \) is the number of targets. The state vector that we have to estimate is made by concatenating the state vector of each target model at time \( k \), \( X_i = [X_1^i, ..., X_r^i] \) follows the state Equation 8 decomposed in \( r \) partial equations [10, 13, 14]:
\[
X_{i+1} = F_{i} \left(X_{i+1,i}^i, V_i^i \right) \quad i = 1, ..., r
\]

\( (V_i^j) \) and \( (V_i^j) \) are supposed white noises, independent for \( i \neq i' \). The measurement vector at time \( k \) is denoted by \( y_i = (y_1^i, ..., y_{n_i}^i) \) and given by:
\[
Y_i = H_i(X_i^i, W_i^i)
\]

\( m_k \) is the available measurements at time \( k \), with \( m_k \) can be different to \( r \). Again, the noises \( (W_i^j) \) and \( (W_i^j) \) are only supposed to be white noises, independent for \( j \neq j' \).

3.2. SMC Methods
The dynamic system is represented by the stochastic process \((X_t) \in \mathbb{R}^n\) whose evolution is given by the state equation [10, 13, 20, 21]:
\[
X_k = F_k(X_{k-1}, V_k)
\]
The objective is to estimate the state vector \((X_k)\) at discrete times via the measurement equation:
\[
Y_k = H_k(X_k, W_k)
\]

Where the functions \( F_k \) and \( H_k \) are not assumed linear and the processes \((V_k) \in \mathbb{R}^{n_v}\) and \((W_k) \in \mathbb{R}^{n_w}\) are assumed independent white noises.

The original particle filter, which is called the bootstrap filter, see [8, 14, 20], estimates the densities obtained by sampling from an appropriate density \( f \). These two steps called Sequential Importance Sampling (SIS) which is given in Algorithm 1. The degeneracy is the principal drawback of the SIS algorithm; most of particles become dispose of very small weights and the others keep high weights. To remedy this drawback, we need another step to eliminate the particles of smaller weights; this latter is called the resampling step.

**Algorithm 1: SIS Algorithm.**

**Initialisation:** \[ s_{n_0}^k = p(X_0) \quad a_{n_0}^k = 1/N \quad n = 1, ..., N. \]

**For** \( k = 1, ..., T_{end} \):

**Proposal:** Sample \( s_n^k \) from

\[
f(X_k|X_{k-1} = s_n^{k-1}, Y_k = y_k) \quad n = 1, ..., N.
\]
\[ \mu_i(k) = \frac{p(y(k) | M_i, Y_{k-1})}{\sum_{j=1}^{r} p(y(k) | M_j, Y_{k-1})} \]

Or

\[ \mu_i(k) = \frac{p(y(k) | M_i, Y_{k-1}) \mu_i(k-1)}{\sum_{j=1}^{r} p(y(k) | M_j, Y_{k-1}) \mu_j(k-1)} \]

Which \( p(y(k) | M_i, Y_{k-1}) \) is the likelihood function, this latter is obtained under gaussian assumptions by:

\[ A_i(k) = p(y(k) | Y_{k-1}, M_i) = N(\gamma_i(k); 0, S_i(k)) \]

Where \( \gamma_i \) and \( S_i \) are the innovation and its covariance. More details in [2, 5, 18].

### 3.3.2. Combined Estimate

We find in the output of each filter the state estimate \( \hat{X}^i \), the covariance \( P^i \) and the likelihood function \( A_i \). After the initialization, the filters run recursively in the same time [2, 25].

The models probabilities \( \mu_i(k) \) are updated using their likelihood functions \( A_i \).

These latters are used to obtain the mean estimate from the elemental estimates:

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

The associated covariance is also obtained by using a weighted average:

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

MM approach with \( r \) filters bank is given in Figure 1.

---

**3.3. MM Approach**

In this approach the system is one of \( r \) models [2, 7, 19], it starts with the prior probabilities of each model to obtain the corresponding posterior probabilities.

The prior probability that \( M_i \) is in mode \( i \) is:

\[ P(M_i | Y_{0}) = \mu_i(0) \]

Where \( Y_0 \) is the prior information and \( \sum_{i=1}^{r} \mu_i(0) = 1 \).

### 3.3.1. Model Probabilities

The posterior probability that the model \( i \) is being correct can be calculated recursively as [2] for more details:

\[ \mu_i(k) = \frac{p(Y_k | M_i) \alpha_i}{\sum_{j=1}^{r} p(Y_k | M_j) \alpha_j} \]

Where \( Y_k \) is the prior information and \( \sum_{i=1}^{r} \mu_i(0) = 1 \).
3.4. NMMPF Algorithm

To study accurately the MTT problem, we present the proposed algorithm called NMMPF algorithm [19].

The combination between MM approach and SMC methods is its basic idea. It consists of running at the same time \( r \) particle filters based on different motion models. For each measurement, we combine the states of the \( r \) particle filters weighted by a probability factor of each filter to obtain an aggregate state estimate for one target. Algorithm 3 presents a general description of NMMPF algorithm for multiple targets.

**Algorithm 3:** NMMPF Algorithm for MTT.

State and covariance initialization filters:
\[
\begin{pmatrix} X_0^1 \ \cdots \ \cdots \ X_0^r \end{pmatrix} \ 	ext{for} \ i = 1, \ldots, r
\]

Initialization of models probabilities:
\[
\text{For} \ j = 1, \ldots, m_0 \ \text{then} \ \mu_j^0(0) = 1/r \ \text{for} \ i = 1, \ldots, r
\]

Model-matched filtering (using a bank of particle filtering)
\[
\begin{align*}
&\text{For} \ i = 1, \ldots, r \\
&\text{Initialization:} \ \left\{ \begin{array}{l}
\hat{x}_{0}^i = p(X_0^i) \ \text{for} \ n = 1, \ldots, N \\
\alpha_0^i = \frac{1}{N}
\end{array} \right. \\
&\text{For} \ j = 1, \ldots, m_0 \\
&f(X_i^j | X_{i-1}^j = x_{i-1}^j, Y_i^j = y_i^j) \ \text{for} \ n = 1, \ldots, N \\
&\text{Weighting:}
\end{align*}
\]

\[
\begin{align*}
&\text{compute un-normalized weights:} \\
&\tilde{\omega}_{n,i}^j = \frac{p(s_i^j | y_i^j, \hat{x}_{i-1}^j, \hat{y}_{i-1}^j)}{f(s_i^j | y_i^j, \hat{x}_{i-1}^j, \hat{y}_{i-1}^j)} \ \text{for} \ n = 1, \ldots, N \\
&\text{normalize weights:}
\end{align*}
\]

\[
\omega_n^j = \frac{\tilde{\omega}_{n,i}^j}{\sum_{i=1}^{r} \tilde{\omega}_{n,i}^j} \ \text{for} \ n = 1, \ldots, N.
\]

Return \( E(g(X_i^j)) = \sum_{n=1}^{N} \omega_n^j g(X_i^j) \)

Calculate \( N_{\text{eff}} = 1 / \sum_{n=1}^{N} (\omega_n^j)^2 \)

Resampling if \( N_{\text{eff}} < N_{\text{threshold}} \)

Model probability update
\[
\begin{align*}
\mu_n^j(k) &= \frac{\Lambda_n^j(k) \mu_n^j(k-1)}{\sum_{i=1}^{r} \Lambda_n^j(k) \mu_i^j(k-1)} \ \text{for} \ i = 1, \ldots, r
\end{align*}
\]

State estimate and covariance combination
\[
\begin{align*}
&\hat{X}^j(\hat{k}) = \sum_{i=1}^{r} \mu_i^j(\hat{k}) \hat{X}^j_i(\hat{k}) \\
&P_{j}(\hat{k}) = \sum_{i=1}^{r} \mu_i^j(\hat{k}) \left[ P^j_i(\hat{k}) + \left[ \hat{X}^j_i(\hat{k}) - \hat{X}^j(\hat{k}) \right] \left[ \hat{X}^j(\hat{k}) - \hat{X}^j_i(\hat{k}) \right]^{T} \right]
\end{align*}
\]

The combination between MM approach and particle filtering methods requires the calculation of additional conditional probabilities \( p(M(k) = M_j | \hat{y}_{k-1}) \) which allow an adequate mixing of the modal estimates to produce an aggregate state and covariance estimates.

4. Simulation Results

In order to evaluate the effectiveness of NMMPF algorithm in MTT, a simulated example is presented in this section using the Matlab simulation. We consider the following scenario where three targets follow:

- **Target 1:** A constant velocity model without acceleration noise.
- **Target 2:** A constant velocity model with acceleration noise.
- **Target 3:** A nearly constant-speed turn model or (CT model).

Initial states vectors that contain positions and velocities of the three targets are:
\[
(X^1_0) = \begin{pmatrix} 3000 \\ 10 \\ 500 \end{pmatrix} \quad (X^2_0) = \begin{pmatrix} 500 \\ 10 \\ 3000 \end{pmatrix} \quad (X^3_0) = \begin{pmatrix} 2000 \\ 10 \\ 1500 \end{pmatrix}
\]

With \( \sigma^2_x = \sigma^2_y = 0.05 m^2/s^2 \).

We suppose that each measurement is related to one target according to Equation 5 at each time \( T = 4 \) s with \( \sigma_{xy} = 0.02 \) rad (about 1.5 degree). The Targets trajectories and their bearings are plotted in Figures 2 and 3.
particle filters, their mean vectors and covariance matrixes initialization according to a gaussian law are:

\[
(X^1)_{\text{meas}} = \begin{pmatrix} 2800 \\ 11 \\ 600 \\ 4 \end{pmatrix}, \quad (X^2)_{\text{meas}} = \begin{pmatrix} 450 \\ 9 \\ 3200 \\ 4.5 \end{pmatrix}, \quad (X^3)_{\text{meas}} = \begin{pmatrix} 2100 \\ 11 \\ 1400 \\ 4 \end{pmatrix}, \quad \text{and} \quad (X^4)_{\text{meas}} = \begin{pmatrix} 0.00323 \end{pmatrix}
\]

\[
P_0^1 = P_0^2 = \begin{pmatrix} 100 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0.01 \end{pmatrix}, \quad P_0^3 = \begin{pmatrix} 100 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0.0001 \end{pmatrix}
\]

For each measurement, the initial probabilities of the three particle filters are:

\[
\mu_i^1 = 1/3; \quad \mu_i^2 = 1/3; \quad \mu_i^3 = 1/3; \quad i = 1, 2, 3
\]

That is to say: The three particle filters models have the same chance to be chosen in the beginning.

The application of the NMMPF algorithm in this example has been done with:

- 100 Monte Carlo runs.
- Particles number \( N = 500 \).
- Resampling threshold \( N_{\text{threshold}} = 0.8N \).

In order to measure the accuracy of NMMPF algorithm in tracking, we have chosen the Root Mean Square Error (RMSE) as the performance measure of this algorithm.

Figure 4. Targets trajectories (real and esteemed).

From Figure 4 it appears that the esteemed trajectories are indistinguishable from the real ones.

The power of NMMPF algorithm appears in the assignment of the measurements to their right targets, it needs to one or two samples to assign each measurement to the right target and these results are plotted in Figure 5-a, b and c.

\[
(X, Y) \text{ position RMSE, } (X, Y) \text{ velocity RMSE and angular velocity RMSE are plotted in Figure 6-a and b, Figures 7-a and b and 8. These smallest RMSE demonstrate and confirm that the NMMPF algorithm is a good estimator for MTT problem and prove the results plotted in Figure 4.}
\]

Figure 5. Models probabilities.

a) First measurement.

b) Second measurement.

c) Third measurement.

Figure 6. RMS positions errors.

a) X Position error.

b) Y Position error.

Figure 7. RMS velocities errors.

a) X Velocity error.

b) Y Velocity error.

c) Angular velocity error.
Figures 9 and 10 present the real and the esteemed trajectories of the three targets for $N=1000$ particles and $N=2000$ particles respectively.

![Figure 9: Targets trajectories (real and esteemed) for $N=1000$ particles.](image)

![Figure 10: Targets trajectories (real and esteemed) for $N=2000$ particles.](image)

Generally, when we increase the particles number $N$, the estimation error decreases, but the time calculation for each iteration or time step $k$ increases. Table 1 presents the necessary time for one iteration when we execute the NMMPF algorithm by PC Pentium IV, 3.40GHz with $N=500$, $N=1000$ and $N=2000$.

![Time calculation for one iteration of NMMPF Algorithm.](image)

Table 1. Time calculation for one iteration of NMMPF Algorithm.

<table>
<thead>
<tr>
<th>Particles Number $N$</th>
<th>500</th>
<th>1000</th>
<th>2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time of One Iteration of NMMPF Algorithm in Milliseconds (ms)</td>
<td>420 ms</td>
<td>945 ms</td>
<td>2300ms</td>
</tr>
</tbody>
</table>

A biased initialisation by the mean vectors:

\[
\begin{pmatrix}
X_1 & \mu_{X_1} \\
X_2 & \mu_{X_2} \\
X_3 & \mu_{X_3}
\end{pmatrix} =
\begin{pmatrix}
3500 & 11 \\
1000 & 9 \\
2500 & 4.5
\end{pmatrix}
\]

Gives us the results plotted in Figure 11.

![Figure 11: Targets trajectories (real and esteemed) with biased initialisation, for $N=500$ particles.](image)

We can say that the initialisation is still a main problem for particle filters and all suboptimal filters. From these results we can say that while maintaining good tracking performance, the NMMPF algorithm is a pertinent solution to nonlinear MTT problem. The RMSE formula is given by the Equation 18.

\[
RMSE(X_j) = \sqrt{\frac{\sum_{k=1}^{200} (X_{real}(k) - X_{esteemed}(k))^2}{200}}
\]

5. Conclusions

Under the combination technique between MM approach and SMC methods, we have presented in this work a NMMPF algorithm for an efficient tracking of three targets with nonlinear state and/or measurement models and non gaussian noises. The SMC methods estimate the targets states vectors, where the MM approach plays the assignment task of measurements to their targets models.

In the case of linear MTT, the JPDAF and the PMHT estimators are the leaders because of the Kalman filter optimality. However, if the process and/or measurement models are nonlinear, the EKF is no longer optimal and presents several drawbacks. We substitute the EKF by the SMC methods to overcome these drawbacks and to combine them with MM
approach. The results simulations of the used example demonstrate that the NMMPF algorithm tracks accurately the targets trajectories and also assigns each measurement to the right target model.

References

Sebbagh Abdennour is a doctoral candidate in electrotechnic and automatic department of 08 May 1945 Guelma University, Algeria. He received his automatic engineer degree from the University of Annaba in 2000 and his Magister degree from the Polytechnic military school (EMP) of Algiers, Algeria in 2004. His research interests include automatic control and tracking targets.

Tebbikh Hicham is a professor in automatic and signal processing of 08 may 1945 Guelma University, Algeria. He received his electronic engineer degree from the University of Annaba in 1984 and his doctorate degree from the I.N.P of Grenoble, France in 1988. His research interests include modeling and fractional control systems, modeling and control of hybrid dynamic systems and automatic pattern recognition. He was in charge of several scientific and administrative responsibilities including Chair of the Scientific Council of the Faculty of Science and Technolog. He also participated, as a member on several national science and technology committees including the Sector Standing Committee (CSP) of the MESRS from 2009 to 2012, the Central Committee of the new technologies of information and communication of the MESRS from 2002. Currently he is a director of automatic and computing laboratory of guelma (LAIG).