Target Tracking Using Interacting Multiple Models with Particle Filtering

Alberto Mariano Corral[1] and Bruno Cernuschi Frias[2]

[1] Universidad de Buenos Aires, Facultad de Ingeniería,
Paseo Colon 850, Buenos Aires, Argentina corral.alberto@gmail.com
[2] CONICET and Universidad de Buenos Aires, Facultad de Ingeniería,
Paseo Colon 850, Buenos Aires, Argentina bcf@ieee.org

Abstract. The problem of modeling accuracy in target tracking has been well studied in the past and is specially important when tracking maneuvering targets. One of the most simple and elegant ways of improving an algorithm in this sense is by using Interacting Multiple Model (IMM). IMM is a method that takes into account more than one model at the same time. This paper describes how it works and how it has been incorporated in tracking algorithms in the past, specially in the Extended Kalman Filter (EKF). We also introduce a novel way of using it with Particle Filters (PF). The original proposal found here is that we estimate the whole target state sampling particles from the Optimal Function.

1 Introduction

Tracking has been proved an important discipline in estimation framework and this can be perceived by the large amount of different algorithms proposed in the literature. Examples of applications where target tracking is extensively used are air traffic control, air defense, vehicle location, surveillance, autonomous underwater vehicles, road vehicle navigation and positioning in network-based wireless systems among others [7,15,8,14,6].

Tracking is decomposed in 2 stages, detection and estimation. This paper focuses on this latter stage. At the same time, estimation of the target's future position, or in any future value that we may be interested in estimating, has two stages: predicting of the next target state using only the past information and updating that prediction using the information available at the present time. That is why the accuracy and the way we model the target's dynamics, along with the measurement process, are so important.

Once the model is written, depending on whether it is a linear and Gaussian model or not, there are basically 3 possible types of filters: Kalman filter for the linear Gaussian situations, Extended Kalman filter or any other filter designed to adapt the linear solutions to non linear situations, and non linear filters like Particle Filter, that handle non linearities as well as non Gaussian noises.

Despite the fact that the usage of only one model is usually preferred in order to describe the target's motion, in common situations targets tend to change the way they move over time, and hence one model is not enough. In this sense, in order to build a proper and more accurate description of how the state of the tracked object evolves, a method called *Interacting Multiple Models* (IMM) is used. This method basically introduces more than one model and delivers an average of more than one filter ponderated by the likelihood of each model being the actual one by which the target moves and is measured.
Although this method has been extensively used in linear filters [14,6,8], and in non linear filters [7,17,9,15], here we propose a way to incorporate that method in Particle Filtering using the Optimal Function. That is why in section 2 we start by defining the classic Particle Filter and the estimation problem. In section 3, a brief description of IMM and how it works with Kalman filter is shown. The Particle Filtering algorithm using Interacting Multiple Models, IMM-PF is developed and explain in section 4. In section 5 this developed algorithm is compared to the IMM - Extended Kalman Filter and finally in section 6 we present this work's conclusions.

2 Problem Definition and Particle Filtering

2.1 Markov Hidden Model and Particle Filtering

Let us consider a discrete Markov Hidden Model where the Markov state process is $\mathbf{x}_{0:k} = \{\mathbf{x}_0, ..., \mathbf{x}_k\}$. This Markov process is hidden in the sense that its true value can't be known. The only way to know something about $\mathbf{x}$ is through the observations $\mathbf{z}_{0:k} = \{\mathbf{z}_0, ..., \mathbf{z}_k\}$. This model is described by an evolution or transition equation and a marginal distribution for the measurements:

$$p(\mathbf{x}_0) \tag{1}$$

$$p(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{z}_{0:k-1}) = p(\mathbf{x}_k|\mathbf{x}_{k-1}) \tag{2}$$

$$p(\mathbf{z}_k|\mathbf{x}_{0:k}, \mathbf{z}_{0:k-1}) = p(\mathbf{z}_k|\mathbf{x}_k) \tag{3}$$

Since we are working in tracking situations, $\mathbf{x}_k$ will be read as the state of the target at time k and it will contain the valuable information about the target (usually position, velocity and acceleration). The target is assume to evolve and the observation is built according to:

$$\mathbf{x}_k = f_k(\mathbf{x}_{k-1}) + v_k \tag{4}$$

$$\mathbf{z}_k = H_k\mathbf{x}_k + w_k \tag{5}$$

where v_k is the process noise and w_k the observation noise. All along this paper, we will assume that the process noise and the observation noise are mutually independent and distributed according to $v_k \sim \mathcal{N}(0, Q_k)$ and $w_k \sim \mathcal{N}(0, R_k)$.

Our aim is to estimate recursively in time any expectation of the form

$$I(f_k) = \mathbb{E}_{p(\mathbf{x}_{0:k}|\mathbf{z}_{0:k})}\left[f_k(\mathbf{x}_{0:k})\right] = \int f_k(\mathbf{x}_{0:k}) \cdot p(\mathbf{x}_{0:k}|\mathbf{z}_{0:k}) d\mathbf{x}_{0:k} \qquad (6)$$

particularly the expectation of $\mathbf{x}_k$. In order to do so, we first need to know hot to calculate or how to estimate the distribution $p(\mathbf{x}_{0:k}|\mathbf{z}_{0:k})$, the so called the a posteriori function.
Most of the times, it is impossible to calculate the integral (6) and hence it can be estimated through Monte Carlo sampling, [4]. If we assume that we can sample N particles out of $p(\mathbf{x}_{0:k}|\mathbf{z}_{0:k})$, with $i = 1, ..., N$, then (6) is estimated by $\overline{I_N}(f_k) = \frac{1}{N} \cdot \sum_{i=1}^{N} f_k(\mathbf{x}_{0:k}^{(i)})$. However, this is often as difficult as it is to calculate (6). Hence, a new function from which to sample particles is introduced, the so called importance function, $\pi(\mathbf{x}_{0:k}|\mathbf{z}_{0:k})$, and then (6) can then be written as

$$\begin{aligned} I(f_k) &= \int f_k(\mathbf{x}_{0:k}) \cdot \frac{p(\mathbf{x}_{0:k}|\mathbf{z}_{0:k})}{\pi(\mathbf{x}_{0:k}|\mathbf{z}_{0:k})} \cdot \pi(\mathbf{x}_{0:k}|\mathbf{z}_{0:k}) d\mathbf{x}_{0:k} \\ &= \int f_k(\mathbf{x}_{0:k}) \cdot w^*(\mathbf{x}_{0:k}) \cdot \pi(\mathbf{x}_{0:k}|\mathbf{z}_{0:k}) d\mathbf{x}_{0:k} \qquad (7) \end{aligned}$$

where $w^*(\mathbf{x}_{0:k}) = p(\mathbf{x}_{0:k}|\mathbf{z}_{0:k})/\pi(\mathbf{x}_{0:k}|\mathbf{z}_{0:k})$ are called the importance weights.
This time, if we sample N particles as $\mathbf{x}_{o:k}^{(i)} \sim \pi(\mathbf{x}_{0:k}|\mathbf{z}_{0:k})$, the estimation results in $\widehat{I_N^*}(f_k) = \frac{1}{N} \cdot \sum_{i=1}^{N} f_k(\mathbf{x}_{0:k}^{(i)}) \cdot w_k^{*(i)}$, where the $w_k^{*(i)}$ are is

$$\begin{aligned} w_k^{*(i)} = w^*(\mathbf{x}_{0:k}^{(i)}) &= \frac{p(\mathbf{x}_{0:k}^{(i)}|\mathbf{z}_{0:k})}{\pi(\mathbf{x}_{0:k}^{(i)}|\mathbf{z}_{0:k})} \qquad (8) \\ &= \frac{p(\mathbf{x}_{0:k}^{(i)}, \mathbf{z}_{0:k})}{\pi(\mathbf{x}_{0:k}^{(i)}|\mathbf{z}_{0:k}) \cdot p(\mathbf{z}_{0:k})} \qquad (9) \end{aligned}$$

Since $p(\mathbf{z}_{0:k})$ is a normalizing constant difficult to evaluate, we introduce the *unnormalized importance weights*, $w_k^{(i)} = p(\mathbf{z}_{0:k}) \cdot w_k^{*(i)}$, and an estimation of the importance weights, the *normalized importance weights* $\widetilde{w}_k^{(i)}$

$$\widetilde{w}_k^{(i)} = \frac{w_k^{(i)}}{\sum_{j=1}^{N} w_k^{(j)}} \qquad (10)$$

$$\widehat{w}_k^{*(i)} = N\widetilde{w}_k^{(i)} \qquad (11)$$

This way, the posterior distribution and (6) becomes estimated by

$$\widehat{I_N}(f_k) = \sum_{i=1}^{N} f_k(\mathbf{x}_{0:k}^{(i)}) \cdot \widetilde{w}_k^{(i)} \qquad (12)$$

$$\widehat{p}_N(\mathbf{x}_{0:k}|\mathbf{z}_{0:k}) = \sum_{i=1}^{N} \widetilde{w}_k^{(i)} \cdot \delta\left(\mathbf{x}_{0:k} - \mathbf{x}_{0:k}^{(i)}\right) \qquad (13)$$

Since in real situations it is highly desirable to calculate the estimations above in a recursive way, at each time $k > 1$ we would like to sample $\mathbf{x}_k^{(i)}$ and calculate $w_k^{(i)}$ only as a function of $\mathbf{x}_{k-1}^{(i)}$ and $w_{k-1}^{(i)}$.

$$w_k^{(i)} = \frac{p(\mathbf{z}_{0:k}) \cdot p(\mathbf{x}_{0:k}^{(i)}|\mathbf{z}_{0:k})}{\pi(\mathbf{x}_{0:k}^{(i)}|\mathbf{z}_{0:k})}$$
$$= \frac{p(\mathbf{x}_{0:k-1}^{(i)}, \mathbf{z}_{0:k-1}) \cdot p(\mathbf{x}_k^{(i)}, \mathbf{z}_k|\mathbf{x}_{0:k-1}^{(i)}, \mathbf{z}_{0:k-1})}{\pi(\mathbf{x}_{0:k-1}^{(i)}|\mathbf{z}_{0:k}) \cdot \pi(\mathbf{x}_k^{(i)}|\mathbf{z}_{0:k}, \mathbf{x}_{0:k-1}^{(i)})}$$
$$= \frac{p(\mathbf{x}_k^{(i)}|\mathbf{x}_{0:k-1}^{(i)}, \mathbf{z}_{0:k}) \cdot p(\mathbf{z}_k|\mathbf{x}_{0:k-1}^{(i)}, \mathbf{z}_{0:k-1})}{\pi(\mathbf{x}_{0:k-1}^{(i)}|\mathbf{z}_{0:k}) \cdot \pi(\mathbf{x}_k^{(i)}|\mathbf{z}_{0:k}, \mathbf{x}_{0:k-1}^{(i)})} \cdots$$
$$\cdots p(\mathbf{x}_{0:k-1}^{(i)}|\mathbf{z}_{0:k-1}) \cdot p(\mathbf{z}_{0:k-1}) \tag{14}$$

If the importance function is chosen such that

$$\pi(\mathbf{x}_{0:k}|\mathbf{z}_{0:k}) = \pi(\mathbf{x}_{0:k-1}|\mathbf{z}_{0:k-1}) \cdot \pi(\mathbf{x}_k|\mathbf{z}_{0:k}, \mathbf{x}_{0:k-1}) \tag{15}$$

then (14) yields

$$w_k^{(i)} = \frac{p(\mathbf{x}_{0:k-1}^{(i)}|\mathbf{z}_{0:k-1}) \cdot p(\mathbf{z}_{0:k-1}) \cdot p(\mathbf{x}_k^{(i)}|\mathbf{x}_{0:k-1}^{(i)}, \mathbf{z}_{0:k})}{\pi(\mathbf{x}_{0:k-1}^{(i)}|\mathbf{z}_{0:k-1}) \cdot \pi(\mathbf{x}_k^{(i)}|\mathbf{z}_{0:k}, \mathbf{x}_{0:k-1}^{(i)})} \cdots$$
$$\cdots \cdot p(\mathbf{z}_k|\mathbf{x}_{0:k-1}^{(i)}, \mathbf{z}_{0:k-1})$$
$$= w_{k-1}^{(i)} \cdot \frac{p(\mathbf{x}_k^{(i)}|\mathbf{x}_{0:k-1}^{(i)}, \mathbf{z}_{0:k}) \cdot p(\mathbf{z}_k|\mathbf{x}_{0:k-1}^{(i)}, \mathbf{z}_{0:k-1})}{\pi(\mathbf{x}_k^{(i)}|\mathbf{z}_{0:k}, \mathbf{x}_{0:k-1}^{(i)})} \tag{16}$$

Another notable consequence of (15) is that at time k there is no need to resample the whole trajectory $\mathbf{x}_{0:k-1}^{(i)}$ and one can simply sample $\mathbf{x}_k^{(i)}$ out of $\pi(\mathbf{x}_k^{(i)}|\mathbf{z}_{0:k}, \mathbf{x}_{0:k-1}^{(i)})$.

2.2 Degeneracy Problem

It is well known that as time flows, the variance of the particle weights starts increasing [4], [1]. This means that almost all of the weights are going to be nearly zero and hence there is a great cost on updating trajectories which will have no impact on the final estimation. The way and speed at which this variance grow depends on the importance function we choose. The ideal case would be to use $p(\mathbf{x}_{0:k}|\mathbf{z}_{0:k})$ and so we would get $\mathbb{E}_{\pi(\mathbf{x}_{0:k}|\mathbf{z}_{0:k})}(w^*(\mathbf{x}_{0:k})) = 1$ and $var_{\pi(\mathbf{x}_{0:k}|\mathbf{z}_{0:k})}(w^*(\mathbf{x}_{0:k})) = 0$.
In order to avoid this, or at least to reduce the variance, there are two different solutions proposed in the literature: resampling and/or using an importance function which minimizes it.

Resampling The basic idea behind resampling is to discard those particles with a very low importance weight and then to replace them with other more *important* particles in a random way. When to replace these particles with negligible weights is a key decision and it is usually decided through a threshold called the effective sample size N_{eff} [4]

$$N_{eff} = \frac{N}{1 + var_{\pi(\mathbf{x}_{0:k}|\mathbf{z}_{0:k})}(w^*(\mathbf{x}_{0:k}))} \leq N$$

It is obvious that the ideal situation is to have N_{eff} as close to N as possible. Using the normalized weights, one can approximate N_{eff} by

$$\widehat{N_{eff}} = \frac{1}{\sum_{i=1}^{N} (\widetilde{w}_k^{(i)})^2} \tag{18}$$

Whenever this estimation decreases bellow the fixed threshold N_{thres}, resampling should occur.

There are several *random ways* to replace this particle trajectories. In this work we use a resampling method based on the probability that a sample from a uniform distribution $\mathcal{U}(0,1)$ has of matching the normalized weights. Since the sum of these normalized weights equals to 1, the value of each weight $\widehat{w}_k^{*(i)}$ can be thought of a region that particle has between 0 and 1. A particle with a higher weight than others will be represented by a greater region in that line between 0 and 1. Because of this, a particle with higher $\widehat{w}_k^{*(i)}$ has more probability of having that sample located in its region. After such process, the expected amount of times a particles was resampled is proportional to its normalized weight. If n_i is the amount of times particle $\mathbf{x}_k^{(i)}$ was resampled at time k and the samples taking from $\mathcal{U}(0,1)$ are independent , $\mathbb{E}[n_i] = N \cdot \widehat{w}_k^{*(i)}$.

Optimal Importance Function The optimal Importance function is the function that minimizes the variance of the importance weights defined above, $var_{\pi(\mathbf{x}_{0:k}|\mathbf{z}_{0:k})}(w^*(\mathbf{x}_{0:k}))$. Recalling that $var[w_k^{(i)}] = \mathbb{E}[(w_k^{(i)})^2] - \mathbb{E}[w_k^{(i)}]^2$, it is possible to show that $\mathbb{E}[w_k^{(i)}] = w_{k-1}^{(i)} \cdot p(\mathbf{z}_k|\mathbf{x}_{0:k-1}^{(i)}, \mathbf{z}_{0:k-1})$. Then, writing $\mathbb{E}[(w_k^{(i)})^2]$ as:

$$\mathbb{E}[(w_k^{(i)})^2] = \left(w_{k-1}^{(i)}\right)^2 \cdot p(\mathbf{z}_k|\mathbf{x}_{0:k-1}^{(i)}, \mathbf{z}_{0:k-1})^2 \cdots$$
$$\cdots \int \frac{p(\mathbf{z}_k|\mathbf{x}_{0:k-1}^{(i)}, \mathbf{z}_{0:k-1})^2}{\pi(\mathbf{x}_k^{(i)}|\mathbf{z}_{0:k}, \mathbf{x}_{0:k-1}^{(i)})^2} \pi(\mathbf{x}_k^{(i)}|\mathbf{z}_{0:k}, \mathbf{x}_{0:k-1}^{(i)}) d\mathbf{x}_k^{(i)} \tag{19}$$

it is easy to see that if $\pi(\mathbf{x}_k^{(i)}|\mathbf{z}_{0:k}, \mathbf{x}_{0:k-1}^{(i)}) = p(\mathbf{x}_k^{(i)}|\mathbf{z}_{0:k}, \mathbf{x}_{0:k-1}^{(i)})$ the integral in (19) equals 1 and hence $\mathbb{E}[(w_k^{(i)})^2] = \left(w_{k-1}^{(i)}\right)^2 \cdot p(\mathbf{z}_k|\mathbf{x}_{0:k-1}^{(i)}, \mathbf{z}_{0:k-1})^2$. Replacing this and $\mathbb{E}[w_k^{(i)}]$ in the $var[w_k^{(i)}]$, it yields $var[w_k^{(i)}] = 0$, and therefore the variance is minimized.

According to the above, the optimal importance function is:

$$\pi(\mathbf{x}_k^{(i)}|\mathbf{z}_{0:k}, \mathbf{x}_{0:k-1}^{(i)}) = p(\mathbf{x}_k^{(i)}|\mathbf{z}_{0:k}, \mathbf{x}_{0:k-1}^{(i)}) \tag{20}$$

Replacing (20) in (16), the recursive form of the unnormalized importance weights becomes

$$w_k^{(i)} = w_{k-1}^{(i)} \cdot p(\mathbf{z}_k|\mathbf{x}_{0:k-1}^{(i)}, \mathbf{z}_{0:k-1}) \tag{21}$$

3 Interacting Multiple Models and IMM-Kalman Filter

3.1 IMM

One of the first steps of object tracking is to model the dynamic law of the target's motion. In real cases, it is certainly impossible to write a model that fits for every possible motion type. This is specially true for maneuvering targets. That is why baring more than one model in the tracking algorithm as an alternative to common algorithms usually yields better results. This method is called *Interacting Multiple Models*, IMM [2,3,5].

Lets suppose $r_k = j$ is the j model that describes the target's motion at time k, where j can be any of the set $\{1, 2, ...M\}$ and M is the amount of possible models. The transition process between models is described by a Markov process:

$$p(r_k = j|r_{0:k-1}) = p(r_k = j|r_{k-1}) \tag{22}$$

$$\pi_{ji} \triangleq p(r_k = j|r_{k-1} = i) \tag{23}$$

$$p(r_0) \tag{24}$$

Originally applied to linear Gaussian tracking, a Kalman Filter is developed for every model at each time, and then the output is calculated as a weighted sum of each filter output:

$$\widehat{\mathbf{x}}_k \triangleq \sum_{j=1}^{M} \widehat{\mathbf{x}}_{j,k} \cdot p(r_k = j|z_k) \tag{25}$$

where $\widehat{\mathbf{x}}_{j,k}$ is the output of each Kalman Filter written using model $r_k = j$.

One interesting application of IMM is Track Formation, [2]. This method aims to decide, based on a threshold, if a couple of measurements inside a window is considered a valid target for tracking or not. However, Track Formation is out of this work's scope.

3.2 IMM - Extended Kalman Filter

The aim of this part is to show briefly how IMM works with Extended Kalman Filter, which will be used in a later section for comparison with the IMM Particle Filtering algorithm proposed here.

The key to understand IMM-KF is that there is a separate Kalman process for each model and that are 3 stages at any given moment k:

1. Updating the likelihood of the models given the past measurements, $p(r_k = i|z_{0:k-1})$, and the marginalized distribution of $\mathbf{x}_{k-1}$ given the past measurements and the present model, $p(\mathbf{x}_{k-1}|r_k = i, z_{0:k-1})$
2. Applying a Kalman Filter per model.
3. Updating the distribution of the model given the present measurement, $p(r_k = i|z_{0:k})$.

Extended Kalman Filter, explained in [16], can be integrated with IMM in the following way. Lets assume that we know every measurement up to time $k-1$ and the past estimations $\widehat{\mathbf{x}}_{i,k-1}$ for each EKF, with $i = 1, ..., M$ where M is the number of different models. Then the first step is to write the likelihood of the actual model given the past measurements

$$p(r_k = i|\mathbf{z}_{0:k-1}) = \sum_j \pi_{ij} p\left(r_{k-1} = j|\mathbf{z}_{0:k-1}\right) \tag{26}$$

and the estimation of $\mathbf{x}_{k-1}$ given the past measurements and the present model,

$$p(\mathbf{x}_{k-1}|r_k = i, z_{0:k-1}) = \mathcal{N}\left(\bar{\mathbf{x}}_{i,k}, \bar{P}_{0,k-1}\right) \tag{27}$$

$$\bar{\mathbf{x}}_{i,k-1} \triangleq \frac{1}{p(r_k = i|\mathbf{z}_{0:k-1})} \sum_j \pi_{ij} p\left(r_{k-1} = j|\mathbf{z}_{0:k-1}\right) \cdot \widehat{\mathbf{x}}_{j,k-1} \tag{28}$$

$$\begin{aligned}\bar{P}_{i,k-1} \triangleq \frac{1}{p(r_k = i|\mathbf{z}_{0:k-1})} \sum_j \pi_{ij} p\left(r_{k-1} = j|\mathbf{z}_{0:k-1}\right) \cdots \\ \cdots \left[P_{j,k-1} + \left[\widehat{\mathbf{x}}_{j,k-1} - \bar{\mathbf{x}}_{i,k-1}\right]\left[\widehat{\mathbf{x}}_{j,k-1} - \bar{\mathbf{x}}_{i,k-1}\right]^T\right]\end{aligned} \tag{29}$$

As stated before, an Extended Kalman Filter is applied per model. The input to each filter is the pair $\left\{\bar{\mathbf{x}}_{i,k-1}, \bar{P}_{i,k-1}\right\}$ and the output is the pair $\{\widehat{\mathbf{x}}_{i,k}, P_{i,k}\}$. The reason why we do not apply the filter straight to $\widehat{\mathbf{x}}_{j,k-1}$ is because the distribution of this estimation is marginalized to the model in the previous instant $k-1$, $p(\mathbf{x}_{k-1}|r_{k-1} = i, z_{0:k-1})$, and hence we would not know which model to use to make the target evolve at time k.

Finally, given the estimation of $\mathbf{x}_k$ from each model, the last step is the update of the distribution from $p(r_k = i|\mathbf{z}_{0:k-1})$ to $p(r_k = i|\mathbf{z}_{0:k})$:

$$p(r_k = i|\mathbf{z}_{0:k}) = C \cdot \frac{p(r_k = i|\mathbf{z}_{0:k-1})}{det\left(B_{i,k}\right)^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{z}_k - \widetilde{\mathbf{z}}_{i,k})^T B_{i,k}^{-1}(\mathbf{z}_k - \widetilde{\mathbf{z}}_{i,k})} \tag{30}$$

where $B_{i,k} = H\widetilde{P}_{i,k}H^T + I$, $\widetilde{\mathbf{z}}_{i,k} = h\left(f\left(\widehat{\mathbf{x}}_{i,k-1}\right)\right)$ and C is a normalization constant so that $\sum_i p(r_k = i|\mathbf{z}_{0:k}) = 1$.

For output purposes, the final estimation of $\mathbf{x}_k$ and the covariance matrix P_k at time k are

$$\widehat{\mathbf{x}}_k = \sum_j \widehat{\mathbf{x}}_{j,k} p(r_k = j|\mathbf{z}_{0:k}) \tag{31}$$

$$P_k = \sum_j p(r_k = j|\mathbf{z}_{0:k}) \left[P_{i,k} + \left[\widehat{\mathbf{x}}_k - \widehat{\mathbf{x}}_{j,k}\right]\left[\widehat{\mathbf{x}}_k - \widehat{\mathbf{x}}_{j,k}\right]^T\right] \tag{32}$$

It is easy to see how simple this method is. However, per model that we add to the tracking, an additional filter has to be calculated every time we want an estimation of the process. This implies that as we add new models, the time it takes to estimate the target position takes more and more time. We will see in the next section how this is improved when we use the IMM Particle Filter.

4 IMM Particle Filter

In the last years, IMM has been introduced in Particle Filter algorithms, [9,10,11,12,17,15,8], and in some variations of it like Variable Rate Particle Filter (VRPF) [15]. In those papers, there is a rich diversity in which IMM was used. For instance, [8] uses IMM as it was conceived in [2] and uses KF, MPF and PF as the estimation filters. Then the outputs are ponderated with the likelihood of the model as in (25). In[15] and [17], the model is thought as part of the particle and a new model is sampled for each particle trajectory. However. in [15] the particle filter used is the bootstrap filter and in [17] Rao-Blackwellization has been used and only the model is estimated using Particle Filtering.
In this work, in order to incorporate the multiple models in the algorithm, our proposal is to have the model as part of the particle. Similar to what was done in [17], with the exception that we are going to estimate the whole particle by Particle Filtering and not just the model. Simply put, we define the new particle $\mathbf{y}_k^{(i)}$ as

$$\mathbf{x}_k^{(i)} \longrightarrow \mathbf{y}_k^{(i)} = \{\mathbf{x}_k^{(i)}, r_k^{(i)}\}$$

The estimation problem continues as defined in Section 2, but replacing $\mathbf{x}_k$ with $\mathbf{y}_k$, and hence $p(\mathbf{x}_{0:k}|\mathbf{z}_{0:k})$ with $p(\mathbf{y}_{0:k}|\mathbf{z}_{0:k})$. The target now is supposed to evolve according to:

$$\mathbf{x}_k = f_{k,r_k}(\mathbf{x}_{k-1}) + v_{k,r_k} \tag{33}$$

$$\mathbf{z}_k = H_{k,r_k}\mathbf{x}_k + w_{k,r_k} \tag{34}$$

where f_{k,r_k}, H_{k,r_k}, v_{k,r_k} and w_{k,r_k} depend now on the model we are using at time k.
As in the classic form of IMM, the model is again described by a Markov process and hence

$$\begin{aligned} p(\mathbf{y}_k|\mathbf{y}_{0:k-1}) &= p(\mathbf{x}_k|r_k, \mathbf{y}_{0:k-1}) \cdot p(r_k|\mathbf{y}_{0:k-1}) \\ &= p(\mathbf{x}_k|r_k, \mathbf{x}_{k-1}) \cdot p(r_k|r_{k-1}) \end{aligned} \tag{35}$$

Up to here, we have been able to write a Particle Filter algorithm which takes into account more than one dynamic and observation model. In the following section we will compare IMM-PF with PF and IMM-EKF.

5 Simulation and Results

The performance of the IMM-PF is evaluated in the following simulations and the results are shown in the figures bellow. Two comparisons are provided: one comparing PF with IMM-PF and the second comparing IMM-PF with the IMM-EKF introduced in section 3.

IMM-PF compared to PF. The aim of the first trail of simulations is to compare classic Particle Filter with the IMM Particle Filter proposed here. Two models are used, a constant velocity model (CV) and a constant acceleration model (CA). These models can be found in [13]. The model transition probabilities are $\pi_{CV\,CV} = 0.9$, $\pi_{CA\,CV} = 0.1$, $\pi_{CA\,CA} = 0.8$ and $\pi_{CV\,CA} = 0.2$. The time interval is $T = 0.5$ and the number of particles used are 200 in each filter. The measurement noise is a zero-mean white Gaussian process with covariance matrix $R_z = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$. The acceleration increment in CA is a white Gaussian random variable with variance $\sigma^2_{acel} = 400$ in both x and y dimensions. As for the white Gaussian noise added the velocity in the CV model, we are going to simulate three different values for the variance: $\sigma^2_{vel\,1} = 1$ for the first simulation, $\sigma^2_{vel\,2} = 10$ for the second and $\sigma^2_{vel\,3} = 100$ for the third and last one. The intention is to show that as the variance of the noise increases, the effect of having 2 models is not distinguishable. A higher noise in the velocity makes CV equivalent to the CA.
In all the three figures, the IMM-PF estimation steps on the Ground Truth path. It is easy to see that only when the variance of the velocity in the CV model is big enough to disguise the difference between CV and CA, Fig. 1c, the PF estimation matches the IMM-PF and the true path.

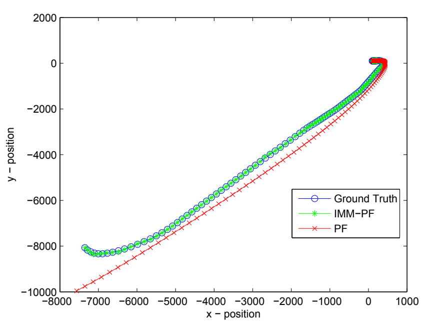
Table 1: RMSE for PF and IMM-PF depending on the CV process noise variance.

σ^2_{vel2}	PF	IMM-PF
1	1217.3	14.85
10	592.99	24.2
100	5.43	4.41

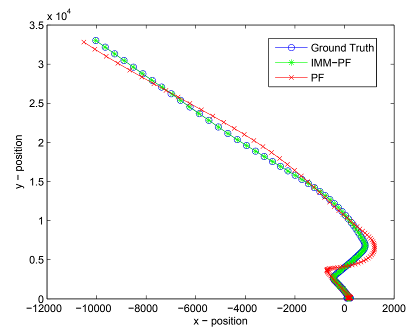
IMM-PF compared to IMM-EKF. The following simulations compare IMM-EKF explained in section 3 with IMM-PF. To achieve this we propose two different models:

$$\begin{aligned} &\text{M1:} \quad x_k = x_{k-1} + v_k \\ &\text{M2:} \quad x_k = \cos(x_{k-1}) \cdot \cos(x^2_{k-1}) + 9 + v_k \end{aligned}$$

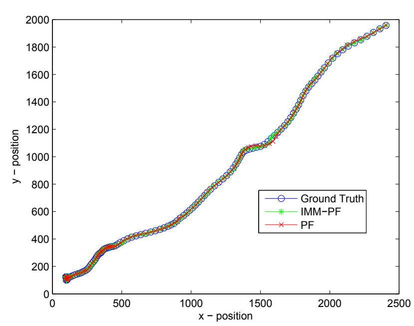
where v_k is the same white Gaussian noise for both models, with variance $\sigma^2_v = 0.1$. The measurement process is the same for both models, $z_k = x_k + w_k$ where w_k is the measurement white Gaussian noise with variance $\sigma^2_z = 0.3$. The algorithm has $n = 500$ steps and we are going to run $S = 100$ simulations three times for different values of particles, $N = 50$ particles for the first one, $N = 150$ for the second and $N = 500$ particles for the last simulation. The probabilities of each model transition are $\pi_{1\,1} = 0.9$, $\pi_{1\,2} = 0.4$, $\pi_{2\,1} = 0.1$ and $\pi_{2\,2} = 0.6$.



(a)



(b)



(c)

Fig. 1: **IMM − PF compared to PF** : Fig. (a) Simulation with $\sigma_{vel}^2 = 1$; Fig. (b) Simulation with $\sigma_{vel}^2 = 10$; Fig. (c) Simulation with $\sigma_{vel}^2 = 100$.

For comparison purposes we are going to evaluate the empirical standard deviation for the filtering estimates $\widehat{\mathbf{x}}_n$ as in [4]:

$$\sqrt{\mathrm{VAR}\left(\widehat{\mathbf{x}}_k\right)} = \frac{1}{n}\sum_{k=1}^{n}\left(\frac{1}{S}\sum_{j=1}^{S}\left(\widehat{\mathbf{x}}_k^j - \mathbf{x}_k^j\right)^2\right)^{\frac{1}{2}}$$

where $\mathbf{x}_k^j$ is the true state value at time k for simulation j and $\widehat{\mathbf{x}}_k^j$ is its estimation.

Table 2: Root Mean Square Deviation $\sqrt{\mathrm{VAR}\left(\widehat{\mathbf{x}}_k\right)}$

	IMM-EKF	IMM-PF
$N = 50$	0.2577	0.2607
$N = 150$	0.2579	0.2439
$N = 500$	0.2569	0.2378

6 Conclusions

In the first part of this paper we described the basic algorithms of Particle Filtering, Extended Kalman Filter and Interacting Multiple Models Extended Kalman Filter. In the second part we developed and proposed a new IMM Particle Filter which we also compared with the algorithms from the first part. From these simulations, it is seen that IMM-PF has less error than both PF and IMM-EKF because, as stated before, it takes into account every model, unlike PF, and deals with non linearities better than EKF. We proposed an IMM-PF algorithm that investigates the model space in the same way it does for the state space and that uses the optimal function to sample particles from, without linearizing them. Future research will also take into account multiple measurements in a Probabilistic Data Association framework.

7 Acknowledgments

The authors thank CONICET, the University of Buenos Aires, and ANPCYT, Argentina for they financial support.

References

1. M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. 50(2):174–188, 2002.
2. Yaakov Bar-Shalom and Xiao-Rong Li. *Multitarget-Multisensor Tracking: Principles and Techniques*. Yaakov Bar-Shalom, 1995.

3. H. A. P. Blom and Y. Bar-Shalom. The interacting multiple model algorithm for systems with markovian switching coefficients. 33(8):780–783, 1988.
4. Arnaud Doucet. On sequential simulation-based methods for bayesian filtering. *Tech. Report*, 1:15, 1998.
5. P. H. Foo and G. W. Ng. Combining the interacting multiple model method with particle filters for manoeuvring target tracking. *IET Radar, Sonar & Navigation*, 5(3):234–255, 2011.
6. C. Fritsche, U. Hammes, A. Klein, and A. M. Zoubir. Robust mobile terminal tracking in nlos environments using interacting multiple model algorithm. In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing ICASSP 2009*, pages 3049–3052, 2009.
7. S. A. Gadsden, S. R. Habibi, and T. Kirubarajan. A novel interacting multiple model method for nonlinear target tracking. In *Proc. 13th Conf. Information Fusion (FUSION)*, pages 1–8, 2010.
8. H. Q. Liu, M. Chitre, and Gao Rui. Auv positioning based on interactive multiple model. In *Proc. OCEANS 2010 IEEE - Sydney*, pages 1–6, 2010.
9. S. Maresca, M. Greco, F. Gini, and L. Verrazzani. Radar tracking of a move-stop-move maneuvering target in clutter. In *Proc. IEEE Radar Conf. RADAR '08*, pages 1–6, 2008.
10. S. Maresca, M. Greco, F. Gini, and L. Verrazzani. Tracking of a move-stop-move target in clutter: A comparison among mm methods. In *Proc. European Radar Conf. EuRAD 2009*, pages 433–436, 2009.
11. P. S. Maybeck, M. C. Kozak, and B. D. Smith. Mixed-model multiple-hypothesis tracking of targets in clutter. 44(4):1402–1415, 2008.
12. M. R. Morelande and S. Challa. Manoeuvring target tracking in clutter using particle filters. 41(1):252–270, 2005.
13. X. Rong Li and V. P. Jilkov. Survey of maneuvering target tracking. part i. dynamic models. 39(4):1333–1364, 2003.
14. Rafael Toledo-Moreo, Miguel A. Zamora-Izquierdo, and Antonio F. Gomez-Skarmeta. Imm-ekf based road vehicle navigation with low cost gps/ins. In *Proc. IEEE Int Multisensor Fusion and Integration for Intelligent Systems Conf*, pages 433–438, 2006.
15. Y. Ulker, B. Gunsel, and S. Kirbiz. A multiple model structure for tracking by variable rate particle filters. In *Proc. 19th Int. Conf. Pattern Recognition ICPR 2008*, pages 1–4, 2008.
16. G Welch and G. Bishop. An introduction to the kalman filter. Technical report, University of North Carolina at Chapel Hill, 2006.
17. Xiaojun Yang and Xiangmo Zhao. Mixed particle filtering for maneuvering target tracking in clutter. In *Proc. Int Computational Intelligence for Modelling Control & Automation Conf*, pages 557–562, 2008.