Abstract— In this paper, we present a filtering algorithm to perform accurate estimation in jump Markov nonlinear systems, in case of multi-target tracking. With this paper, we aim to contribute in solving the problem of model-based body motion estimation by using data coming from visual sensors. The Interacting Multiple Model (IMM) algorithm is specially designed to track accurately targets whose state and/or measurement (assumed to be linear) models changes during motion transition. However, when these models are nonlinear, the IMM algorithm must be modified in order to guarantee an accurate track. In order to deal with this problem, the IMM algorithm was combined with the Unscented Kalman Filter (UKF) [8]. Even if the later algorithm proved its efficacy in nonlinear model case; it presents a serious drawback in case of non Gaussian noise. To solve this problem we propose to substitute the UKF with the Particle Filter (PF). To solve the problem of data association, we propose the JPDA algorithm. The latter is then combined with the IMM-PF algorithm; the derived algorithm is called JPDA-IMM-PF.

Index Terms— Estimation, Kalman filtering, Particle filtering, JPDA, Multi-Target Tracking, Visual servoing, data association.

I. INTRODUCTION

This paper hope to be a contribution within the field of visual-based control of robots, especially in visual-based tracking [10]; tracking manoeuvring targets, which may themselves be robots, is a complex problem, to ensure a good track when the target switches abruptly from a motion model to another is not evident. Because of the complexity and difficulty of the problem, a simple case is considered. The study is restricted to 2-D motions of a point, whose position is given at sampling instants in terms of its Cartesian coordinates. This point may be the center of gravity of the projection of an object into a camera plane, or the result of the localisation of a mobile robot moving on a planar ground.

Several of maneuvering target tracking algorithms are developed. Among them, the interacting multiple model (IMM) method based on the optimal Kalman filter, yields good performance with efficient computation especially when the measurement and state models are linear with Gaussian noise. However, if the later are nonlinear and/or non Gaussian noise, the standard Kalman filter should be substituted, in our study we choose the Particle Filter (PF). The algorithm derived from this combination is called IMM-PF. The other problem treated in this paper, is about the data association. Effectively, at each sample time, the sensor (camera) present, several measures and observations, coming from different targets; the problem is how to affect each measure to the correct target, to solve this problem we choose the JPDA algorithm. The algorithm derived from the combination of the later approach and the nonlinear IMM algorithms is noted JPDA-IMM-PF.

The paper is organized as follows. In section II the mathematical formulation of 2-D motion is presented. In section III We describe the IMM algorithm PF based. In section IV we present the JPDA-IMM-PF algorithm. In section V we present and discuss the results of simulations. Finally in section VI we draw the conclusion.

II. MATHEMATICAL FORMULATION OF 2-D MOTION

The mathematical formulation of 2-D motion used is mainly inspired from Danes, Djouadi, and al in [4]. They make the hypothesis that the measurements are only the 2-D Cartesian coordinates of the moving point.

Let s(.) denote the curvilinear abscissa of M over time onto its trajectory, the origin of curvilinear abscissae is set arbitrarily. Functions x(.) and y(.) represent the Cartesian coordinates of M. The measurement equation may be written as:

\[
\begin{pmatrix}
    x(t) \\
    y(t)
\end{pmatrix} = h(s(t), p(t))
\]  

(1)

Where \( p(.) \) is a parameter vector function of minimal size. We can see that equation (1) is independent of the type of the motion of M onto its trajectory. The state equation could be written as:

\[
\dot{X}(t) = AX(t)
\]  

(2)

with \( X(t) = \begin{pmatrix} s(t) \\ p(t) \end{pmatrix} \)
A equals \( \begin{pmatrix} A_s & 0 \\ 0 & 0 \end{pmatrix} \), with \( A_s \) the \( n \times n \) zero matrix with ones added on its first upper diagonal, and 0 the matrices of convenient sizes. The continuous time state equation (2) is linear time invariant and independent of M’s trajectory, except on the sizes of \( \mathbf{s}^{(i)} \) and \( \mathbf{p}^{(i)} \). Moreover, it may be shown that the fundamental matrix \( \mathbf{F} \) involved its exact discretization at the period \( T \) takes the form

\[
\mathbf{F} = \exp(\mathbf{A}T) = \sum_{i=0}^{n-1} \frac{(\mathbf{A}T)^i}{i!}.
\]

The dynamic and measurement noises are supposed to be stationary, white and Gaussian, non inter-correlated with known covariances.

A. Canonical motion equations

The point M is supposed to move on straight or circular trajectories at constant or uniformly time-varying speed (constant speed or constant acceleration). Those motions belong to the set of the possible behaviours of a non-holonomic robot whose wheels are driven at constant velocities or accelerations.

1) Output equations: One minimal description of a straight line is defined by the vector \( \mathbf{p} = (\alpha, d)^T \) shown in figure 1(a), which is related to Plucker coordinates. Concerning a circular trajectory one minimal description is defined by the vector \( \mathbf{p} = (R_0, x_0, y_0)^T \) shown in figure 1(b). The origin of curvilinear abscissa is uniquely defined from those parameterizations.

The output equations are as follows (trajectory parameter are considered time-invariant):

**Straight Line:**

\[
z(k) = \begin{bmatrix} x(k) \\ y(k) \end{bmatrix} = \begin{bmatrix} d \cos \alpha + s(k) \sin \alpha \\ d \sin \alpha - s(k) \cos \alpha \end{bmatrix} + \nu(k) \tag{3}
\]

**Circle:**

\[
z(k) = \begin{bmatrix} x(k) \\ y(k) \end{bmatrix} = \begin{bmatrix} x_0 + R_0 \cos s(k) \\ y_0 + R_0 \sin s(k) \end{bmatrix} + \nu(k) \tag{4}
\]

with \( s(k) \) distance covered by the target and \( \nu(\cdot) \) zero mean Gaussian noise, \( E[\nu(k)] = 0, E[\nu(k) \cdot \nu^T(k)] = R \delta_{\nu,\nu} \).

2) State Equations

**Constant velocity**

\[
\left\{ \begin{array}{l}
\mathbf{s}(k+1) = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \mathbf{s}(k) + w_s(k) \\
\mathbf{p}(k+1) = \mathbf{p}(k) + w_p(k)
\end{array} \right. \tag{5}
\]

**Constant acceleration**

\[
\left\{ \begin{array}{l}
\mathbf{s}(k+1) = \begin{bmatrix} 1 & \frac{T^2}{2} \\ 0 & 1 \end{bmatrix} \mathbf{s}(k) + w_s(k) \\
\mathbf{p}(k+1) = \mathbf{p}(k) + w_p(k)
\end{array} \right. \tag{6}
\]

where the random vectors \( \mathbf{x}(0) = [\mathbf{s}(0)^T, \mathbf{p}(0)^T]^T \), \( \mathbf{w}(\cdot) = [w_s(\cdot)^T, w_p(\cdot)^T]^T \) and \( \nu(\cdot) \) are jointly Gaussian and:

\[
E[\mathbf{x}(0)] = \tilde{\mathbf{x}}(0), \quad E[\mathbf{w}(k)] = 0, E[\nu(k)] = 0,
\]

\[
E: \begin{bmatrix} \mathbf{x}(0) - \tilde{\mathbf{x}}(0) \\ \mathbf{w}(k) \\ \mathbf{v}(k) \end{bmatrix} = \text{diag}[\mathbf{P}(0), Q \delta_{x,x}, R \delta_{x,x}]
\]

\[
\mathbf{z}(k) = \begin{bmatrix} \mathbf{s}(k) \\ \dot{s}(k) \\ \ddot{s}(k) \\ \mathbf{p}(k) \end{bmatrix} \quad \text{vector of the point M dynamics}
\]

trajectory parameters vector

**III. The IMM-Particle Filter Algorithm**

The basic idea is to combine the IMM approach [2], with a particle filter one. In the derivation of the standard IMM filter, a merging and filtering process are defined. We adopt a regularized particle filter for this filtering step, and perform
the merging step on the probability densities, represented by a Gaussian mixture. One consequence of the discrete nature of the approximation of the a posteriori density is that it cannot directly be applied to an IMM framework as it is used in [1].

To obtain a good continuous approximation of the a posteriori density, we use a regularized version of the bootstrap filter as first reported in [11,12] for tracking targets in clutter. In this hybrid version of the bootstrap filter, the probability density function, that has been computed as a point mass probability density on a number of grid points in the state space, is fitted to a continuous probability density function that is a sum of a prefixed number of Gaussian density functions. Moreover, by using a hybrid type of sampling filter as an alternative for direct resampling, degeneracy in the effective number of particles is avoided [11,12]. The main advantages of the new method that we propose here are:

- the method is able to deal with nonlinearities and non-Gaussian noise in a mode;
- the method uses a fixed number of particles in each mode, independent of the mode probability.

**Algorithm**

Let a system be described by the equations:

\[
\begin{align*}
x(k) &= f(x(k-1), k-1, M(k)) + w[k-1, M(k)] \\
z(k) &= h(x(k), M(k)) + v[k, M(k)]
\end{align*}
\]  

(7)

The process noise and the measurement noise are possibly mode-dependent. Their densities are denoted by: \(d_{w[k,M(k)]}(w)\) and \(d_{\nu[k,M(k)]}(\nu)\).

Where \(M(k)\) denotes the model at time \(k\). It’s a finite state Markov process tacking values in \(\{M_j\}_{j=1}^J\) according to a Markov transition probability matrix \(p\) assumed to be known.

The probability density of the initial state is known, \(x(0) \sim P_0(x)\). Define the information up to and including time step \(k\) as:

\[Z(k) = \{z(1), ..., z(k)\}\]

The filtering problem that has to be solved is: Given a realization of \(Z(k)\) associated with (7) compute \(p(x(k)|Z(k))\); i.e. the conditional probability density of the state \(x(k)\) given the set of measurements \(Z(k)\). A cycle of the IMM algorithm could be summarized in four steps:

- **Interaction stage**
  Compute Mixing probabilities
  \[
  \mu_{ij}(k-1/k-1) = \frac{1}{c_j} p_j \mu_i(k-1)
  \]  
  \[c_j = \sum_{i \in M} p_j \mu_i(k-1)\]  

(8)

Compute Normalizing factors

\[
\sum_{i \in M} \mu_{ij}(k-1/k-1)
\]

Compute A priori probability density in mode \(j\)

\[
p_{ij}'(x_{ij}(k-1)/Z(k-1)) = \sum_{i \in M} \hat{p}_i'(x_{ij}(k-1/Z(k-1))) \ast \mu_{ij}(k-1/k-1)
\]

(10)

- **Filtering stage**

\[\forall j \in M \text{ draw } N \text{ samples } \tilde{x}_j^{(k-1)} \text{ according to } \hat{p}_j'(x_{uj}(k-1)/Z(k-1))\]

The predicted samples are:

\[
\hat{x}_j^{(k)} = f(\tilde{x}_j^{(k-1)}, k-1, j) + w^j(k-1, j) \]

(11)

Where \(w^j(k-1, j)\) are samples obtained from \(d_{w(k-1,j)}(w)\)

The predicted output

\[
z_j^{(k/k-1)} = h(\hat{x}_j^{(k)}, k, j)
\]

(12)

The probability weight

\[
q_j^i = d_{\nu(k,j)}(z(k) - \hat{z}_j^{(k/k-1)})
\]

(13)

Normalizing

\[
\tilde{q}_j(k) = \sum_{i=1}^N \tilde{q}_j^i(k)
\]

(14)

Normalized probability masses

\[
q_j^* = \frac{\tilde{q}_j^i(k)}{\tilde{q}_j(k)}
\]

(15)

Mean of the state over the sample set

\[
\tilde{x}_j(k) = \sum_{i=1}^N q_j^i \tilde{x}_j^{(k)}
\]

(16)

Covariance of the state over the sample set

\[
\hat{P}_j(k) = \sum_{i=1}^N q_j^i (\tilde{x}_j^{(k)} - \bar{x}_j(k))(\tilde{x}_j^{(k)} - \bar{x}_j(k))^T
\]

(17)

From the conditional probability density function for the state in mode \(j\) based on a mixture of \(N\) Gaussian densities

\[
\hat{p}_j'(x_{ij}(k)/Z(k)) = \sum_{i=1}^N q_j^i N(\tilde{x}_j^{(k)}, \nu_j, \hat{P}_j(k))
\]

(18)
Where \( \nu_j = 0.5N^{-2/d_j} \), and \( d_j \) is the dimension of the state space.

We obtain the probability density function for the state in mode \( j \) after mixture reduction, i.e. based on a mixture of \( N_j \leq N \) Gaussian densities.

\[
\hat{p}^i(x_j(k)/Z(k)) = \sum_{j=1}^{N} q_j \ast N(\hat{x}_j^r(k), \nu_j, \hat{P}_j^r(k))
\]  

(19)

The mean of predicted output over the sample set

\[
\overline{H}_j(k) = \sum_{j=1}^{N} h(\hat{x}_j^r(k), k, j)
\]  

(20)

Residual covariance over the sample set

\[
\hat{S}_j(k) = \sum_{j=1}^{N} q(h(\hat{x}_j^r(k), k, j) - \overline{H}_j(k))(h(\hat{x}_j^r(k), k, j) - \overline{H}_j(k))^T
\]  

(21)

Innovations

\[
\gamma'_j(k) = z(k) - h(\hat{x}_j^r(k), k, j)
\]  

(22)

Probability density function for the innovations

\[
\hat{p}^i(\gamma'_j(k)/Z(k)) = \sum_{j=1}^{N} q_j \ast N(0, \hat{S}_j(k))
\]  

(23)

Likelihoods

\[
L'_j(k) = N(\gamma'_j(k); 0, \hat{S}_j(k))
\]  

(24)

Mode probabilities

\[
\mu_j(k) = \frac{1}{c} L'_j(k)c_j
\]  

(25)

Where

\[
c = \sum_{j \in M} L'_j(k)c_j
\]  

(26)

\( \checkmark \) Combination stage

The a posteriori conditional probability density function for the state

\[
\hat{p}(x(k)/Z(k)) = \sum_{j \in M} \hat{p}^i(x_j(k)/Z(k)) \mu_j(k)
\]  

(27)

IV. JPDA-IMM ALGORITHM

The principle of the JPDA algorithm is the computation of probabilities association for each track and new measurement. These probabilities are then used as weighting coefficients in the formation of the averaged state estimate, which is used for updating each track. For a better description of the JPDA algorithm, see \cite{1,7}.

The combination of the JPDA and the IMM-PF algorithms done as follows. A single set of validated measurements for JPDA-IMM-PF is obtained by considering the intersection \( Z_k \) of \( r \) sets of measurements corresponding to individual models:

\[
Z_k = \bigcap_{j=1}^{r} Z_k^j
\]

Where \( Z_k^j \) represents the set of validated measurements under the assumption that model \( j \) is effective. The combined likelihood functions for the \( r \) modes of the IMM-PF algorithm are computed as in \cite{12}.

The prior mixed state estimates for model \( j \) and the validation regions for individual models are also computed as in \cite{11,8}. The new mode probabilities, output state estimates, and corresponding error covariances are obtained as in \cite{11,12}.

V. SIMULATIONS AND RESULTS

In this section, we perform some simulations to evaluate our algorithm (JPDA-IMM-PF).

The motion models considered are: constant velocity on straight line (M1), constant acceleration on straight line (M2), constant velocity on circle (M3), constant acceleration on circle (M4).

To explore the capability of our JPDA-IMM-PF algorithm to track maneuvring targets, various scenarios are considered; among of them we select the typical case of three highly maneuvring targets with crossing trajectories.

We assume that the target is in a 2-D space and its position is sampled every \( T=1s \). we run the JPDA-IMM-PF with 1000 samples in each mode.

- The probability transition matrix of four models is

\[
p = \begin{bmatrix}
0.97 & 0.01 & 0.01 & 0.01 \\
0.01 & 0.97 & 0.01 & 0.01 \\
0.01 & 0.01 & 0.97 & 0.01 \\
0.01 & 0.01 & 0.01 & 0.97
\end{bmatrix}
\]

- The initial probability of selecting a model is 0.25, that’s to say, at the start all models have the same chance to be selected.

- The curvilinear abscissa \( s(.) \) remains continuous even if a trajectory jump occurs.

A. Considered scenario

We consider that we have to track simultaneously three maneuvring targets. In order to complicate the scenario, we suppose that the targets follow during there movements crossing trajectories.

a) **Target 1 (blue):**
The target starts moving according to model M3 until the 50th sample when an abrupt acceleration about 0.2 m/s² occur and still moving according to this during 50 samples (switching from model M3 to M4).

b) **Target 2 (black):**
The target starts moving according to model M1 until the 50th sample when an abrupt trajectory change occur and still moving according to this during 50 samples (switching from model M1 to M3).

c) **Target 3 (green):**
The target starts moving according to model M1 until the 50th sample when an abrupt acceleration about 0.2 m/s² occur and still moving according to this during 50 samples (switching from model M1 to M3).
B. Results interpretation:

Figure 2 shows that the estimated and the real trajectory for the three targets are superposable and almost identical even if an abrupt change occurs on the tracked target dynamic. This result is confirmed by the figures (3,4,5,6,7), from this we can say that the tracker based IMM-PF algorithm is a pertinent solution to the problem of visual-based tracking highly maneuvering targets. In the other hand figure 2 shows also that the data association is correctly done even if the trajectories cross each other. This should permit us to say that the JPDA algorithm computes perfectly and its combination with the IMM-PF algorithm would be an efficient solution to the problem of highly maneuvering multi-target visual-based tracking.

VI. CONCLUSION

The model-based body motion estimation by using data coming from visual sensors still an open problem on which we try to provide a contribution. In this paper we presented an algorithm which attempts to track efficiently a highly maneuvering target whose trajectory and/or dynamic could change abruptly and the distribution noises are not necessary Gaussian, the algorithm proposed is noted IMM-PF. To extend this algorithm to multi-target case, we combined the later with the JPDA algorithm to ensure good data association. Simulations show that the JPDA-IMM-PF is a good investment while we are asked to track a highly maneuverable targets whose measurement and/or state models present a strong nonlinearities and the noise distributions are not Gaussian, and when there different trajectories cross each other.

REFERENCES