Multi-sensor Multi-target Tracking with Robust kinematic data based Credal Classification

Samir Hachour*, François Delmotte†, David Mercier† and Éric Lefèvre†
Univ. Lille Nord de France, UArtois, EA 3926 LGI2A, Béthune, France
Emails:*samir_hachour@ens.univ-artois.fr
†firstname.lastname@univ-artois.fr

***Abstract*—Multi-target tracking using multiple sensors is an important research field in application areas of mobile systems and military applications. This paper proposes a decentralized multi-sensor, multi-target tracking and belief (credal) based classification approach, applied to maritime targets. A given number of sensors, considered as unreliable, are designed to locally predict and update targets states using Interacting Multiple Model (IMM) algorithms (one IMM for one target). Targets IMMs are updated by sequentially acquired measurements. The measurements are assigned to the targets by the means of a generalized Global Nearest Neighbor (GNN) algorithm. The generalized GNN algorithm is able to provide information on the newly detected or non-detected targets and these information is used by score functions which manage the targets appearances and disappearances. In addition to the tracking task of multiple targets, each sensor performs a local classification of each one of the targets. The unreliability of the sensors makes the local classifications weak. In this article, a global classification method is shown to improve the sensors classification performances.**

I. Introduction

Combining multiple unreliable sensors information can provide more accurate results than using single sensor [1]. This paper proposes a decentralized multiple sensors architecture which aims to correctly track and classify a set of randomly appearing and disappearing multiple maneuvering targets.

To deal with a such complex problem, each sensor performs a complete tracking solution. This solution consists on targets states estimation task, sequentially acquired measurements to known targets assignment and targets appearances and disappearances management.

Concerning the adaptive estimation of the targets states, the designed solution uses the Interacting Multiple Model (IMM) algorithm, one of the best approaches related to multi-target estimation theory [2], [3]. Spite of that, the estimation accuracy still depends on the manner that the real measurements are assigned to the known targets [3]–[5]. In our solution this issue is ensured by the means of a generalized Global Nearest Neighbor (GNN) algorithm. In literature, multiple other assignment approaches exist, namely, probabilistic ones like: Joint Probability Data Association (JPDA), Integrated Probability Data Association (IPDA) [6], [7], etc. These approaches update the different targets states with a weighted sum of all the real measurements, due to their probabilistic aspect these methods are known to have some difficulties to handle targets appearances and disappearances. Multi-Hypotheses Tracking (MHT) algorithm [3] has a different principle, it is a multi-scan approach that holds on the assignment decision until having clearer data. Considered as the best assignment approach, it is also the most computationally intensive. More recently, Markov Chain Monte Carlo (MCMC) based assignment algorithms are proposed [8], [9]. They are based on Monte Carlo sampling and their performance depends on the number of the performed samples. As well as MHT, IPDA and JPDA, MCMC methods are considered inefficient in a dense targets environment, due to their high computational complexity. The assignment algorithm adopted in this work, namely the generalized GNN, belongs to the deterministic approaches, beyond its its limits in conflicting scenarios, it is appreciated for its simplicity and low computation complexity, it is largely used in real time applications, even in a dense targets environment. In addition to this, its generalized version can provide information on the newly detected and non-detected targets. Using score functions representing targets tracks quality, these information is smartly used to handle targets appearances and disappearances.

Concerning the classification problem, in the single target framework, the work of Smets and Ristic [10] about credal classification, based on belief functions [11], reveals a capital weakness of the already existing Bayesian solutions [12], [13], namely, imprecise classification situations which are handled by the belief theory but not by the Bayesian one. The same credal classification method is extended to multi-target framework in [14]. The credal classification is highly performing when data are acquired by a reliable sensor. However, it is shown in this article that the classification performance decreases when the sensor's data are quit imprecise, this is why a multi-sensor approach is adopted. Simulation results shows that the multi-sensor approach with an adequate fusion strategy achieve a classification performance better than the local classifications achieved by all the sensors.

Section II of this article provide an introduction to multi-target tracking problem and the local tracking solution is depicted in Section III. Section IV describes the local classification algorithm and Section V proposes a high level sensors' classifications fusion. Finally, in Section VI, simulation results on multiple maritime targets are given to highlight the advantage of the adopted decentralized approach.

II. Multi-target tracking problem position

Targets in their environment can perform different maneuvers, their evolution can be seen as a switching Markov process. One possible modeling of this process consists on a finite set of linear models. Details on the derivation of the targets motion models can be found in [15]–[17]. For a given target t being in evolution model m_l, with $l = \{1, ...r\}$ and r being the number of possible known acceleration models, the state vector evolution can be given as follows:

$$x_k^t = F x_{k-1}^t + G u_k^t(m_l) + w_k^t \quad (1)$$

where, $x_k^t \in \mathbb{R}^p$ is the t^{th} target state vector at time k, with F being the $(p \times p)$ state matrix and u_k^t represents the t^{th} target deterministic input, which represents simply a known acceleration mode m_l, where $l = 1, ..., r$ and r represents the number of possible accelerations modes. The parameter w_k^t represents the state Gaussian noise with covariance matrix Q. The input matrix is denoted G. For simplicity, the measurements are taken according to a linear model given as follows:

$$z_k^j = H x_k + v_k, \quad (2)$$

where, $z_k^j \in \mathbb{R}^q$ is the j^{th} received observation at time k, with $j \in \{1, 2, ..., m\}$. The observation matrix of dimension $(q \times p)$ is noted H and v_k represents the measurement error, it is considered as a Gaussian noise with zero mean value and covariance matrix R. Note that the number of targets n is not constant over the time and a given measurement z_k^j at time k can come from a known target t, a new target t' or a clutter. The number of known targets n or/and of acquired measurements m can be different from a sensor to an other, so the number of targets known by a sensor i are noted n_i and the number of measurements acquired by the same sensor is noted m_i. For simplicity, the sensor's index is not used to describe the local tracking and classification algorithms. The set of measurements taken by the sensor i at time k is noted $Z_i = \{z_k^1, z_k^2, ..., z_k^{m_i}\}$, with $i \in \{1, 2, ..., S\}$ and S is the number of sensors. The optimal Bayesian estimation of the t^{th} target state at time k requires the calculation of the following probability density function:

$$p(x_k^t | z_{1,...,k}^t, u_k^t(m_{1,...,r})), \ i = 1, ..., n, \quad (3)$$

where, $z_{1,...,k}^t$ represents the cumulative measurement for the target t until the time k.

III. Local tracking algorithm

The Gaussian assumption of the state and measurement noises allows the probability density functions, in Equation (3), of n maneuvering targets, to be estimated by n different Interacting Multiple Model (IMM) algorithms. Each target's IMM is composed of r Kalman filters (one Kalman filter for one evolution model m_l, with $l = 1, ..., r$). A switching process between the different Kalman filters keeps track of the complex targets maneuvers. At each time step k, each IMM performs two main steps, namely: state and measurement prediction step and state update step. The prediction step of the IMMs provides a set of predicted measurements $\hat{z}^t, \ t = \{1, 2, ..., n\}$ with n the number of already known targets at time $k - 1$. In addition to this information the prediction step provides n covariance matrices P^t concerning the expected measurement prediction errors. At time step k a set of real measurements $z^j, \ j = \{1, 2, ..., m\}$ are received. They are compared to the predicted ones following the resolution of an assignment problem. The assignment solution is a generalized GNN algorithm which is performed on two main steps, namely:

- generalized assignment matrix calculation: the assignment matrix elements are normalized Euclidean distances calculated as follows:
$$d_{t,j} = (z^j - \hat{z}^t)^T (P^t)^{-1} (z^j - \hat{z}^t), \quad (4)$$
with $t = \{1, 2, ..., n\}$, $j = \{1, 2, ..., m\}$.
- Assignment matrix resolution, using the generalized Munkres algorithm [18] in such a way that: targets that have received an observation will be updated following their IMMs update process. Targets that have not received any observation are considered as non-detected. The non-assigned measurements are used to initialize new targets.

It is mentioned in [19] that the Euclidean normalized distances calculated in Equation (4) follows a χ^2 distribution with a degree of freedom q (dimension of the measurement vector). For example, a measurement j satisfying the following test is candidate to initialize a new target.

$$\text{If } d_{t,j} > T, \quad \forall\, t \in \{1, 2, ..., n\}, \quad (5)$$

where T, is a threshold drawn from the χ^2 table, basing on an *a priori* probability that the measurement corresponds to a new target and the degree of freedom q. The information about newly detection measurements and non-detected targets are smartly handled by score function in order to manage targets appearances and disappearances. Score function is a log-likelihood $L(t)$ ratio sequentially calculated for each target, it has an increasing evolution when the target is detected and a decreasing evolution when the target is not detected. The quantity $L(t)$ is compared to two thresholds T_1 and T_2, respectively, the deletion and confirmation thresholds. For example, a non-detected target is not immediately deleted, the algorithm continue to predict its trajectory (maybe the non-detection is caused by a simple occlusion). The non-detected target is left when its score function reaches the deletion threshold T_1. The use of the score function is described in [14] and detailed in [15]. The last tracking step is the IMMs update step, it uses the managed measurements to provide the targets state estimation noted $\hat{x}^t, \ t = \{1, 2, ..., n\}$.

IV. Local classification

It is considered that the IMM algorithms contain an exhaustive list of all the targets' possible evolution models. The list of models is given by:

$$M = [m_1, m_2, ..., m_r], \quad (6)$$

where r represents the total number of models.

An *a priori* knowledge on the tracked targets classes' and their behaviors allows to cluster the r different models in M. The models are clustered in such a way to define the set models belonging to each specific behavior. The set of possible behaviors can be defined by: $B = [b_1, b_2, ..., b_{nb}]$, where nb is the number of behaviors. The set of models belonging to the behavior b_i, for example, is defined by $M_{b_i} \subseteq M$, with $i = 1, ..., nb$. The number of models in M_{b_i} is noted by r_{b_i}.

Ones the different behaviors are defined, their likelihoods $l(b_i)$ are calculated basing on the different models likelihoods $\lambda(m_j)$ and probabilities $\mu(m_j)$, with $j = 1, ..., r$ which are provided from the IMMs update steps.

A. Behaviors likelihoods calculation

Calculation of the behaviors likelihoods is performed as follows:

$$l_{b_i} = \sum_{j:m_j \in M_{b_i}} \mu'_j \lambda_j \ , \quad i = 1, ..., nb \ , \tag{7}$$

with:

$$\mu'_j = \frac{\mu_j}{\sum\limits_{j:m_j \in M_i} \mu_j} \ , \quad j = 1, ..., r_{b_i} \ . \tag{8}$$

Based on the behaviors likelihoods and using the Generalized Bayesian Theorem [10], [20], a mass function on the behaviors space B can be calculated. In order to obtain the classes probabilities or pignistic probabilities, the calculated behaviors likelihoods, can either be processed by a Bayesian or credal classifiers, respectively.

B. Bayesian classifier

The following Bayesian classifier is first presented in [12], it is based on the Bayesian rule.

$$P(b_i/z_{1,...,k}) = \frac{l_{b_i}}{\sum_{j=1}^{S} l_{b_j} P(b_j|z_{1,...,k-1})} P(b_i/z_{1,...,k-1}), \tag{9}$$

where, $P(b_i/z_{1,...,k-1})$ represents the *a priori* probability of the behavior b_i.
The initial *a priori* probabilities can be taken equal to an equiprobability, then they are simply the previous values.

C. Credal classifier

This step is ensure by the Generalized Bayesian Theorem, it proceeds on two mean steps:

- Mass function calculation using the following equation:

  $$m_k(D) = \prod_{b_i \in D} l(b_i) \prod_{b_i \in \bar{D}} (1 - l(b_i)), \tag{10}$$

  where $D \subseteq B$.
- Recursive mass functions combination: the mass function m_k of time k is combined with the mass function m_{k-1} of time $k-1$, using the conjunctive combination [21]:

  $$m_k(D) = \sum_{D_1, D_2 | D_1 \cap D_2 = D} m_k(D_1) m_{k-1}(D_2), \tag{11}$$

  where the initial belief m_0 being a vacuous mass function [11].

D. Classes mass function calculation

According to relations known *a priori*, the behaviors mass function can be precisely transferred on a classes mass function, with $C = \{c_1, c_1, ..., c_{nc}\}$ is the set of possible classes and nc is the number of classes. For example, a go-fast boat's class is related to a specific behavior which is complex maneuvers, so the mass function on complex maneuvers is transferred to the go-fast boat's class. The belief is transferred from the mass function on 2^B to a mass function on 2^C using the following equation:

$$m_k^C = \bar{M} \times m_k^B , \tag{12}$$

where $\bar{M}$ is a matrix expressing the relations between behaviors and classes, it contains the conditional masses $m(A|D)$, with $A \subseteq C$ and $D \subseteq B$. This step is application depending, more details are given in the example of Section VI.

The resulting mass function m_k^C is supposed having all the available information. In order to make a decision, the mass function is simply transformed to pignistic probabilities using the following equation:

$$BetP(c_i) = \sum_{c_i \in A} \frac{1}{|A|} \frac{m_k^C(A)}{1 - m_k^C(\emptyset)}, \tag{13}$$

where $A \subseteq C$ and $\emptyset$ being the empty set.

Note that the described classification algorithm is executed by each sensor, for each tracked target. For a global credal classification purpose, the local decision making can be avoided and the classification mass functions of the n targets are used to perform a global classification. The set of mass functions given by a sensor i at time k is denoted $M_i = \{m_k^1, m_k^2, ..., m_k^n\}$, where $i = 1, ..., S$ and S is the number of sensors.

As it was shown in [10] for a single target classification, and extended to multiple targets classification in [14], the credal kinematic data based classification outperforms the Bayesian one, therefore, the interest of this article is focused on the application of the credal classification, in multiple unreliable sensors. In Section VI, a comparison between the credal and Bayesian classifications is given at a local level, but for lack of space, only the most interesting multi-sensor results will be given, namely, the global credal classification results.

V. GLOBAL CLASSIFICATION

Figure 1 illustrates the complete multi-sensor and multi-target algorithm. Each sensor i among the S designed ones makes a set of estimated state vectors $\hat{X}_i = \{\hat{x}_k^1, \hat{x}_k^1, ..., \hat{x}_k^n\}$ basing on the set of taken measurements Z_i at time k, by performing the described local tracking algorithm. Using the local classification algorithm, the sensor provide a set of mass functions M_i concerning targets classifications.

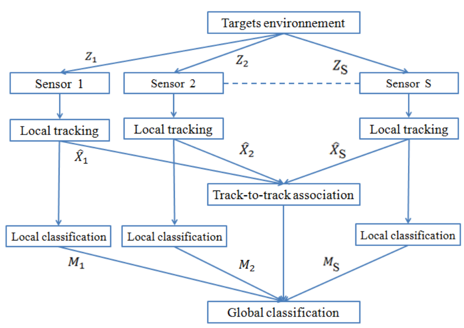

Fig. 1. Decentralized classification approach flowchart.

Box "Track-to-track association" in Figure 1 aims to establish a matching between sensors estimates sets $\hat{X}_{1,\ldots,S}$ elements in order to recognize in which order the belief sets $M_{1,\ldots,S}$ have to be combined for a global classification purpose. The matching step is ensured by the means of a generalize GNN algorithm, as described in Section III, it is executed for each pair of sensors. The cost of assigning target t of sensor i to target l of sensor j is defined as follows:

$$D_{t,l} = (\hat{x}_i^t - \hat{x}_j^l)^T (P^t + P^l)^{-1} (\hat{x}_i^{t_1} - \hat{x}_j^l), \qquad (14)$$

where $t = 1, \ldots, n_i$ and $l = 1, \ldots, n_j$, with n_i and n_j are, respectively, the number of targets tracked by sensors i and j. The distances of Equation (14) [15] form the track-to-track generalized assignment matrix, which is resolved by the generalized Munkres algorithm.

Once a consensus is reached between sensors about the commonly tracked targets, their local classifications are fused using Dempster's conjunctive rule and disjunctive rule of combination.

A. Fusion using Dempster's rule

According to the provided matching solution, the elements of all the local sets $\{M_1, M_2, \ldots, M_S\}$ have to be combined using Dempster's conjunctive combination in Equation (15) and give a common set of mass functions $M_{1\oplus 2\oplus \ldots \oplus S}$. Dempster's normalized conjunctive combination of two mass functions m_1 and m_2, provided by sensors 1 and 2, is defined as follows:

$$m_{1\oplus 2}(A) = \frac{m_{1\textcircled{\cap}2}(A)}{1 - m_{1\textcircled{\cap}2}(\emptyset)} = \frac{\sum_{A_1, A_2 | A_1 \cap A_2 = A} m_1(A_1) m_2(A_2)}{1 - m_{1\textcircled{\cap}2}(\emptyset)}, \qquad (15)$$

with $A \subseteq C$ and $\emptyset$ being the empty-set.

In order to make a global classification decision, the elements of the set $M_{1,2,\ldots,S}$ has to be transformed on pignistic probabilities using Equation (13).

B. Fusion using disjunctive rule

According to the matching solution, the mass functions in the sets $\{M_1, M_2, \ldots, M_S\}$ are distinctively combined. The disjunctive rule of combination of two mass functions m_1 and m_2 which are provided by sensors 1 and 2, is defined as follows:

$$m_{1\textcircled{\cup}2}(A) = \sum_{A_1, A_2 | A_1 \cup A_2 = A} m_1(A_1) m_2(A_2). \qquad (16)$$

Note that all mass function contained in the sets M_i, with $i = 1, \ldots, S$, are combined using the above described fusion rules according to the order given by the track-to-track assignment step. In order to take a decision concerning the target global classifications, the obtained global mass functions are transformed to pignistic probabilities using Equation (13).

VI. Maritime piracy multi-target tracking and classification example

A. Description

This section describes an example of piracy targets environment. Targets are randomly appearing, disappearing and performing different maneuvers. The identification of the targets type (cargo, military boat, go-fast boat (generally pirates boats), etc.) is based on the complexity of the performed maneuvers. The distinctive behaviors of the considered targets are given as follows:

- behavior 1 (b_1): behavior of targets having a low maneuvering capacities (e.g. cargo).
- Behavior 2 (b_2): behavior of targets having a medium maneuvering capacities (e.g. military boat).
- Behavior 3 (b_3): behavior of targets having a high maneuvering capacities (e.g. go-fast boat).

Where the set of targets' possible classes $C = \{c_1, c_2, c_3\}$ corresponds to $C = \{\text{Cargo}, \text{Military boat}, \text{Go-fast boat}\}$.

The state vector of all the targets is represented by $x = [x \quad \dot{x} \quad y \quad \dot{y}]$, it represents the position and the velocity on (x, y) directions. The state vector of each target evolves following the model in Equation (1), with a sate matrix F given by:

$$F = \begin{bmatrix} 1 & \Delta T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta T \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where ΔT is the sampling time.

The deterministic input vector $u(m_l) = [a_x \quad a_y]^T$ in Equation (1) representing the targets different acceleration modes. For example $u(m_1) = [0 \quad 0]^T$ represents a constant velocity mode. The differences in the acceleration capabilities allow the distinction to be made between the targets different classes' behaviors. The acceleration limitations for the *a priori* known classes are expressed by: $-L_i \leq \{a_x, a_y\} \leq L_i$, where $L_i = 0g, 0.2g$ and $0.4g$ respectively, for the classes c_1, c_2 and c_3, with $g = 9.81 \ m/s^2$ is the gravitational acceleration.

In the performed simulation, each target's IMM is composed of 13 evolution models according to the different maneuvers which can be made in x and y directions (see the example described in [12]). The different evolution models are distributed over the three possible targets' behaviors, as follows:
$M_{b_1} = [m_1]$: models belonging to the behavior b_1.
$M_{b_2} = [m_1, \ldots, m_5]$: models belonging to the behavior b_2.
$M_{b_3} = [m_1, \ldots, m_{13}]$: models belonging to the behavior b_3.

Once the different behaviors are defined and their mass function is calculated (based on the local classification algorithm described in Section IV), it is transformed on classes mass according to the following relation:

- relation 1: target in behavior 1 can correspond to cargo, military or go-fast boats. All of them can evolve with a constant velocity. This relation can be written as: $b_1 = \{c_1, c_2, c_3\}$.

- Relation 2: target in behavior 2, which has performed a medium maneuver, may corresponds to a military or go-fast boats only. Cargos is supposed unable to perform any maneuver. This relation can be written as: $b_2 = \{c_2, c_3\}$.
- Relation 3: target in behavior 3, which has performed a sharp maneuver, can only be a go-fast boat, because cargos and military boats can not perform sharp maneuvers. This relation can be written as: $b_3 = \{c_3\}$.

The belief transfer is performed using Equation (12). It transfers the belief on the power set 2^B of behaviors to the power set 2^C of classes, according to the relations described above. The corresponding complete transfer matrix $\bar{M}$, in Equation (12), has a size $(2^3 = 8) \times (2^3 = 8)$, and is given by:

$$\bar{M} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

Details concerning the derivation of the above matrix can be found in [10]. This matrix enables to compute the mass functions on the classes space C, in order to take a decision on the targets' classifications.

B. Simulation and results

Two unreliable sensors taking measurements according to Equation (2) are observing, all or some, of multiple maneuvering and randomly appearing and disappearing targets. The trajectories of the targets on (x, y) space, are depicted in Figure 2.

Note that target 2 is observed by the two designed sensors and only its classification results are presented in this section (lack of space).

As it can be seen in Figure 2 and described above, target 2 evolution consists on three constant evolution segments separated by two maneuvers. First, it begins its trajectory with a constant velocity, which corresponds to zero acceleration vector $u = [0 \quad 0]^T$. Then in time window $[62, 66]$, it performs a medium acceleration in x direction (according to the input vector $u = [0.2 \quad 0]^T$), after that it takes a constant evolution until its second sharp maneuver in y direction (according to the input vector $u = [0 \quad 0.4]^T$) during the time window $[80, 86]$. Finally, it finishes its evolution with a constant velocity.

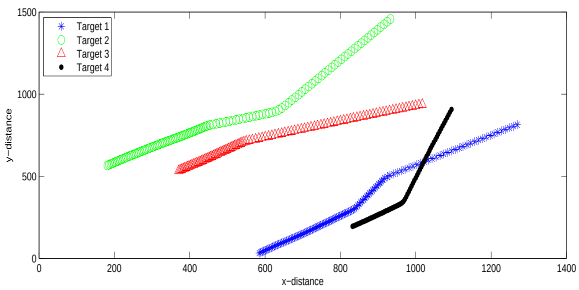

Fig. 2. Multi-target trajectories.

Normally, we expect that target 2 has to be in doubt between the three classes during the first constant velocity evolution step, and has to be in doubt between the second and third classes after its medium maneuver, because targets of class 1 can not perform any maneuver, and finally, it has to be classified in class 3 after its sharp maneuver.

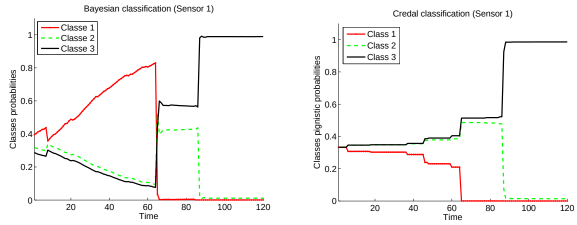

Fig. 3. Sensor 1 and 2, respectively, local classification results.

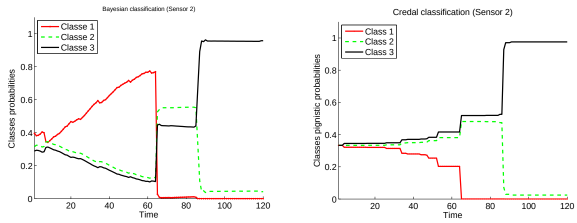

Fig. 4. Sensor 1 and 2, respectively, local classification results.

As expected, Figure 3 shows that both sensors succeed to classify target 2 as a go-fast boat after its sharp maneuver, this using the Bayesian or the credal algorithm. It can also be shown that in the double situation, where it is supposed to have a perfect doubt, in the first movement step for example, the Bayesian classifier favors the class c_1 which leads quickly to a wrong decision, the explanation is given in [14]. Due to the sensors unreliability, the sensors' credal classification are also deteriorated. Indeed, as it is shown in the Figures 3 and 4 the doubt in the first and second movement steps is badly managed, even in the credal results. The following results try to enhance the sensors classifications by fusing them. It is shown, with an explanation, that the disjunctive fusion gives the best classification result.

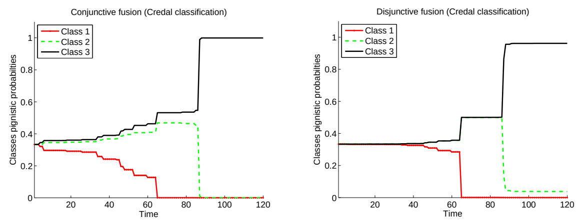

Fig. 5. Conjunctive and disjunctive combinations, respectively, global classification results.

Figure 5, shows the classification results obtained after global classification performing, using conjunctive and disjunctive rules of combination, respectively. It can be seen that the conjunctive combination accentuates the classification deterioration. This is due to the nature of the conjunctive combination which favors singletons or specific subsets over the doubt. In fact, for example, if sensor' 1 belief is given by: $m_1(\{c_2, c_3\}) = 0.4$ and $m_1(\{c_1, c_2, c_3\}) = 0.6$, and sensor' 2 belief is given by: $m_2(\{c_2, c_3\}) = 0.5$ and $m_2(\{c_1, c_2, c_3\}) = 0.5$, the conjunctive combination gives: $m_{1 \textcircled{\cap} 2}(\{c_2, c_3\}) = 0.7$ and $m_{1 \textcircled{\cap} 2}(\{c_1, c_2, c_3\}) = 0.3$. This corresponds, in the simulated example, to classification deterioration accentuation.

In the other hand, it can be seen that the disjunctive global classification is more corresponding to the expected classification result. It gives a more refined classification result than both sensors local ones and also the conjunctive global one. It can be remarked that the doubt is well managed in the first and second steps of movement so, the deterioration caused by the sensors high noises is reduced. This is taking advantage of the prudence of the disjunctive rule of combination. Indeed, for example, the disjunctive combination of the belief masses defined above gives: $m_{1 \textcircled{\cup} 2}(\{c_2, c_3\}) = 0.2$ and $m_{1 \textcircled{\cup} 2}(\{c_1, c_2, c_3\}) = 0.8$. This illustrates the cautious nature of the disjunctive combination and its utility in the considered classification problem.

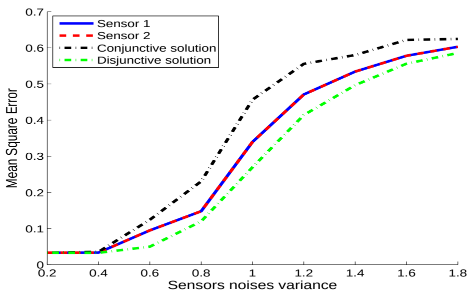

Fig. 6. Classification mean square error.

Figure 6 gives the local and global classifications Mean Square Errors (MSE) according to different sensors' noises variance values. The MSE represents the difference between the expected classes pignistic probabilities and the calculated ones, it is calculated by: $MSE = (\hat{Betp} - Betp)'(\hat{Betp} - Betp)$, where $\hat{Betp}$ is the theoretical expected pignistic probabilities. For each sensors' noise value the MSE is averaged over 20 simulations with different noises distributions. It can be seen, in Figure 6, that indeed, the disjunctive combination brings the best performance when it comes to fuse unreliable and independent sensors information.

VII. Conclusion

This paper proposes a multi-sensors architecture where each sensor is able to, locally, track and classify multiple targets basing on its sequentially acquired measurements. A large imprecision on measurements taking is simulated, which corresponds to the unreliability of the sensors. Due this unreliability, the local classifications performed by the senors are deteriorated. The aim is to combine the local weak classifications in order to obtain a global more robust one.

Many combination rules are presented in the belief theory framework. In this work, the two most general ones are taken, namely: the conjunctive and disjunctive rules of combination. The disjunctive rule, which suppose any information on the sensors reliability, provides the most robust classification results.

The capacity of the belief theory to model both the uncertainty and imprecision makes its use, in multiple target tracking and classification context, challenging and promising. Our future works will aim to integrate belief functions in the estimation theory and reformulate the assignment (observations-to-tracks or tracks-to-tracks) problem in the same framework.

References

[1] D. Smith and S. Singh. Approaches to multisensor data fusion in target tracking: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 18(12):1696–1710, 2006.

[2] Y. Bar-Shalom, X.R. Li, T. Kirubarajan, and J. Wiley. *Estimation with applications to tracking and navigation*. Wiley Library, 2001.

[3] S.S. Blackman. Multiple hypothesis tracking for multiple target tracking. *Aerospace and Electronic Systems Magazine*, 19(1):5–18, 2004.

[4] D. Mercier, É. Lefèvre, and D. Jolly. Object association with belief functions, an application with vehicles. *Information Sciences*, 181(24):5485–5500, 2011.

[5] Y. Bar-Shalom. *Tracking and data association*. Academic Press Professional, San Diego, USA., 1987.

[6] D. Musicki, R. Evans, and S. Stankovic. Integrated probabilistic data association. *IEEE Transactions on Automatic Control*, 39(6):1237–1241, 1994.

[7] H.A.P. Blom and E.A. Bloem. Combining IMM and JPDA for tracking multiple maneuvering targets in clutter. In *Proceedings of the 5th International Conference on Information Fusion.*, volume 1, pages 705–712, 2002.

[8] Z. Khan, T. Balch, and F. Dellaert. MCMC-based particle filtering for tracking a variable number of interacting targets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1805–1819, 2005.

[9] S. Oh, S. Russell, and S. Sastry. Markov Chain Monte Carlo data association for general multiple-target tracking problems. In *43rd IEEE Conference on Decision and Control*, volume 1, pages 735–742, 2004.

[10] P. Smets and B. Ristic. Kalman filter and joint tracking and classification based on belief functions in the TBM framework. *Information Fusion*, 8(1):16–27, 2007.

[11] P. Smets and R. Kennes. The Transferable Belief Model. *Artificial intelligence*, 66(2):191–234, 1994.

[12] B. Ristic, N. Gordon, and A. Bessell. On target classification using kinematic data. *Information Fusion*, 5(1):15–21, 2004.

[13] W. Mei, G. Shan, and Y. Wang. A second-order uncertainty model for target classification using kinematic data. *Information Fusion*, 12(2):105–110, 2011.

[14] S. Hachour, F. Delmotte, E. Lefèvre, and D. Mercier. Tracking and identification of multiple targets. In *Proceedings of the 7th Workshop Interdisciplinaire sur la Sécurité Globale*, Troyes, France, 2013, available online at: http://www.agence-nationale-recherche.fr/Colloques/WISG2013/articles/Article_Delmotte.pdf.

[15] S.S. Blackman and R. Popoli. *Design and analysis of modern tracking systems*. Artech House, Norwood, MA, USA, 1999.

[16] G.A Watson and W.D Blair. IMM algorithm for tracking targets that maneuver through coordinated turns. In *Proceedings of Signal and Data Processing of Small Targets*, volume 1698, pages 236–247, 1992.

[17] R.A. Singer. Estimating optimal tracking filter performance for manned maneuvering targets. *IEEE Transactions on Aerospace and Electronic Systems.*, (4):473–483, 1970.

[18] F. Bourgeois and J-C. Lassalle. An extension of the Munkres algorithm for the assignment problem to rectangular matrices. *Communications of the ACM*, 14(12):802–804, 1971.

[19] GJ. McLachlan. Mahalanobis distance. *Resonance*, 4(6):20–26, 1999.

[20] F. Delmotte and P. Smets. Target identification based on the Transferable Belief Model interpretation of Dempster-Shafer model. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 34(4):457–471, 2004.

[21] P. Smets. Belief functions: the disjunctive rule of combination and the Generalized Bayesian Theorem. *International Journal of Approximate Reasoning*, 9(1):1–35, 1993.