

A Fuzzy-Logic based non cooperative target recognition

Thomas Boulay, Julien Lagoutte, Ali Mohammad-Djafari, Nicolas Gac

► **To cite this version:**

Thomas Boulay, Julien Lagoutte, Ali Mohammad-Djafari, Nicolas Gac. A Fuzzy-Logic based non cooperative target recognition. Signal Image Technology and Internet Based Systems 2012, Nov 2012, Naples, Italy. pp.410-145, 10.1109/SITIS.2012.66 . hal-00778560

HAL Id: hal-00778560

<https://hal.archives-ouvertes.fr/hal-00778560>

Submitted on 21 Jan 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Fuzzy-Logic based non cooperative target recognition

Thomas Boulay, Julien Lagoutte
Surface Radar
Thales Air Systems
Limours, France
thomas.boulay@thalesgroup.com

Ali Mohammad-Djafari, Nicolas Gac
Laboratoire des Signaux et Systèmes
UMR 8506 CNRS-SUPELEC-UNIV PARIS SUD
Gif-sur-Yvette, France
Ali.Mohammad-Djafari@lss.supelec.fr

Abstract—In this paper, we present the problem of Non Cooperative Target Recognition (NCTR) as a supervised classification problem. After a brief presentation on the radar acquisition system of range profiles and the problem of recognition, we use a Fuzzy-Logic based algorithm to do this classification maximizing the recognition rate while controlling the error rate. Unlike classical NCTR algorithms, this new algorithm allows to control error rate under a fixed value and maximize the recognition rate.

Keywords—NCTR, KNN, HRR, RADAR, CLASSIFICATION, RECOGNITION, FUZZY LOGIC

I. INTRODUCTION AND CONTEXT

During major conflicts, cooperative identification techniques are not enough reliable and Non Cooperative Target Recognition (NCTR) [1], [2] is increasingly seen as essential. One of these techniques is to compare target signature under test with target signature contained in training set. Therefore, the problem of Non Cooperative Target Recognition can be considered as a supervised classification problem [3], [4].

Many classification methods allow to get good performance of classification when the Signal to Noise Ratio (SNR) is in favor. However, when SNR decreased, performances of classification deteriorate significantly. Algorithms used in NCTR applications must guarantee an error rate below a specified value. Therefore, in this paper, a new algorithm based on Fuzzy-Logic is proposed to reach objective of maximizing the recognition rate while controlling the error rate. This objective is a crucial point for NCTR applications because every decision incurs human lives or may cause, to a lesser extent, losses of material.

II. NON COOPERATIVE TARGET RECOGNITION (NCTR)

Unlike cooperative applications, non-cooperative applications run without cooperation from the target at the time of detection. When the application is a target recognition application, it is called Non-Cooperative Target Recognition application. Several techniques can be used for NCTR (Kinematics, Doppler, ISAR, HRR) but the technique we use for this paper is High Range Resolution (HRR) technique.

High Range resolution [5] offers a very simple and rapid way to characterise a target through the use of radar range profile. A range profile is essentially a one-dimensional radar image of the target (cf. Figure 1). The HRR profile is a representation of temporal response of target to an high resolution radar impulse.

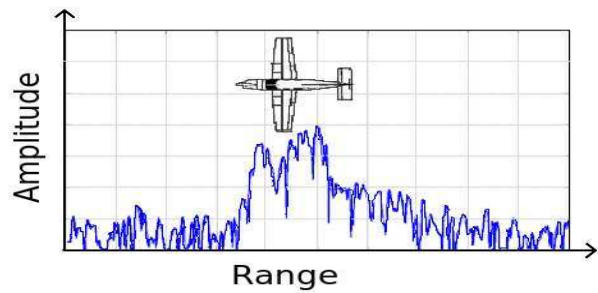


Figure 1. Example of range profile

In [6], steps to get a range profile are detailed. To summarize, after acquisition of a complex signal in phase quadrature, a Fourier Transform (with "zero-padding") and a Hamming weighting (in order to limit influence of side lobes) is applied. Finally, we keep L_b samples corresponding to the useful signal and we only keep the module to get what is called a *Range Profile* (cf. Figure 2). In a last step, range profiles are transformed in logarithmic scale so as to highlight points of low amplitude.

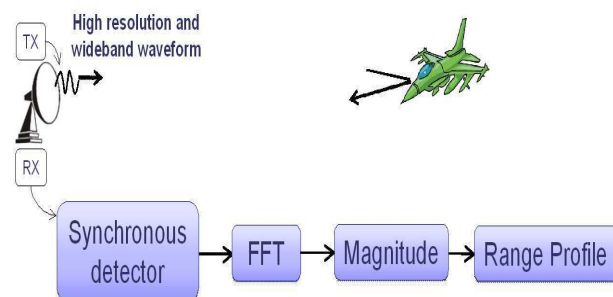


Figure 2. Formation of range profile

III. DATABASE

The data used in this article are synthetic data. They are divided into two bases: a training set and a test set.

A. Training set

The training set contains L_{tr} range profiles in $K = 3$ different classes ($L_{tr}^1 = L_{tr}^2 = L_{tr}^3$). They were generated from target modeling (CAD) on which were applied techniques to calculate RCS (Radar Cross Section). For each configuration (frequency, elevation angle, azimuth) is calculated the RCS complex value. Angular increment was determined so as to have all RCS representative values. Frequency increment in turn is a compromise between the size of observation window ($T = \frac{1}{\Delta_f}$ with Δ_f the frequency increment) and the computation time. Indeed, RCS calculations are generally quite long so we usually adjust the frequency to have a sufficiently large observation window and not to have an prominent computation time. Note that the training set is independent of the radar. There is not radar chain modeling.

If all possible aspect angle configurations¹ are considered, the total number of range profiles in training set would equal $N_{tot} = 1440 \times 720 = 1036800$. The learning set will actually be restricted by limiting variation in azimuth and elevation angle around azimuth α_t and elevation angle θ_t of range profiles of test set. Therefore all range profiles corresponding to an azimuth belonging to $[\alpha_t - 5^\circ, \alpha_t + 5^\circ]$ and an elevation angle belonging to $[\theta_t - 0.75^\circ, \theta_t + 0.75^\circ]$ are selected. Our training set will be made up $L_{tr} = 861$ range profiles.

L_{tr}^k	the number of range profiles belonging to the class k for training set.
$L_{tr} = \sum_{k=1}^K L_{tr}^k$	the total number of range profiles for training set.
$\mathbf{x}_{tr,l}^k$	the l -th range profile of training set belonging to the class k , with $k = \{1, \dots, K\}$ et $l = [1, \dots, L_{tr}^k]$.
\mathbf{X}_{tr}^k	the matrix of range profiles of training set belonging to the class k .
$\mathbf{x}_{tr,l}$	the l -th range profile of training set with $l = [1, \dots, L_{tr}]$.
\mathbf{X}_{tr}	the matrix of range profiles of training set.

Table I
SUMMARY OF NOTATIONS USED FOR THE TRAINING SET

B. Test set

In test set, we have $L_{te} = 150$ range profiles also in 3 different classes too ($L_{te1} = L_{te2} = L_{te3} = 50$). Range profiles of test set have the same aspect angle (α_t, θ_t). They were obtained from simulation of the radar signal. Therefore these signals contain some imperfections related to the radar. Unlike range profiles of learning set, range profiles of test set are noisy.

¹ The aspect angle of an aircraft can be expressed as a pair of coordinated (α, θ) where α is azimuth and θ is elevation angle ($\alpha = [0, \dots, 360^\circ]$, $\theta = [0, \dots, 180^\circ]$). The azimuth is the angle between direction of the nose of aircraft and direction of the radar line of sight projected onto the plane formed by the nose and wings of the aircraft. The elevation angle is the angle between the line of sight of the radar and the plane formed by the nose and wings of the aircraft.

L_{te}^k	the number of range profiles belonging to the class k for test set.
$L_{te} = \sum_{k=1}^K L_{te}^k$	the total number of range profiles for test set.
$\mathbf{x}_{te,l}^k$	the l -th range profile of test set belonging to the class k , with $k = \{1, \dots, K\}$ et $l = [1, \dots, L_{te}^k]$.
\mathbf{X}_{te}^k	the matrix of range profiles of test set belonging to the class k .
$\mathbf{x}_{te,l}$	the l -th range profile of test set with $l = [1, \dots, L_{te}]$.
\mathbf{X}_{te}	the matrix of range profiles of test set.

Table II
SUMMARY OF NOTATIONS USED FOR THE TEST SET

IV. FUZZY-LOGIC BASED ALGORITHM

A. Background on Fuzzy Logic

Fuzzy Logic was initially proposed by L.Zadeh in a 1965 paper [7] to solve the problems of uncertain data in automatic control applications. Fuzzy Logic has been applied in many fields like robotic, air traffic control, meteorology, medicine and pattern recognition. Fuzzy logic is a form of many-valued logic or probabilistic logic, it deals with reasoning that is approximate rather than fixed and exact. In contrast with traditional logic theory, where binary sets have two-valued logic (true or false), fuzzy logic variables may have a truth value that ranges in degree between 0 and 1.

Fuzzy Logic for pattern recognition applications is mainly relied on two functions:

- “Membership Function” $M(\theta)$: this function is used to describe how a given class can be represented by a given feature θ . This function reflects the available knowledge about this feature. In our case, as we will see, θ represents a distance between two range profiles. $M(\theta)$ is then related to the probability distribution $P(\theta|X_{tr}^k)$.
- Density of Possibility $D(\theta)$: it describes how the value of a given feature can be distributed, given a measurement of this feature and an estimation of its accuracy. This function is related to $P(\theta|x_{te,l}^k, x_{tr,l}^k)$.

By combining the above functions, two numbers (ranging in $[0,1]$) are computed for each class:

- Possibility: this number describes the level of overlapping between the measurement and reference data for each class. Let θ be a feature. Knowing the Membership Function and the Density of Possibility linked to this feature, the value of possibility is deduced as follow:

$$P = \max_{\theta} \{ \min [M(\theta), D(\theta)] \} \quad (1)$$

- Necessity: this number describes the level of inclusion of the density of possibility within the membership function.

$$N = \min_{\theta} \{ \max [M(\theta), 1 - D(\theta)] \} \quad (2)$$

Figure 3 illustrates possibility and necessity computing from Membership Function and Density of Possibility.

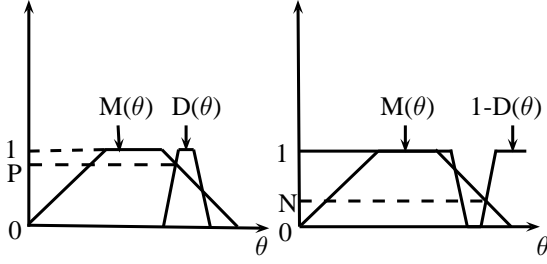


Figure 3. Possibility (P) and Necessity (N) computing

B. Fuzzy Logic based algorithm

Objective of this algorithm is to make target recognition keeping control of error rate and maximizing success rate. Error rate control means we want to ensure probability to take the decision "class $\neq k$ " knowing "class = k" is less than 5% for example. The purpose of this section is to present a method of radar target recognition based on Fuzzy Logic principles allowing to achieve this objective. Feature used to describe different classes in our target recognition algorithm is the distance between two range profiles. The first step is to compute membership functions of each class. Idea of membership functions is that classes can be distinguished through a feature. As it happens, we want to model the fact that distances between range profiles of the same class i are differently distributed than distances between range profiles of other classes $j \neq i$.

1) *Membership functions computing*: Let Θ^k , the matrix of distances between range profiles of the matrix X_{tr}^k . Each θ_{ij}^k is the distance between the i -th range profile and the j -th range profile of X_{tr}^k . The diagonal of Θ^k is equal to 0.

$$\theta_{ij}^k = \|x_{tr,i}^k - x_{tr,j}^k\|^2 \quad (3)$$

Let $\mathbf{m}\Theta^k = [m\theta_1^k, \dots, m\theta_{L_{tr}^k}^k]$, the vector of minima of distances between range profiles of training set belonging of class k .

$$m\theta_i^k = \min_j(\theta_{ij}^k), \forall i \neq j \quad (4)$$

To compute membership functions of each class, the histogram of each $\mathbf{m}\Theta^k$ is initially computed. From this histogram, the distribution functions for each class (DF_k) can be determined. For each DF_k , we want to get the value of the 95 percentile, ie. we seek $m\theta_{lim}^k$ such as $P(m\theta_i^k \leq m\theta_{lim}^k) = 0.95$. For each class, we get a distance $m\theta_{lim}^k$ that allows us to build the membership function of each class. Indeed, if the test range profile belongs to the class k , there is 95% chance that distance between the test range profile and training range profile is less than $m\theta_{lim}^k$. We want to use the fact that difference between two range profiles of different classes is a priori more important than the difference between two range profiles of the same class.

Let M_k , the membership function of the class k .

$$M_k(m\theta_i^k) = \begin{cases} 1 & \text{if } m\theta_i^k \leq m\theta_{lim}^k \\ 0 & \text{else.} \end{cases} \quad (5)$$

Figure 4 summarizes the different steps to build membership functions.

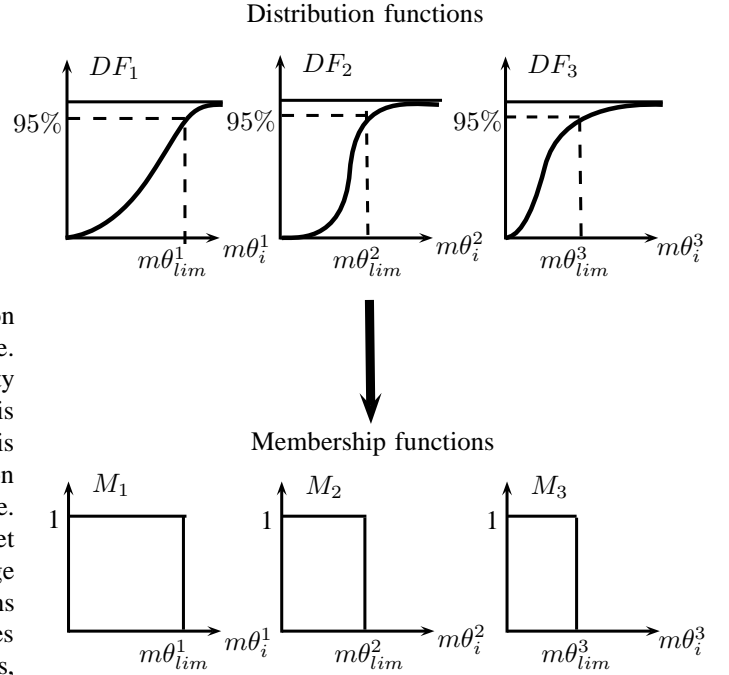


Figure 4. Building of membership functions

2) *Probability distribution of the distance measurement error*: Until now, degree of membership to a class k has been described through the distance between two range profiles. However, accuracy of this measurement has not been considered. To take into account this accuracy, density of possibility concept is introduced. The used distance in our algorithm is the euclidian distance. Let x and y two range profiles of size (number of range bins) equal to L_b :

$$d_{euclidian} = \sum_{i=1}^{L_b} (x_i - y_i)^2 \quad (6)$$

It is assumed that measurement of RCS in a range bin follows a normal distribution, ie. $x_i \sim \mathcal{N}(m, \sigma_n^2)$ and $y_i \sim \mathcal{N}(m, \sigma_n^2)$. Let $z_i = x_i - y_i$, then $z_i \sim \mathcal{N}(0, 2\sigma_n^2)$.

Therefore,

$$\sum_{i=1}^{L_b} \frac{z_i^2}{2\sigma_n^2} \sim \chi_{nc}^2(L_b, \lambda) \quad (7)$$

with χ_{nc}^2 , the noncentral chi-squared distribution and $\lambda = \sum_{i=1}^{L_b} \left(\frac{m_i}{\sigma_i}\right)^2$. In our case, all $m_i = 0$ and all $\sigma_i = 2\sigma_n^2$, thus $\lambda = 0$.

Equation (7) can be rewritten:

$$\alpha = \frac{d_{euclidian}}{2\sigma_n^2} \sim \chi^2(L_b) \quad (8)$$

The random variable α follows chi-squared distribution with L_b degrees of freedom. It should nevertheless take into account the fact that range profile under test and all training range profiles are thresholded according to a threshold that varies according to the noise level of each range profile under test. The number of points used to calculate the distance varies from one range profile under test to another and, therefore, for each range profile under test, the degree of freedom of the chi-square distribution will vary.

3) *First step of algorithm: Distances computing:* The first step of algorithm is to compute distances between range profile under test (one range profile of X_{te}) and all range profiles of training set X_{tr} . The distances calculated are splitted into three vectors, one for distances between range profile under test and range profiles of X_{tr}^1 and so on. These vectors are noted \mathbf{d}^1 , \mathbf{d}^2 and \mathbf{d}^3 . Figure 5 illustrates in 2D this first step.

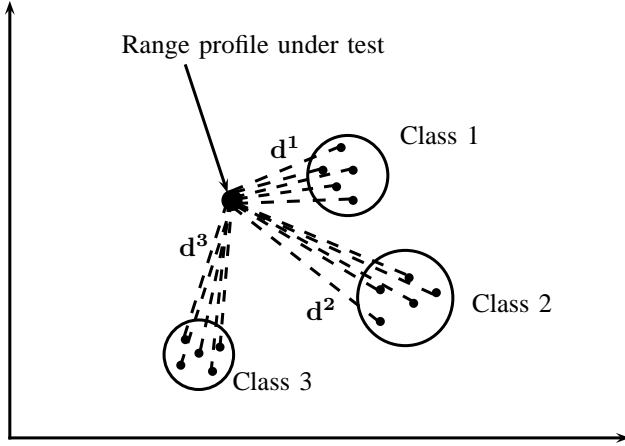


Figure 5. Distances computing

4) *Second step: Density of possibilities computing:* The second step is to compute densities of possibilities associated with distances computed in first step. Let d_j^k the distance between range profile under test j and range profile k of training set. In IV-B2, we saw that $\alpha_j^k = d_j^k / (2\sigma_n^2)$ followed chi squared distribution with L_b degrees of freedom.

Density of possibilities associated with α_j^k can be defined as follows:

$$DP_{i,j}^k = \alpha_j^k - t_i \quad (9)$$

with $i \in \{1, 2\}$. t_1 and t_2 are respectively 99.5 percentile and 95 percentile of chi-squared distribution with degree of freedom depending on the noise level of range profile under test. Densities of possibility are not centered around distance value because distance measurement can not be under-estimated. Figure 6 summarizes this second step.

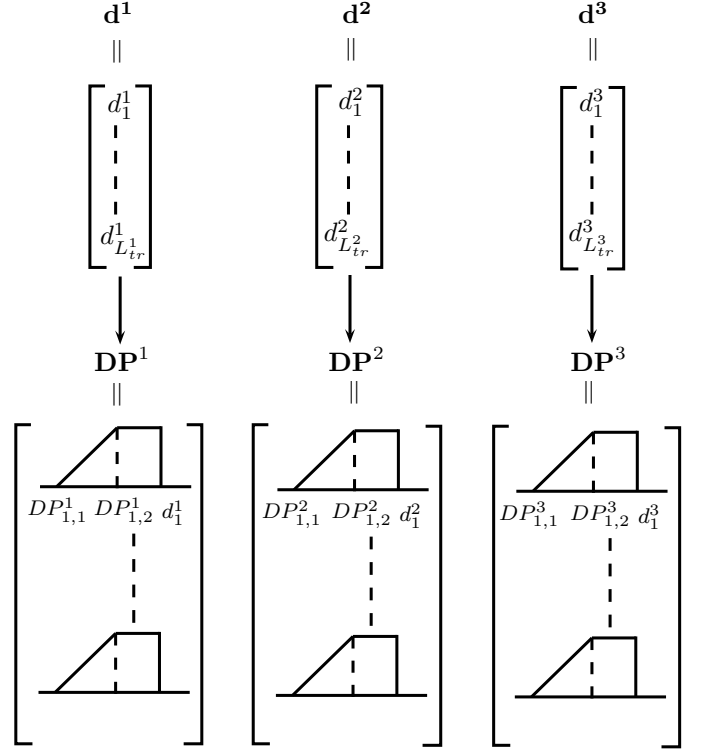


Figure 6. Densities of possibilities computing

5) *Third step: Possibilities computing:* When densities of possibility are calculated, each of them (DP^1 , DP^2 and DP^3) are respectively superposed to membership functions M_1 , M_2 and M_3 to determine the possibilities. Intersection of DP^1 , DP^2 and DP^3 with M_1 , M_2 and M_3 provides the three vectors of possibility \mathbf{p}^1 , \mathbf{p}^2 and \mathbf{p}^3 (cf. Figure 7).

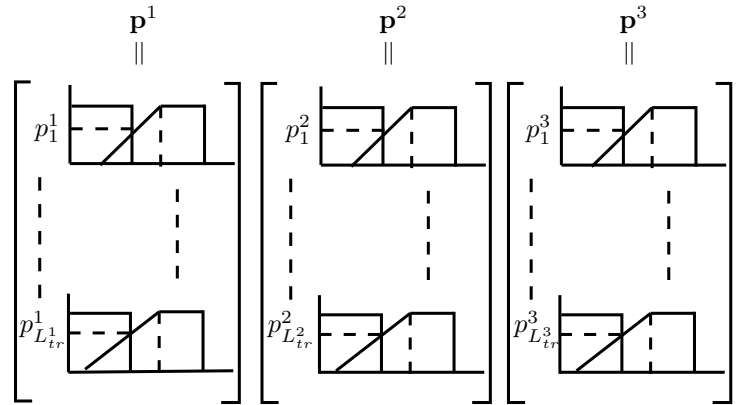


Figure 7. Computing of possibilities

6) *Fourth step: Decisions:* The decision is based on possibilities calculated in the previous step. Foremost, possibilities calculated for each class (\mathbf{p}^1 , \mathbf{p}^2 , \mathbf{p}^3) are merged to get one fused value of possibility by class.

$$mP^k = \max(\mathbf{p}^k) \quad (10)$$

Then, decision is taken from these values as follows:

- Class k is granted for range profile under test, if $mP^k \geq TG$ (**Granted**).
- Class k is dubious for range profile under test, if $TD \leq mP^k \leq TG$ (**Dubious**).
- Class k is denied for range profile under test, if $mP^k \leq TD$ (**Denied**).

where TG et TD are adjusted to obtain the desired error rate.²

With this algorithm, several classes can be granted, dubious or denied for a range profile under test. The decision is taken independently for each class. Most algorithms take a single decision for each range profile under test even if the probability of belonging to a class is small. Our algorithm prefers to grant several classes when indecision is too high.

V. PERFORMANCE OF FUZZY-LOGIC BASED ALGORITHM

Performance of Fuzzy-Logic based algorithm are evaluated according to error rate and success rate. The error rate is the number of times the correct class is rejected divided by the total number of test range profile. The success rate is the number of times the correct class and only the correct class is granted divided by the total number of test range profile. Our algorithm is compared to a classical K Nearest Neighbor (KNN) algorithm. Behavior of these two algorithms is observed when value of Signal to Noise ratio (SNR) on range profile under test decreases.

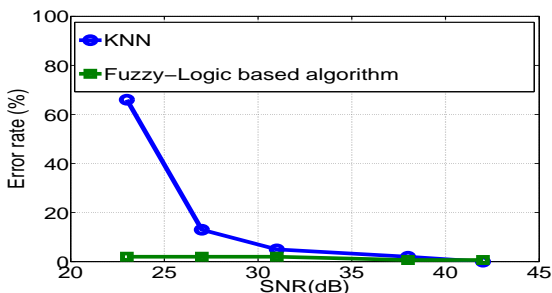


Figure 8. Evolution of error rate as a function of SNR

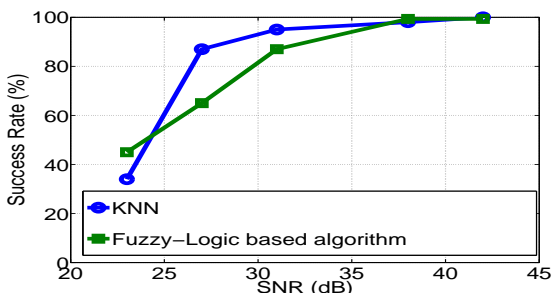


Figure 9. Evolution of success rate as a function of SNR

² Using rectangular membership function leads to a better definition of TG and TD (thanks to analytical formula).

Figure 8 and Figure 9 respectively show evolution of error rate and success rate as a function of SNR for KNN algorithm and Fuzzy-Logic based algorithm described in this paper. Figure 8 shows error rate remains at a constant level close to 0% for Fuzzy-Logic based algorithm while it significantly increases with KNN algorithm. This is a crucial point for NCTR algorithm because we must be able to ensure the user of our recognition system that error rate will remain below a given value. This constraint is respected at the expense of success rate as shown in Figure 9. Indeed, it substantially reduces for KNN algorithm or our Fuzzy-Logic based algorithm when the SNR decreases. For KNN algorithm, success rate is opposite of error rate because only one class is selected by range profile under test. It is therefore natural that success rate decreases when error rate increases. For Fuzzy-Logic based algorithm, success rate decreases because multiple-decision are taken when the SNR decreases. Instead of retaining only the correct class, several classes including the correct class are selected. The average number of successful classes and rejected classes by decision is plotted on Figure 10. We can see that in many cases at least one class is rejected even if algorithm retains several classes. This information can be interesting for the operator. For example, in case where class 1 and 2 are friendly targets and class 3 is hostile, if our algorithm rejects class 1 and 2 but accepts class 3, operator can react accordingly.

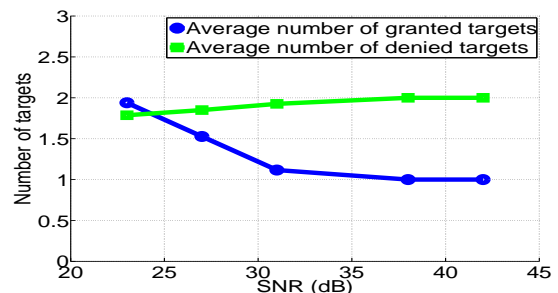


Figure 10. Evolution of average number of successful targets as a function of SNR

VI. CONCLUSION

This paper presents an NCTR algorithm based on Fuzzy-Logic. A major constraint with NCTR application is error rate control. Performances of the proposed algorithm are compared with classical K Nearest Neighbour (KNN) algorithm. We show that our algorithm allows to control error rate even when SNR decreases sharply in contrast to KNN algorithm. Although success rate is reduced when SNR decreases, evolution of mean number of successful targets by decision demonstrates that in majority of cases, our algorithm allows us to reject at least one target while keeping an error rate below a fixed value. Another constraint related to NCTR application is real-time execution. In fact, to be efficient, NCTR application must issue a decision as quickly as possible. One way to comply this constraint is to use Graphics Processing

Units (GPU) and parallelize algorithm. As KNN algorithm [8], the Fuzzy-Logic based algorithm proposed in this paper is massively parallel and its implementation on GPU would allow to improve the execution time. In fact, each distance, each density of possibilities and each possibility could be computed independently of others, in parallel on the different computing units of GPU. This perspective would provide a truly operational algorithm for air defense systems.

REFERENCES

- [1] M. Moruzzis and N. Colin, "Radar target recognition by Fuzzy Logic," *IEEE Aerospace and Electronic Systems Magazine*, vol. 13, no. 7, pp. 13–20, 1998.
- [2] V. Nimier and al., "MILORD, an application of multifeature fusion for radar NCTR," in *Proc. Third Int. Conf. Information FUSION 2000*, 2000.
- [3] Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification (2nd Edition)*, Wiley-Interscience, 2 edition, Nov. 2001.
- [4] A. K. Jain, Robert P. W. Duin, and Jianchang Mao, "Statistical pattern recognition: a review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 4–37, 2000.
- [5] Donald R. Wehner and Bruce Barnes, *High-Resolution Radar*, Artech House Publishers, 1994.
- [6] P. Tait, *Introduction to Radar Target Recognition*, IEE Radar, Sonar and Navigation series 18. The institution of Electrical Engineers, 2005.
- [7] Lofti A. Zadeh, "Fuzzy Sets," *Information and Control*, vol. 8, pp. 338–353, 1965.
- [8] T. Boulay and al., "Algorithmes de Reconnaissance Non Coopérative de Cibles et implémentation sur GPU," in *GRETSI*, Bordeaux, France, Sept. 2011, p. ID146.