# Neural design procedure for an ATTR system based on video imagery usage

Constantin-Iulian VIZITIU, Petrica CIOTIRNAE
*Communications and Electronic Systems* Department
Military Technical Academy
George Cosbuc Avenue 81-83, Bucharest
ROMANIA
vic@mta.ro, ciotirnae@mta.ro http://www.mta.ro

*Abstract*: - It is well known that the design and effective development of ATTR systems incorporating in their architecture some characteristic elements belong to artificial intelligence represents and will represent a very important research direction in this area. Accordingly, in the current study a specific design algorithm for a full neural implementation of an ATTR system based on video imagery usage is described. Using an actual database, the theoretical and experimental results obtained confirm the great potential of the proposed design algorithm to be implemented into concrete ATTR application.

*Key-Words:* - ATTR system, pattern recognition, neural networks, video imagery

## 1 Introduction

It is known that the design and development of ***automatic target tracking and recognition*** (ATTR) systems integrating in their architecture some specific elements belong to artificial intelligence (e.g., neural networks, pattern recognition and other computational intelligence techniques) represents and will continue to represent an important research direction because of naturally proximity by the human operator abilities. Generally speaking, an ATTR system using the principles of ***artificial intelligence*** (AI) become a ***smart*** system or weapon and its new functions are substantial improved than similar standard systems [1], [2].

The ***basic*** functions of ATTR systems are target ***detection*** and ***tracking***, and target ***recognition***, respectively [2]. The concrete modality to implement these functions depend a lot by the spectral domain particularities of its internal sensors (e.g., video, FLIR, thermal, SAR/ISAR etc.). If we reffer also to application spectrum of these systems than we can discuss about ***civilian*** (e.g., surveillance systems, the efficiency increase of the flight traffic guidance, the navigation safety increase in air, ground and naval transportation, for other scientific goals etc.) or ***military*** (e.g., the weapon system guidance, the ground and naval survey, air and antibalistic defence etc.) potential applications. However, these several applications are another important reason to continue the reaserching work to develop new and more effective ATTR systems.

According to [3], one of the most often used way to design and implement an ATTR system consists in ***video imagery*** usage (see Fig.1). In the specialized literature assigned to video ATTR system theory, a lot of standard methods for tracking and recognition function implementation are mentioned [1], [4], [5], [6], [7]. However, the performance level offered by all these methods depend on the specific procedures used for video image acquisition and preprocessing, on the information accuracy proccesed in recognition chain and the influence of perturbation factors.

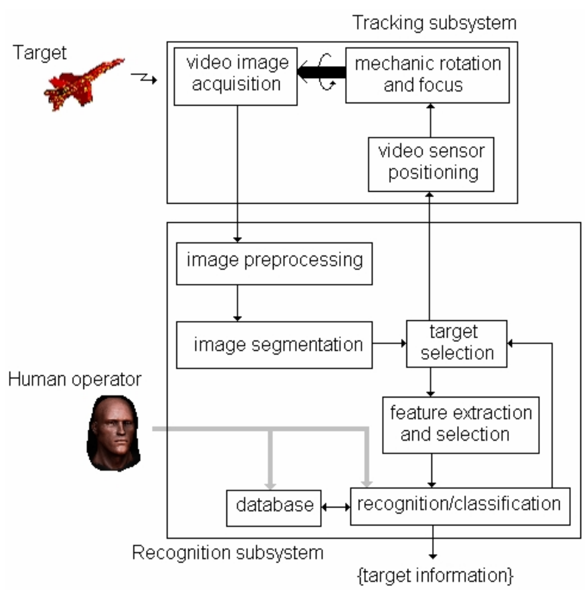



Fig.1: ATTR system based on video imagery usage

Starting from the indubitable advantages offered by conexionist paradigm of AI (e.g., their adaptive nature, learning capacity, noise increased robustness, generalization capacity etc.), a very interesting and performed approach concerning the basic function implementation of (video) ATTR systems is represented by the *neural network* (supervised and/or clustering techniques) usage [8], [9], [10] [11], [12]. Generally speaking, the greatest part of neural network applications into ATTR systems refers to *automatic target recognition* (ATR) system class, and these applications do not use very complex neural topologies and learning rules [5], [10]. In ATTR system literature cannot be also identified many concrete examples of *full* neural ATTR systems [8], [13].

The main *goal* of this paper is to propose a suitable design solution for a more efficient video ATTR system, based on *advanced* artificial neural network (i.e., topologies and learning rules) usage.

Our proposed solutions in the current study will be checked by using an actual video database. Therefore, a theoretical approach of the ATTR system basics is done in the first part of the paper. Then we will describe the neural design procedure used for implementation of tracking and recognition functions. In the experimental section, are shown the results that confirm the broached theoretical aspects from beginning. At the end of the paper, some conclusions and future research directions in this action field are included.

## 2 Design of neural ATTR system

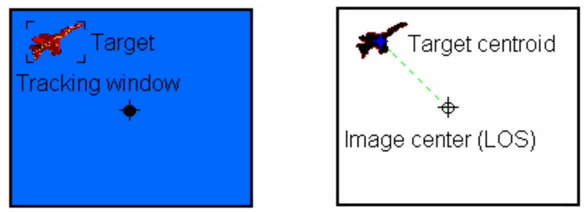
As one can see on Fig.1, the two basic components of an video ATTR system are *tracking* and *classification/recognition* subsystems. More details about the internal structure and functions of each subsystem can be found in [13]. However, it is important to stretch out the point of view by that the presence of the human operator in the internal working algorithm of this system is one *limited*: the interaction with the database occurs only in order to increase its informational content, and with the recognition/classification block only in the training phase of supervised neural networks or in critical situations. Thus, the *automatic* character of ATTR system is assured.

Let us describe the neural architectures used for implementation of ATRR system functions:

*A.Tracking function*

The positioning block of video sensor is the driving force element of the tracking subsystem and it ensures an efficient dynamic positioning of the sensor view direction to the tracked target.

The working algorithm of this block can be resume as it follows: let us consider an image or frame sequence $\{I_r\}_{r=0,1,2,...}$ along the certain period of time. These input images are processed in order to determinate the centroid position of involved target referring to video image center (see Fig.2). It is desired to use a real time method determining the evolution to zero of the differences between pattern centroid and video image center. Obviously, the correction signals $(c_x, c_y)$ are proportional with these calculated differences.




actual display mode error calculus

Fig.2: Display of ATTR system

Accordingly, the basic structure of the algorithm that ensures the target tracking consists of:

– the calculus of the target centroid, $(dx, dy)$;

– the calculus of the differences between this centroid and video image center, $(d_x, d_y)$;

– the calculus of the associated correction signals, $(c_x, c_y)$.

The coordinate system used in the above calculus has its origin in the video image center (see Fig.2), and for effective determination of the correction signals, an improved version of standard RBF neural network will be used.

Fig.3 depicts a possible way for initial positioning of RBF network receiving fields by σ variance (in our case, we chose nine centers). The vector $\mathrm{d} = (d_x, d_y)$ represents the characteristic input for neural network, and the response of hidden neuron *m* is expressed by the equation:

$$y_m = f(d_x, d_y) = \exp\left(\frac{-\left\|\mathrm{d} - \mathrm{w}^{(m)}\right\|^2}{2\sigma^2}\right), \quad m = \overline{1, M} \quad (1),$$

where the notations used are consacrated [12]. The network output is the vector $c = (c_x, c_y)$ of correction commands.

According to [11], in order to increase the tracking accuracy, the RBF neural network must use at least *two* feed-back inputs. The followed strategy is to direct the current positioning errors of target to zero by using as input vector assigned to $r$ cycle, the $r_0$ feed-back errors. Accordingly, the error vector can be written as it follows:

$$e = \left( dx^{(r)}, dy^{(r)}, dx^{(r-1)}, dy^{(r-1)}, \ldots, dx^{(r-r_0)}, dy^{(r-r_0)} \right), r_0 > 1 \quad (2).$$

Consequently, RBF neural network will be trained to make a correct transformation from the space of error vectors $e$ to the space of correction signals $c$.

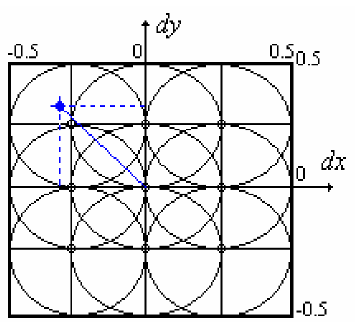



Fig.3: Positioning way of RBF network centers

During of RBF network training process, the selection and training of centers are done by using an unsupervised training procedure that is underlain on the algorithm proposed by Fritzke [13]. This algorithm allows the automatic determination of the RBF number and parameters by using a growing cell process that can be stopped when the performances of RBF network are satisfactory.

After the training phase, RBF neural network will be able to control the position of system video sensor. The neural weigth values to output layer are random initialized, and $\sigma$ has a constant value. It is important to say also that neural network is trained having as primary objective the simultaneous fitting of center positions and weigth values to output layer. In order to do that, it was used the minimization of ***sum-squared error*** (SSE) along the target trajectories, error given by equation:

$$\mathrm{SSE} = \sum_{(x,y)\in \Im} \left( dx^2 + dy^2 \right) \to \min \quad (3).$$

According to [10], the most important ***advantages*** given by RBF neural network usage comparative with the case of standard MLP neural networks, led to the choice of this neural architecture as possible solution for implementation of the ATTR tracking function. Generally speaking, these advantages can be formulated as it follows: the training times are lower, the absence of local minimum points, the influence of atypical points is diminishing, oportunity to use fast training algorithms, relative easy ways for VLSI hardware next implementation etc.

More details regarding neural procedure proposed for implementation of the tracking function can be found in [13] and [14].

*B. Recognition function*

According to [13], by view of recognition block, the three basic ***working*** regimes of video ATTR system can be synthetized as it follows:

– ***dependent*** working regime when the system is able to recognize only the targets stored into its internal database. Naturally, the neural classifier that will be used in this situation is one ***supervised***. The information from database are used in the training and testing phases of each neural classifiers;

– ***independent*** working regime when the system is able to recognize all the targets from its field of view. In this case, a neural ***clustering*** technique is more suitable to be used. The information from database are used only to assign a correct label or purport to each cluster descovered into input space;

– ***semi-independent*** working regime when the neural classifier is by ***hybrid*** type. The information from databse are used only to assign a proper label to each input cluster, too.

Consequently, the modalities used for video database design, and for proper implementation of the recognition function will be discussed in the next sections. All these applications had as starting point the current working regime assigned to ATTR recognition subsystem.

*B1. Database design*

The logical diagram used to generate the input video database for supervised neural network training and testing, is presented in Fig.4.

As it can be seen from Fig.4, the video database used in this application was obtained from a digital photographical survey of ***five*** military aircraft models (see Table 1) scaled at 1:48. The survey was taken using a $5^0$ increment in the azimuthal plane, using a range of $\left[0^0, 180^0\right]$ justified by the geometric aircraft shape simmetry.

Each image from the input video database has a resolution of 520×160 pixels, in an uncompressed BMP format.

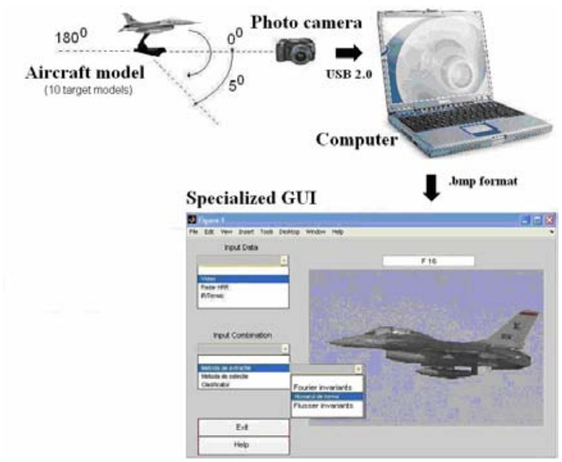



Fig.4: Logical diagram used for database design

After the acquisition and preprocessing step, a number of 37 video images per class is obtained. As feature extraction method, the invariant set described in [15] was used. According to this reference, the used invariants represent an ***improved*** version of the standard Flusser moments. Also, the option to use this set was determined by the important ***properties*** offered by these descriptors, namely: invariance at elementary geometric transformations, robustness to the action of noisy factors inside of a recognition system, better than the usage of standard Flusser moments, form an independent base of pattern invariants etc.

Consequently, after applying of the feature extraction method, the feature vector matrix has the dimension of (11×37) for each input model class.

More details about video database design can be found in [11] and [13].

*B2. Case of dependent working regime*

In this case, a set made by *three* supervised neural classifiers, i.e. {MLP, SART (*Supervised ART*) and FuNN (*Fuzzy Neural Network*)}, was used for classification purposes.

It is known that SART classifier is developed using q* standard algorithm [16]. A set of prototypes, which approximates the probability density modes of the vectors of each input class, was used. The NN (*Nearest Neighbour*) classification rule is then used to classify the new vectors as compared to these prototypes. Generally speaking, this classifier generates in a *supervised* manner a new prototype if the distance between the new vector and the existing prototype exceeds a certain threshold (according to *Follow the leader* principle from ART artificial neural network theory).

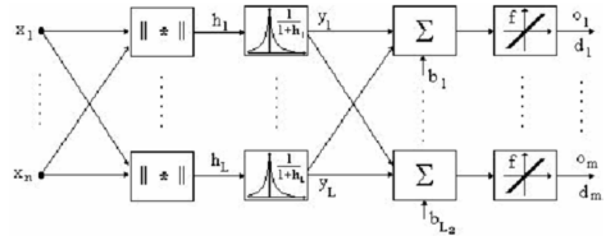


Fig.5: Internal structure of SART classifier

According to [10], FuNN networks proposed by Yamakava and Uchino, are *multilayer feedforward architectures* that achieve a set of fuzzy rules and an engine of fuzzy inferences using a connexionist manner. The basic architecture of these neural networks contains *five* neural layers (see Fig.6), namely: first neural layer gets the outside information, second layer calculates the degrees of fuzzy membership, third layer elaborates the set of fuzzy rules, fourth layer calculates the matching degree between the input data and the output membership functions and finally, fifth layer achieves the defuzzyfication and value calculus of the output variables.

Consequently, FuNN networks have some important *advantages* in comparison with the standard connexionist or fuzzy systems: accuracy and promptitude in their training process, very good capacity to generalize, the mixed character of the training process, increased robustness to the obliviousness phenomenon of the patterns already learned etc.

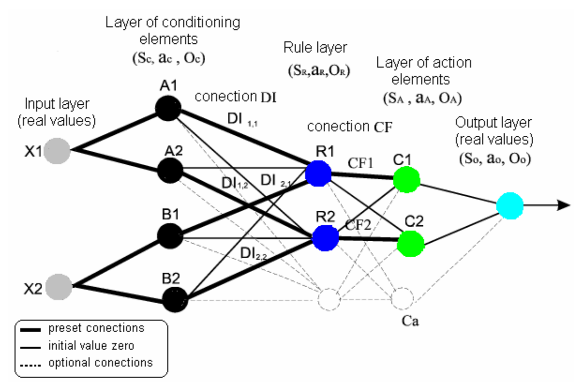



Fig.6: The basic architecture of FuNN networks

In order to design and optimize the used FuNN architecture, the algorithm proposed by Watts and Kasabov was applied [10].

More details about topology and training rule assigned to SART and respectively, FuNN networks can be found in [10] and [13].

In order to increase the classification/recognition performances (as main performance indicator the *classification rate* (CR) has been computed. It is well known that the significance of CR (in %) is the ratio between the number of correct classified input patterns and the total number of the patterns used), we applied a *decision fusion* method based on fuzzy-evolutive integral usage.

According to [16], the *fuzzy-evolutive* integral (see Fig.7) is a hybrid method used to optimize the mixing mode of the outputs assigned to more (neural) classifiers. This procedure uses the Sugeno's fuzzy integral to realise the suitable output combination of some distinctive neural networks based on importance assigned those by a proper genetic algorithm.

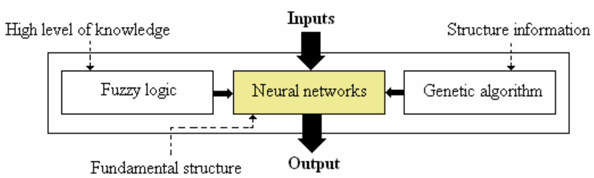



Fig.7: The basic diagram of fuzzy-evolutive integral

In order to finalize the database structure, after feature extraction step and for each input class, an interlacing splitting algorithm was applied on resulted matrix. Accordingly, a number of 19 feature vectors were used for classifier training while for testing 18 vectors were used.

The logical diagram used for implementation of the recognition function in case of dependent working regime for ATTR system, is synthetically shown in Fig.8.

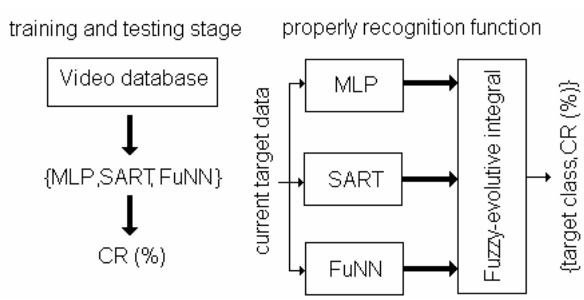



Fig.8: Logical diagram used in case of ATTR dependent working regime

Generally speaking, the applying of proposed decision fusion method is possible because its calculus is not a very expensive time process, and the property of *real time* for ATTR system can be thus preserved. According to [10], the option for the above neural classifier set can be explained by very good CRs that were obtained using the avaible video database.

More details about this classification diagram can be found in [10] and [13].

*B3. Case of independent working regime*

As we mentioned in the beginning, in this case is properly to use a neural (unsupervised) clustering techique. Consequently, starts from the theoretical and experimental results indicated in [10], authors proposed for usage a particular neural architecture belong to FOSART (*Fully self-Organizing Simplified ART*) class.

It is well known that FOSART neural networks are *on-line* training algorithms combaining the properties of SOM, GNG (*Growing Neural Gas*), FCM (*Fuzzy c-Means*), FART (*Fuzzy ART*), and FSART (*Fuzzy Simplified ART*) architectures. According to [13], FOSART network uses a fuzzy membership function derived from one used in case of FCM and FSART algorithms in order to calculate the values of activation function. Also, this network uses a function by GBF (*Gaussian Basis Function*) type in order to calculate the absolute (possible) values by fuzzy.

Generalizing the competitive training strategy used by GNG algorithm, FOSART supposes that entire output map whereupon a winner unit belongs, represents a resonant vicinity of this. In the same manner as GNG networks, FOSART algorithm uses an improved version of the competitive training rule by hebbian type in order to generate synaptic conections using a training example. Also, these conections are eliminated using a uncontinuous parametric adaptation scheme. Generally speaking, it can be considered that FOSART neural networks belong to class of FSONN (*Fully Self-Organizing artificial Neural Network* ) conexionist models.

More details about training rule assigned to FOSART networks can be found in [10] and [13].

Consequently, related to specific requirements of ATTR system class, the basic *advantages* offered by using of these networks can be formulated as it follows: increased robustness at the action of the perturbation factors, self-fitting possibility of its topology in concordance with the complexity of the clustering task, lower calculus effort in relation to reference algorithms, classification performances better than standard clustering techniques etc.

In our application, in order to train the FOSART network, the algorithm indicated in [10] was used.

The logical diagram that describes the modality used to implement recognition function in this ATTR working regime, is illustrated in Fig.9.

In order to understand this logical diagram, it is important to know that after classification stage at the output of FOSART network will obtain CRs for each input class, and unlabeled data clusters, respectively. Consequently, to assign a proper label or purport for each descovered cluster, we calculated the centroids of each data clusters, and compared there with the known class prototypes stored into database. Thus, the target identification process is accomplished.

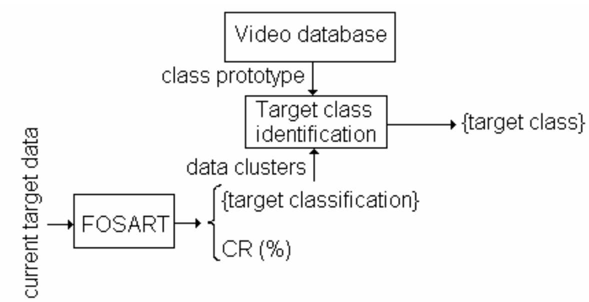



Fig.9: Logical diagram used in case of ATTR dependent working regime

More details about this classification diagram can be found in [10] and [11].

*B4. Case of semi-independent working regime*

In this regime, we used a hybrid neural classifier. It is well known that at macro-stuctural level, a hybrid neural network is made by joining an unsupervised, and supervised topology, respectively (see Fig.10).

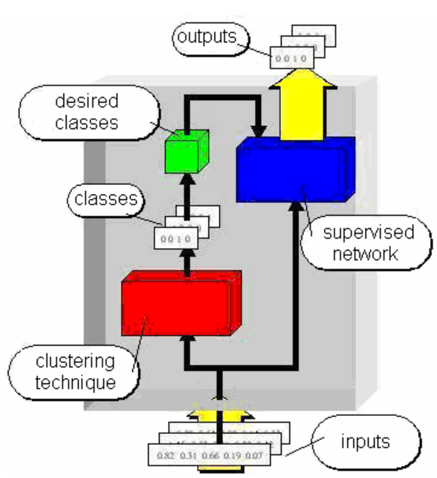



Fig.10: General structure of hybrid neural network

In our study case, we chose to use a hybrid architecture by FOSART-SART type. Accordingly, the logical diagram that describes the working algorithm of this neural hybrid architecture is depicted in the Fig.11.

As it can be seen from Fig.11, the FOSART network will help us to identify the main clusters from input dataspace, and in the training process assigned to SART classifier. The video database is used also only for target class identification, in order to obtain the desired class for SART training step.

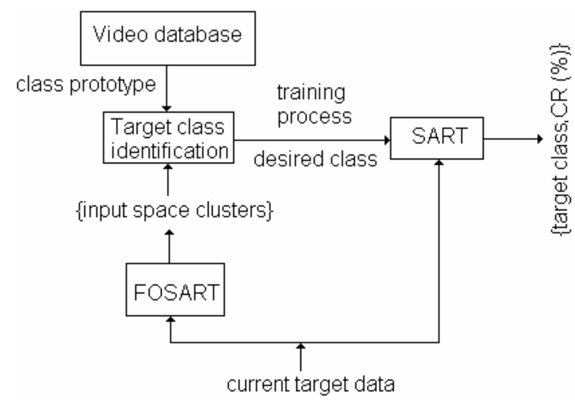



Fig.11: Logical diagram used in case of ATTR semi-dependent working regime

More details about this classification diagram can be found in [10].

Consequently, as a conclusion, in this moment we dispose of concrete procedures for a *full neural* network implementation of the two basic functions assigned to video ATTR systems.

## 3 Experimental results

The main *objectives* of these experiments are:

– to demonstrate that the usage of the proposed neural algorithm to implement the tracking function of an ATTR system is a correct option;

– to demonstrate that the usage of the proposed neural networks to implement the recognition function results in more increased classification rates than in other similar applications indicated in specialized literature.

In order to analyse and quantify the RBF network tracking abilities, the *simulation conditions* used were following:

– the neural network topology is determined by the choice of the values for parameters $r_0$ and $M$, so that we chose to use $r_0 = \{1,2,3\}$ and $M = 9$;

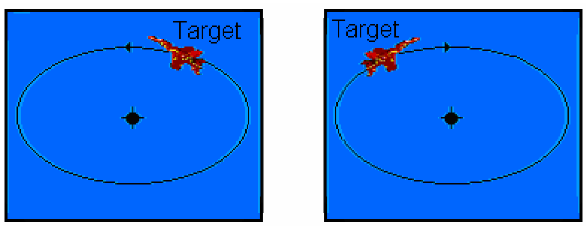
– the initial neural weigths to output layer were randomly initialized in the range of $[-1,1]$;

– the initial positioning of RBF network centers and sensor field of view was similar with one indicated in Fig.3;

– the RBF variance $\sigma$ was chosen constantly, and for a stable working regime of the tracking subsystem, this parameter had the value 0.4;

– 25 training cycles were used, and in each cycle, the main task was to fit the center positioning, and the neural weigths to output layer, respectively by using the minimization of SSE error along the whole trajectories of tracked target.

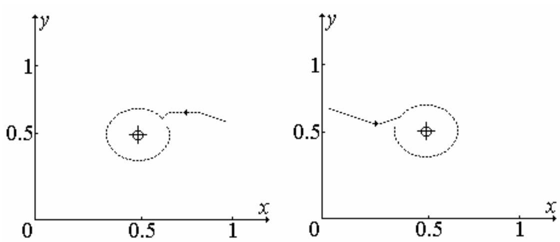
In order to test the tracking performances of RBF neural network, two targets displacing with the great velocity along the similar trajectories with ones indicated in Fig.12, were simulated. As well to occlude the tracking loop inside of ATTR system, the video sensor was simple modellated, i.e. the assigned correction signals $\left(c_x, c_y\right)$ will determinate the moving of its LOS axis with angles $\left(\theta_x, \theta_y\right)$ which are proportional with maximal moving angles $\left(\theta_{x_{\max}}, \theta_{y_{\max}}\right)$ of its cardanic subsystem.




anti-clockwise rotation clockwise rotation

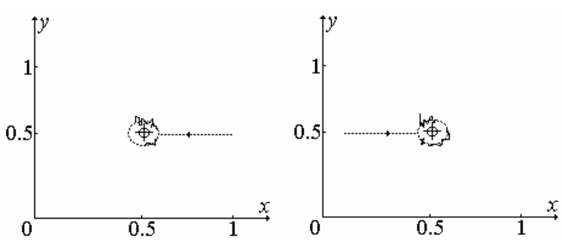
Fig.12: Target trajectories used in simulation

Using simulation conditions from above and the target trajectories depicts in Fig.12, the *dynamic positions* of the target relative to video sensor LOS are suggestive illustrated in Fig.13.



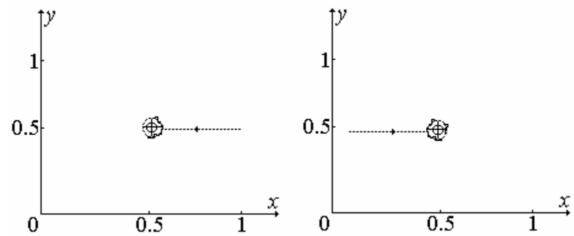
anti-clockwise rotation clockwise rotation

(a) case of $r_0 = 1$



anti-clockwise rotation clockwise rotation

(b) case of $r_0 = 2$



anti-clockwise rotation clockwise rotation

(c) case of $r_0 = 3$

Fig.13: Dynamic positions assigned to the two simulated target

On the other hands, the average *training time* that is necessary in the two simulated situations has a value less than 0.1 s, and increases simultaneously with the increase of the reaction order used.

More details about the RBF network simulation can be found in [10].

Another interesting point that was treated in this paragraph refers to CRs (and other performance indicators) which have been obtained in case of neural implementation of ATTR system recognition function. Naturally, the classification results are indicated in concordance with the working regimes assigned to ATTR system.

Accordingly, in case of the dependent working regime, and video database usage, the classification results are synthetically presented in Table 1.

Table 1

| **Target class** | **Classification rate, CR(%)** | | |
|---|---|---|---|
| | MLP | SART | FuNN |
| Mirage 2000 | 92 | 92 | 93 |
| Rafale | 91 | 93 | 94 |
| F16 | 93 | 95 | 95 |
| F117 | 96 | 95 | 97 |
| Tornado | 94 | 95 | 94 |
| Harrier | 96 | 94 | 98 |
| Eurofigther 2000 | 98 | 96 | 100 |
| Suchoi 35 | 92 | 94 | 95 |
| Mig 21 | 95 | 95 | 96 |
| Mig 29 | 91 | 94 | 94 |
| **Average** | **93.8** | **94.3** | **95.6** |
| **Fuzzy-evolutive integral** | **98.5** | | |

Classification results obtained in the first working regime of ATTR system

As one can see in Table 1, after fuzzy-evolutive integral applying results higher values of CR than the individual performances of neural classifier used. Also, after decision fusion usage, a snapshot of its output value variation is presented in Fig.14.

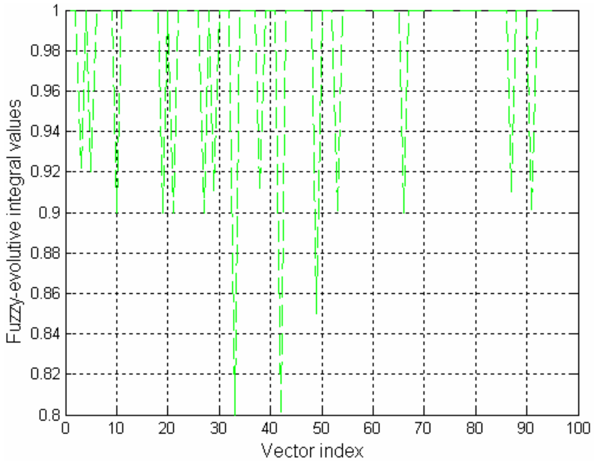
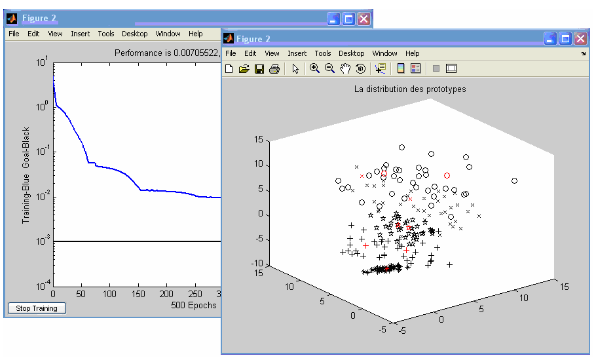
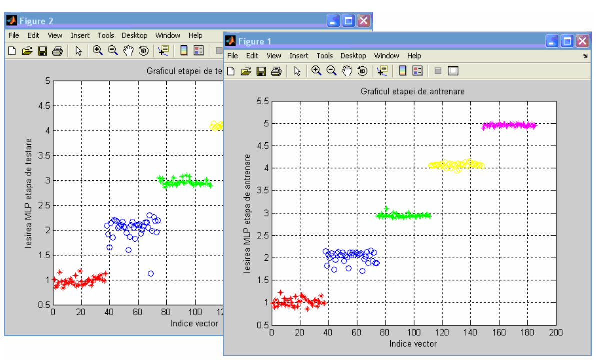



Fig.14: Fuzzy-evolutive integral value during decision fusion

Also, Fig.15 depicts some partial classification results obtained in case of training and testing phases assigned to SART neural classifier.




(a) training and testing phases




(b) visualization of the classification results

Fig.15: Some classification results obtained in case of SART neural classifier usage

It is known that in case of independent working regime, FOSART neural classifier was used. In order to quantify the classification performances, the testing dataset was obtained by mixture of data belong to the same five input classes. Adding to the data cluster structure generated by FOSART classifier usage the information about each class prototypes assigned to avaible video database, the classification results are indicated in Table 2.

Table 2

| **Target class** | **Classification rate, CR(%)** |
|---|---|
| | FOSART |
| Mirage 2000 | 88 |
| Rafale | 90 |
| F16 | 91 |
| F117 | 94 |
| Tornado | 90 |
| Harrier | 94 |
| Eurofigther 2000 | 94 |
| Suchoi 35 | 90 |
| Mig 21 | 91 |
| Mig 29 | 90 |
| **Average** | **91.2** |
| **Running parameters** | Training time=2.1 s; Vigilance threshold, $\rho = 0.8$; Min-max synapse threshod, $l_r = 2.5$; Max number of testing cycles, $n_{\max} = 50$ |

Classification results obtained in the second working regime of ATTR system

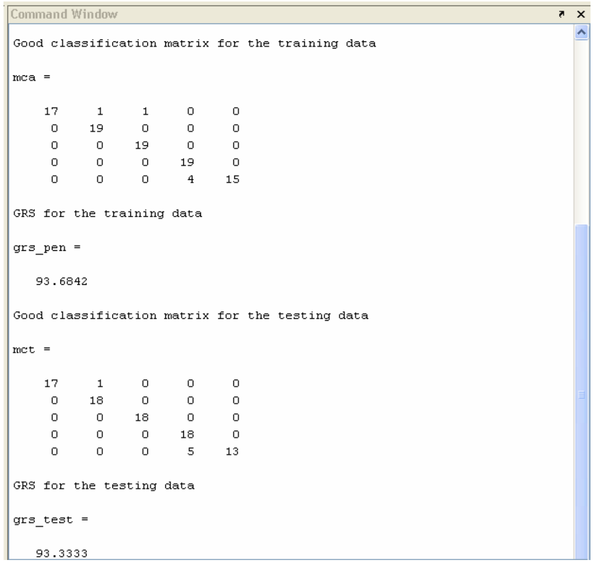
As one can see in Table 2, the usage of FOSART neural classifier leads to very good values of CR taking into account that we tested an unsupervised neural network, but these values are indeed less than ones obtained in the supervised study case. These classification rates are also positive comparable with ones indicated in another ATR applications using neural clustering techniques [13].

In the last working regime of ATTR system, using a similar way to design the input dataset, and a hybrid neural network based on FOSART-ART combination usage (see Fig.11), the classification results are synthetically indicated in Table 3. Also, Fig.16 depicts an example of partial classification results (CR) obtained in this study case.

The rules used to mix the two classifier results are specific to hybrid classification techniques, and are largely described in [10].

The classification results indicated in Table 3 allow us to conclude that these are very good, and indeed comparable with ones presented in another pattern recognition applications [8] and [9].

All applications presented in this section were developed using MATLAB™ *nnet* and *images* toolboxes.



```
Command Window

Good classification matrix for the training data

mca =

    17     1     1     0     0
     0    19     0     0     0
     0     0    19     0     0
     0     0     0    19     0
     0     0     0     4    15

GRS for the training data

grs_pen =

   93.6842

Good classification matrix for the testing data

mct =

    17     1     0     0     0
     0    18     0     0     0
     0     0    18     0     0
     0     0     0    18     0
     0     0     0     5    13

GRS for the testing data

grs_test =

   93.3333
```

Fig.16: A MATLAB *command window* capture

Table 3

| **Target class** | | **Classification rate, CR(%)** |
|---|---|---|
| | | FOSART-ART |
| Mirage 2000 | | 92 |
| Rafale | | 93 |
| F16 | | 94 |
| F117 | | 95 |
| Tornado | | 92 |
| Harrier | | 95 |
| Eurofigther 2000 | | 94 |
| Suchoi 35 | | 93 |
| Mig 21 | | 92 |
| Mig 29 | | 93 |
| **Average** | | **93.3** |
| **Running parameters** | FOSART | Training time=1.8 s; Vigilance threshold, $\rho = 0.8$; Min-max synapse threshod, $l_r = 2.5$; Max number of testing cycles, $n_{\max} = 50$ |
| | SART | Training time=3.1 s; Goal=$10^{-4}$; Maxepochs=10; Maxprototypes=10 |

Classification results obtained in the third working regime of ATTR system

More details regarding experimental aspects treated in this section can be found in [10], [11], [13] and [15].

## 4 Conclusion and Future Work

The theoretical and experimental results presented in this paper leads to the following *remarks* concerning the proposed neural design procedure for a video ATTR system:

– referring to *tracking function*:

1) the neural procedure proposed for tracking function allows a real time implementation of this, with very good tracking performances;

2) accordingly, this choice of RBF network for target tracking is one correctly. In order to have a very fast and diminishing oscillated tracking process, it is necessary that order of the reaction error to be great than one, but for decreasing training times, this order must be less than three.

– referring to *recognition function*:

1) the classification results obtained by using of tested neural networks are also very good, and generally more than 98% in case of supervised network usage, 90% in case of neural clustering technique, and 93% in case of hybrid neural network, respectively;

2) all proposed neural network allow for ATTR system a real-time implementation of its recognition function. This conclusion is confirmed once again by the hardware models existing in specialized literature for these neural architectures [11];

3) as a result of applying this neural design procedure, the classification rates are similar with ones indicated in literature [5], [8], [10] and [11], i.e. generally more than 90%.

Summarizing, an ATTR system that will use the neural design algorithm proposed in this study will become a *smart* system, will be more *effective*, and will have improved functions than standard systems.

The neural design procedure described in this paper is intended to be extended for the case of *noisy* and *variable resolution* images used in input database design.

It is very important also to analyze and quantify the positive influence on ATTR system performances of *hardware implementation* (the logical step after neural network simulation) of neural techniques tested.

Another interesting point for a further development refers to the design of entirely *new* neural tracking and classification algorithms, which means not only more improved neural networks for tracking and recognition, but for feature extraction and selection, and for 2D or 3D target classification using different types of imagery, particularly video imagery.

Finally, in the same scientific research direction it will be interesting to make a thoroughly experimental analysis concerning the neural design procedures of ATTR system functions using *other types* of input sensors (e.g., IR, thermal or video $L^3$).

*References*:


[1] W.H. Licata, "Automatic Target Recognition Beyond the Year 2000", *RTO SCI Lecture Series on Technologies for Future Precision Strike Missile Systems*, Stockholm, Sweden, RTO-EN-018, 2001

[2] H.N. Nasr, *Automatic Target Recognition: Principles, Algorithms and Applications*, Course at University of Houston, USA, 2005

[3] C. Arcelli, *Visual form analysis and recognition*, Plenum Press Printing House, New York, 2002

[4] A. Mahalanobis and F. Goudail, "Methods for Automatic Target Recognition by Use of Electro-Optic Sensors: Introduction to the Feature Issue", *Applied Optics*, vol. 43, pp. 207-209, 2004

[5] A. Nicula, *Missile autoguidance systems*, MTA Publishing House, Bucharest, 2003

[6] Integrated Sensors Inc., *Video tracking and recognition capabilities*, 2005

[7] P. Ragothaman, T. Yang and W. B. Mikhael, "Automatic Target Recognition Using a Novel Adaptive Technique for Rayleigh Quotient Quadratic Correlation Filters", *WSEAS Transactions on Signal Processing*, issue 7, vol.2, pp.949-955

[8] M. Roth, "Survey of neural network technology for automatic target recognition", *IEEE Transactions on Neural Networks*, pp. 28–43, 1990

[9] G.C. Vasconcelos, P.J. Adeodato, and E.C. Carvalho Filho, "Neural network based automatic target processing and recognition", *Proc. of Int. IEEE Conf. on Image Processing and Its Applications*, vol. 1, pp. 280-284, 1999

[10] I.C. Vizitiu, "Video Pattern Recognition using Neural Networks", Ph.D. thesis, MTA, Bucharest, Romania, 2003

[11] I.C. Vizitiu and A.Stoica, "An application of the neural networks to target tracking and recognition using video images", *Proc. of Int. Conf. on Defence Technology*, Budapest, Hungary, pp. 231-236, 2004

[12] H.K. Sulehria, Y. Zhang, "Hopfiled Neural Networks – A Survey", *Proc. of 6th Int. Conf. WSEAS Artificial Intelligence, Knowledge Engineering and Data Bases*, 2007, Corfu Island, Greece, pp.125-130

[13] I.C. Vizitiu, *Neural networks used in video pattern recognition*, MTA Publishnig House, Bucharest, 2005

[14] R.O. Duda, P.O. Hart, and D.G. Stork, *Pattern Classification*, John Wiley&Sons, New York, USA, 2001

[15] I.C. Vizitiu, D. Munteanu, "A new version of Flusser moment set for pattern feature extraction", *Proc. of 6th Int. Conf. WSEAS CIMMACS*, 2007, Tenerife, Spain, pp.456-461

[16] I.C Vizitiu, I. Nicolaescu, P. Ciotirnae, "Target recognition improvement using the decision fusion between HRR and video imagery", *Proc. of Int. Conf. NAUN Communications and IT*, 2008, Marathon, Greece, pp.234-239