Evolutive Neural Fuzzy Filtering: An Approach

[1]J. C. García Infante, [2]J. J. Medel Juárez, [1]J. C. Sánchez García

[1]Professional School of Mechanical and Electrical Engineering
Av. Santa Ana No. 1000, Col. San Francisco Culhuacan
Del. Coyoacan, México D.F.

[2]Centre of Computing Research, National Polytechnic Institute
Vallejo D. F. 07738, México
jcnet21@yahoo.com, jjmedelj@yahoo.com.mx, jcsanchezgarcia@gmail.com

Abstract: - The paper, is a description of the evolutive neural fuzzy filtering with real time conditions, giving the basics of its operation based on a back propagation fuzzy neural net, which adaptively choose and emit a decision according with the reference signal changes in order to loop the correct new conditions for a process. This work is an approach about the operation of the evolutive neural fuzzy digital filters (*ENFDF)*. Using the neural fuzzy mechanism select the best parameter values into the knowledge base (*KB*), updating the filter weights to give a good answers with respect to the desired signal in natural linguistic sense. Additionally, the filtering architecture includes a decision making stage using an inference into its structure to deduce the filter decisions in accordance with the previous and actual filter answer in order to updates the new decision with respect to the new reference system conditions. The process requires that all of its states bound into *ENFDF* time limit as a real time system. In this paper, the characterization of the membership functions building the knowledge base in a probabilistic way with respect to the rules set in order to describe the reference system and the inference to selects the new filter decision. Moreover, the work describes in schematic sense the neural net architecture with the decision-making stages in order to integrate the filtering stages as an evolutive system. The results expressed in formal sense using the concepts into the paper references. Finally, we present the simulation of the *ENFDF* operation using the Matlab© software. The paper has eight sections conformed as follows: 1. Introduction, 2. Filter description, 3. Neural net structure, 4. Decision stage, 5. Rules strategy, 6. Real time constrains, 7. Simulations, 8.Conclusion and References.

Key-Words: - Digital filters, neural net, evolutive systems, fuzzy logic, inference mechanism, real time.

1 Introduction

The complex natural life, which is discovering through time in order to be apply its characteristics into the technological advances, as example in electronics, systems and communications. All of them based on natural systems operation and its different kinds of intelligence, which is being structuring in accordance with the natural processes as biological systems and the thinking process like artificial intelligence.

There are different tools as artificial neural networks, evolutive processes and fuzzy systems. In order to develop novel filtering systems with more reasoning capacities, that could infers its external condition when interacts with a reference system in order to give an specific good response in accordance with a decision maker into its internal processing architecture in order to obtain an autonomous capacities with different levels.

Some problems for develop a system as the restrictions into its capacity to model high evolutive changing processes. The new systems should use into its architecture evolutive tools in order to get its own perception and give the best decision answers, using this to solve complex problems, actualizing and adapting its perceptions and answers in accordance with a reference dynamical system.

The use of learning techniques based on searching the optimal solutions represents a classical alternative manner to obtain knowledge. The evolutive systems into the neural fuzzy digital filtering is an option for to obtain different decision answer levels, in accordance with a system dynamics or reference model, adapting its answers to the possible changes by selecting the best values in order to get the necessary convergence conditions, which should has the best operation at each time.

An evolutive neural fuzzy digital filter has a solution searching process based on the reference model conditions having at each moment a regulation inference mechanism that describes the desire conditions, selecting the best answer into the knowledge base in accordance with it, emitting the new answer condition for the best adaptation.

The evolutive neural fuzzy digital filtering is constructing with the capacity to operate with different kind of decision answers, considering a set of characteristics based on a minimal reference model region limited by its operation. To characterize and infer a system that has uncertainties in its operation, actually it is a complex task, because many natural processes with different operation levels need accurate answers.

The system that describes the natural process as a rule needs a feedback law in order to follow the basic properties respectively to a desired input signal adjusting its parameters in order to give a correct solution using the dynamics conditions.

Having this perspective, the paper integrates the concept of Real time neural fuzzy digital filtering [6] and [15] with statistical methods in order to integrate them into the Kalman filter structure to give answers with operation levels in natural way making an specific decision in order to follow the natural reference model [5], and [16].

An evolutive system with artificial neural network properties is a computational model imitating natural processes as the biological systems that has processing elements called neurons, all of them being connected constituting a neural structure and a decision stage to loop the process with the best answer [19], and [25].

The evolutive digital filter has an internal neural net structure that can classify, search and associate information [5] and [8] using the knowledge base giving the corresponding answer value according to the desired input signal building the control area described as $T_N = \{(y(k), \hat{y}(k))\}$ [1] within a membership intervals limited inside a region described by the knowledge base. The set of membership function takes the correct response from the Knowledge Base (*KB*) [8], [10], [13], and [26] according to objective law in order to take the best decision process, predefined by this natural reference.

The neural net filter stage [1] works as a parallel fuzzy neurons in loop form, which has an iterative searching methodology used for evolutive algorithms and based on the back propagation (*BP*) algorithm since its parameters are updated dynamically ([2], [4], [7], and [15]) at each iteration by degrees [8]. This process refers to a back propagation parameter adaptation [21], using supervised learning by the knowledge base according with the error *e(k)* ([6], [7], and [14]) described by the difference between the desired response *y(k)* and the actual signal $\hat{y}(k)$ ([6], [7], [15])[1].

The criterion described as the error minimization *e(k)*[2] is the difference between the desired input *y(k)* and the actual output filter $\hat{y}(k)$, allowing this to find the respective membership function. Which is the best approximation from the signal $\hat{y}(k)$ to *y(k)* in order to adjust the parameters of the filter and get the correct answer describing the reference system path in order to take the decision into the decision stage. The error value *e(k)* should be close to γ that is a limited interval [0, ε] and ε is described as $\inf\{|e(k)| : i, k \in Z_+\}$ (i represents an index, with k interval [13], and [18])[3].

Each decision stage is in accordance with the actual and previous values of $\hat{y}(k)$ described as $\hat{y}(k-1)$ doing a comparison between these values in order to select the corresponding instruction answer to the reference system, all the instruction answers are previously defined and using the logic connectors *IF-AND-THEN* the filter selects the corresponding instruction to actualize the reference system conditions:

a. *Input Fuzzy Inference*: In this stage, the natural desired signal *y(k)* from the reference system to the input filter has a description in metric sense [4].

[1] $T_N : Y_N \times \hat{Y}_N \to \{(y(k), \hat{y}(k))\}\big|_{k=1}^{N}$

[2] $e(k) := y(k) - \hat{y}(k)$, is a fuzzy value.

[3] Inf is the greatest lower bound error value into the error set.

b. *Rule base*: Dynamical rank intervals with respect to the input of the filter use the logical binary connector known as *IF*.

c. *Inference* : The expert action with respect to the rule base known as consequence uses the logical binary connector *THEN* to find the correct weight answer as $\hat{a}(k)$ called membership function in accordance with its respective variable.

d. *Activation function:* This is the filter stage, which takes the digital answer, converted in a natural response. This is the closest distance value to the desired signal, based on the predefined knowledge base.

e. *Natural feedback:* the filtering process takes the correct linguistic value and feedbacks the filter parameters, updating its operation according to the natural evolution of the reference system considering the error differences between $y(k)$ and $\hat{y}(k)$ dynamically and using a metric rank of the error functional $J(k)$[4].

f. *Decision stage:* Finally, the filter decision mechanism takes the best answer option according with the values of $\hat{y}(k)$ and $\hat{y}(k-1)$ to actualize, the decision described as $ds(k)$.

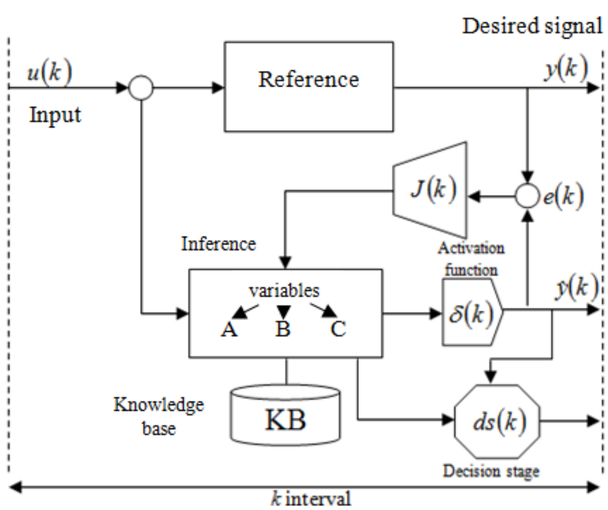

Fig. 1. Evolutive neural fuzzy filter process

[4] The functional $J(k)$ describes the convergence relation among the real observed and its estimation; symbolically in recursive form: $J(k)=\frac{1}{k}\left[\left((k-1)J(k-1)^2+e(k)^2\right)\right]^{\frac{1}{2}}$.

2 Filter description

The evolutive neural fuzzy filter (see: Fig. 1) requires a knowledge base (*KB*); because, it has all the possible responses in accordance with the reference system or process in which the filter is interacting. The evolutive neural fuzzy filter has its state variables bounded by a rank of values described by fuzzy sets (for example, the temperature classified by the fuzzy logic as high, medium or low (see: [1], [3], [11], [21], [23] and [24]).

This kind of filter adaptively modifies its response $\hat{y}(k)$ according to the dynamic changes at the filter desired input *y(k)* from the reference process alteration (produced by neural excitation). Nevertheless, it requires that the error loop inference (comparison between desired input *y(k)* and actual output $\hat{y}(k)$) changes the filtering responses using the knowledge base (*KB*) with real time conditions (see: [2], [6], and [7]).

The evolutive neural fuzzy filter classifies its different levels of operation by membership functions, based on the error minimum function J_{min} to give a corresponding near specific answer for each kind of desired input *y(k)* limited by the error distribution function, which is also limited by γ. The classification of the reference system responses realized by the filter is by the error distribution function considering different levels of response (membership functions), each level delimited by specific metrics bounded all into the error distribution region. This classification represents the characterization of the reference process operation in interaction with the filter (see: [9], and [13]).

Using this kind of filters avoids the initial instability of the error convergence because it has all information required and limited by an interval into a distribution region as supervised learning technique.

To do a classification of the membership functions the filter uses an error criterion function using metric intervals to delimit its operational levels. A membership function established inside a distribution function corresponding to the error criterion; this classification includes also several membership functions according to each error interval (as operation linguistic levels).

This classification based on the triangular form [23] gives different and simple conditions, allowing to express multiple functions in an iterate form and describing separately each of them as operation level (see: [3], [4], [10], [11], [12], [13], [17], and [23]).

Meanwhile, respecting to the Fig. 1, $u(k)$ is an input natural linguistic value from the reference process interacting with the filter. $T(k)$ is the control area, which is all the filtering processes corresponding with an individual iteration of a process set, in order to obtain an output response $\hat{y}(k)$. The desired signal described as $y(k)$ is the fuzzy natural reference variable given by the reference system. $e(k)$ is a fuzzy value where the set $\{\gamma_i : \gamma_i > 0, \ \forall i \in Z_+\}$ has $inf\{\gamma_i\} \to |\lambda|, |\lambda| > 0$ and the $sup\{\gamma_i\} \to |\lambda|, |\lambda| < 1$[5], with $\gamma*$ that must be the minimum value between $y(k)$ and $\hat{y}(k)$, thus, both values should be almost equal; in other words, $e(k)$ is the difference between $\hat{y}(k)$, and $y(k)$, being the criterion in order to select the membership function for each case[1].

3 Neural net structure

The ENFDF structure is develop in order to work as a neural net using back propagation properties ([2], [4], [7], [8], [15], and [21]). The neural net characterizes the operational levels of a linguistic desired variable set $\{y(k)\}$, expressing as a basic estimation $\hat{a}(k)$ result[6], in accordance with the inference rules with respect to the error value criterion selecting the corresponding membership function. The neural net works with an activation function representing the neural filter process (see: [5], [8], [16], [19], and [25]).

The characteristics of the neural net is (see: [8], [16], [19], and [25]):

1. The desired input $y(k)$ represents fuzzy labels, and each one has different levels of operation.

2. The weights requiring for the internal adaptive adjustments into the filter parameters to get the correct response expressed by the membership functions renewing the values of $\hat{a}(k)$ parameter from the knowledge base (The $\hat{a}(k)$ value in a conventional filter is a constant parameter).

3. The connection into the filter according to the error differences[1] has a minimum operation criterion, selecting the membership function based on the minimum cost of the filter response.

4. It uses supervised learning into the knowledge base, therefore, the neuron has previously all the information included in T_N.

5. The activation function $\delta(k)$, which represents the filtering transfer function ([8], [19] and [25]).

Globally, according to Figure 1, the filter without the reference process has three neural layers:

The first with $p \times n,\ n, p \in Z_+$ input neurons representing a set of desired inputs $\{y(k)\}$, a single hidden layer with p processing units in which the inference mechanism operates in order to find the respective membership function $\hat{a}(k)$, and one output $\{\hat{y}(k)\}$, that is the answers set of the neural process (see: Fig. 2). The filter additionally has other layer stage called *decision stage* as evolutive condition in order to give control instructions for an external process according with the previous filter answers.

The neurons of the neural net architecture are simple processors, which get information from the first input layer from the outside world as desired signal $\{y(k)\}$ to the next neurons into the hidden filter layer.

Therefore, the neural net structure proceed the result of their previous processing step to nodes in the next higher layers in order to obtain the nearest desired signal value of $y(k)$ at the filter output $\hat{y}(k)$. In consequence the decision stage in the top-layer communicate their decision results to the outside world as $ds(k)$, which its instructions bounded inside the external process operation in order to give the best control value.

Subsequently, a rule-based fuzzy system performs the classification at both stages: filtering and decision; its rules are generated automatically by the evolutive neural fuzzy filtering evaluating features which are difficult to extract or to evaluate directly delimited by the filtering criterion.

[5] Sup is the least upper bound of a partially ordered set

[6] The desired signal commonly has the basic and explicit description as $y(k) = a(k)y(k-1) + \omega(k)$, where a_k is known as stability parameter (see: [5] and [14]), $\omega(k)$ is the perturbation output noise, $y(k)$ is the desired reference signal.

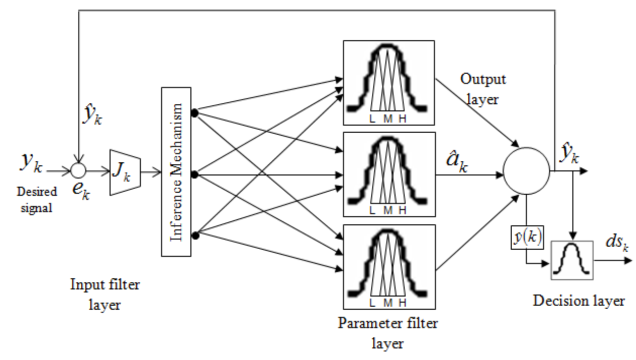

Fig. 2. Evolutive neural filter layers (L-low, M-medium, H-high).

4 Decision stage

The evolutive strategies according with a set of instructions limited to control an external process performance in order to adjust its condition into the best operation state. This is a fundamental strategy to solve problems according with the previous states variations. One of the most important characteristic of an evolutive neural fuzzy filtering is its capacity to adapt its answers into a dynamic environment by its decision stage answer set $ds(k)$.Which selects a condition updating an external process according with an instruction described as $ds(k)$given by the inference of the filter answer $\hat{y}(k)$ and its previous value $\hat{y}(k-1)$.

The ENFDF uses supervised learning by fuzzy inference as training mechanism and characterizing the decision stage response previously by a set of answers in a metric form, given for the process control.

The decision stage has a set of candidate solutions to choose the best control answer, according with the quality of response to solve a problem, the variable $ds(k)$ change its values to match the corresponding instruction of control according with the evolution of the process in order to maintain the performance of its operation that corresponds with the environment changes.

The variable $ds(k)$ as operator creates the diversity needed to give the correct control answer to loop a process to a new condition, this set of answers limited in probabilistic form into a distribution function characterized by its set of decision values, describing different metric levels in order to characterize all its possible answers into membership functions.

The inference decision mechanism infers an answer according with specific change of the environment, in order to use the filtering level answer to get the best control instruction. This mechanism selects the corresponding $ds(k)$ answer, limited into a maximum value and a minimum value as extremes values of its set of characteristic values considered in the process to control in order to get different options to select the best control answer in real time.

The evolutive neural fuzzy filter analyzes the changing characteristics of the environment by the desired signal at the filter input. Probabilistically classified into decisions levels using rules base as a combinations of different variations of each environment characteristic as possible scenario to give the best decision answer as evolutive condition described previously as $y(k)$, which generates the filter output $\hat{y}(k)$ and its previous value in order to selects the corresponding $ds(k)$ answer.
Considering the set of decision answers into the decision stage mechanism, we have the next condition:

$$P(ds_{max} \le ds(k) \ge ds_{min}) = 1 \qquad (1)$$

This forms the set of candidate decisions that the filter choose by intervals scale, according of its total number values, describing the condition as:

$$\sum_{k \in ds} ds_i(k) = \frac{ds_{max} - ds_{min}}{l}, \ \forall A_i \in A, \qquad (2)$$

Where l is the linguistic variables number and $A = \{ A_1 ,...A_l \}$, is the set of it.

5 Rules strategy

This kind of filtering has operational properties defined by the rules base to learn, recall, associate and compares the new information delimited before according to variance limits predefined. The fuzzy rules conditions established as logical connectors (*IF-THEN*) constitute the rules base with respect to the error intervals and its respective response described as membership function.
In an evolutive neural fuzzy filter, the rules generated constitute the inference error $e(k)$[1] as the logical

connector *IF*, and the logical connector *THEN* selecting automatically the parameter weighs of $\hat{a}(k)$ according to the knowledge base (see: [13], [14], [15], [23], [24], and [26]):

The KB has an automatic classification of the filter conditions: having the knowledge of the filtering operation levels.

The KB generates a membership function of the knowledge base according to the error value $e(k)$ (obtained by the logic connector *IF*); thus, according to it, the filter selects the corresponding membership function (value of $\hat{a}(k)$) (with the logical connector *THEN*) into the knowledge base.

Using the KB estimates the parameter value $\hat{a}(k)$, adapting its responses weighs to give a correct answer $\hat{y}(k)$, close to reference model $y(k)$.

The decision stage described as $ds(k)$ selects the best instruction answer to control a process according with the filter answer $y(k)$, and its previous value $y(k-1)$, with the decision inference mechanism by fuzzy rules.

A rules base filter characterized by a set of desired signals $\{y(k)\}$ at the filter input, generates an error difference set *{e(k)}* classified by intervals delimited previously in order to select the corresponding membership function, which has the corresponding $\hat{a}(k)$ value giving a correct response $\hat{y}(k)$ seeking that it is the closest distance to $y(k)$:

$$T_N = \{(y(k), \hat{y}(k))\}_{k=1}^{N} \tag{3}$$

The rules set constituted a simple operation filter definition, based on a set error values *{e(k)}* as indicator of the corresponding membership function set adapting the $\hat{a}(k)$ parameter dynamically.

The neural net represents a set of services, activating a specific neuron means the use of specific service with different levels of response. The filter requires a variable value according to the inference classification of the error set (it has a broad of values per neuron), according to the changes of the error *e(k)* per iteration, one neuron is activating; moreover, the next iteration the filter could renew the neuron or only change its value by degrees. The value of *e(k) defines* a neuron to activate or to select, in order to use its own set of operation levels (membership functions by degrees) to give the corresponding value of $\hat{a}(k)$, and gives a correct natural answer (see: [8], [16], [19], and [25]).

In addition, mathematically correspondence of $y(k)$ and $\hat{y}(k)$, expressed with respect to the second probability moment, has the infimum value, described as:

$$J_{\min} = \inf_{N} \{\min J(y_0, \hat{y})\}_N \tag{4}$$

For the decision stage, which is the finally inference of the evolutive neural fuzzy filter by using a set of fuzzy rules to get the correct instruction of control from its knowledge base (KB). Firstly considering the $\hat{y}(k)$ value and its previous value $\hat{y}(k-1)$ in order to consider the actual environment by the $\hat{y}(k)$ and $\hat{y}(k-1)$ levels and select the corresponding condition or strategy from the knowledge base to loop an external process as $ds(k)$.

To describe the decision stage mechanism, we use fuzzy rules to infer and select the corresponding value of the KB. in natural manner probabilistically this stage has the characterization of the set of structures If-Then directly, the filtering system selects and gives a specific instruction answer as $ds(k)$ in order to improve the control instruction performance minimizing the error process.

The cardinality of the decision stage representation is according with the set values of $\{\hat{y}(k)\}$ and the previous set $\{\hat{y}(k-1)\}$ in order to choose the correct instruction answer into the $\{ds(k)\}$ set, according with this the fuzzy rule structure will have the below form:

If $\hat{y}(k)$ *value is____ and* $\hat{y}(k-1)$ *value is____ Then the* $ds(k)$ *value is ____.*

The figure 3, illustrates the decision stage mechanism as fuzzy rules as:

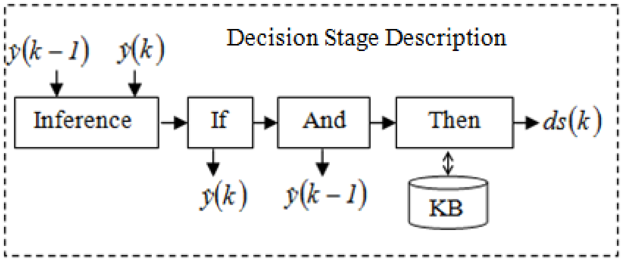

Fig. 3. Decision stage inference

6 Real time constrains

DEFINITION 1 (Real time properties of the evolutive neural fuzzy digital filter - ENFDF). *The real time properties of an evolutive neural fuzzy digital filter, which is an adaptive filter performed according to ([2], [6], [7], and [14]):*

I. To take and to emit signals with fuzzy information sense limited intervals with respect to the reference process response according to the stability criteria (see: [4], [6], and [7]).

II. To take and to emit information through semi-open time intervals [4], synchronized with evolution time of the process [7] described in a relative way by semi-open time intervals, considering the criteria (see: [18], [20], and [22]).

III. A membership functions group forms a control area, according to the properties considered in a) and b) points, respectively (see: [11], [24], and [26]) .

IV. A set of fuzzy rules builds the knowledge base according to the fuzzy desired signal from the reference model $y(k)$, obtaining a specific filter answer $\hat{y}(k)$(see: [3], [8], [9], and [26]).

V. The adaptation algorithm updates the filter coefficients according to the selected membership function respectively t to the established error criterion symbolically depicted as γ_*.

VI. The decision stage mechanism selects the best instruction answer as $ds(k)$, in order to perform the external process operation at each time.

In the evolutive neural fuzzy filter, the knowledge base has all information that the filter requires for to adjust its gains in optimal form and gives a satisfying answer accomplishing the convergence range, inside of the time interval (indexed with $k \in \mathbf{Z}_+$) in agreement with the Nyquist sense, without loss of stability properties [7], [11], [14], [22], [24], and [26]:

i) $y(k)$ is a variable with measurable value and it is classified in metric ranks by degrees in a linguistic sense (described all of them into a state space variable bounded symbolically in a linguistic natural expressions as high, medium or low values),

ii) $T(k)$ is the control area described in pairs formed by $\hat{y}(k)$ and $y(k)$ limited by the time interval (it has a velocity change bounded in the sense exposed by [15]),

iii) $e(k)$ is the fuzzy value defined by the difference among $\hat{y}(k)$ and $y(k)$, which is bounded by the set $\{\gamma_i : \gamma_i > 0,\ \forall i \in Z_+\}$, with $\inf\{\gamma_i\} \to |\lambda_*|$, such that $|\lambda_*| > 0$, $\sup\{\gamma_i\} \to |\lambda^*|$, $|\lambda^*| < 1$, means that $\hat{y}(k)$ is approximately equal to $y(k)$ metrically speaking, nevertheless, in linguistic sense both are the same natural value.

iv) $ds(k)$ is the control instruction answer of the filter according with the output of the filter $\hat{y}(k)$ and its previous value in fuzzy sense, to selects the corresponding candidate function classified in probabilistic sense into an knowledge base to loops an external process into a new condition actualized.

DEFINITION 2 (Local and global description) *An ENFDF in local and global temporal sense, has quality of response according to the convergence criterion[1] with respect to real time conditions [18].*

Global characteristics: *The convergence intervals defined by $[0, \varepsilon \pm \alpha)$ with measures up to zero through error functional J(k) considering [7] and the convergence relation[1], temporally parameterized to the membership function according to the linguistic variables values ([1 and 17]), without loss that /e(k)/<1 in agreement to [7].*

According to the evolutive neural fuzzy concepts, the global characteristics specified in stochastic sense according to [18], where $J(\tau_m) = inf\{min\{\tilde{J}_k\}\} \le \varepsilon$ (see: (3)), with $\{\tilde{J}_k\} \subseteq \{J_k\}$ and $P(\tilde{J}_k \le \varepsilon \pm \sigma) = 1,\ \sigma << \varepsilon$ without loss that its natural evolution described by [22]:

$$\tau_{min} = 0.5 f_{max}^{-1} \qquad (5)$$

7 Simulations

The simulation of the evolutive neural fuzzy filter built in this case using the Kalman filter [7] with a transition matrix described by the knowledge base according to the error functional criterion ([4] and [7]).

The evolution times inside of a soft system into *PC* with *AMD* Sempron processor 3100+ and k intervals having a mean evolution time of 0.004 sec ± 0.0002 sec.

The basic system in discrete state space expressed by first order difference, as:

$$x(k+1)=a(k)x(k)+w(k) \qquad (5)$$

In accordance whit the system the output is:

$$\begin{aligned} & x(k), w(k), v(k) \in R^{n\times 1} \\ & y(k)=x(k)+v(k): \end{aligned} \qquad (6)$$

Where: {x(k)}= is the set of internal states, {a(k)}= is the parameters sequence, {w(k)}= is the noise set system perturbation, {y(k)}= is the set of desired signal from the system reference, v(k)= is the output noises.

The different operation levels are in probability sense in order to match with the functional error[1] according to the desired signal *y(k)* and the Kalman filter answer $\hat{y}(k)$. Bounded all by the second probability moment, establishing for each of them a linguistic natural variables expressed as low, medium and high levels.

According to the parameters $\hat{a}(k)$ selected by the rule strategy, the Fig. 4, shows the output answer $\hat{y}(k)$ of the filter with respect to the desire signal *y(k)* system:

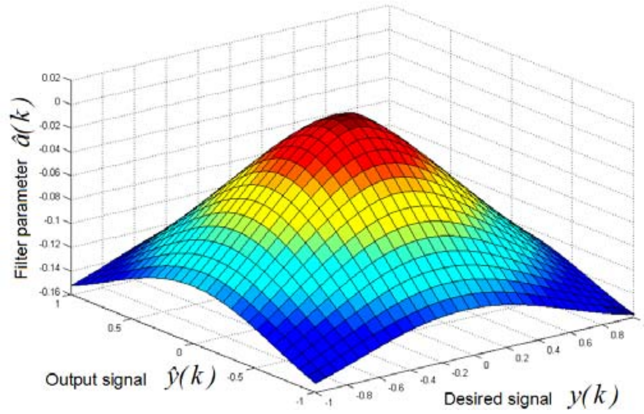

Fig. 4. $\hat{y}(k)$ Estimation in accordance with $\hat{a}(k)$ value and $y(k)$ reference.

Figure 5 shows the decision stage process to obtain the instruction answer as $ds(k)$ according with the $\hat{y}(k)$ values and its previous stage described as $\hat{y}(k-1)$:

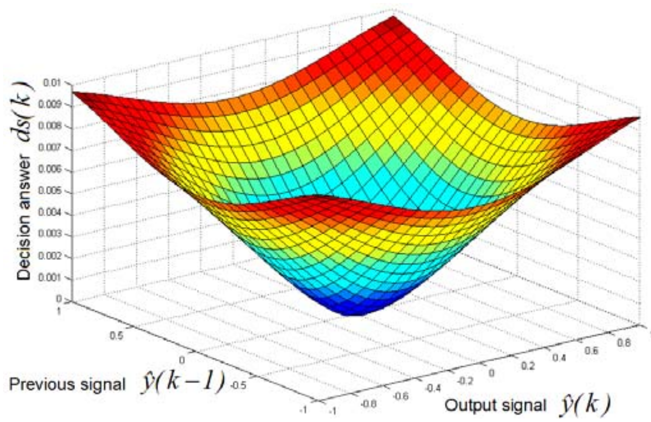

Fig. 5. Decision answer $ds(k)$ with respect to $\hat{y}(k)$ and $\hat{y}(k-1)$.

The figure 6, shows the decision answer stage as decision levels according with the filtering process in order to give different decision answers level described as DA#:

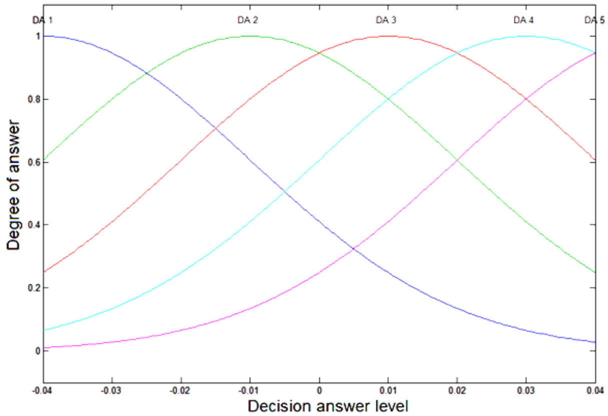

Fig. 6. Decision answer as linguistic levels

Figure 7 shows the functional described as $J(k)$ with respect to the filter:

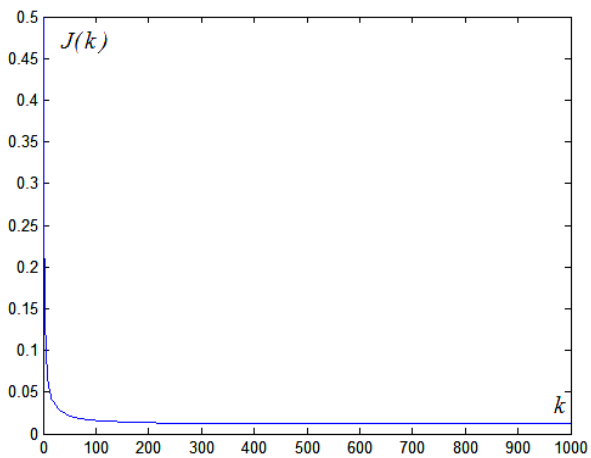

Fig. 7. Parameter of convergence γ^* illustratively by the functional $J(k)$

7 Conclusion

This paper was about the analysis of the evolutive neural fuzzy filtering operation and its real time conditions, in order to apply it into dynamical systems, we gave a description of the evolutive neural fuzzy digital filters (*ENFDF*) operation.

In addition, the paper presented the adaptive inference mechanism that classifies and deduces the filter answers by the error value, in order to search the adaptive weights of the filter and update its parameters to give a correct response dynamically as a natural linguistic answer.

Moreover, we establish how to construct and characterize the membership functions into the knowledge base in a probabilistic manner with the decision rules set, making a description of the real time conditions that the evolutive neural fuzzy filter (*ENFDF*) has to perform described as local and global characteristics and how its architecture works as a neural net.

The decision stage according with the filter operation makes a fuzzy inference in order to select the corresponding decision answer value, in accordance with the actual and previous filter answers.

The evolutive strategies selected in the decision stage according with a set of instructions limited in order to control an external process performance in order to adjust its condition into the best operation state.

The results are in formal sense and described using the definitions considered in the papers referenced here; finally this work showed a simulation of the *ENFDF* operation using the Matlab tool.

References:

[1] Abdul, S.: Fuzzy Logic and its uses, http://www.doc.ic.ac.uk, (2006).
[2] Ali, H. S.: Fundamentals of Adaptive Filters, (Complex systems, 2003).
[3] Amble, T.: Logic Programming and Knowledge Engineering, (Addison Wesley, 1987).
[4] Ash R.: Real Analysis and Probability, (Ed. Academic Press, USA, 1970).
[5] García, J., Medel, J., Guevara, L.: RTFDF Description for ARMA Systems, (WSEAS International Conferences, Canada, 2007).
[6] García, J., Medel, J., Sánchez, "Real-time neuro-fuzzy digital filtering: Approach", Computer and Simulation in Modern Science, volume I, (WSEAS press selected papers, 2008).
[7] Haykin, S.: Adaptive Filtering, (Prentice Hall, 2001).

[8] Huang, G., Zhu, K., Siew, C . : Real-Time Learning Capability of Neural Networks, (IEEE Transactions on Neural Networks, 2006), Vol. 17, 863-878.
[9] Jantzen, J., A tutorial on Adaptive Fuzzy Control, (Technical University of Denmark 2002).
[10] F. Gustafsson, “Adaptive Filtering and Change Detection”, John Wiley and Sons, Ltd, 2000.
[11] Mamdani, E.: Applications of Fuzzy Algorithms for Control of Simple Dynamic Plant, (Proc. IEEE, 1974), Vol. 121, 1585-1588.
[12] Manuel, L.: Teoría de la Medida e Integral de Lebesgue, (Universidad Nacional del Rosario 2003).
[13] Margaliot, M., and Langholz, G.: New Approaches to Fuzzy Modeling and Control Design and Analysis, (World Scientific 2000).
[14] Medel, J., and Guevara, P.: Caracterización de Filtros Digitales en Tiempo-real para Computadoras Digitales, (Computación y Sistemas, 2004), Vol. VII, No. 3.
[15] Medel, J., García, J. and Sánchez, J. “Real-time Neuro-Fuzzy Digital Filtering: Basic concepts”, (WSEAS Transactions on Systems and Control, 2008), Vol.8.16.
[16] Marcek, D.: Stock Price Forecasting: Statistical, Classical and Fuzzy Neural Networks, (Modeling Decisions for Artificial Intelligence, Springer Verlag, 2004), 41-48.
[17] Montejo, M.: Lógica Difusa y Control Difuso, http://www.redeya.com, (2006).
[18] Morales, G.: Introducción a la Lógica Difusa, (Cinvestav-IPN, 2002).
[19] Piotr, C.: Evolution Fuzzy Rule Based System with Parameterized Consequences, (International Journal of Applied Mathematics and Computer Science, 2006)Vol. 16, No. 3, 373-385.
[20] Passino, K. M.: Fuzzy Control, (Addison Wesley, 1998).
[21] Rajen, B., and Gopal, M.: Neuro-Fuzzy Decision Trees, International Journal of Neural Filters, 2006), Vol. 16, 63-68.
[22] Shannon, M.: A Mathematical Theory of Communication, Bell Systems Technical Journal, 1948), Vol. 27, 379-423 and 623-656.
[23] Smith, J. and Eiben, A.: Introduction to Evolutionary Computing, (Springer, 2003).
[24] Takagi, T., and Sugeno, M.: Fuzzy Identification of Systems and its Applications to Modelling and control, (IEEE Transactions and Systems, man, and cybernetics, 1986), Vol. 15, 116-132,
[25] Yamakawa, F.: Fuzzy Neurons and Fuzzy Neural Networks, (1989).
[26] Zadeh, L.: Fuzzy Sets, (Information and control, 1965), Vol. 8, 338-353.