Journal of Intelligent Learning Systems and Applications, 2010, 2: 12-18
doi:10.4236/jilsa.2010.21002 Published Online February 2010 (http://www.SciRP.org/journal/jilsa)

Evolutive Neural Net Fuzzy Filtering: Basic Description

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

[1]Professional School of Mechanical and Electrical Engineering, National Polytechnic Institute, Col. San Francisco Culhuacan Del. Coyoacan, México D.F.; [2]Centre of Computing Research, National Polytechnic Institute, Vallejo, México D. F.
Email: jcnet21@yahoo.com, jjmedelj@yahoo.com.mx, jcsanchezgarcia@gmail.com

Received October 29th, 2009; accepted January 20th, 2010.

ABSTRACT

The paper describes the operation principles of the evolutive neuro fuzzy filtering (ENFF) properties, which based on back propagation fuzzy neural net, this filter adaptively choose and emit a decision according with the reference signal changes of an external reference process, in order to actualize the best correct new conditions updating a process. This neural net fuzzy filter mechanism selects the best parameter values into the knowledge base (KB), to update the filter weights giving a good enough answers in accordance with the reference signal in natural sense. The filter 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 filtering process states require that bound into its own time limit as real time system, considering the Nyquist and Shannon criteria. The characterization of the membership functions builds the knowledge base in probabilistic sense with respect to the rules set inference to describe the reference system and deduce the new filter decision, performing the ENFF answers. Moreover, the paper describes schematically the neural net architecture and the decision-making stages in order to integrate them into the filter architecture as intelligent system. The results expressed in formal sense use the concepts into the paper references with a simulation of the ENFF into a Kalman filter structure using the Matlab© tool.

Keywords: *Digital Filters, Neuro Fuzzy Systems, Evolutive Systems, Real Time*

1. Introduction

The development of new intelligent systems requires mechanism that deduces its own external environment. Having a characterization of its answers in order to discover dynamically the changes of an external process; interpreting the operation levels to select the best response updating its answers weights, in order to follow the best approximation condition in accordance to the external process changes and taking a decision about the actual condition to correct or update a system parameters.

These kinds of systems related to artificial intelligent mechanisms, which have some problems to develop capacities with high evolutive changing processes.

The new systems should use into its architecture evolutive tools in order to get its own perception giving 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 net fuzzy digital filtering is an option for to obtain different decision answer levels, in accordance with a reference model dynamics, 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 each time.

An evolutive system with artificial neural network properties is a computational model imitating natural processes as the biological systems. Constituting in this case the processing elements called neurons, all of them connected forming the neural net structure and a decision stage to loop the process with the best answer [1,2].

The evolutive filtering is constructing with different kind of decision answers, considering a set of characteristics limited by the reference operation region. The goal of this kind of filter is to characterize and infer a system that has uncertainties in its operation, describing the natural process, with a rule set; needing a feedback law in order to follow the basic properties in accordance with a desired input signal, adjusting its parameters to give a correct solution dynamically. According to this, the filter chooses the best decision instruction to update the system.

The neural net filter stage [3] 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 ([4–6]) at each iteration by degrees [7]. This process refers to a back propagation parameter adaptation [8], using supervised learning by the knowledge base according with the error *e(k)* ([9,10]) described by the difference between the desired response *y(k)* and the actual signal $\hat{y}(k)$ [11][1].

With this perspective, the paper integrates the *ENFF* concept with its real time restrictions [9], using statistical methods in order to characterize the *Kalman* filter internal structure to give answers with respect to the operation levels in natural way making a specific decision in order to follow the natural reference model ([12,13]).

2. Filtering Stages

The criterion described as the error minimization *e(k)*[3] describes the difference between the desired input *y(k)* and the actual output filter $\hat{y}(k)$, allowing to find the corresponding 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 with a limited interval [0, ε] and ε is described as $\inf\{|e(k)| : i,k \in Z_+\}$ (*i* represents an index, with k interval [14,15])[2] Below, there is the stages description of the evolutive filtering, the first three stages are the inference mechanism; the stages d and e are the filtering process and the decision stage is the instruction decision:

- *Input Inference*: In this stage, the natural desired signal *y(k)* from the reference system to the input filter has a description in metric sense, is the first step of the inference mechanism [5].
- *Rule base*: Dynamical rank intervals with respect to the input of the filter use the logical binary connector known as *IF*, representing the second step of the inference mechanism.
- *Output 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 is the third step of the inference mechanism.
- *Filter algorithm:* 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.
- *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 $J(k)$.[3]
- *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 $ds(k)$.

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 using the logic connectors *IF-AND-THEN* the filter selects the corresponding instruction to actualize the reference system conditions.

Figure 1, shows the filter architecture interacting with a reference process to describe its operation and take a decision instruction.

3. Filtering Description

The evolutive neural net fuzzy filter (see: Figure 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 by neural net fuzzy filter has its state variables bounded by a

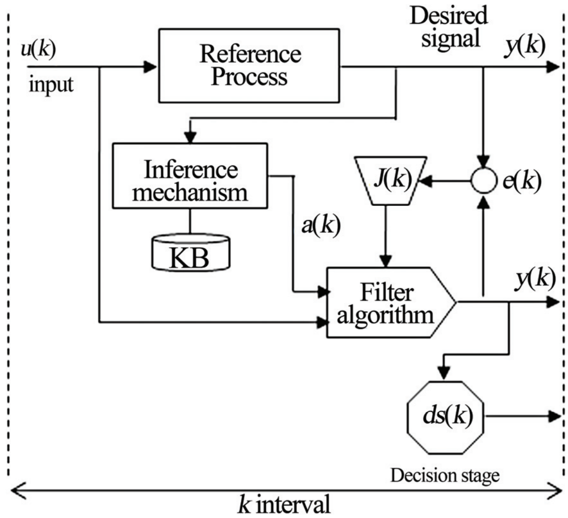

Figure 1. Evolutive neural net fuzzy filter process

---

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

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

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

rank of values described by fuzzy sets (for example, the temperature classified by the fuzzy logic as high, medium or low ([3,8,17,18]).

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). 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: [4,6,9]).

The filter classifies its different operation levels 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 ([14,19]).

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. The membership function set established inside a distribution function corresponding to the error criterion; this classification with several membership functions is in accordance to each error interval (as operation linguistic levels).

Respect to Figure 1, $u(k)$ is an input natural linguistic value from the reference process interacting with the filter. $T(k)$ is the control area, which include all the filtering processes corresponding to one 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 $\hat{y}(k)$; both values should be almost equal; in other words, $e(k)$ is the difference between $\hat{y}(k)$, and $y(k)$, being this the criterion distance in order to select the membership function for each case[1].

4. Fuzzy Neural Net Structure

The *ENFF* architecture is develop in order to work as neural net using back propagation properties ([11,19]). This characterizes the operational linguistic levels of the desired variable set $\{y(k)\}$, expressing as a basic estimation $\hat{a}(k)$ result[5], in accordance with the inference rules with respect to the error value criterion selecting the corresponding membership function.

The neural net conditions are [1]:

- The desired input $y(k)$ represents fuzzy labels, and each one has different operation levels.
- The input inference recognizes the variable type (A, B or C).
- The weights required for the internal adaptive adjustments into the filter architecture, to get a correct response described as membership functions, renewing the values of $\hat{a}(k)$ parameter using the knowledge base.
- 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 to update its weights.
- It uses supervised learning into the knowledge base, therefore, the neuron has previously all the information included in T_N.
- The decision stage compares the previous output and the actual value of the filter to select a instruction decision [7].

Globally, according to Figure 2, the filter has four neural layers:

The first layer 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 $\hat{a}(k)$; the output$\{\hat{y}(k)\}$, that is the answers set of the neural process (see: Figure 2). The filter has a decision *stage* layer as evolutive condition in order to give control instructions for an external process according with the previous filter answers.

The neurons 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.

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 reference process operation in order to give the best control value.

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

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 desired signal set, according to the changes of the error rank *e(k)* per iteration, one neuron is activating; next iteration the filter could renew the neuron or only change its value by degrees.

The value of *y(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 [2].

5. Rule Base 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 the error 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.

The rules generated constitute the inference of the desired signal $y(k)^5$ as the logical connector *IF*, and the logical connector *THEN* selecting automatically the parameter weighs of $\hat{a}(k)$ according to the knowledge base ([18,20]):

The *KB* has an automatic classification of the filter conditions: having the knowledge of the filtering operation levels. In addition, selects the corresponding membership function (value of $\hat{a}(k)$) of the knowledge base, adapting its responses weighs to give a correct answer $\hat{y}(k)$, near to $y(k)$.

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

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

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:

$$J_{\min} = \inf_{N} \{\min J(y_0, \hat{y})\}_N \quad (2)$$

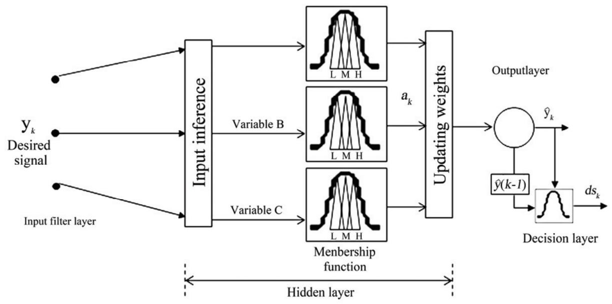

Figure 2. Evolutive neural filter layers (L-low, M-medium, H-high)

The rules set constitute a simple filter operation, based on the error set {e(k)} as limit indicator of the corresponding membership function set to adapt the parameter dynamically.

The decision stage described as selects the best instruction answer to control a process according with the filter answer, and its previous value, with the decision inference mechanism by fuzzy rules.

For the decision stage, which is the finally inference of the evolutive neural net fuzzy filter, using a set of fuzzy rules to get the correct control instruction from its own knowledge base (KB). First, consider the value level and its previous value as the actual environment and select the corresponding condition or strategy from the knowledge base to loop an external process as.

To describe the decision stage mechanism, we use fuzzy rules to infer and select the corresponding value with If-then rules; the filtering system selects and gives a specific instruction answer as in order to update and improve the control instruction performance minimizing the error process, in accordance with the fuzzy rule structure will have:

If value is____ and value is____ Then value is ____.

This filter (ENFF) has two knowledge bases, one for to select the parameter (as in Figure 1) and the second used for the decision stage to select the instruction answer (as in Figure 3).

The Figure 3 illustrates the decision stage mechanism as fuzzy rules as:

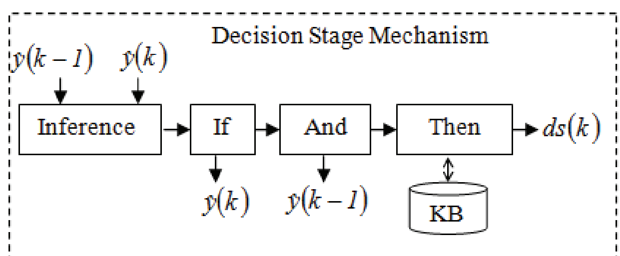

Figure 3. Decision stage inference

6. Real Time Scheme

DEFINITION 1 The real time properties of an evolutive neural net fuzzy digital filter, which is an adaptive filter performed according to ([6] and [10]):

- To take and to emit signals with fuzzy information into intervals limited in accordance to the reference process response according to the stability criteria ([5,6]).
- To take and to emit information through semi-open time intervals, synchronized with the process evolution time, described in a relative way by semi-open time intervals, considering the criteria into references [15,21,22].
- A membership functions group forms a control area, according to the properties considered in a) and b) points, respectively ([18,20,23]) .
- 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 $\hat{y}(k)$ ([16,20]).
- The adaptation algorithm updates the filter coefficients according to the selected membership function respectively to the established error criterion symbolically expressed as γ_*.
- The decision stage mechanism selects the best instruction answer as $ds(k)$, in order to perform the external process operation at each time.

The knowledge base has all information that the filter requires to adjust its gains in optimal form and gives an 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 the stability properties [6,20,23]):

- $y(k)$ is a measurable variable classified in metric ranks as degrees in linguistic sense (described into a state space variable bounded symbolically into linguistic natural expressions as high, medium or low values),
- $T(k)$ is the control area described in pairs formed by $\hat{y}(k)$ and $y(k)$ limited in time interval (has a velocity change bounded in the sense exposed by [11]),
- $e(k)$ 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 as linguistic variables, both are the same natural value.
- $ds(k)$ is the 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 a knowledge base to update an external process into a new condition.

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

[5]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: [12] and [10]), $\omega(k)$ is the perturbation output noise, $y(k)$ is the desired reference signal.

6.1 Local and Global Description

DEFINITION 2 An ENFF in local and global temporal sense has quality of response according to the convergence criterion[1] with respect to the real time conditions [15].

Global characteristics: The convergence intervals defined by $[0, \varepsilon \pm \alpha)$ with measures up to zero through error functional $J(k)$ considering and the convergence relation[1], temporally parameterized to the membership function according to the linguistic variables values ([3,24]), without loss that /e(k)/<1 in agreement to [6].

According to the evolutive neural net fuzzy concepts, the global characteristics specified in stochastic sense according to [15], where $J(\tau_m) = inf\{min\{\mathcal{J}_k\}\} \leq \varepsilon$ (see: (2)), with $\{\mathcal{J}_k\} \subseteq \{J_k\}$ and

$$P(\mathcal{J}_k \leq \varepsilon \pm \sigma) = 1, \ \sigma << \varepsilon$$

without loss that its natural evolution described by [22]:

$$\tau_{min} = 0.5 f_{max}^{-1} \tag{3}$$

7. Simulations

For the simulation of the *ENFF*, in this case we use the Kalman filter structure to integrate the filter stages, with a transition matrix described by the knowledge base in accordance to the error functional criterion [5]. The evolution times into a soft system integrated in a *PC* with *AMD* Sempron processor 3100+ with *k* intervals, having a mean evolution time of 0.004 sec ± 0.0002 sec.

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

$$x(k+1) = a(k)x(k) + w(k) \tag{4}$$

In accordance with the system (4) the output is:

$$x(k),\ w(k),\ v(k) \in R^{n \times 1}$$

$$y(k) = x(k) + v(k) \ : \tag{5}$$

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.

There are different operation levels described in probability sense in order to match with the functional error[1] limit, in accordance with the desired signal *y(k)* and the *Kalman* filter answer $\hat{y}(k)$. The filter operation bounds into the second probability moment, establishing the filter classification as linguistic natural variables expressed as low, medium and high levels.

According to the parameter $\hat{a}(k)$ selected by the rule strategy, the Figure 4 shows the output answer $\hat{y}(k)$ of the filter with respect to the desire signal *y(k)* at the input filter:

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 $\hat{y}(k-1)$:

The evolution time is less than the reference process proposed as 0.092 sec., satisfying the condition described in (3).

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

The global convergence time of the filter is 0.08 sec, which is less than the evolution condition of the system, known as system maximum evolution time, oscillating around 0.092 sec.

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

8. Conclusions

The paper was about the analysis of the evolutive neural net fuzzy filtering and its real time conditions, in order show the applicability conditions into dynamical systems. The paper describes the adaptive inference mechanism that classifies and deduces the filter answers by the error value as limit, in order to search the adaptive weights and update its parameters to give a correct response dynamically as a natural linguistic answer.

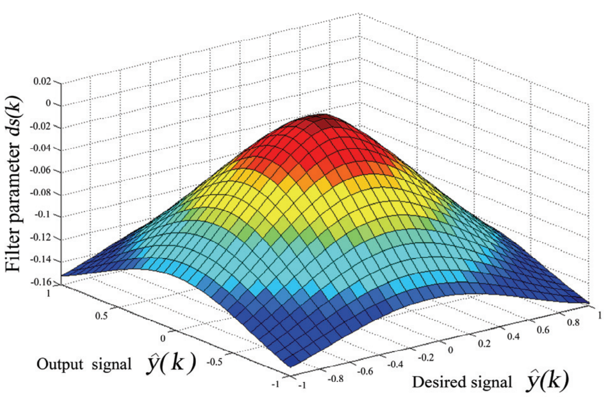

Figure 4. $\hat{y}(k)$ **Estimation in accordance with** $\hat{a}(k)$ **value and** $y(k)$ **reference**

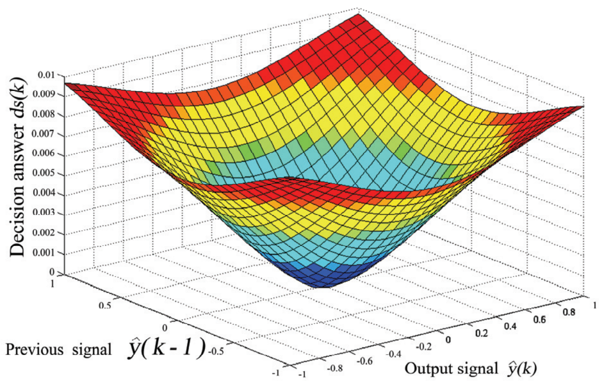

Figure 5. Decision answer $ds(k)$ **with respect to** $\hat{y}(k)$ **and** $\hat{y}(k-1)$

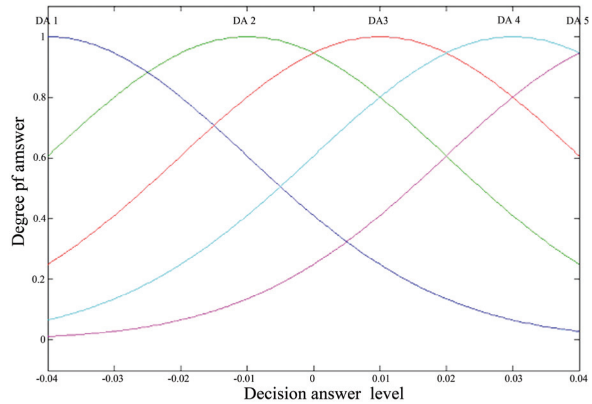

Figure 6. Decision answer as linguistic levels (DA)

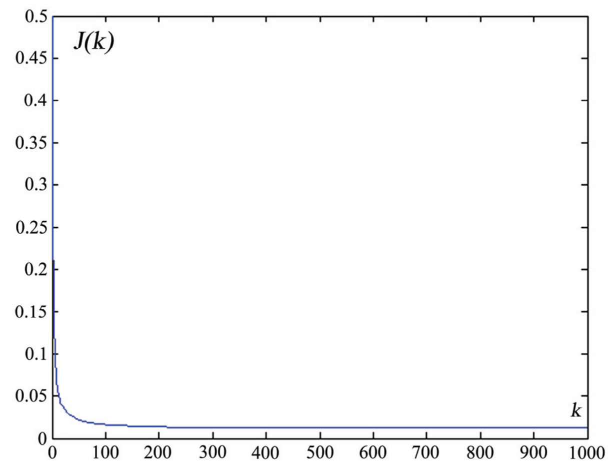

Figure 7. Parameter of convergence γ^* **illustratively by the functional** ***J(k)***

The paper establishes how to construct and characterize the membership functions of the knowledge base in a probabilistic manner with the decision rules set, making a description of the real time conditions that the evolutive neural net fuzzy filter (*ENFF*) has to perform, which architecture works as a neural net. The filter has a final decision stage in accordance with the filter operation making a fuzzy inference between the actual and previous filter answers in order to select the corresponding decision answer value.

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 *ENFF* operation using the Matlab tool and the *Kalman* filter structureto integrate the evolutive properties, considering five decision answers into the decision stage, having an accurate filtering time response with respect to the reference system in accordance with the real time properties proposed in this work.

9. Acknowledgments

The authors thank the sponsors for the support of this work: Conacyt and National Polytechnic Institute (*IPN*).

REFERENCES

[1] C. Piotr, “Evolution fuzzy rule based system with parameterized consequences,” International Journal of Applied Mathematics and Computer Science, Vol. 16, No. 3, pp. 373–385, 2006.

[2] F. Yamakawa, “Fuzzy neurons and fuzzy neural networks,” 1989.

[3] S. Abdul, “Fuzzy logic and its uses,” http://www.doc.ic.ac.uk.

[4] H. S. Ali, “Fundamentals of adaptive filters,” Complex Systems, 2003.

[5] R. Ash, “Real analysis and probability,” Ed. Academic Press, USA, 1970.

[6] S. Haykin, “Adaptive filtering,” Prentice Hall, 2001.

[7] G. Huang, K. Zhu, and C. Siew, “Real-time learning capability of neural networks,” IEEE Transactions on Neural Networks, Vol. 17, pp. 863–878, 2006.

[8] B. Rajen and M. Gopal, “Neuro-fuzzy decision trees,” International Journal of Neural Filters, Vol. 16, pp. 63–68, 2006.

[9] J. García, J. Medel, and J. Sánchez, “Real-time neuro-fuzzy digital filtering: Approach,” Computer and Simulation in Modern Science, WSEAS press selected papers, Vol. 1, 2008.

[10] J. Medel and P. Guevara, “Caracterización de filtros digitales en tiempo-real para computadoras digitales,” Computacióny Sistemas, Vol. 7, No. 3, 2004.

[11] J. Medel, J. García, and J. Sánchez, “Real-time neuro-fuzzy digital filtering: Basic concepts,” WSEAS Transactions on Systems and Control, Vol. 8, No. 16, 2008.

[12] J. García, J.Medel, and L. Guevara, “RTFDF description for ARMA systems,” WSEAS International Conferences, Canada, 2007.

[13] D. Marcek, “Stock price forecasting: Statistical, classical and fuzzy neural networks,” Modeling Decisions for Artificial Intelligence, Springer Verlag, pp. 41–48, 2004.

[14] M. Margaliot and G. Langholz, “New approaches to fuzzy modeling and control design and analysis,” World Scientific, 2000.

[15] G. Morales, “Introducción a la lógica difusa,” Cinvestav-IPN, 2002.

[16] T. Amble, “Logic programming and knowledge engineering,” Addison Wesley, 1987.

[17] J. Smith and A. Eiben, “Introduction to evolutionary computing,” Springer, 2003.

[18] T. Takagi and M. Sugeno, “Fuzzy identification of systems and its applications to modelling and control,” IEEE Transactions and Systems, Man, and Cybernetics, Vol. 15, pp. 116–132, 1986.

[19] G. Huang, K. Zhu, and C. Siew, “Real-time learning capability of neural networks,” IEEE Transactions on Neural Networks, Vol. 17, pp. 863–878, 2006.

[20] L. Zadeh, “Fuzzy sets,” Information and Control, Vol. 8, pp. 338–353, 1965.

[21] K. M. Passino, “Fuzzy control,” Addison Wesley, 1998.

[22] M. Shannon, “A mathematical theory of communication,” Bell Systems Technical Journal, Vol. 27, pp. 379–423, 623–656, 1948.

[23] E. Mamdani, “Applications of fuzzy algorithms for control of simple dynamic plant,” Proceedings IEEE, Vol. 121, pp. 1585–1588, 1974.

[24] M. Montejo, “Lógica difusay control difuso,” 2006, http: //www.redeya.com.

