Real-time Neuro-fuzzy Digital Filtering: Basic Concepts

J. J. Medel Juárez, J. C. García Infante, J. C. Sánchez García

Centre of Computing Research, National Polytechnic Institute
Vallejo D. F. 07738, México

Microelectronics Research Department
Professional School of Mechanical and Electrical Engineering
Av. Santa Ana No. 1000, Col. San Francisco Culhuacan
Del. Coyoacan, México D.F.

jjmedelj@yahoo.com.mx, jcnet21@yahoo.com, jcsanchezgarcia@gmail.com

Abstract: - In this paper we describe the neural fuzzy filtering properties in real-time sense; giving an approach about the real-time neuro-fuzzy digital filters, defined in acronym form as *RTNFDF*. This kind of filters require the adaptive inference mechanism into the fuzzy logic structure to deduce the filter answers in order to select the best parameter values into the knowledge base (*KB*), actualizing the filter weights to give a good enough answers in natural linguistic sense; this require that all of the states bound into *RTNFDF* time limit as a real-time system, considering the Nyquist criteria. In this paper we characterize the membership functions into the knowledge base in a probabilistic way respect to the rules set decisions without lost its real-time description, performing the *RTFNDF*. Moreover, the paper describes in schematic sense the neurons set architecture into the filter description. The results expressed in formal sense use the concepts exposed in the papers included into the references. Finally, we present in illustrative manner the *RTNFDF* operations using as a tool the Matlab© software. Explicitly, the paper has eight sections conformed as follows: 1. Introduction, 2. Neural Architecture, 3. Rule Base Dynamics, 4. Neural Rules, 5. Real-time Descriptions, 6. Restrictions for RTNFDF, 7. Simulation, Conclusions and References.

Key-Words: Digital filters, adaptive, neural networks, fuzzy logic, inference systems, real-time.

1 Introduction

One of the technological tools most used to interact with dynamical processes are the digital filters, which work for different utilities as: the error elimination of a system, get specific data, reconstruct and predict the system operation. Digital filters have many applications to use in many systems: control, medicine, instrumentation, electronics, computation and communications [1] and [2].

The technological systems are designed to be developed with capacities as the ability to follow a process in a natural way. In the real life we can find systems with a dynamical operation that requires technological applications with expert mechanisms into its internal construction in order to give answers dynamically to the different operation states of a system, moreover, having a characterization of it into its own structure and using an inference mechanism to select the correct response to perform a system trajectory from a previous condition to a balance state with the interaction process.

Actually, the digital filters systems have many developing topic areas; therefore the study about this topic is important. The characterization of a filter responses and the inference mechanism in order to allow us to find a correct action at each time is one of the requirements to solve in order to apply the digital filters in the future into systems with more advanced properties related with mobility, velocity, interoperability, integration [1], [2] and [3].

Is difficult to characterize and infers a system that has uncertainties in its operation, actually is a complex task because many natural processes with different operation levels need accurate answers, thus, in many senses.

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

From this perspective, the best tool used is a digital filter in recursive sense, the problem is to adjust in dynamical form its parameters and gives a limited answers in temporal sense; before answer this question, the main problem observed in Kalman filter is the transition matrix characterized in dynamical sense, following the natural evolution of the reference system considered to interact with the filter [1] and [2].

Moreover, if the answer is in linguistic form, a conventional digital filter can't express its operation into levels range. Thus, the answer correct level is very difficult to find.

In this paper, we integrate concepts like fuzzy logic, real time and neural networks using statistical models in order to integrate them into the Kalman filter mechanism to give answers operation levels in a natural way following the natural model process reference [5], and [16].

An artificial neural network 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 [19], and [25].

A fuzzy neuron can classify, search and associate information [5] and [8] giving the corresponding answer value according to the desired output signal building the control area described as $T_N = \{(y(k), \hat{y}(k))\}$[1] within a membership intervals limited inside a region called as 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, predefined by this natural reference.

The neuro-fuzzy filter [1] works as a parallel set of fuzzy neurons in loop form, which has an iterative searching methodology used for learning algorithms and based in 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 $e(k)^2$ that 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 approximates the signal $\hat{y}(k)$ to *y(k)* in order to adjust the parameters of the filter and get the correct answer.

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 rule of the neuro-fuzzy filtering determines the specific membership function value, which is inside a limited region according to the correct operation of the reference system rank; the reference distribution function defined previously and its desired inputs limited inside of a region in order to build the control area of the filter.

The neuro-fuzzy filter based on the next elements considers the concepts studied in [1], [2], [6], [7], [8], [10], [11], [14], [15], and [18]:

i. *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].

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

iii. *Inference Mechanism*: The expert action respect to the rule base known as consequence uses the logical binary connector *THEN*.

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

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

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

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

v. *Natural feedback:* Finally, 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].

2 Neural Architecture

The RTNFDF structure developed in order to work as a group of neurons using backpropagation properties ([8], [16], and [19]), where each single neuron characterizes the operational levels of a linguistic desired variable set $\{y(k)\}$, expressing as a basic estimation $\hat{a}(k)$ result[5], in according with the inference rules from respect to the error value criterion selecting the membership function corresponding.
The neuron works with an activation function representing the neural filter process (to see: [5], [8], [16], [19], and [25]).

The characteristics of a single neuron are (to see: [8], [16], [19], and [25]):

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

b. 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).

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

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

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

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

Layer 1, with $p \times n, n, p \in Z_+$ input neurons representing a set of desired inputs $\{y(k)\}$ of the reference process interacting with the filter,

Layer 2, is a single hidden layer with *p* processing units in which the inference mechanism operates in order to find the respective membership function $\hat{a}(k)$ in order to adjust the filter mechanism

Layer 3, is the filter output $\{\hat{y}(k)\}$, that is the answers set of the neural process (to see: Fig. 1).

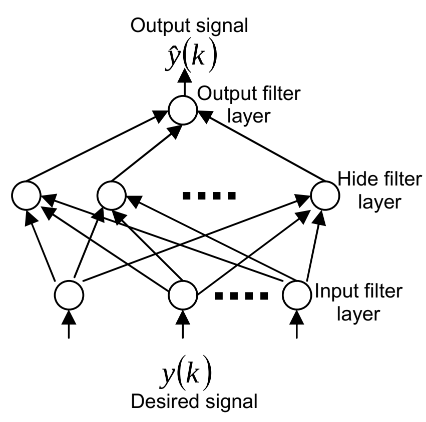

Fig. 1. Layers of the neural filter

3 Rule Base Dynamics

This kind of filtering has an operational properties defined by the rules base in order t learn, recall,

---

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

[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 (to see: [5] and [14]), $\omega(k)$ is the perturbation output noise, $y(k)$ is the desired reference signal.

associates and compare the new information delimited before according to variance limits predefined.

The fuzzy rules conditions established as logical connectors (*IF-THEN*) constitutes the rules base respect to the error intervals and its respective response described as membership function.

In a neuro-fuzzy filter, the generated rules constitute the inference error $e(k)^1$ as the logicals connector *IF*, and *THEN* either, selecting automatically the parameter weighs of $\hat{a}(k)$, according to the knowledge base (to see: [7], [8], [16], [23], [24], and [26]):

1. It has an automatic classification of the filter conditions: having the knowledge of the filtering operation levels.

2. It 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.

3. The estimation parameter value $\hat{a}(k)$, adapting its responses weighs to give a correct answer $\hat{y}(k)$, close to reference model $y(k)$.

4. 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 $y(k)$:

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

4 Neural Rules

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.

Each neuron into RTNFDF represents a specific service with different levels of response, activating a specific neuron.

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 neighbor value $\hat{a}(k)$, and gives a correct natural answer (to see: [8], [16], [19], and [25]). Fig. 2 shows the neural filter process.

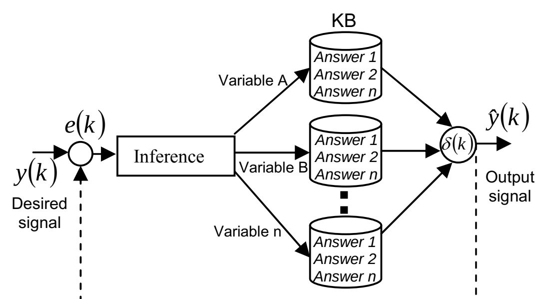

Fig. 2. Neural filter process

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

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

5 Real-time Descriptions

In neural fuzzy filter, the knowledge base has all information that the filter requires for adjustment of its gains in an 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 lose the 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 $\{\gamma_i : \gamma_i > 0, \quad \forall i \in Z_+\}$, 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.

THEOREM 1: *The set of inputs and outputs of the RTNFDF must obey the stability criteria, in other case the Real-time neuro-fuzzy digital filter entropy not converges to a region previously defined.*

PROOF: *The basic energy of the system is bounded, if the inputs and outputs system are not bounded, the system could give an instability infinite response or it could not have a response limited in an entropy sense, considering in the recursive form, the entropy:* $H(y_k) = [H(y_{k-1}) - y_k \ln y_k]$.

THEOREM 2: *The velocity change respect to the input and output signals are bounded into semi-open intervals with respect to frequency iteration time, input and output respectively.*

PROOF: *In the Shannon sense and the agreement to Ash measure theory, each time interval is described as semi-open right because in other case the entropy increase with respect to intersections of the interval limits neighbors.*

Each time interval into the input and output signals is bounded in the Nyquist sense, in other case the Shannon entropy increase too, because the information increase and the system collapse.

THEOREM 3: *The discourse universe into de neuro-fuzzy filter is bounded in temporal sense, because in other case the output responses would be obtained in an infinite time.*

PROOF: *The discourse universe must be bounded into time intervals respect whit the system properties established in Nyquist and Shannon criteria, in other that the time of the filter response will be infinite.*

LEMMA 1: *The neuro-fuzzy digital filter is an adaptive digital filter and it requires a desired signal with respect to the discourse universe to actualize and adjust its response in agreement to the knowledge base; then the filter has a basic properties respect to adaptive fuzzy criteria.*

PROOF: *The basic energy of the system is bounded; if the inputs and outputs system are not bounded, the system could give an instability infinite response or it could not have a response limited in an entropy sense: considering in recursive form, the entropy:* $H(y_k) = [H(y_{k-1}) - y_k \ln y_k]$, *the difference between* $0 < |H(y_k) - H(y_{k-1})| < 1$, *such that in other case the change velocity is not bounded, meaning that* $y_k \uparrow \infty$. *In fuzzy sense, this velocity is described as disjoint neighbors' intervals accomplishing in linguistic sense that the change between them is limited numerically by* $\{\delta_i : \delta_i > 0, \quad \forall i \in Z_+\}$, *with* $\inf\{\delta_i\} \to |\lambda_*|$, *such that* $|\delta_*| > 0$, $\sup\{\delta_i\} \to |\delta^*|$, $|\delta^*| < 1$, *means that linguistic difference is bounded metrically speaking, by an interval* $(0,1)$.

REMARK 1: *Respect to the lemma 1, the null condition implies that the system doesn´t work, having* $y_k = 0$.

REMARK 2: *In linguistic sense, the output answer is defined positive,* symbolically speaking has three linguistic natural expressions defined its intervals as high, medium or low.

The stability region by natural answer with life operation, is defined into *an interval* $(0,1)$.

This kind of systems has an interaction with real process, obeying its time restriction and required to give a synchronized answer, seeking a natural communication between its.

6 Restrictions for RTNFDF

DEFINITION 1 (Local and global description) *An RTNFDF in local and global temporal sense, has quality response according to the convergence criterion[1] respect to real time conditions [18], without loss the synchrony and interaction in the same interval evolution, bounded in the Niquist and Shannon sense as:*

$$\tau_{min} = 0.5 f_{max}^{-1} \quad (3)$$

DEFINITION 2 (Global characteristics) *The convergence intervals defined by* $[0, \varepsilon \pm \alpha)$ *with measures up to zero respect to the error functional J(k) considered in [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], remembering* an error difference set *{e(k)}* classified by intervals also, delimited previously in order to select the infimum membership function in agreement to the infimum value region convergence, depicted by this filter structure.

According to the neural fuzzy concepts, the global characteristics specified in stochastic sense [18], where $\mu\{J(\tau_m)\} = \inf\{\mu\{\min \tilde{J}_k\}\} \subseteq [0,\varepsilon)$ *(to see: (3)), with* $\{\tilde{J}_k\} \subseteq \{J_k\}$ *and* $\mu(\tilde{J}_k \le \varepsilon \pm \sigma) = 1,\ \sigma << \varepsilon$ *without loss that its natural evolution described by [22].*

The conversion time occurred when the functional error expressed metrically has a variations bounded or in probability sense, the variations are inside interval with stationary conditions, it means that it is the region in where the functional evolution has the first two moments as constant for any time. In this sense, the converging region starts when the first two probability moments accomplish the stationary properties.

The Fig. 3, shows the global filter operation illustrating the convergence region by the error functional expressed metrically in agreement to the entropy function considered in the proof of lemma 1 and described illustratively into the Fig. 3; watching that the invariance conditions region coincident respect to critical properties imposed by dynamical reference system, temporally speaking as *LDmax*:

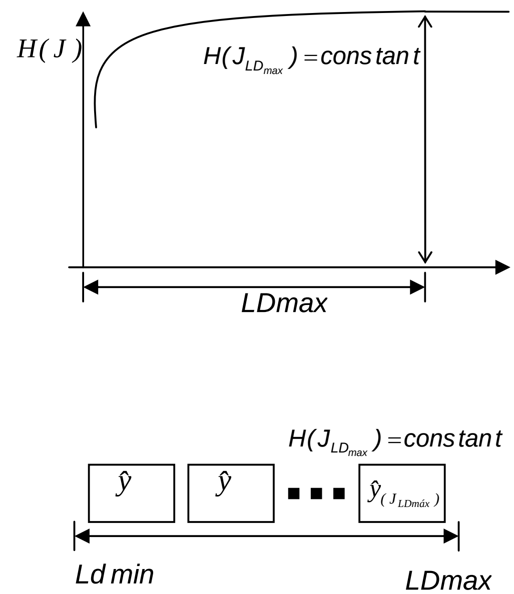

Fig. 3. Global filter operation.

Local characteristics: *Implies the process convergence through the neuro-fuzzy digital filter adjusting its parameters respect to a knowledge base containing all possible parameters that the filter require for some condition evolution, it means that* $\{a(k)_i\} \subseteq T_N$ *, is a parameter whole base, inside of it with linguistic conditions selecting each of its through the fuzzy variables, without loss conditions, objectivity answer and finalized times* $\{f(k)_i\}$ *establishing for filter criteria, delimited temporally by its corresponding limits* $[ld(k)_{i\_min}, LD(k)_i)$ *in Nyquist and Shannon sense [22] locally, according to [14] and [15] thus* $\mu[ld(k)_{i\_min}, LD(k)_i) < \tau_{min}$*:*

$$\lim\{y(k)\}_i \to [0, \gamma_* \pm \alpha),\ k \to m \quad (4)$$

The Fig. 4, shows the neuro fuzzy filter as task in global sense with an execution, described the whole elements by time interval depicted illustratively as element. The measure interval is result generated by the difference between the limits evolution imposed temporally by dynamical reference system.

Then the whole operations that compound the fuzzy filter could be expressed as a basic task, symbolically depicted in the Fig. 4, so that, each filter evolution is temporally bounded by time dynamical reference system in the Nyquist sense [22]. It means that the feedback, fuzzy logic and the rest operations need to be fulfilling in each interval time according to [14] and [15].

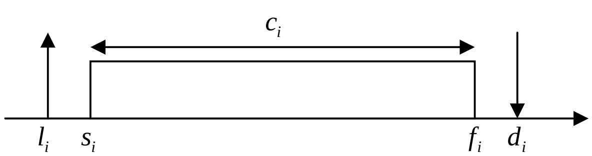

Fig. 4. Sub-process operation.

The Fig. 4, has a lot of symbols used into real-time schemes as arrival time (l_i), start time (s_i), execution time (c_i), end time (f_i), maximal time limit (d_i), bounded temporality by (5).

For the simulation section had choosing the Kalman filter because is a mathematical power tool that is playing an increasingly important role in computer applications and has been the subject of extensive scientific research of different kind of systems.

Essentially the Kalman filter is a set of mathematical equations that implement a predictor-corrector type estimator that is optimal because it minimizes the estimated error covariance [6] and [7].

Basically in the simulation had included the neuro-fuzzy mechanism conditions in order to select the corresponding parameter $\hat{a}(k)$ and the from the knowledge base according with the corresponding operation level of the reference system that is interacting with it, classifying all its described internal mechanism into operation ranks and updating the filtering process according to basic dynamics previously considered.

According with the Kalman literature [6] and [7], the identifier has the next equation:

$$\hat{x}(k) = \hat{a}(k)\hat{x}(k-1) + K(k)\hat{w}(k) \tag{5}$$

Where: $\hat{x}_t$ is the state identified, $\hat{a}_t$ is the parameter coefficient, x_t is the state variable, K_t is the Kalman gain and $\hat{w}_t$ the internal noise.

In the next section, we present the simulation of the RTNFDF using the Matlab tool, illustrating its operation with different levels of response, showing its graphics that describes this new perspective about digital filtering.

7 Simulations

The neuro-fuzzy filter simulation is built considering the Kalman filter [7] with a transition matrix described by the knowledge base according to the error functional criterion ([4] and [7]) through the linguistic inference rules, selecting the best element about it respect to criterion defined as infimum value functional error convergence.

The simulation evolution times inside into soft PC system with Sempron processor 3100+ and k intervals having a mean evolution time of 0.004 sec ± 0.0002 sec, using the MatLab resources, allowing to evaluate the performance fuzzy filter evolution, respect to define system.

The basic system in discrete space states as (AR-1) is expressed as first order difference, as:

$$x(k+1) = a(k)x(k) + w(k) \tag{6}$$

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

And the output described as:

$$y(k) = x(k) + v(k)\ : \tag{7}$$

with $y(k), v(k) \in R^{n\times 1}$.

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, and v(k) is the output perturbations with normal description and first four moments bounded.

The different operation levels as inference rules are expressed in probability sense in order to match with the functional error[1] according to the desired signal *y(k)* and the Kalman filter answer $\hat{y}(k)$ dotty obtains.

Bounded all sequence elements by the second probability moment, establishing each of its with linguistic natural variables expressed basically as low, medium and high levels.

In Fig. 5, could be observe the transition parameters estimation, numerically expressed:

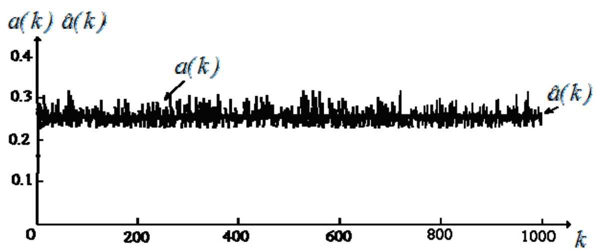

Fig. 5. Parameter estimation

According to the parameters estimation $\hat{a}(k)$, the Fig. 6, shows the output answer $\hat{y}(k)$ of the filter in respect to the desire signal *y(k)* system. Having a surface with optimum point and with it respective projections in each axis.

The

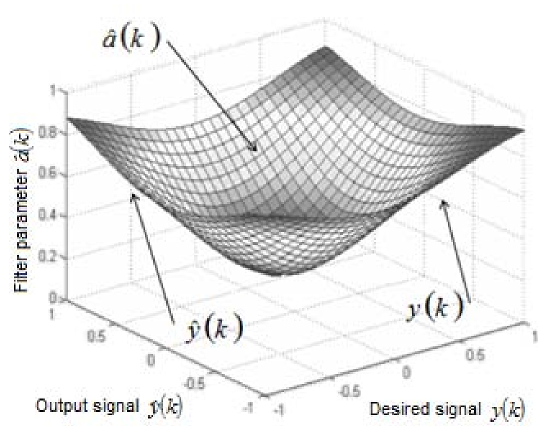

Fig. 6. Estimation of $\hat{y}(k)$, according to $\hat{a}(k)$ and $y(k)$.

In the same way, considering the result exposed in the Figure 5, we obtain the internal identification *x(k),* with high levels of convergence, such that is shown in the Figures 7 and 8.

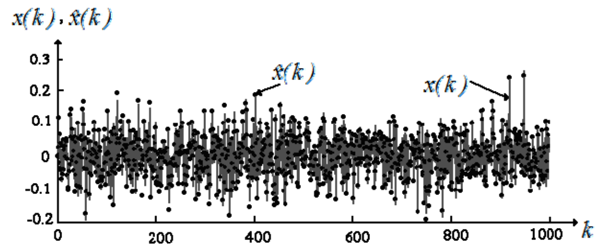

Fig. 7. Internal identification *x(k)*

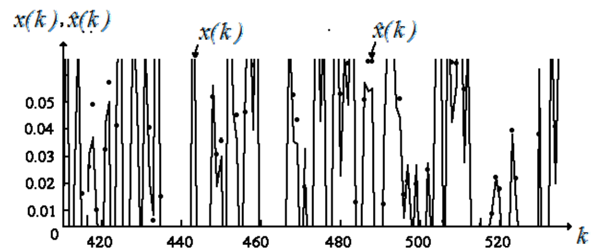

Fig. 8. Internal identification zoom

Figure 9, shows the linguistic variable classification of the different subsystems of the reference model, according to the desired signal $y(k)$ value as:

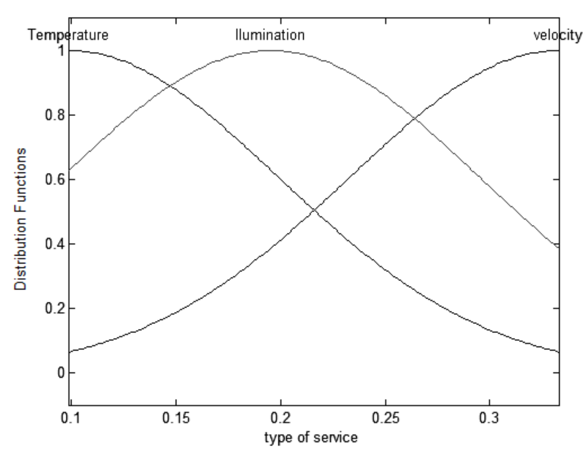

Fig. 9. Classification of the linguistic subsystems

The evolution time is less than the reference system proposed as 0.09 sec., satisfying the condition described in (3).

According to the temperature linguistic variable, the Fig. 10, shows its response linguistic levels as:

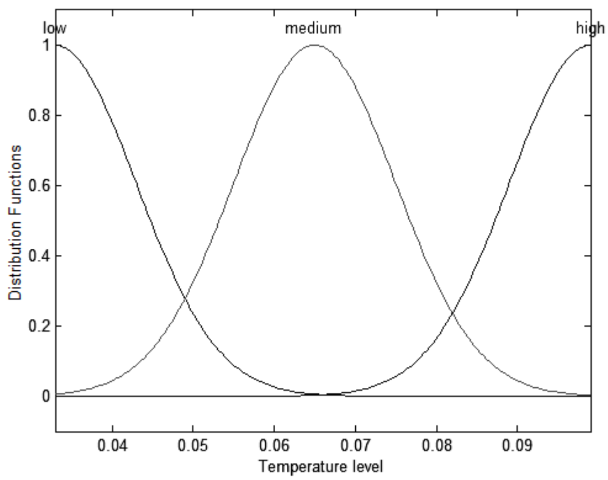

Fig. 10. Temperature linguistic levels

Figure 11, shows the Functional Levels in distribution sense:

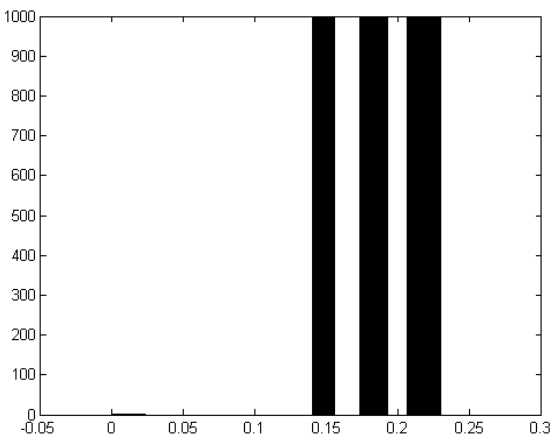

Fig. 11. Distribution of error function

Figure 12 shows the functional described in (2) respect to the filter:

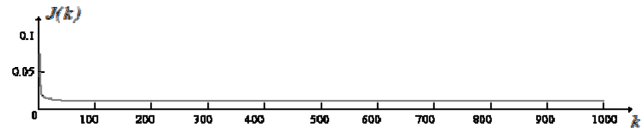

Fig. 12. Parameter of convergence γ^* illustratively by the functional *J(k)*

Figure 12, shows functional levels in distribution sense with respect to *J(k),* bounded it by the first two probability moments:

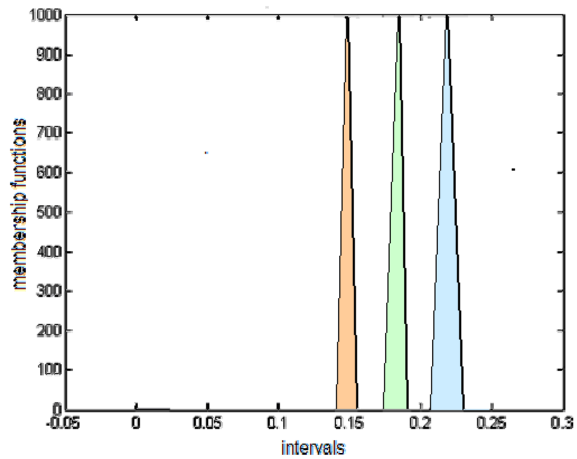

Fig. 12. Membership functions in fuzzy logic sense

Figure 13, shows the graphic of the entropy according with the error functional of the filter process:

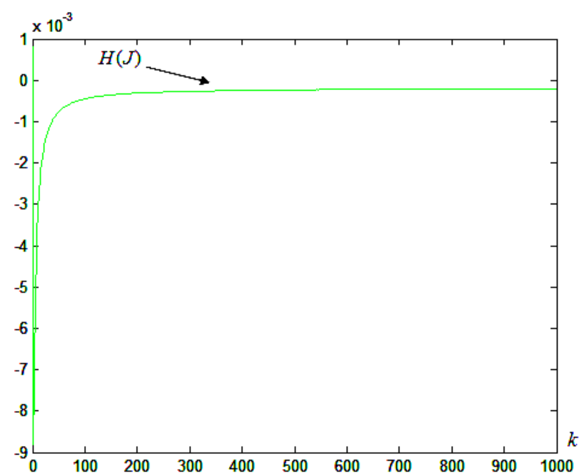

Fig. 13. Entropy of the filter

Thus, the global convergence time is 0.08 sec, which is less than the evolution condition of the system known as *LDmax*, oscillating around 0.09 sec accordingly to its properties previously described into definitions.

Conclusions

The paper treated about the analysis of the neuro-fuzzy filtering and its Real-time conditions, in order to apply it into dynamical systems, so that, we gave a basic description respect to the real-time neuro-fuzzy digital filters (RTNFDF) operations. Also, we presented the adaptive inference mechanism that classifies and deduces the filter answers by the error values, in order to search the adaptive weights and update its parameters to give a correct response dynamically accepted as a natural linguistic answer. Moreover, we establish how to construct and characterize the membership functions according to the knowledge base developed in probabilistic sense with decision rules set, making a description of the Real-time conditions that the neuro-fuzzy filter (RTNFDF) has to perform, and how its architecture works as a set of neurons: 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 RTNFDF operation using 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] Gustafsson, F.: Adaptive Filtering and Change Detection, (John Wiley & Sons, Ltd, 2000).
[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.
[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., Hernández, G., Guevara, P.: Real Time Flexible Manufacturing System. (WSEAS International Conferences, Turkey, 2008).
[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] Nikola, K.: Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering, (The MIT Press, 1996).
[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] Toledo, M., Medel, J.: Priority Task Allocation through the Maximum Entropy Principle (WSEAS International Conferences, Canada, 2007).
[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.