UDC 004.4’2, DOI: 10.2298/csis0902141E

The Adaptive Fuzzy Designed PID Controller using Wavelet Network

Ibrahiem M. M. El Emary[1], Walid Emar[2], and Musbah J. Aqel[3]

[1]Faculty of Engineering, Al Ahliyya Amman University, Amman, Jordan
omary57@hotmail.com
[2]Faculty of Engineering, Al Israa University, Amman, Jordan
walidemar23@yahoo.com
[3]Faculty of Engineering, Applied Science University , Amman, Jordan
musbahaqel@yahoo.com

Abstract. During the past several years, fuzzy control has emerged as one of the most active and fruitful areas for research in the applications of the fuzzy set theory, especially in the realm of the industrial processes, which do not lend themselves to control by conventional methods because of a lack of quantitative data regarding the input-output relations i.e., accurate mathematical models. The fuzzy logic controller based on wavelet network provides a means of converting a linguistic control strategy based on expert knowledge into an automatic strategy. In the available literature, one can find scores of papers on fuzzy logic based controllers or fuzzy adaptation of PID controllers. However, relatively less number of papers is found on fuzzy adaptive control, which is not surprising since fuzzy adaptive control is relatively new tool in control engineering. In this paper, fuzzy adaptive PID controller with wavelet network is discussed in subsequent sections with simulations. An adaptive neural network structure was proposed. This structure was used to replace the linearization feedback of a second order system (plant, process). Also, in this paper, it is proposed that the controller be tuned using Adaptive fuzzy controller where Adaptive fuzzy controller is a stochastic global search method that emulates the process of natural evolution. It is shown that Adaptive fuzzy controller be capable of locating high performance areas in complex domains without experiencing the difficulties associated with high dimensionality or false optima as may occur with gradient decent techniques. From the output results, it was shown that Adaptive fuzzy controller gave fast convergence for the nonparametric function under consideration in comparison with conventional Neural Wavelet Network (NWN).

Keywords: Wavelet network, fuzzy logic, PID controller.

1. Introduction

It is known that PID controller is employed in every facet of industrial automation. The application of PID controller span from small industry to high technology industry. Using Fuzzy controller to perform the tuning of the controller will result in the optimum controller being evaluated for the system every time. In this paper, the selected model is a second order system which may concern the control of a robot arm position or dc motor position and so on. Intelligent systems cover a wide range of technologies related to hard sciences, such as modelling and control theory, and soft sciences, such as the artificial intelligence (AI). Intelligent systems, including neural wavelet networks (NWN), fuzzy logic, and wavelet techniques, utilize the concepts of biological systems and human cognitive capabilities.

The above three systems have been recognized as a robust and attractive alternative to the some of the classical modeling and control methods. The major drawbacks of these architectures are the curse of dimensionality, such as the requirement of too many parameters in NWNs, the use of large rule bases in fuzzy logic, the large number of wavelets, and the long training times, etc. These problems can be overcome with network structures, combined two or all these systems [20].

Wavelets are mathematical functions that cut up data into different frequency components, and then study each component with a resolution matched to its scale. The fundamental idea behind wavelets is to analyze the signal at different scales or resolutions, which is called multiresolution. Wavelets are a class of functions used to localize a given signal in both space and scaling domains. A family of wavelets can be constructed from a mother wavelet. Compared to Windowed Fourier analysis, a mother wavelet is stretched or compressed to change the size of the window. In this way, big wavelets give an approximate image of the signal, while smaller and smaller wavelets zoom in on details. Therefore, wavelets automatically adapt to both the high-frequency and the low-frequency components of a signal by different sizes of windows. Any small change in the wavelet representation produces a correspondingly small change in the original signal, which means local mistakes will not influence the entire transform. The wavelet transform is suited for nonstationary signals (signals with interesting components at different scales) [19]. This makes wavelets interesting and useful. The sine and cosine functions which comprise the bases of Fourier analysis are non-local (and stretch out to infinity). They therefore do a very poor job in approximating sharp spikes. But with wavelet analysis, we can use approximating functions that are contained nearly in finite domains. Wavelets are well-suited for approximating data with sharp discontinuities [20].

Fuzzy controller with wavelet network is one of the succeed controller used in the process control in case of model uncertainties. But it may be difficult to fuzzy controller to articulate the accumulated knowledge to encompass all circumstance. Hence, it is essential to provide a tuning capability [2],[3]. There are many parameters in fuzzy controller can be adapted. The Speed control of turbine unit construction and operation will be described. Adaptive controller is suggested here to adapt normalized fuzzy controller, mainly output/input scale factor. The algorithm is tested on an experimental model to a robot arm of second order transfer function. A comparison between

Conventional method and Adaptive Fuzzy Controller with wavelet network is done.

This paper is organized as follows: in Section two, a brief introduction to wavelet analysis is presented and a wavelet network is constructed. Section three illustrates Proposed Neural Wavelet Network (NWN) structure and learning. PID is shown in section four, and its principle of operation and the design of its parameters using Ziegler-Nichols frequency method is shown in section five. Based on the proposed Neural Wavelet Network, the adaptive PID control used to achieve good performance is described in Section six. The simulation results and the conclusion are shown in section seven and eight.

2. Review of Wavelet Network

The origin of wavelet networks can be traced back to the work by Daugman (1988) in which Gabor wavelets were used for image classification. Wavelet networks have become popular after the work by Pati (1991, 1992), Zhang (1992), and Szu (1992). Wavelet networks were introduced as a special feed-forward neural network. Zhang applied wavelet networks to the problem of controlling a robot arm. As mother wavelet, they use the following function [8]:

$$\psi(a,b)(x) = \frac{1}{\sqrt{a}}\psi(\frac{x-b}{a})$$

Where ***a*** is dilation (i.e. a>0) and **b** is translation.

The wave-net algorithms consist of two processes: the self-construction of networks and the minimization of error. In the first process, the network structures applied for representation are determined by using wavelet analysis. The network gradually recruits hidden units to effectively and sufficiently cover the time-frequency region occupied by a given target. Simultaneously, the network parameters are updated to preserve the network topology and take advantage of the later process. In the second process, the approximations of instantaneous errors are minimized using an adaptation technique based on the LMS algorithms. The parameter of the initialized network is updated using the steepest gradient-descent method of minimization. Each hidden unit has a square window in the time-frequency plane. The optimization rule is only applied to the hidden units where the selected point falls into their windows. Therefore, the learning cost can be reduced [9], [10], [11].

3. Proposed neural Wavelet Network (NWN)

The term “wavelet” as it implies means a little wave. This little wave must have at least a minimum oscillation and a fast decay to zero, in both the positive and negative directions, of its amplitude. This property is analogous to an admissibility condition of a function that is required for the wavelet transform. Fig.1a is an example of a wavelet called “Morlet wavelet” named after Jean Morlet, the inventor, in 1984 [3]. Sets of “wavelets” are employed to approximate a signal and the goal is to find a set of daughter wavelets constructed by a dilated (scaled or compressed) and translated (shifted) original wavelets or mother wavelets that best represent the signal. So, by “travelling” from the large scales toward the fine scales, one “zooms in” and arrives at more and more exact representations of the given signal. Figs. (1a, b, c and d) display various daughter wavelets where *a* is a dilation and *b* is a translation corresponding to the Morlet mother wavelet [3].

The mother wavelet must satisfy the following admissibility condition and any admissible function can be a mother wavelet.

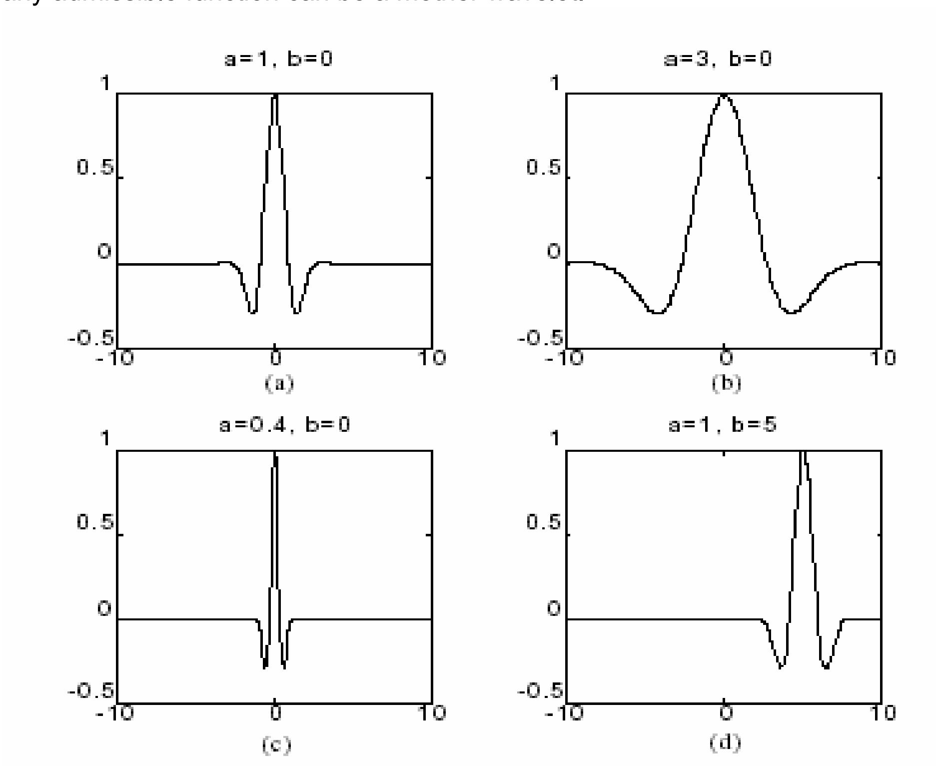


Fig. 1. Dilated and Translated Morlet Mother Wavelets.

$$c_h = \int_{-\infty}^{\infty} \frac{|H(\omega)|^2}{|\omega|} d\omega \quad < \infty \tag{1a}$$

Where: - $H(\omega)$ is the Fourier transform of $h(t)$.

The constant Ch is the admissibility constant of the function h(t). The wavelet transform of a function f with respect to a given admissible mother wavelet $h(t)$ is defined as:

$$W_f(a,b) = \int_{-\infty}^{\infty} f(t) h^*{}_{a,b}(t) dt \tag{1b}$$

Where * denotes the complex conjugate. However, most wavelets are real valued. The daughter wavelets are generated from a single mother wavelet $h(t)$ by dilation and translation:

$$h_{a,b}(t) = \frac{1}{\sqrt{a}} h(\frac{t-b}{a}) \tag{2}$$

Where a>0 is the dilation factor and b is the translation factor [3].

In 1958, Rosenblatt demonstrated some practical applications using the perceptron. The perceptron is a single level connection of McCulloch-Pitts neurons sometimes called single-layer feedforward networks. The network is capable of linearly separating the input vectors into a pattern of classes. In such an application, the network associates an output pattern (vector), and information is stored in the network by virtue of modifications made to the synaptic weights of the network [3]. Fig.2 illustrates perceptron, which is described by:

$$y_i = f\left(\sum_{j=1}^{N} w_{ij} x_j - v_i \right) \tag{3}$$

Where i = 1, 2... M (output nodes), j = 1, 2… (Inputs).

Rosenblatt derived a learning rule based on weights adjusted in proportion to the error between the output neurons and the desired output (target). The weight adaptations are given by:

$$\Delta w_{ij}(n) = \mu \; [\hat{y}_i(n) - y_i(n)] * x_j(n) \tag{4}$$

Where i = 1, 2… M (output nodes), j = 1, 2... N (inputs), $\hat{y}_i$ is the desired output at node i of time n and μ is a learning rate.

Wavelet transform have proven to be very efficient and effective in analyzing a very wide class of signals because of their attractive feature. Wavelet transform describes signals in terms of their local shifts. Thus, they provide a time-frequency representation, the generation of wavelets and a calculation of all wavelet expansions employing summation, not integrals, that is well matched to be implemented by digital computers [20].

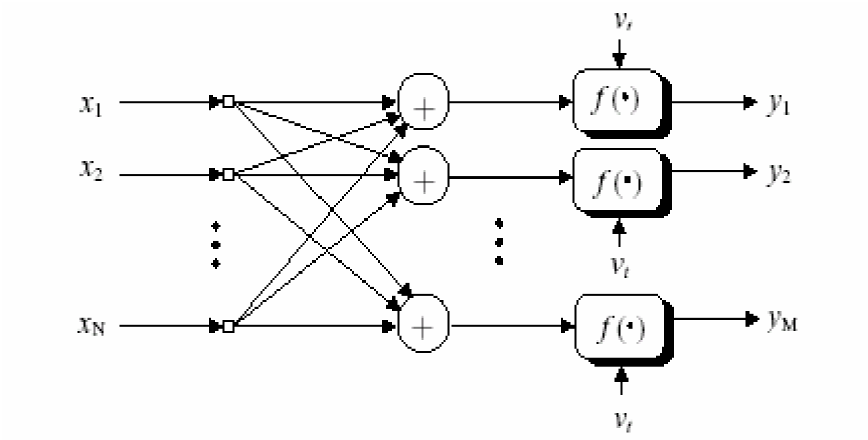


Fig. 2. Single-Layer Perceptron Feedforward

4. PID Controller

Here, an adaptive characteristic was suggested to improve a fuzzy NWN algorithm and reduce the learning cycle time. PID controller consists of: Proportional Action, Integral Action and Derivative Action. It is commonly refer to Ziegler-Nichols PID tuning parameters. It is by far the most common control algorithm [1]. In this paper, PID controller's algorithm is mostly used in feedback loops. PID controllers can be implemented in many forms. It can be implemented as a stand-alone controller or as part of Direct Digital Control (DDC) package or even Distributed Control System (DCS). The latter is a hierarchical distributed process control system which is widely used in process plants such as pharceumatical or oil refining industries. It is interesting to note that more than half of the industrial controllers in use today utilize PID or modified PID control schemes. A simple diagram illustrating the schematic of the PID controller feeding a second order system (plant, process) with a suitable signal is shown in Fig.3a.

In proportional control, the relationship between the output of the controller u(t) and the actuating error signal e(t) is

$$u(t)=K_P\,e(t)$$

Where:-

K_P is the proportional gain of the controller. It uses proportion of the system error to control the system. In this action an offset is introduced in the system.

In Integral control, $u(t)=K_I\int_0^t e(t)dt$

Where:-

K_I is the integral gain of the controller. It is proportional to the amount of error in the system.

In this action, the I-action will introduce a lag in the system. This will eliminate the offset that was introduced earlier on by the P-action. In derivative control action, $u(t)=K_D\frac{d\,e(t)}{dt}$, where K_D is the derivative gain of the controller. The value of the controller output here is changed at a rate of the actuating error signal e(t). Derivative control action, when added to a proportional controller, provides a means of obtaining a controller with high sensitivity.

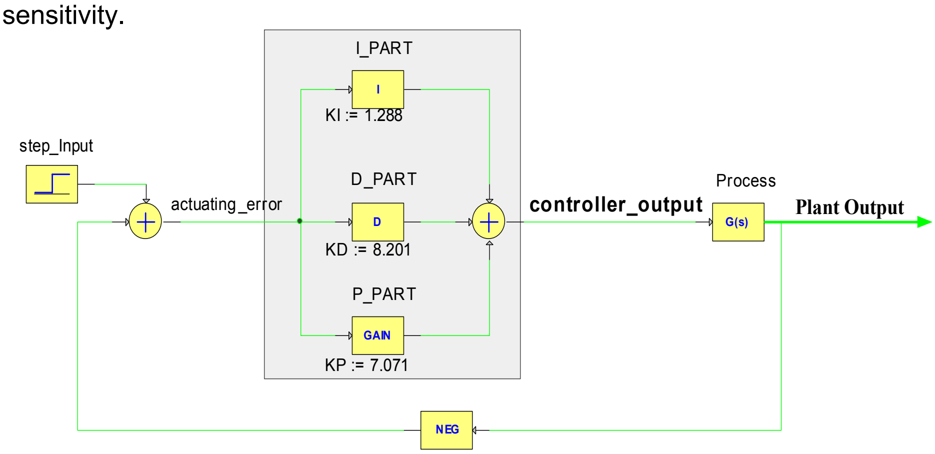


Fig. 3a. Schematic of the PID Controller – Non- Interacting Form

5. Experimental Process Identification and Design of PID Parameters

For the system under study, Ziegler-Nichols tuning rule based on critical gain, critical period and a certain deadbeat will be used. In this method, the integral time Ti will be set to infinity and the derivative time Td to zero. This is used to get the initial PID setting of the system. This PID setting will then be further optimized using the “steepest descent gradient method” (Fig.3b). The transfer function of the PID controller is

$$G(s) = K_p (1 + T_i + T_D) \tag{5}$$

The objective is to achieve a unit-step response curve of the designed system that exhibits a maximum overshoot of 25 %. In this method Z-N, only the proportional control action will be used. The K*p* will be increased to a critical value at which the system output will exhibit sustained oscillations as it is shown in Fig. 3c.

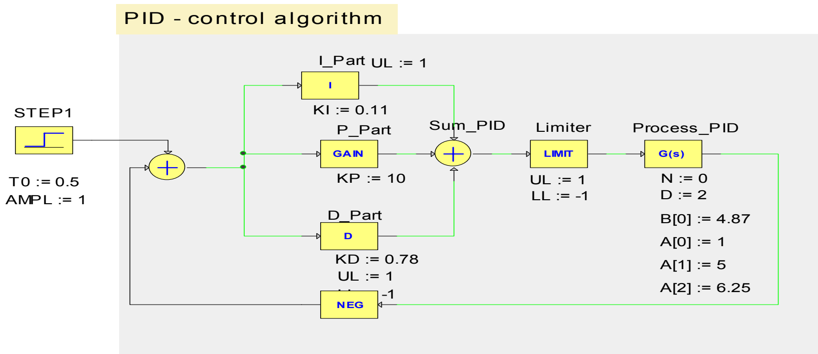


Fig. 3b. PID Control Algorithm

If the maximum overshoot is excessive, let's say about 50%, fine tuning should be done to reduce it to less than 25%. The transfer function of the process-PID can be approximated in the form of a second order transfer function $G(s) = \frac{4.75}{1 + 5s + 6.25s^2}$. The identified model is approximated as a linear model, but exactly the closed loop is nonlinear due to the limitation in the control signal. From Ziegler-Nichols frequency method of the second method [1], table (1) suggested tuning rule according to the formula shown. From these we are able to estimate the parameters of Kp, Ti and Td.

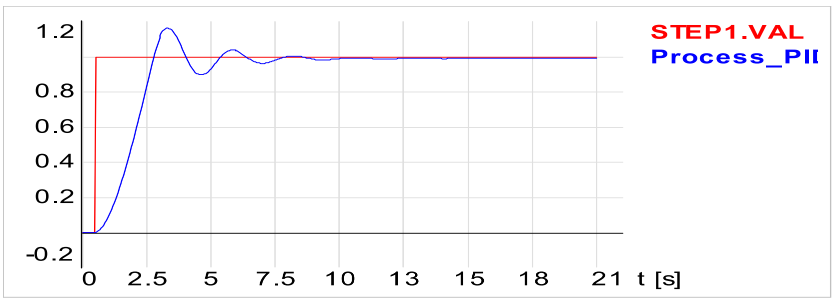
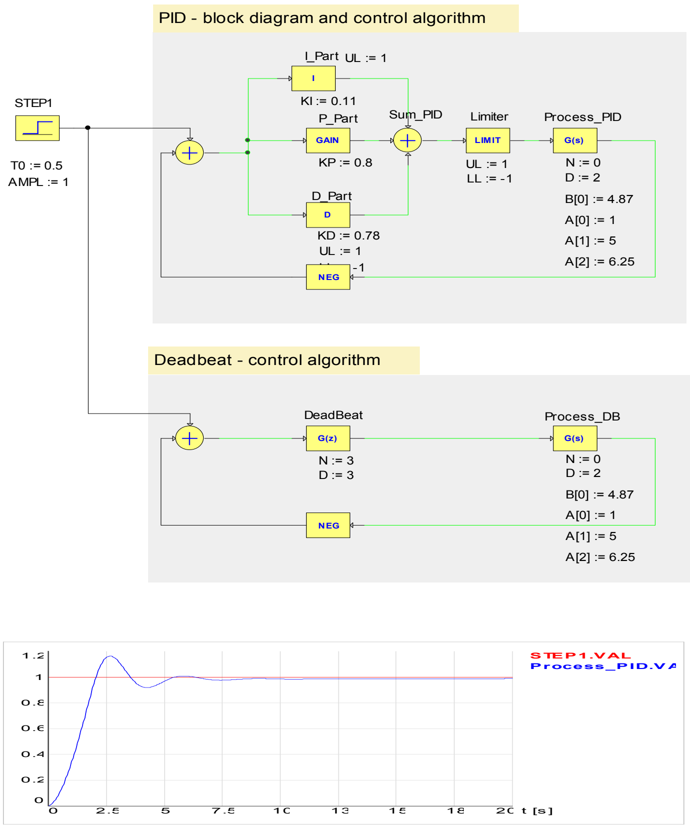


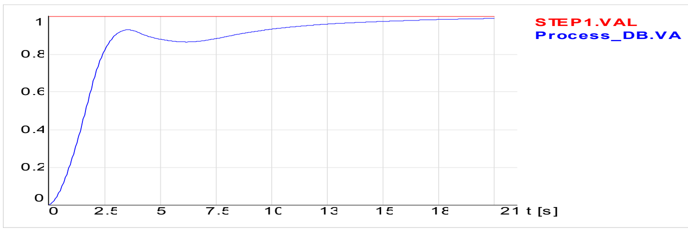
Fig. 3c. Illustration of Sustained Oscillation of PID with Classical Method

Table 1. Various Suggested Tuning Values

Controller	Kp	Ti	TD
P	0.5Kr	50ms	50ms
PI	0.5Kr	1/1.2Per	50ms
PID	0.8Kr	0.5Per	0.12Per



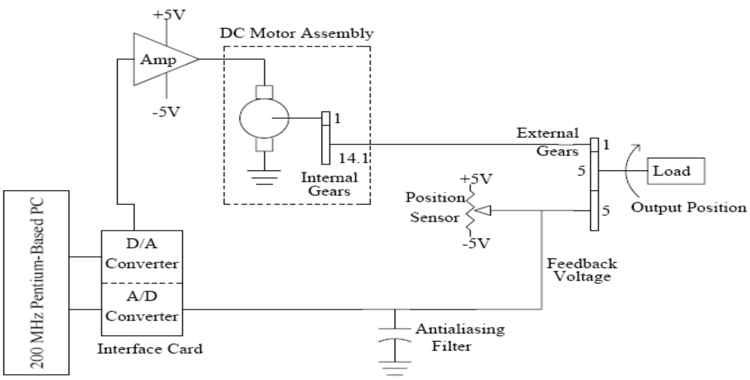
(a)



(b)

Fig. 4. Comparison between: (a) PID-Control Algorithm and (b) Deadbeat-Control Algorithm

In this paper, it is shown that the inefficiency of designing PID controller using the classical method will be improved by using deadbeat control algorithm and neural wavelet network algorithm. Fig. (4) shows the different dynamic behaviour of a (classical) PID algorithm and a time-discrete deadbeat controller with the same process. Now, the PID controller shown in Fig. (3) is replaced with fuzzy PID controller with NWN structure. Fuzzy controllers are designed with one specific set of rules (a rule base) that indicates what the plant input should be, given the current inputs to the controller. The fuzzy controller inputs are fuzzified to form fuzzy sets that can be used by the inference mechanism. The schematic circuit and block diagram of the fuzzy controller is shown in Fig.5.



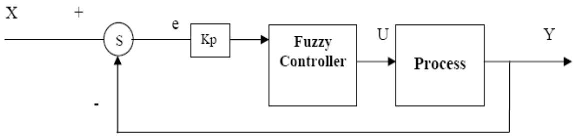


Fig. 5. Fuzzy Control System

The inference mechanism then decides what rules to apply for these inputs by matching the fuzzified inputs to the premises of the rules in the rule base. The inference mechanism provides a fuzzy set that indicates the certainty that the plant input should take on various values. Then de-fuzzification is used to convert the fuzzy set produced by the inference mechanism into a crisp output to be used by the plant. For this purpose we simply use the wavelet network.

6. Wavelet Network Algorithm

The wavelet architecture, shown in Fig.6 approximates any desired output signal $y(t)$ by generalizing a linear combination of a set of daughter wavelets $h_{a,b}(t)$, where the daughter wavelets $h_{a,b}(t)$ are generated by dilation *a* and translation *b* from a mother wavelet $h(t)$. Table (2) gives several mother wavelet filters and their derivatives [3].

$$h_{a,b}(t) = h(\frac{t-b}{a}) \tag{6}$$

Where *a>0* … Dilation factor, *b* … Translation factor.

A wavelet network is a 3-layer feed forward neural network [14]. First the WN parameters, dilation *a's*, translation *b's*, and weight *w's* should be initialized and the desired sets of data, the input signal *x(t)*, the desired output (target) *y(t)* and the number of wavelons *k* are given. Assuming that the network output function satisfies the admissibility condition and the network sufficiently approximates the target. The approximated signal of the network $\hat{y}(t)$ can be represented by equation (7):

$$\hat{y}(t) = x(t) \times \sum_{i=1}^{k} w_i h_{a_i, b_i}(t) \tag{7}$$

Where *x(t)* is the input signal, *k* is a number of wavelons, w_i is the weight coefficients between hidden and output layer, i=1,2,…,k., and $h_{a,b}(t)$ is a set of daughter wavelets generated from a mother wavelet $h(t)$ as described in equation (7).

Table 2. Derivatives of the various wavelet filters with respect to its translation

Linear	$f(x) = kx$
Step (commonly: $\beta = 1,\ \delta = 0,\ x_k = 0$)	$f(x) = \begin{cases} \beta & \text{if } x \geq x_k \\ \delta & \text{if } x < x_k \end{cases}$
Ramp	$f(x) = \begin{cases} \rho & \text{if } x \geq \rho \\ x & \text{if } \lvert x \rvert < \rho \\ -\rho & \text{if } x \leq -\rho \end{cases}$
Sigmoid	$f(x) = \frac{1}{1+e^{-\alpha x}}, \quad \alpha > 0$
Hyperbolic Tangent	$f(x) = \tanh(\gamma x) = \frac{1-e^{-2\gamma x}}{1+e^{-2\gamma x}}, \quad \gamma > 0$
Rational	$f(x) = \begin{cases} \frac{x^2}{1+x^2} & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$
Gaussian	$f(x) = \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]$

After constructing the initial WN and after calculating output signal of the network, the training of WN starts. It is further trained by the gradient descent algorithms like least mean squares (LMS) to minimize the mean-squared error. During learning, the parameters of the network are optimized. The wavelet network parameters w_i, a_i, and b_i can be optimized in the LMS algorithm by minimizing a cost function or the energy function, E, over all function interval. Thus by denoting

$$e(t) = y(t) - \hat{y}(t) \tag{8}$$

The energy function is defined by:

$$E = \frac{1}{2} \sum_{t=1}^{T} e^2(t) \tag{9}$$

$$E = \frac{1}{2} \sum_{t=1}^{T} (y(t) - \hat{y}(t))^2 \quad (10)$$

Where T the total interval of function is, $y(t)$ is the desired output (target) and $\hat{y}(t)$ is the actual output signal of NWN.

To minimize E, the method of steepest descent is used which requires the gradients $\frac{\partial E}{\partial w_i}$, $\frac{\partial E}{\partial a_i}$, and $\frac{\partial E}{\partial b_i}$ for updating the incremental changes to each particular parameter w_i, a_i, and b_i, respectively. The gradients of E are given by the following expressions:

$$\frac{\partial E}{\partial w_i} = -\sum_{t=1}^{T} e(t) h(\tau) x(t) \quad (11)$$

$$\frac{\partial E}{\partial b_i} = -\sum_{t=1}^{T} e(t) x(t) w_i \frac{\partial h(\tau)}{\partial b_i} \quad (12)$$

$$\frac{\partial E}{\partial a_i} = -\sum_{t=1}^{T} e(t) x(t) w_i \tau \frac{\partial h(\tau)}{\partial b_i} = \tau \frac{\partial E}{\partial b_i} \quad (13)$$

Where $\tau = \frac{t - b_i}{a_i}$ The derivatives of the various wavelet filters with respect to its translation $\frac{\partial h(\tau)}{\partial b_i}$ are given in Table (2). The incremental changes of each coefficient are simply the negative of their gradients:

$$\Delta w = -\frac{\partial E}{\partial w} \quad (14)$$

$$\Delta b = -\frac{\partial E}{\partial b} \tag{15}$$

$$\Delta a = -\frac{\partial E}{\partial a} \tag{16}$$

Thus each coefficient w, b, and a of the network is updated in accordance with the rule:

$$w(t+1) = w(t) + \mu_w \Delta w \tag{17}$$

$$b(t+1) = b(t) + \mu_b \Delta b \tag{18}$$

$$a(t+1) = a(t) + \mu_a \Delta a \tag{19}$$

Where μ is the fixed learning rate parameter [3]. The network parameters are modified using the gradient descent algorithm till one of the stopping conditions is satisfied. The algorithm is stopped when one of several conditions is satisfied: the Euclidean norm of the gradient, or of the variation of the gradient, or of the variation of the parameters, reaches a lower bound, or the number of iterations reaches a fixed maximum, whichever is satisfied first. The final performance of the wavelet network model depends on whether: (i) the assumptions made about the model are appropriate, (ii) the training set is large enough, (iii) the family contains a function which is an approximation of *f* with the desired accuracy in the domain defined by the training set, (iv) an efficient training algorithm is used [20].

7. Morlet Mother Wavelet Basis Functions

In this section, it will be shown through experimental simulations how various types of mother wavelet basis functions perform their learning ability. As we know, wavelets are orthogonal basis functions, thus they can be added or removed one at a time without having to update the parameters of the previously placed basis functions resulting to an incrementally improvement at low computational cost. Moreover, wavelets are local basis functions that provide less interfering than global ones, leading to a noncomplex dependency in the neural network (NN) parameters. The following sections will confirm this idea by providing several observations derived from the results of the MATLAB simulations.

Assuming the training data are stationary and sufficiently rich, good performance can usually be achieved with a small learning rate. Thus, all learning rate parameters for weights, dilations, translations and feedback coefficients are fixed at 0.01, 0.05, 0.05, 0.02, and 0.02, respectively. All initial weights w_k and dilations a_k are set to 0 and 10, respectively. Note that if the dilation parameters are set too wide, they can cause several overlapping partitions and thus cannot be realized.

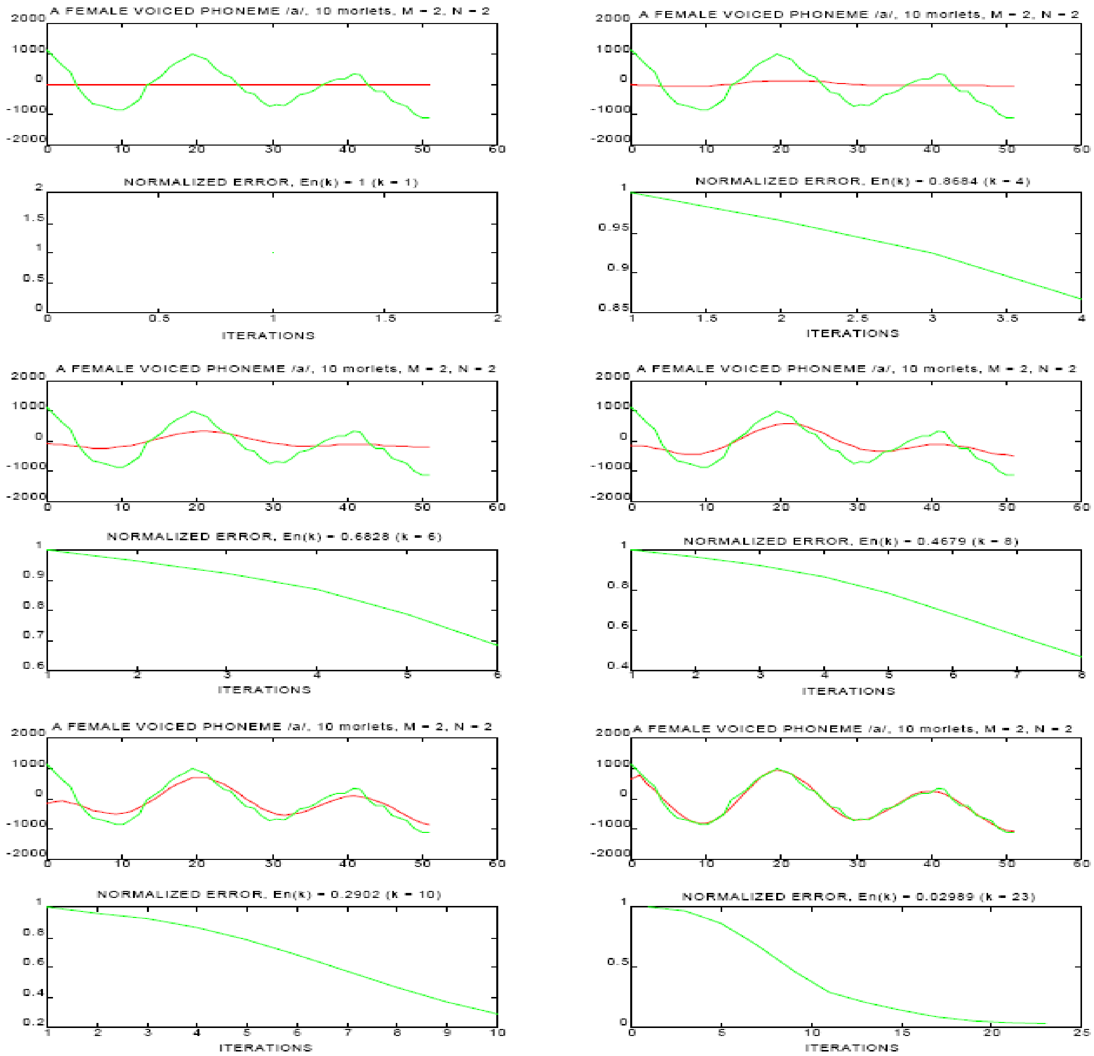


Fig. 6a. Wavenet Simulations with 30 Morlet Wavelets: Voiced model output (green), NN APPRX. OUTPT (red)

Setting a_k too narrow may result in longer convergence. Initial translation parameters b_k are spaced equally apart throughout the training data to provide non-overlapping partitions throughout the neighbouring intervals. Finally, the initial coefficients *c* and *d* should be set so that the system has poles inside the unit circle, thus both are set to 0.1. The number of coefficients for each feedforward and feedback *m* and *n* are both set to 2 as well. The learning epoch will terminate when the desired normalized error of 0.03 is reached. Figure 6 will describe the results of the *wavenet* network performance employing Morlet.

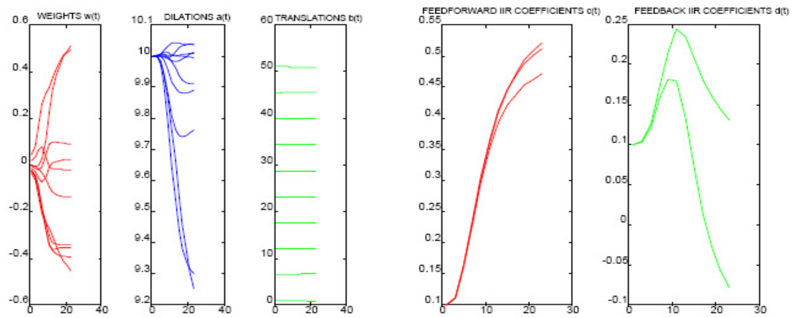


Fig. 6b. Wavenet Parameter Updates with 10 Morlet Wavelets

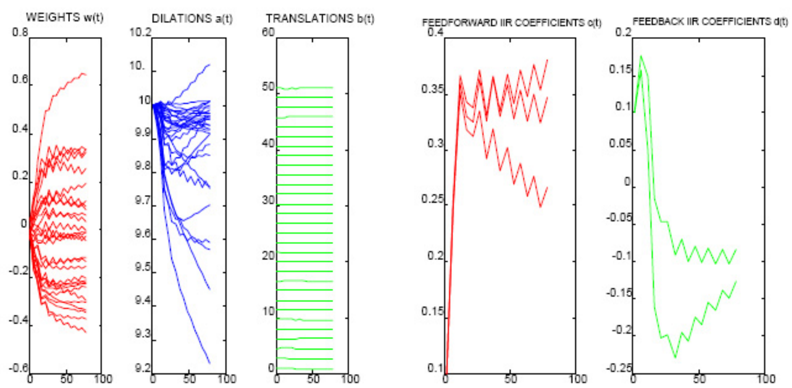


Fig. 6c. Wavenet Parameter Updates with 30 Morlet Wavelets

Figs.6b and 6c capture the learning performance of the *wavenet* network using 10 and 30 Morlet wavelets, respectively. We can conclude that the *wavenet* network composed of more wavelets can reach initial convergence with reference to the number of iterations very rapidly. However, to reach the desired error goal 0.03, networks with a large number of wavelets cannot converge easily and the error performance starts to oscillate. This behavior may be caused by the rate of learning stepsize not being small enough, causing the iteration process to bounce between the two opposite sides of a valley rather than following the natural gradient contour (as shown in Fig. 6c). Fig. 6d and Table 3 provide information on the Morlet *wavenet* characteristic. As we can see, when the number of wavelets K is small, for example, for K = 3, it takes 28 iterations to reach error of 0.8 while it takes 3 iterations for K = 35, but when the error of 0.03 is the target, K = 8 takes 23 iterations while K = 30 takes 432 iterations. Large K is also undesirable to the expense of more coefficients to be updated. Small K can also take a large amount of time to compute as for K = 3 take more than 2000 iterations to reach error of 0.04. In conclusion, the number of Morlet wavelets between K = 8 to K = 16 is sufficient to approximate the unknown voice model.

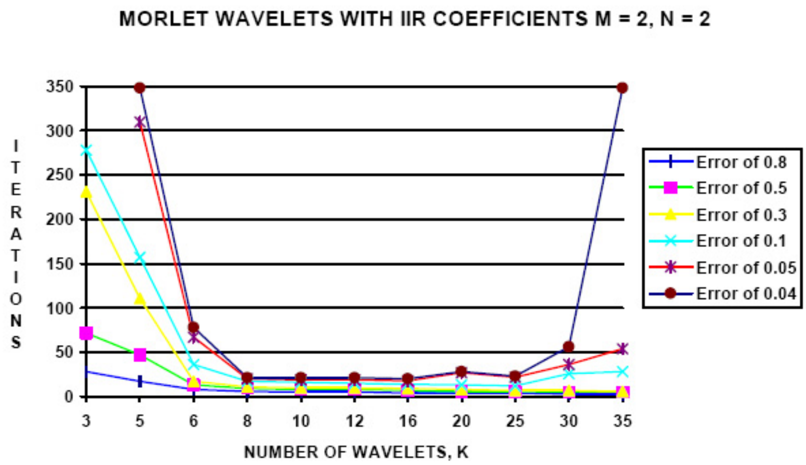


Fig. 6d. Iterations vs. Number of Morlet Wavelets per Normalized Errors

Table 3. Number of Iterations vs. Number of Morlet Wavelets Employed

				Number of Wavelets, K				**Number of**
35	**30**	**25**	**20**	**10 12 16**	**8**	**6**	**3 5**	**Iterations**
3	3	4	4	5 5 4	6	8	28 17	**Error of 0.8**
5	5	6	6	8 8 7	9	13	72 47	**Error of 0.5**
6	7	7	8	10 10 9	11	17	231 111	**Error of 0.3**
28	26	12	13	16 15 14	17	36	278 157	**Error of 0.1**
54	36	22	27	19 19 18	20	67	853 310	**Error of 0.05**
348	56	23	28	21 21 20	21	78	2000+ 348	**Error of 0.04**
432	78	27	29	23 24 23	23	94	671	**Error of 0.03**

8. Simulation OF Configuration and Results

The fuzzy NWN structure and its learning algorithm are used for development controller to control of dynamic plant. The structure of control system is given in figure 7. Neural control system synthesis is performed in the closed control system. For learning of fuzzy NWN the error between target characteristic of control system and output of control object $\Delta(y,t) = K_e\,(g(t) - y(t))$ is used. This error is used for correction network parameters for adjusting of controller. Using learning algorithm the optimal values of weight coefficients of fuzzy NWN controller are found.

The NWN algorithm, with one hidden layer of twenty [20 Morlet, 20 Slog1] wavelons in the hidden neurons (k = 20) and fixed learning rate of 1 is implemented to identify this hard nonlinear function:

$$\begin{aligned} f(x) = 2\cdot 0025\ (1-x) + e^{2x-1}\sin(3\pi(x-0.6)^2) \\ + e^{3(x-0.5)}\sin(4\pi\ (x-0.9)^2) \end{aligned} \tag{20}$$

Initial *w's* and dilations *a's* are set to 0 and 9 respectively. *b's* are spaced equally apart throughout the training data. The block diagram of Fuzzy controller with a second order system is shown in Fig. 8. The simulation results shown in Fig. (9a, b, c). Figure (9a) shows the MSE against the number of iterations for off-line training of the network. Fig. 10 illustrates the performance results of the network identifying the function given above.

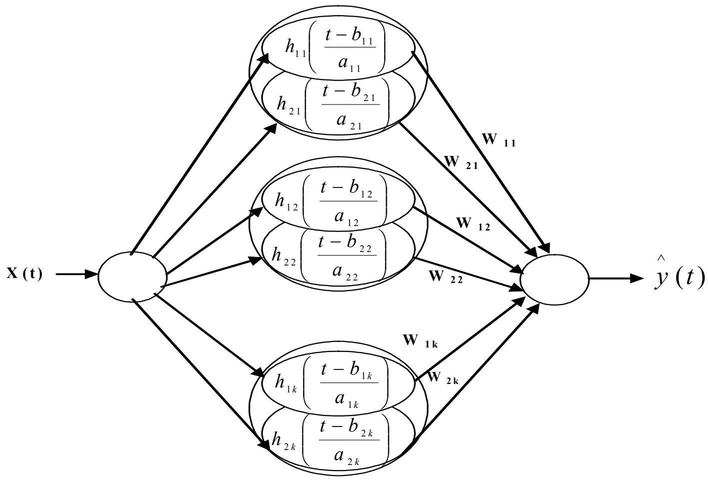


Fig. 7. Structure of NWN

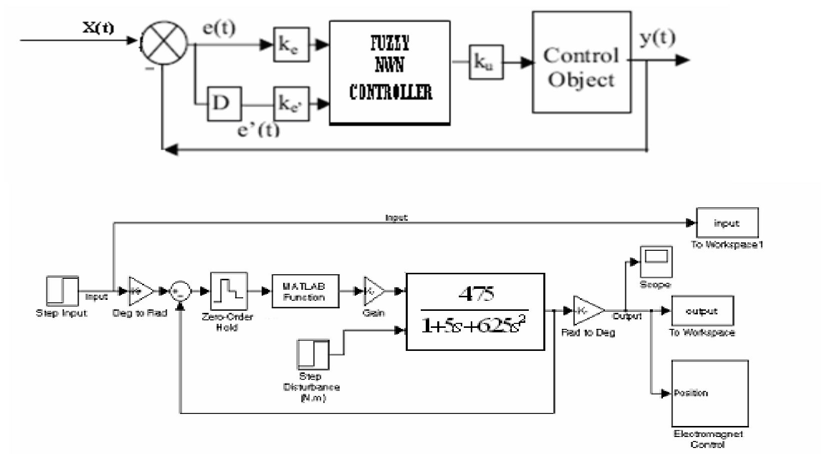
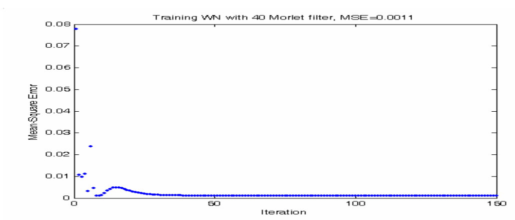
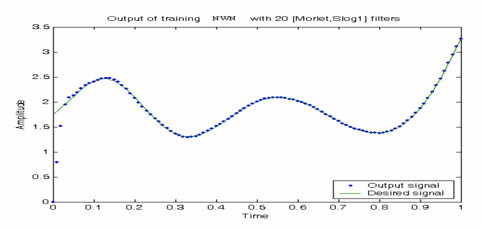


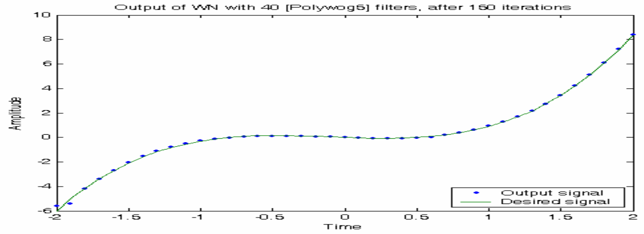
Fig. 8. The Simulink Control Model of a Second Order System with NWN Controller



(a) Mean-Square Error per learning iteration

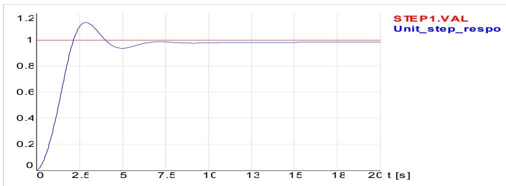


(b) Desired and Identified output signal per time.

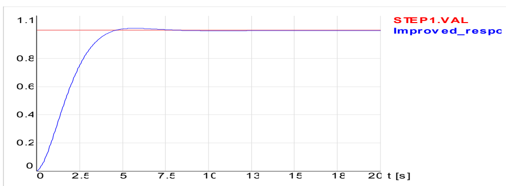


(c) Desired and Identified output signal per time

Fig. 9. Simulation results of NWN training



(a) Step Response of Designed System



(b) Improved system

Fig. 10. Input – output data set

9. Conclusion

In this paper, an advanced wavelet network, called Neural Wavelet Network is presented as an interesting alternative to wavelet networks. This technique absorbs the advantage of high resolution of wavelets and the advantages of learning and feed-forward of neural networks. The algorithm of function identification is designed and implemented using Matlab 7 and Simplorer 6 tool. The Neural Wavelet Network (NWN) structure is implemented and an example of a second order system is carried out to verify this implementation.

It can be concluded that this structure achieves an approximation assuming reasonable choice of the number of wavelets and mother wavelet basis functions. The Neural Wavelet Network is proved to be a controller analogous to PID controller. After the off-line training of the NWN controller, it shows the ability to get the specified position response exactly at the specified time when it's embedded in the control system. No significant difference between the position responses for both PID and the proposed NWN controllers, indicating further the validity of the idea of this research.

10. References

1. Panrong, X.: Image Compression by Wavelet Transform. A Thesis presented to the faculty of Computer and Information Sciences East Tennessee State University, August 2001.
2. Sgarbiy, M., Colla, V., and Reyneri, L.M.: A Comparison Between Weighted Radial Basis Functions and Wavelet Networks. Department of Electronic, University of Torino, C.so Duca degli Abruzzi 24, 10129 TORINO, ITALY, 1997.
3. Anthony J., and Rolf T.: Nonlinear Adaptive Flight Control using Neural Networks", Georgia Institute of Technology School of Aerospace Engineering Atlanta, GA, 30332, 1996.
4. Xin, Y., Senior member, IEEE: Evolving Artificial Neural Networks", Proceedings of the IEEE, Vol. 87, No. 9, September, 1999.
5. Dongbing, G. and Huosheng H..: Neural Predictive Control for a Car-like Mobile Robot. Department of Computer Science, University of Essex Wivenhoe Park, Colchester CO4 3SQ, UK, International Journal of Robotics and Autonomous Systems, Vol. 39, No. 2-3, May, 2002.
6. Oussar, Y., Rivals, I., Personnaz, L., and Dreyfus, G.: Training Wavelet Networks for Nonlinear Dynamic Input-Output Modeling. Laboratory of Electronic Superior School of Physical and Chemistry Industrial 10, rue Vauquelin F - 75231 PARIS Cedex 05, FRANCE, 1996.
7. Yacine, O., Gerard D.: Initialization by Selection for Wavelet Network Training. Laboratory of Electronic Superior School of Physical and Chemistry Industrial 10, rue Vauquelin F - 75231 PARIS Cedex 05, FRANCE, 1996.
8. Zhang, Q. and Benveniste A. :Wavelet Networks. IEEE Transactions on Neural Networks, Vol. 3, No. 6, November 1992.

9. Zhang, Q.: Using Wavelet Network in Nonparametric Estimation. IRISA, Institut de Recherche en Informatique et System Aleatoires, Campus de Beaulieu-35042 Rennes Cedex-France, 1994
10. Parvez S.: Advanced Control Techniques for Motion Control Problem. Paper pp. 401-408, at ISA 2003.
11. Kobayashi K. and Torioka T.: A Wavelet Neural Network for Function Approximation and Network Optimization. Proceeding of ANNIE'94 (Vol.4).
12. Oussar Y., Rivals I., Personnaz L., Dreyfus G.: Training Wavelet Networks for Nonlinear Dynamic Input-Output Modeling. Neurocomputing, in press, F - 75231 PARIS Cedex 05, FRANCE. Phone: 33 1 40 79 45 41 Fax: 33 1 40 79 44 25, E-mail: Yacine.Oussar@espci.fr.
13. Carranza M.: Verifying a Knowledge-Based Fuzzy Controller. Computational Logic Inc., 1717 W. 6th St. Suite 290 Austin, Texas 78703, Technical Report 82, September 1992.
14. Kelkar N.D.: A Fuzzy Controller for Three Dimensional Line Following of an Unmanned Autonomous Mobile Robot. University of Cincinnati, Department of Electromechanical, Industrial, and Nuclear Engineering, A master thesis 1997.
15. Marc T.: A REVIEW OF WAVELET NETWORKS, WAVENETS, FUZZY WAVENETS AND THEIR APPLICATIONS. email: Marc.thuiUard @ cerberus.ch
16. Brain E. Miller, J. Edward C.: Using A Wavelet Network to Characterize Real Environments for Haptic Display. e-mail : blemillerfgnwu.edu .
17. Lekutai G.: Adaptive Self-Tuning Neuro Wavelet Network Controllers. Virginia Polytechnic Institute PHD thesis, Blacksburg, Virginia, 1997.
18. Mosavi, M. R.: A wavelet based neural network for DGPS corrections prediction. WSEAS Transaction on Systems, 3(10), pp. 3070-3075, 2004.
19. Mosavi, M. R.: Comparing DGPS corrections prediction using neural network, fuzzy neural network, and Kalman filter. International Journal of GPS Solutions, 10(2), pp. 97- 107, 2006a.
20. Parker, T. E., and Matsakis, D.: Time and frequency dissemination advances in GPS transfer techniques. GPS World Magazine, pp. 32-38, 2004.

Ibrahim M. El Emary received the Dr. Eng. Degree in 1998 from the Electronic and Communication Department, Faculty of Engineering, Ain shams University, Egypt. From 1998 to 2002, he was an Assistant Professor of Computer science in different faculties and institutes in Egypt. Currently, he is a Visiting Associate Professor at Al Ahliyya Amman University. His research interests include: analytic simulation techniques, performance evaluation of communication networks, application of intelligent techniques in managing computer communication network, bioinformatics and business intelligence as well as performing a comparative studies between various policies and strategies of routing, congestion, sub netting of computer communication networks. He published more than 100 articles in international refereed journals and conferences.

Musbah J. Aqel obtained his M.Sc. in Computer Engineering from Aligarh Muslim University – India and his PhD from I.T, Banaras Hindu University-India in Computer Engineering. He is currently associated professor of computer engineer, and chairman of Electrical and Computer Engineering

Dept, Applied science university-Jordan. He joined Electronic Imaging and media communication department at the University of Bradford as a post doctoral. He is an editor in International journal of soft computing and applications. His research interest includes knowledge-base system design, image processing, computer networks, and simulation techniques.

Walid Emar was born in Palestine, in 1970. He received the Dipl.-Ing. (M.Sc.) degree in industrial electronics and the Dr.-Ing. (Ph.D.) degree in power electronics and control from the University of West Bohemia, Pilsen, Czech Republic. In 2001, he joined Škoda Transportation, Škoda Company , Pilsen, Czech Republic. In 2002, he joined Birzeit University, Electrical and Computer Department. Currently, He is now teaching at Al-Isra Private University as a full-time assistant professor. He is engaged in research and development of high power multilevel converters for industrial and traction applications, as well as dc/dc converters and regulators.

Received: October 31, 2007; Accepted: October 05, 2009.