Al-Khwarizmi Engineering Journal,Vol. 13, No. 1, P.P. 107- 122 (2017)

Al-Khwarizmi Engineering Journal

Model Reference Adaptive Control based on a Self-Recurrent Wavelet Neural Network Utilizing Micro Artificial Immune Systems

Omar Farouq Lutfy* **Maryam Hassan Dawood****

*,**Department of Control and Systems Engineering/ University of Technology

*Email: 60157@uotechnology.edu.iq

**Email: mariam.hassan93@yahoo.com

(Received 18 September 2016; accepted 4 January 2017)

https://doi.org/10.22153/kej.2017.01.006

Abstract

This paper presents an intelligent model reference adaptive control (MRAC) utilizing a self-recurrent wavelet neural network (SRWNN) to control nonlinear systems. The proposed SRWNN is an improved version of a previously reported wavelet neural network (WNN). In particular, this improvement was achieved by adopting two modifications to the original WNN structure. These modifications include, firstly, the utilization of a specific initialization phase to improve the convergence to the optimal weight values, and secondly, the inclusion of self-feedback weights to the wavelons of the wavelet layer. Furthermore, an on-line training procedure was proposed to enhance the control performance of the SRWNN-based MRAC. As the training method, the recently developed modified micro artificial immune system (MMAIS) was used to optimize the parameters of the SRWNN. The effectiveness of this control approach was demonstrated by controlling several nonlinear dynamical systems. For each of these systems, several evaluation tests were conducted, including control performance tests, robustness tests, and generalization tests. From these tests, the SRWNN-based MRAC has exhibited its effectiveness regarding accurate control, disturbance rejection, and generalization ability. In addition, a comparative study was made with other related controllers, namely the original WNN, the artificial neural network (ANN), and the modified recurrent network (MRN). The results of these comparison tests indicated the superiority of the SRWNN controller over the other related controllers.

Keywords: Artificial neural network, micro artificial immune system, model reference adaptive control, self-recurrent wavelet neural network , Wavelet neural network.

1. Introduction

Adaptive control techniques have gained widespread popularity among researchers due to their ability in handling different types of complex and nonlinear control problems. Model reference adaptive control (MRAC) is a major type of adaptive control strategies. Owing to its ability to guarantee the global asymptotic stability, the MRAC scheme has been successfully employed to control different linear systems [1-5]. However, it is well-known that most real systems are inherently nonlinear in nature, and hence, it is impractical to describe such systems by linear mathematical models. This fact limits the effectiveness of utilizing the conventional MRAC scheme to control such nonlinear systems. In order to mitigate this difficulty, many researchers have utilized the nonlinear function approximation capability of artificial neural network (ANN) to form a nonlinear ANN-based MRAC scheme to control nonlinear systems [6-10]. Despite this widespread utilization of ANN, recently a more powerful neural network structure, namely the wavelet neural network (WNN), has received extensive attention in the literature, especially in handling various control problems [11-14]. The strength of WNN approximation capability is realized by combining the theory of wavelets and the ANN into a unified

framework. Hence, WNNs possess both the localization property of wavelet transform together with the learning and generalization abilities of ANNs. In spite of these desirable properties of WNNs, only few efforts have been made to employ WNNs in MRAC schemes. For instance, Wai and Chang [15] employed a WNN-based MRAC scheme to control an induction motor drive. In addition, the authors used the back stepping approach to design another feedback control action to get the required performance. In this method, to optimize the above structure, several learning rates had to be chosen according to the controlled system. Therefore, the above method is characterized by its difficulty in implementation due to the involvement of two complex control methods. As another MRAC system, Yoo et al. [16] used two SRWNNs, one of which acts as an identifier and the other as a controller, in a MRAC structure to control nonlinear systems. Both of these SRWNNs were trained by a gradient descent (GD) method to achieve the desired response. However, in addition to the complexity of this control approach, GD techniques are characterized by their slow convergence rates and the inclination to get stuck at local minima in the search space [3]. In order to avoid these limitations in GD methods, more attention has been given to evolutionary algorithms, due to their ability in finding the global optimal solution of a particular problem. In this regard, genetic algorithms (GAs) are regarded as the most essential types of evolutionary algorithms [17, 18]. However, in recent years, a more powerful evolutionary algorithm has received considerable attention among researchers. This relatively new algorithm is the artificial immune system (AIS), which was built based on some concepts from the natural immune system. Compared to the GA, the AIS algorithm has a more efficient mutation operator which results in a better diversity of populations [19, 20]. In this regard, utilizing this promising optimization method, Lutfy [21] proposed to use a modified micro artificial immune system (MMAIS) to train a WNN as the main controller in the MRAC scheme. However, in the above work, the WNN controller was treated as a black box approximator without using an initializing phase, which can affect the WNN approximation capability. In this context, it is worth noticing that the wavelet is a rapidly vanishing function which is fully defined by dilation and translation factors [22]. Consequently, suitable initialization of these factors plays an important role in improving the WNN approximation capability. However, despite this importance for the initialization process, several researchers did not consider a specific initialization approach for these parameters [11, 13, 16, 21, 23].To this end, aiming at enhancing the performance of the WNN controller, the motivation of the present work was to propose a more efficient version of the WNN structure. This modified WNN structure encompasses two amendments, namely; adopting an initialization phase to enhance the convergence to the optimal weights and including self-feedback connections to the wavelons in the wavelet layer. These modifications have contributed in improving the approximation capability of the proposed controller compared to other related controllers, as will be seen in the simulation results of the present work. Moreover, an on-line MMAIS-based adaptive control design is used to further enhance the performance of the SRWNN controller. To the best of the authors' knowledge, this is the first utilization of the MMAIS in an on-line adaptive control design. The remaining content of this paper is arranged as follows: Section 2 discusses the utilization of the SRWNN controller within the MRAC. The SRWNN structure is elucidated in Section 3. Basic concepts of the AIS and the MMAIS algorithms are given in Section 4. To demonstrate the efficiency of the proposed SRWNN-based MRAC, several performance tests along with two comparative studies are presented in Section 5. Finally, the main conclusions are drawn in Section 6.

2. The SRWNN-based MRAC Structure

The main idea behind the MRAC is to specify the desired behavior of the controlled system by a reference model. This reference model is simply a plant of known dynamical structure whose task is to provide the desired output by applying a given input signal. Then, by applying a suitable optimization method, the main goal of MRAC design is to regulate the controller parameters by minimizing the difference between the outputs of the reference model and the controlled system [24]. Utilizing this basic MRAC design approach, the SRWNN structure is employed in this work as the main controller to constitute a nonlinear MRAC scheme. Figure 1 depicts a block diagram of the proposed SRWNN-based MRAC scheme.

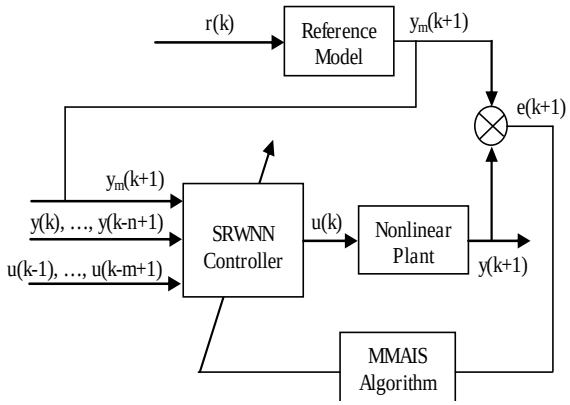


Fig. 1 Structure of the SRWNN- based MRAC system.

In particular, the design of the SRWNN controller in the framework of MRAC is based on the generalized inverse control technique. In other words, the SRWNN is trained to work as an inverse controller, as illustrated in Fig. 1. In order to clarify this design approach, consider the following nonlinear autoregressive moving average (NARMA) model, which is utilized to describe single-input single output (SISO) discrete-time nonlinear systems [9, 21, 24]:

$$(k+1) = f\big(y(k), y(k-1), \dots, y(k-n+1), u(k), \dots, u(k-m+1)\big) \quad \dots (1)$$

Where $y(k)$ and $u(k)$ are the system output and input, respectively, $f(.)$is a smooth nonlinear function, n and m are the system orders, and k represents the discrete-time instant. Subsequently, In order to deduce the control law, it is assumed that Equation (1) above is invertible resulting in the following equation:

$$(k) = h\big(y_m(k+1), y(k), y(k-1), \dots, y(k-n+1), u(k-1), \dots, u(k-m+1)\big) \quad \dots (2)$$

Where,$y_m(k+1)$ represents the output of the reference model at time instant $(k+1)$ and $h(.)$is the inverse function of f in Equation (1), such that:$h(.) = f^{-1}(.)$. In order to realize the control law given in Equation (2), a suitable function approximator must be used to approximate the inverse function $h(.)$. In particular, due to its remarkable nonlinear function approximation capability, the SRWNN is employed in this work to achieve this task. As can be seen from Fig. 1, the training of the SRWNN structure is accomplished by minimizing the error signal between the outputs of the actual system and the reference model. More precisely, this error signal is described by the following formula [22]:

$$E = \frac{1}{2}\sum_{k=1}^{N}(y(k+1) - y_m(k+1))^2, \quad \dots (3)$$

Where N represents the number of samples. In particular, this error signal is utilized as the performance index to be minimized viaan appropriate optimization method. In the present work, the MMAIS algorithm is employed for this task, as illustrated in Fig. 1.

3. The SRWNN Structure

As mentioned before, the proposed SRWNN structure, which is shown in Fig. 2, is an improved version of the previously reported WNN structure [21].

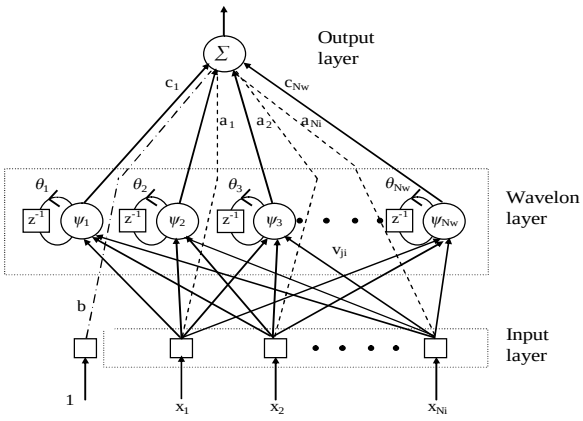


Fig. 2. Structure of the SRWNN controller.

As depicted in Fig. 2, the SRWNN is a multi-input single-output network which has an input layer, a mother wavelet (wavelon) layer, and an output layer. These layers work together to generate the final output of the SRWNN structure. In the following, the functions of these layers are discussed in detail [16, 21]:

Layer 1: The function of this layer, which is called the input layer, is to convey the input variables as they are to the next layer. As explained in Section 2, the utilization of the SRWNN as the controller in the MRAC requires the selection of the input variables to be of the form:

$$[y_m(k+1), y(k), \dots, y(k-n+1), u(k-1), \dots, u(k-m+1)] \quad \dots (4)$$

Therefore, the number of nodes in this layer depends on the orders of the controlled system.

Layer 2: This layer is referred to as the mother wavelet or the wavelon layer. Each node (wavelon) in this layer possesses a mother wavelet and a self-feedback connection with an adjustable parameter. As the mother wavelet, the Mexican hat function is employed in this work, and this function has the following form:

$$(x) = (1 - x^2)e^{-\frac{1}{2}x^2} \quad \ldots (5)$$

To compute the wavelon outputs, the wavelet y_j of each node is obtained from its mother wavelet function ψ according to the following expression:

$$\psi_j(x) = \psi(z_j), \quad with\ z_j$$
$$= d_j\left(\sum_{i=1}^{N_i} v_{ji}x_i + \psi_j(k-1).\theta_j\right)t_j \quad \ldots (6)$$

where d_j and t_j demarcate dilation and translation factors of the SRWNN, respectively, Ni is the number of input nodes, x_i denotes the i^{th} input variable, v_{ji} is the link connection from node i in the input layer to wavelon j in the mother wavelet layer, $y_j(k-1)$ represents the memory term, whose task is to store the previous state of the j^{th} wavelon, and q_j is the link value of the self-feedback connection belonging to the j^{th} wavelon. Specifically, this weight value is responsible for deciding the rate of information storage. Finally, the j^{th} wavelon's output is calculated by making use of the Mexican hat function as expressed in the following:

$$(z_j) = (1 - z_j^2)exp\left(-\frac{1}{2}z_j^2\right) \quad \ldots (7)$$

It is worth to highlight that as an advantage of the SRWNN over the WNN, the self-feedback connections in the wavelon layer have the ability to reserve the previous network information. In other words, the current network state contributes in producing the network output for the next sample of time. This feature enhances the overall approximation ability of the SRWNN compared to the conventional WNN, as will be demonstrated in the simulation results of this work.

Layer 3: This layer, known as the output layer, has only one node. The task of this node is to generate the SRWNN output, using the following equation:

$$y = \sum_{j=1}^{N_w} c_j\psi_j(x) + \sum_{i=1}^{N_i} a_i\, x_i + b \quad \ldots (8)$$

where, as can be observed from Fig. 2, Nw and Ni denote the number of nodes in the wavelon layer and the input layer, respectively, c_j is the weight connecting wavelon j and the output node, a_i is the weight connecting node i in the input layer and the output node, and b represents a bias connection to the output node. Since the SRWNN structure described above is utilized as a controller in this work, the SRWNN output in Equation (8), y, represents the control action applied to the controlled system at a given sample of time.

3.1. Initialization of the SRWNN Parameters

In conventional ANN, the network parameters are randomly initialized to small values and the whole network functions as a black box approximator. This initialization process may lead to getting stuck at local minima and decrease the network convergence rate [25]. However, unlike ANNs, the WNN structure consists of dilation and translation factors which affect the shape of the wavelet. More precisely, a wavelet is a waveform defined by suitable dilation and translation parameters which determine the waveform effectiveness during only a limited duration. Therefore, a random initialization for dilation and translation parameters may generate ineffective wavelons which negatively affect the network convergence rate. For example, a very small value for the dilation factor may result in a wavelet with too local property that makes it inappropriate for the training data under consideration. On the other hand, if the translation factor is not initialized properly, the wavelet may materialize outside the domain of the training samples. Nevertheless, despite this influence for dilation and translation factors, several researchers did not consider a specific initialization process for these parameters [11, 13, 16, 21, 23].

In this work, in order to handle this issue, dilation and translation factors are initialized according to the available information from the dataset according to the following procedure [22, 25].

Let the variables a and b be the minimum and the maximum values that the input variables can take, respectively. Then, the translation and the dilation factors are initialized as follows:

$$t_j = \frac{1}{2}(a + b) \quad \ldots (9)$$

and

$$d_j = 0.2(b - a) \quad \ldots (10)$$

where t_j and d_j are translation and dilation factors of wavelon j in the wavelet layer and j = 1, 2, …, Nw. As stated before, this initialization process takes into account the input domain defined by the training samples, and hence, it ensures the coverage of the whole input domain by the initial wavelets. On the other hand, initial value selection for the remaining SRWNN parameters is less crucial; therefore, they are initialized to small random values. As can be noticed, the above initialization procedure is very simple, yet it has considerably improved the approximation capability of the SRWNN.

3.2. Training of the SRWNN

In the framework of MRAC, training of the SRWNN controller dictates finding the optimal values for the SRWNN parameters by minimizing the error between the outputs of the reference model and the controlled system. As discussed in Section 3, the SRWNN consists of different adjustable parameters, which can be summarized by the following set

$$S = [c_j \; d_j \; t_j \; v_{ji} \; a_i \; b \; \theta_j], \quad \ldots (11)$$

Where the adjustable parameters in the above equation were defined in Section 3. These parameters are optimized by the MMAIS algorithm in this work, as discussed in the following section.

4. Artificial Immune System

The underlying principle behind artificial immune system (AIS) is based on the information processing capability of the biological immune system which is a parallel, highly evolved, and distributed adaptive system. As a result, the AIS algorithm possesses powerful exploration and exploitation operators. In particular, compared to other evolutionary techniques, the AIS algorithm has more efficient mutation operators, and moreover, it can achieve better diversity of populations which leads to faster convergence rates [19, 20]. In order to understand the basic operators of the AIS algorithm and relate them to their biological counterparts, the main functions of the biological immune system are discussed below.

4.1. Biological Immune System

The biological immune system is a powerful defense network whose responsibility is to detect the incursion of foreign antigens and destroy them to protect the human body from these disease-causing substances. For this purpose, the biological immune system employs an enormous variety of antibodies, which are produced by the B cells, to counteract the effect of these harmful antigens [26]. More specifically, each antibody type has the ability to identify a specific antigen type using a correlation measure between the antibody and the antigen. This correlation measure is called the affinity value [27]. Based on this affinity value, once an external antigen is recognized, the B cells with high affinity to that antigen will explosively generate antibodies (cloned cells) to fight the antigen in a process known as the cloning operator. Moreover, these cloned cells undergo certain somatic hypermutation to make sure that the cloned cells will differ from their parent cell to adapt the immune system against other possible variations of the stimulating antigen [20]. To globally enhance the population of cells, the cells that are seldom stimulated by the attacking antigen are removed and replaced by other recruited cells. Then, after eliminating the antigen, mature B cells are repressed gradually to certain regular level except some cells which become memory cells. These memory B cells facilitate the immune system response when the same or similar antigen attacks again, since all the information on the attacking antigen are already stored in the memory cells [26, 28, 29].

4.2. Micro-AIS

Utilizing certain concepts from the biological immune system described above, several variations of the AIS algorithm have been effectively employed to solve various optimization problems [27, 30, 31]. However, the cloning operator in the traditional AIS algorithm has a drawback represented by the increase in the population size which entails a long execution time and substantial memory exploitation. To alleviate this problem, the authors in [32] suggested another version of the AIS algorithm, which was called the Micro-AIS algorithm, based on utilizing a reduced population size. The assumption behind the Micro-AIS was that decreasing the antibody number in a population results in a decrease in the number of cost function evaluations, which consequently increases the convergence speed and reduces the memory usage. However, in the original Micro-AIS algorithm, the mutation operator did not offer the necessary variable variations for solving a

particular problem. Thus, to enhance the searching capability of the original method, a MMAIS algorithm was presented in [21] by improving the mutation operator of the original method. Due to its effectiveness in several applications [21, 33, 34], the MMAIS algorithm is exploited in this work to optimize the parameters of the proposed SRWNN controller based on the MRAC strategy, as will be elucidated in the following section.

4.3. Procedure of Applying the MMAIS Algorithm to Train the SRWNN Controller

The following procedure illustrates the application of the MMAIS algorithm to train the SRWNN controller in the framework of MRAC strategy.

Step 1: Initialize the maximum number of iterations and the probability of mutation, P_m.

Step 2: Produce a starting (initial) population. This population consists of five antibodies which are randomly generated within preselected ranges. Each of these antibodies represents a single SRWNN controller which is defined by the adjustable parameters given in Equation (11). These five antibodies enter a nominal convergence loop, with 10 iterations, as described below.

Step 3: Set the counter of the nominal convergence loop to 1, and do the following:

Step 3-1: For each antibody in the current population, compute the objective function using the performance index defined in Equation (3). Next, for each antibody, find the affinity value according to the following equation:

$$ffinity = \frac{1}{Objective\ function + \varepsilon} \quad \dots (12)$$

Where ε is a small constant which is used to evade the division by zero.

Step 3-2: Sort descending the antibodies based on their affinity values. As a result, the antibody with the largest affinity value, which is called BestAb, becomes the first one.

Step 3-3: Generate a specific number of clones from the current five antibodies utilizing the following expression:

$$N_c = \sum_{i=1}^{n} \big(n - (i-1)\big) \quad \dots (13)$$

Where $N_c$the number of clones to be produced, n isis the number of antibodies in the current population, and I is the antibody index. Thus, by using Equation (13), a five-antibody population produces a 15-clone population. In particular, BestAb produces five clones; the second antibody produces four clones; and so on until the fifth antibody is encountered which produces one clone.

Step 3-4: Perform the maturation of clones using the mutation operator. First, determine the mutation probability of each group of clones generated from a given antibody. This probability is computed according to the antibody affinity value and then it is uniformly reduced with iterations. Therefore, the clones related to BestAb are mutated less than other groups of clones. More specifically, the mutation probability is determined according to the following equation:

$$pro_{mutation(i)} = \frac{Aff(i)}{\sum_{i=1}^{n} Aff(i)} \quad \dots (14)$$

Wherei is the antibody index and n is the number of antibodies in the old population. Moreover, to reduce the mutation probability within the nominal convergence loop, the following expression is utilized:

$$if\ p_m \geq \frac{prob\_mutation(i)}{iteration} \quad \dots (15)$$

$$, then\ apply\ the\ mutation\ operator$$

Where iteration in the above equation is the current nominal convergence loop iteration. Next, perform the mutation operator using the following expression:

$$\frac{(rand.rang)}{(iteration.group\_N_c)} \quad \dots (16)$$

where x' and x are the mutated decision variable and the decision variable to be mutated, respectively, rand is a random number generated within the range [0, 1], iteration is the current nominal convergence loop iteration, and group_N_c represents the number of clones for each antibody group. Regarding the five BestAB clones, the variable range in Equation (16) represents a random number generated between lower and upper bounds of the decision variables. For the remaining clones, other than the first five clones, range is the equivalent decision variable from BestAb and the mutation operator is performed using Equation (16).

Step 3-5: For each clone in the new population, calculate the objective function and the affinity value using Equations (3) and (12), respectively.

Step 3-6: Sort the 15 clones of the new population based on their affinity values in a descending order.

Step 3-7: From the sorted 15 clones, constitute a new population of five clones. In particular, as an elitism strategy, the first two clones are copied

into the new population. The other three clones are chosen randomly from the population of the 15 clones.

Step 3-8: If the counter of the nominal convergence loop has reached to its maximum limit, go to Step 4. If not, set counter = conter + 1 and go back to Step 3-1.

Step 4: After completing the nominal convergence loop, a new population of five antibodies is constituted by selecting the best two antibodies along with three other randomly generated ones. This new population enters the nominal convergence loop, specifically, Step 3 above. The whole algorithm, including Steps 3 and 4, continues until the stopping criterion is satisfied. Particularly, the stopping criterion in the present work is defined by reaching the maximum iteration number.

As an off-line design approach, the objective of the above training procedure is to optimize the SRWNN parameters defined in Equation (11). However, to realize adaptive control, the best controller found by the off-line design approach is further tuned by the MMAIS algorithm using the on-line training procedure described below.

4.4. On-line Training Procedure

In the on-line design stage, the MMAIS algorithm starts with an initial population of five antibodies which include the best controller found from the off-line design stage along with other related controllers. As the on-line adapted parameters, only the weights between the wavelon and the output layers are allowed to be adjusted by the optimization algorithm. At each time step, the MMAIS algorithm selects the best controller, from a set of several candidate controllers, to control the plant. The general idea of the proposed on-line design approach was adopted from [35]. The following procedure illustrates the on-line training steps used in this work.

$$\text{affinity} = \frac{1}{|J|} \quad \ldots (18)$$

Step 1: As the starting population in the on-line design stage, constitute an initial population of five antibodies in which the first two antibodies represent the exact copies of the off-line designed controller. The remaining three antibodies are initialized by randomly generating the weights between the wavelon and the output layers. In this way, the best controller obtained from the off-line design stage is effectively utilized to provide useful information for the adaptive design stage.

Step 2: Set the simulation time, k, to zero.

Step 3: Collect the current reference input, r(k), the current system output, y(k), and the current reference model output, ym(k).

Step 4: Calculate the next reference model output, ym(k+1), using the desired reference model equation.

Step 5: For each antibody in the current population, which consists of five antibodies, calculate the objective function using the following performance index:

$$J = y(k+1) - y_m(k+1) \quad \ldots (17)$$

Where y(k+1) is the system output at sample (k+1). After that, the affinity value of each antibody is found according to the following equation:

$$ffinity = \frac{1}{|J|} \quad \ldots (18)$$

Step 6: The antibody with the largest affinity value is chosen as the controller for the current sampling time. However, due to the random operators in the MMAIS algorithm, the selected controller might not be the best solution for the current sampling time. To alleviate this difficulty, the error produced by the on-line designed controller is compared with that of the off-line designed one at a given sampling time. Based on this error, which is defined by Equation (17), if the on-line designed controller achieves less error value, it is used to control the plant at the next sampling time. Otherwise, the off-line designed controller is selected as the controller for the next sampling time. From several simulation tests, this strategy has resulted in the best on-line controller performance as will be seen from the simulation results of the next section.

Step 7: Produce the next population of the MMAIS algorithm using the procedure described in Section 4.3, in particular Steps 3-1 to 3-8. However, instead of Equations (3) and (12), Equations (17) and (18) are used to evaluate the performance of each antibody. After completing the nominal convergence loop, a new population of five antibodies is generated.

Step 8: If the simulation time, k, has reached to its maximum value, the algorithm is terminated. Otherwise, set k = k +1 and go back to Step 3.

5. Simulation Results

This section is dedicated to assess the performance of the proposed SRWNN-based MRAC in terms of control accuracy, robustness ability, and generalization to various input signals. In addition, a comparative study with other

controllers within the framework of MRAC is conducted in this section. Utilizing the MMAIS algorithm, the initial learning and the on-line design procedure discussed in Sections 4.3 and 4.4, respectively, were employed to control all the considered plants. As the main parameters in the MMAIS algorithm, the mutation probability and the maximum number of iterations were set to 0.3 and 500, respectively. Moreover, for all the controlled plants, only six wavelons (with self-feedback connections) were used to constitute the wavelon layer in the SRWNN structure. These settings for the optimization algorithm and the SRWNN structure were adequate to attain the required control performance. As mentioned before, the main contribution of this work is represented by the performance improvement achieved by the SRWNN compared to the original WNN structure, as will be seen in Section 5.5. Furthermore, unlike the training method of the original WNN controller, an on-line training procedure is adopted in this work to further enhance the control performance and to realize an effective adaptive control strategy.

5.1. Control Performance Tests

To show the applicability of the proposed SRWNN-based MRAC to control various dynamical systems, three different nonlinear systems were adopted. These systems include a nonlinear non-minimum phase system, a water bath temperature control system, and a nonlinear minimum phase system.

Plant 1: This plant represents a nonlinear non-minimum phase system which is described by the following discrete-time equation [9]:

$$(k+1) = \frac{y(k)k(k-1)}{1+y^2(k)+y^2(k-1)} + u(k) + 1.5(k-1) \quad \ldots (19)$$

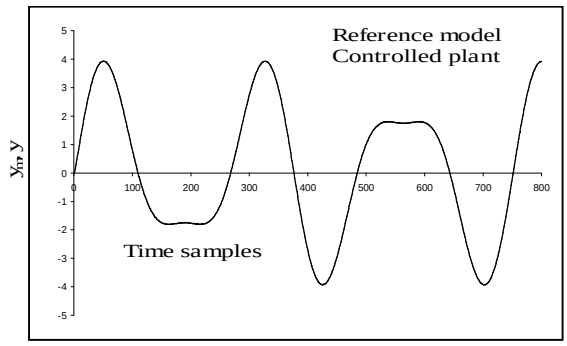
As the reference model for the above system, the following equation is used:

$$y_m(k+1) = 0.6y_m(k) + r(k) \quad \ldots (20)$$

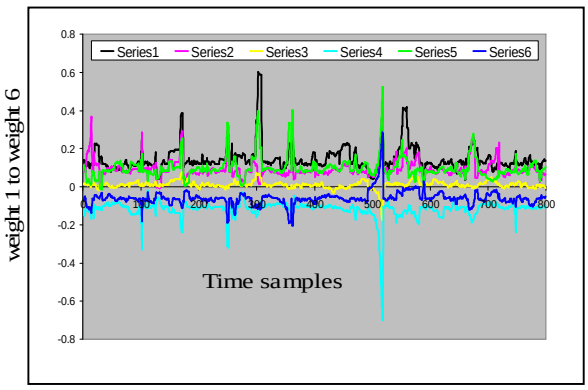
The SRWNN-based MRAC, shown in Fig. 1, is required to track the following reference signal:

$$(k) = 0.5\sin\left(\frac{2\pi k}{150}\right) + 1.2\sin\left(\frac{2\pi k}{250}\right) \quad \ldots (21)$$

Figure 3 (a) illustrates the result of controlling Plant 1, while Fig. 3 (b) shows the on-line adaptation made to the six weights connecting the wavelon and the output layers in the SRWNN structure.



a



b

Fig. 3. Plant 1 (a) outputs of the reference model and the controlled plant (b) on-line adaptation of the six weights in the SRWNN structure.

From Fig. 3 (a), it is obvious that the proposed control approach has achieved an accurate control performance by following the desired reference model output. On the other hand, Fig. 3 (b) demonstrates the on-line adaptation ability of the MMAIS algorithm in finding the optimal value for the six weights at each sampling time. Furthermore, it is worth mentioning that the initial learning for the SRWNN controller has resulted in a value of 0.1 for the performance index of Equation (3) after 500 iterations, while the on-line training procedure has further reduced the above value to 0.015, which indicates the effectiveness of the on-line training procedure.

Plant 2: As the second nonlinear controlled plant, the water bath temperature control system is considered. This system is governed by the following discrete-time equation [36, 37]:

$$y(k+1) = a(T_s)\mathrm{y(k)} + \frac{b(T_s)}{1+e^{0.5y(k)-y}}u(k) + [1-a(T_s)]Y_0 \quad \ldots (21)$$

where y(k) is the output temperature of the system in ^{0}C , u(k) is the system control input, which is limited by the following bounds$0 \leq$ u(k) ≤ 5 V, Y0 is the room temperature in ^{0}C

$a(T_s) = e^{*aT}$, and $b(T_s) = \frac{\beta}{\alpha}(1 - e^{-\alpha T})$. As for the parameters of this system, the following settings were used: $a = 1.00151 \cdot 10^{-4}, b = 8.67973 \cdot 10^{-3}, g = 40, Y_0 = 25^{\circ}C$ and $T_s = 30$ s. These values have been adopted from a real water bath system [37]. The desired reference signal, which represents the desired water temperature in $^{\circ}C$, is given by the following signal:

$$r(k) = \begin{cases} 34, & k \le 30, \\ 34 + 0.5(k - 30), & 30 < k \le 50, \\ 44 + 0.8(k - 50), & 50 < k \le 70, \\ 60 + 0.5(k - 70), & 70 < k \le 90, \\ 70, & 90 < k \le 120. \end{cases} \quad \ldots (23)$$

The desired system output is specified by the following first-order reference model equation [38]:

$$y_m(k + 1) = 0.6y_m(k) + 0.4r(k) \quad \ldots (24)$$

The actual system output, the reference model output, and the resulting control signal are exhibited in Fig. 4 (a), while the on-line adaptation achieved on the six network weights is depicted in Fig. 4 (b).

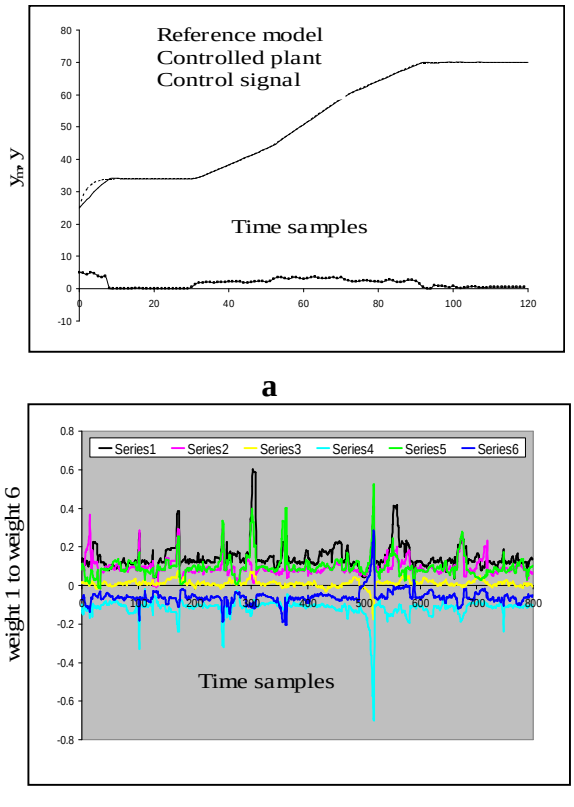


Fig. 4. Plant 2 (a) outputs of the reference model and the controlled plant (b) on-line.

As it is evident from Fig. 4 (a), the actual system output has accurately followed the reference model output. Figure 4 (b) clearly shows the on-line adaptation ability of the MMAIS algorithm at each sampling time. The initial learning for the controller after 500 iterations has achieved a performance index of 53.965, while the on-line training has further reduced this value to 21.393, signifying the efficiency of the on-line algorithm. Bearing in mind the control signal limitation within a specific rang [0,5], it can be concluded that the proposed controller has achieved the desired performance even with the existence of such constraints.

Plant 3: This is a nonlinear plant expressed by the following difference equation [24]:

$$y(k + 1) = \frac{y(k) + y(k - 1)}{1 + y^2(k) + 2y^2(k - 1)} + u(k) \quad \ldots (25)$$

The desired system behavior is given by the following reference model equation:

$$y_m(k + 1) = 0.5y_m(k) + 0.3y_m(k - 1) + r(k) \quad \ldots (26)$$

While the reference signal is as follows:

$$r(k) = 0.5[\sin(10\pi k) + \sin(25\pi k + 0.5)] \quad \ldots (27)$$

Figure 5 (a) shows Plant 3 output together with the reference model output, and Fig. 5 (b) illustrates the on-line adaptation of the six network weights.

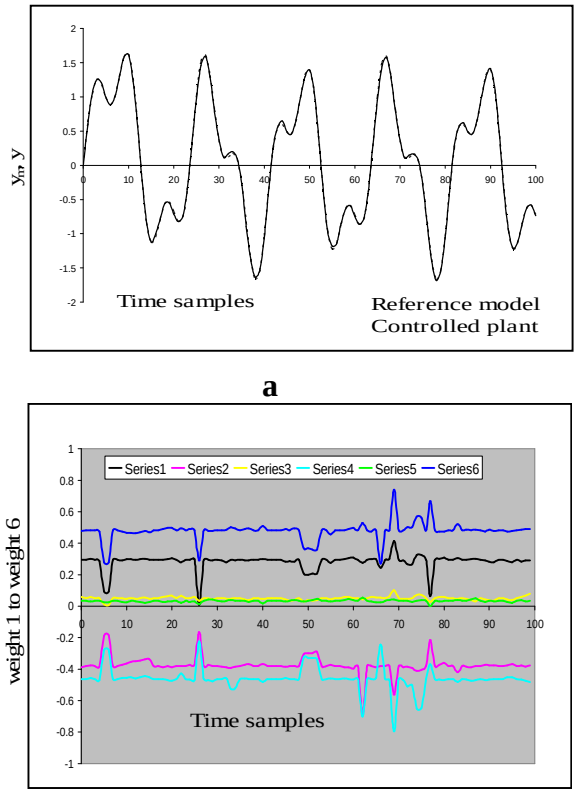
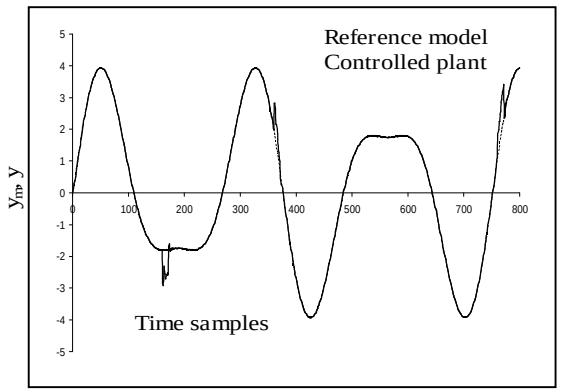


Fig. 5. Plant 3 (a) outputs of the reference model and the controlled plant (b) on-line adaptation of the six weights in the SRWNN structure

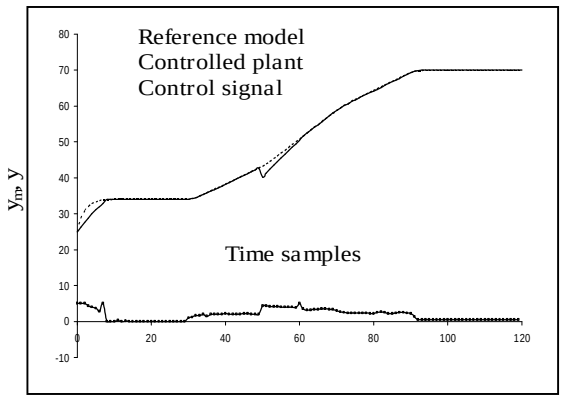
From Fig. 5 (a), it can be seen that the SRWNN-based MRAC has done well in tracking the desired reference model output. The ability of the optimization method in adapting the network weights can be observed in Fig. 5 (b). As with Plants 1 and 2, the on-line training algorithm has achieved a smaller performance index value of 0.027, compared to the initial learning value of 0.057.

5.2. Robustness Tests

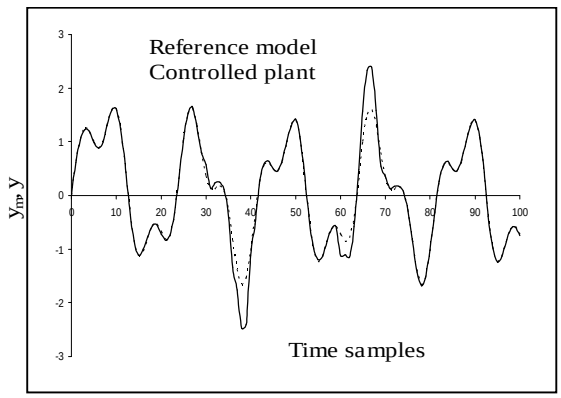
These tests aim at investigating the robustness ability of the proposed SRWNN-based MRAC by conducting a disturbance rejection test on each plant considered in the previous section. For Plants 1 and 3, a bounded disturbance of 50% from the controlled system output was applied for 10 samples at different periods of the simulation time. In particular, the disturbance was applied at the intervals $161 \pounds k \pounds 171$, $361 \pounds k \pounds 371$, and $761 \pounds k \pounds 771$, for Plant 1, and at the intervals $30 \pounds k \pounds 39$ and $60 \pounds k \pounds 69$, for Plant 3. For the water bath temperature control system, a disturbance of $-3\,^{o}C$ was applied at the 50th sample. It is interesting to notice that all of these disturbances were applied during only the controller testing phase of each plant. This means that the SRWNN controller was not trained to handle these disturbances. Nonetheless, the SRWNN controller managed to cope with the unexpected effects of these disturbances by maintaining the desired output response for all the plants. Figures 6 (a), (b), and (c) show the results of these tests for Plants 1, 2, and 3, respectively.



a



b



c

Fig. 6. Outputs of the reference model and the controlled plant when a disturbance is applied at the output of (a) Plant 1 (b) Plant 2 (c) Plant 3 objective functions against generations.

It is noteworthy that the large disturbance amplitude, specifically the 50% of system outputs for Plants 1 and 3, indicates that the SRWNN controller possesses the ability to counteract large disturbances even when such disturbances are not encountered during the controller training phase.

5.3 Parameter Variation Test

This test is conducted to evaluate the controller ability in handling inherent changes in the system parameters. To perform this test, certain variations are made on a parameter in the water bath temperature control system at different time samples. More specifically, an increase of 10% in the original value of $b(T_s)$ in Equation (22) was considered at sample numbers 30, 50, and 70. As was done in the previous section for disturbance rejection tests, the parameter variations were made only during the controller testing phase, which means that the controller was not trained to face these dynamic changes in the system. Figure 7 illustrates the outputs of both the system and the

reference model along with the control action of the controller.

$$r(k) = 0.5[\sin(15\pi k) + \sin(35\pi k + 0.5)] \quad \ldots (29)$$

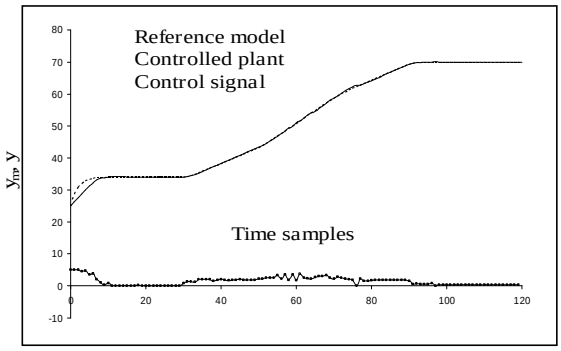


Fig. 7. Outputs of the reference model and the controlled plant when a parameter variation test is applied on Plant 2 objective functions against generations.

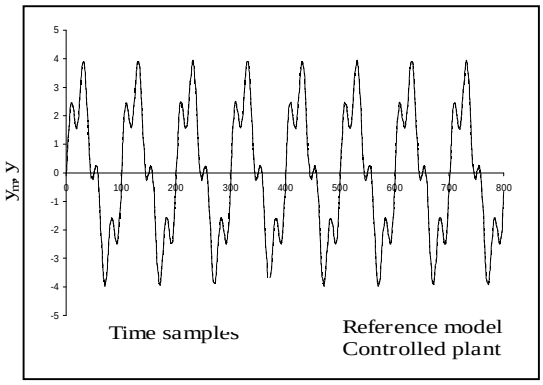
As can be clearly seen from Fig. 7, the SRWNN controller has done well in tracking the desired reference model output even with the existence of an unexpected time-variant parameter in the system model. The control signal in Fig. 7 indicates the adaptation made on the controller behavior to deal with such nonlinear time-variant system.

5.4 Generalization Tests

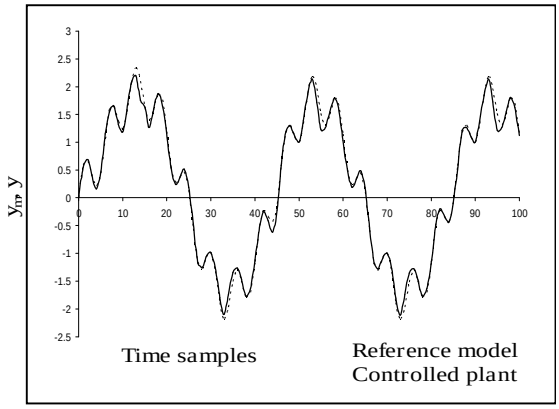
The objective of these tests is to demonstrate the ability of the SRWNN controller in handling reference signals different from those used in the controller training stage. This controller feature is known as the generalization ability. The tests were conducted on Plants 1 and 3 by using the same training signals of Equations (21) and (27) for Plants 1 and 3, respectively. While, for the testing stage, the following signals were used:

$$r(k) = 0.5sin\left(\frac{2\pi k}{25}\right) + 1.2sin\left(\frac{2\pi k}{100}\right) \quad \ldots (28)$$

Equations (28) and (29) represent the testing signals for Plants 1 and 3, respectively. The results of these tests are shown in Fig. 8, where it is clear that the SRWNN controller has appropriately generalized its learning to handle input signals which were not encountered throughout the controller training phase for Plant 1, Fig. 8 (a), and for Plant 3, Fig. 8 (b).



a



b

Fig. 8. Outputs of the reference model and the controlled plant resulted from the generalization tests applied on (a) Plant 1 (b) Plant 3 objective functions against generations.

5.5. A Comparative Study with other Related Controllers

In this section, the performance of the proposed SRWNN controller is compared against those of other related controllers in terms of control accuracy and processing time. Specifically, the controllers under consideration include the original WNN controller, the ANN controller, and the modified recurrent network (MRN) controller. These controllers are used within the same MRAC structure shown in Fig. 1. For a fair comparison, the same MMAIS algorithm was employed to train each of the above controllers using the off-line procedure described in Section 4.3. The structure of the original WNN controller [21] is similar to that of the SRWNN discussed in Section 3, with the exception of the wavelons' self-feedback connections and the initialization phase in the SRWNN structure. On the other hand, the ANN structure consists of an input layer, a hidden layer

with hyperbolic tangent as the neurons' activation functions, and an output layer. Finally, the structure of the MRN, which was proposed in [39], has an input layer, a hidden layer, a context layer, and an output layer. The context layer has the same number of nodes as the hidden layer. Each node in the context layer receives two inputs, the first of which represents a self-feedback connection with an adjustable weight, b, while the second input is the previous output of the corresponding hidden node multiplied by an adjustable weight, x. As the hidden nodes' activation functions, the hyperbolic tangent functions were used. In fact, the MRN is an improved version of the modified Elman network [40]. It is worth to highlight that only 6 neurons were used in each hidden layer of the networks mentioned above. Plants 1 and 3 used before are considered in this comparative study, along with the following time-delay nonlinear plant [21]:

$$(k+1) = \frac{1.2(1-e^{-0.8})y(k)}{1+y^2(k)} + u(k-1) \quad \dots (30)$$

The MMAIS is classified as a stochastic algorithm, since it employs several random operators. As a result, the performance of a final optimized controller might vary for different simulation runs. Hence, to achieve a reliable comparative study, 10 runs were conducted for each plant and the average result was taken. Table 1 summarizes the results of this comparative study. As can be concluded from Table 1, the proposed SRWNN-based MRAC has attained the best results with respect to the other controllers. In terms of control accuracy, the SRWNN controller has achieved the least values for the performance indices for all the plants. Regarding processing time, the SRWNN controller took the shortest times except for Plants 1 and 2 controlled by the original WNN controller. However, this slight increase in processing time for the SRWNN, which was the result of including the self-feedback weights, can be ignored in light of the superior results of the proposed controller compared to the original WNN controller.

5.6. A Comparative Study on the Optimization Methods

As stated earlier, it has been shown that the AIS algorithm is superior to the GA with regard to maintaining good population diversity due to the efficient mutation operator employed by the AIS algorithm [19, 20]. In this context, it is essential to compare the performances of both the MMAIS and the GA in training the proposed controller, and this section is dedicated for this purpose. As the controlled systems, the same plants considered in the previous section are used in this comparative study. Similar to the comparison procedure followed in Section 5.5, 10 runs were conducted for each plant and the average result was considered in order to attain a reliable comparative study. Table 2 exhibits the comparison results. Obviously, Table 2 signifies the advantage of the MMAIS algorithm over the GA. In particular, the MMAIS algorithm has accomplished the least values for the performance indices and took the shortest processing times for all the plants compared to the GA.

Table 1,
Comparison results of the WNN, the ANN, the MRN, and the proposed SRWNN.

MRAC type	Criterion	Controlled plant		
		Plant 1	Plant 2	Plant 3
WNN-based MRAC	Average Performance Index	0.636	1.261	0.386
	Average Time (sec.)	57.775	57.709	9.603
ANN-based MRAC	Average Performance Index	1.625	2.258	0.085
	Average Time (sec.)	71.341	75.574	12.085
MRN-based MRAC	Average Performance Index	0.843	2.758	0.155
	Average Time (sec.)	103.482	110.048	16.740
Proposed SRWNN-based MRAC	Average Performance Index	0.052	0.913	0.039
	Average Time (sec.)	58.928	62.028	9.327

Table 2,
Results of comparing the performances of the GA and the MMAIS in training the proposed SRWNN-based MRAC

Controlled Plant	Genetic algorithm		MMAIS algorithm	
	Average performance index	Average time (sec.)	Average performance index	Average time (sec.)
Plant 1	0.338	116.787	0.052	58.928
Plant 2	1.832	112.918	0.913	62.028
Plant 3	0.15	60.614	0.039	9.327

6. Conclusions

In this paper, a SRWNN-based MRAC scheme is proposed to control nonlinear dynamical systems. As an improved version of a previously reported WNN structure, a SRWNN structure is put forward by adopting a specific initialization phase and utilizing self-feedback weights in the wavelon layer. In addition, an on-line training procedure is followed to further enhance the control accuracy of the SRWNN controller. To train the above controller, the newly developed MMAIS algorithm is employed to optimize the SRWNN parameters. Several nonlinear dynamical systems are used to show the effectiveness of the proposed control approach via an extensive evaluation tests, including control performance tests, robustness tests, parameter variation test, and generalization tests. All these tests have indicated the effectiveness of the SRWNN-based MRAC. Furthermore, a comparative study with other related controllers has shown the superiority of the proposed controller. Finally, as compared to the GA, the MMAIS algorithm has achieved better results with regard to control accuracy and processing time.

7. References

[1] S. Pankaj, J.S. Kumar, and R.K. Nema, "Comparative analysis of MIT rule and Lyapunov rule in model reference adaptive control scheme," Innovative Systems Design and Engineering, Vol. 2, No. 4, pp. 154-162, 2011.

[2] N. Aguila-Camacho, and M. A. Duarte-Mermoud, "Fractional adaptive control for an automatic voltage regulator," ISA Transactions, Vol. 52, No. 6, pp. 807–815, 2013.

[3] K. A. Mohideen, G. Saravanakumar, K. Valarmathi, D. Devaraj, T. K. Radhakrishnan, "Real-coded genetic algorithm for system identification and tuning of a modified model reference adaptive controller for a hybrid tank system," Applied Mathematical Modelling, Vol. 37, No. 6, pp. 3829-3847, 2013.

[4] R. Al-ghamri, "Design of an adaptive controller for magnetic levitation system based bacteria foraging optimization algorithm," M.Sc. Thesis, Electrical Engineering Department, The Islamic University of Gaza 2014.

[5] S. E. Oltean, M. Dulau, and A. V. Duka, "Model reference adaptive control design for slow processes," A case study on level process control. Procedia Technology, Vol. 22, pp. 629-636, 2016.

[6] A. C. Huang, "Model reference adaptive control of a class of non-autonomous systems using serial input neuron," Neurocomputing, Vol. 6, pp. 413-423, 2003.

[7] I. Douratsos, and J. B. Gomm, "Neural network based model reference adaptive control for processes with time delay," International Journal of Information and Systems Sciences, Vol. 3, No. 1, pp. 161-179, 2007.

[8] R. Prakash, and R. Anita, "Neuro-PI controller based model reference adaptive control for nonlinear Systems," International Journal of Engineering, Science and Technology, Vol. 3, No. 6, pp. 44-60, 2011.

[9] S. Li, J. Li, J. Qiu, H. Ji, and K. Zhu "Control design for arbitrary complex nonlinear discrete-time systems based on direct NNMRAC strategy," J Process Control, Vol. 21, No. 1, pp. 103-110, 2011.

[10] A. Errachdi, andM. Benrejeb, "Model reference adaptive control based-on neural networks for nonlinear time-varying system," Proceedings of the International Conference on Systems, Control and Informatics, Italy, pp. 73-77, 2013.

[11] C. -M. Lin, C. –S. Hsueh, and C. –H. Chen, "Robust adaptive backstepping control for a class of nonlinear systems using recurrent wavelet neural network,"

Neurocomputing, Vol. 142, pp. 372-382, 2014.

[12] F. M. Mohammed, S. A. Aziez, and H. N. Abdul-Rihda, "Comparison between wavelet and radial basis function neural networks for GPS prediction," Engineering and Technology Journal, Vol. 33, No. 3, pp. 560-572, 2015.

[13] Y. Zhao, X. Du, G. Xia, L. Wu, "A novel algorithm for wavelet neural networks with application to enhanced PID controller design," Neurocomputing, Vol. 158, pp. 257-267, 2015.

[14] N. Wang, andH. Adeli, "Self-constructing wavelet neural network algorithm for nonlinear control of large structures," Engineering Applications of Artificial Intelligence, Vol. 41, pp. 249-258, 2015.

[15] R. -J. Wai, andH. H.Chang, "Backstepping wavelet neural network control for indirect field-oriented induction motor drive," IEEE T Neural Networ, Vol. 15, No. 2, pp. 367-382, 2004.

[16] S. J. Yoo, J. B. Park, and Y. H. Choi, "Indirect adaptive control of nonlinear dynamic systems using self recurrent wavelet neural networks via adaptive learning rates," Inform Sciences, Vol. 177, pp. 3074-3098, 2007.

[17] M. Z. Othman, and A. A. Al-Qassar, "Study of principle component analysis and learning vector quantization genetic neural networks," Engineering and Technology Journal, Vol. 27, No. 2, pp. 321-331, 2009.

[18] M. H. Miry, "An efficient approach combining genetic algorithm and neural networks for Eigen value grads method (EGM) in wireless mobile communications," Engineering and Technology Journal, Vol. 29, No. 13, pp. 2590-2600, 2011.

[19] S. Wu, H. –D. Wan, S.K. Shukla, and B. Li "Chaos-based improved immune algorithm (CBIIA) for resource-constrained project scheduling problems," Expert Syst. Appl., Vol. 38, No. 4, pp. 3387-3395, 2011.

[20] R. G. Kuo, W. L.Tseng, F. C. Tien, and T. W. Liao "Application of an artificial immune system-based fuzzy neural network to a RFID-based positioning system," Comput Ind Eng, Vol. 63, No. 4, pp. 943-956, 2012.

[21] O. F. Lutfy, "Wavelet neural network model reference adaptive control trained by a modified artificial immune algorithm to control nonlinear systems," Arab. J. Sci. Eng., Vol. 39, pp. 4737-4751, 2014.

[22] Y. Oussar, and G. Dreyfus, ".: Initialization by selection for wavelet network training," Neurocomputing, Vol. 34, pp. 131-143, 2000.

[23] S. J. Yoo, J. B. Park, and Y. H. Choi, "Stable predictive control of chaotic systems using self-recurrent wavelet neural network," International Journal of Control, Automation, and Systems, Vol. 3, No. 1, pp. 43-55, 2005.

[24] F. M. Ham, and I. Kostanic, "Principles of Neurocomputing for Science and Engineering," McGraw Hill, New York, NY, 2001.

[25] A. K. Alexandridis, and A. D. Zapranis, "Wavelet neural networks: A practical guide," Neural Networks, Vol. 42 pp. 1-27, 2013.

[26] K. Leung, F. Cheong, and C. Cheong, "Generating compact classifier systems using a simple artificial immune system," IEEE T Syst Man Cy B, Vol. 37, No. 5, pp. 1344-1356, 2007.

[27] L. Montechiesi, M. Cocconcelli, and R. Rubini, "Artificial immune system via Euclidean distance minimization for anomaly detection in bearingsMechanical Systems and Signal Processing, Vol.76-77, pp. 380-393, 2016.

[28] W. Zhang, G. G.Yen, and Z. He, "Constrained optimization via artificial immune system," IEEE Transactions on Cybernetics, Vol. 44, No. 2, pp. 185-198, 2014.

[29] E. N. Dragoi, C. A.Horoba, I. Mamaliga, and S. Curteanu, "Grey and black-box modelling based on neural networks and artificial immune systems applied to solid dissolution by rotating disc method," Chem. Eng. Process.: Process Intensification, Vol. 82, No. 7, pp. 173-184, 2014.

[30] M. -H. Chen, C. –H. Teng, and P. -C. Chang, "Applying artificial immune systems to collaborative filtering for movie recommendation," Advanced Engineering Informatics, Vol. 29, No. 4, pp. 830-839, 2015.

[31] M. -H. Chen, P. –C Chang, and J. L. Wu, "A population-based incremental learning approach with artificial immune system for network intrusion detection," Engineering Applications of Artificial Intelligence, Vol. 51, pp. 171-181, 2016.

[32] J. C. Herrera-Lozada, H. Calvo, and H. Taud, "Polibits, Vol. 43, pp. 107-111, 2011.

[33] O. F. Lutfy, "Control of nonlinear systems utilizing a wavelet neural network controller

optimized by micro artificial immune systems," The Second Engineering Conference of Control, Computers, and Mechatronics, Baghdad-Iraq, pp. 113-119, 2014.

[34] O. F. Lutfy, and H. Selamat, "Wavelet neural network-based NARMA-L2 internal model control utilizing micro artificial immune techniques to control nonlinear systems," Arab. J. Sci. Eng., Vol. 40, No. 9, pp. 2813-2828, 2015.

[35] L. M. L.Porter, and K. M. Passino, "Genetic model reference adaptive control," Proceedings of the IEEE International Symposium on Intelligent Control, Columbus, USA, pp. 219-224, 1994.

[36] C. -F. Juang, and C.Lo, "Zero-order TSK-type fuzzy system learning using a two-phase swarm intelligence algorithm," Fuzzy Sets and Systems, Vol. 159, No. 21, pp. 2910-2926, 2008.

[37] C. -T. Lin, C. –H.Chen, and C. -J. Lin, "Nonlinear system control using functional-link-based neuro-fuzzy networks," In: Yu, W. (editor) Recent Advances in Intelligent Control Systems. Springer-Verlag London Limited, 2009.

[38] H. Husain, M. Khalid, and R. Yusof, "Direct model reference adaptive controller based-on neural-fuzzy techniques for nonlinear dynamical systems," American Journal of Applied Sciences, Vol. 5, No. 7, pp. 769-776, 2008.

[39] N. A. Shiltagh, "A training algorithm for partial recurrent neural networks and its applications for system identification and control," M.Sc. Thesis, Department of Control and Computer Engineering, University of Technology, Baghdad, Iraq, 2001.

[40] H. -W. Ge, F. Qian, Y. -C. Wellstead, W. –L. Du, and L. Wang "Identification and control of nonlinear systems by a dissimilation particle swarm optimization-based Elman neural network," Nonlinear Analysis: Real World Applications, Vol. 9, No. 4, pp. 1345-1360, 2008

نظام سيطرة متكيف ذو موديل مرجعي مبني على شبكة عصبية مويجية ذاتية التكرار باستخدام أنظمة المناعة الصناعية الدقيقة

عمر فاروق لطفي* **مريم حسن داود****

*،**قسم هندسه السيطرة والنظم / الجامعة التكنولوجية

*البريد الالكتروني :60157@uotechnology.edu.iq

**البريد الالكتروني :mariam.hassan93@yahoo.com

الخلاصـة

يقدم هذا البحث نظام سيطرة متكيفا ذا موديل مرجعي ذكي باستخدام شبكة عصبية مويجية ذاتية التكرار للسيطرة على الأنظمة اللاخطية. الشبكة المقترحة هي نسخة محسنة لشبكة عصبية مويجية منشورة سابقا. وبالتحديد، هذا التحسين تم انجازه بتبني تعديلين على هيكل الشبكة الاصلي. وهذان التعديلات يتضمنان أولا استخدام مرحلة محددة لتوليد الأوزان لتحسين الاقتراب نحو قيم الأوزان المثلى، وثانيا تضمين أوزان ذاتية الإشارة العائدة لمويجات الطبقة المويجية. فضلا عن ذلك، تم اقتراح طريقة تعليم انية لتحسين أداء نظام السيطرة المقترح. وبوصفها طريقة تعليم، تم استخدام نظام المناعة الصناعي الدقيق المعدل والذي طور حديثا لايجاد القيم المثلى لمعاملات الشبكة المستخدمة. وقد تم عرض كفائة الطريقة المستخدة بالسيطرة على عدة انظمة ديناميكية لاخطية. وقد تم اعتماد عدة اختبارات تقييم لكل نظام مسيطر عليه وهذه الاختبارات تتضمن اختبارات اداء السيطرة و اختبارات المتانة واختبارات التعميم. ومن هذه الاختبارات اظهر النظام المقترح كفائته من حيث دقة السيطرة و رفض المؤثرات الخارجية وقابلية التعميم. بالاضافة لهذه الإختبارات، تم اجراء دراسة مقارنة مع مسيطرات اخرى ذات صلة وبالتحديد الشبكة العصبية المويجية الأصلية و الشبكة العصبية الصناعية والشبكة التكرارية المعدلة. وقد اظهرت نتائج هذه الدراسة تفوق المسيطر المقترح على المسيطرات الأخرى.