# HYBRID GENETIC ALGORITHM AND PARTICLE SWARM OPTIMIZATION WAVELET NEURAL NETWORK FOR DIRECT CONTROLLER OF ROBOT MANIPULATOR [+]

**Osama Kareem M.** [*]


**Abstract:**

**An wavelet neural network collected some properties of neural network and also the properties of wavelet transform which used replaced of sigmoidal function.**
**In this paper used wavelet neural network (WNN) in control of the robot arm where used the first method by direct control which contain two types of neural network one of them wavelet neural and the other adoptive linear wavelet neural which is the best because need for less number of wavelet in the hidden layers.**
**Also training the weights of each of them by using Hybrid Genetic Algorithm and Particle Swarm Optimization (HGAPSO) and used also in the operation of training of the weight of network which controlled on robot arm instead of Gradient Decent (GD) its fast but may in local minimum . The properties of this algorithm are achieve best results where can get on least mean square error between the target signal and the output of the robot arm ,also can achieve best result with minimum number of learning cycles also can be used in complex systems which cannot derivative. .**
**Two types of network and learning method are tested by using simulation and control on Robot arm and tested these method under different load also compared the previous methods.**


## الخوارزميه الهجينه *GAPSO* والشبكات العصيبيه المويجيه للسيطره المباشره على ذراع الربوت

**أسامة كريم محمد***


**المستخلص:**

**إن الشبكات العصبية المويجيه تجمع بعض الخصائص المتوفرة في الشبكات العصبية بالاضافه إلى الخصائص التي تمتلكها الشبكات العصبية المويجيه والتي استخدمت في الطبقات الوسطى بدلا"من Sigmoidal Function**
**في هذا البحث استخدمت الشبكات العصبية المويجيه في السيطرة والتحكم بذراع الربوت ، حيث استخدمنا " السيطرة المباشرة " وفيها نوعين من الشبكات المويجيه احدهما العصبية المويجيه والأخرى الشبكات العصبية المويجيه ذات التكييف المحلي الخطي حيث إن الاخيره تكون أفضل لأنها تحتاج اقل عدد من المويجات في الطبقات الداخلية.**
**لقد تم تدريب أوزان كلا الشبكتين باستخدام خوارزمية التدريب :**
**Hybrid Genetic Algorithm And Particle Swarm Optimization ( HGAPSO )**
**وقد استخدمت في عمليات التدريب لأوزان شبكة المسيطر على ذراع الربوت بدلا من استخدام خوارزمية التدريب ( Gradient Decent (GD حيث إنها تكون سريعة ولكنها غالبا ما تسقط في المنخفض المحلي ،ومن خصائص هذه الخوارزمية إنها تحقق نتائج افضل حيث تم الحصول على اقل مربع خطا بين الاشاره المرغوبة وإخراج الربوت وكذلك تم الحصول على افضل النتائج بأقل عدد من دورات التدريب وكذلك إنها تستخدم مع الانظمه المعقدة والغير قابله للاشتقاق .**


---

[+] Received on 18/5/2010 , Accepted on 6/3/2011 .
[*] Assist Lecture / Technical Institute / Aumara .

إن كلا النوعين من الشبكات وطرق التدريب المذكورة أعلاه قد اختبرت من خلال المحاكاة والسيطرة علـــى ذراع الربوت ، وقد اختبر أداء جميع الطرق أعلاه تحت شروط أحمال مختفله وتم مقارنـة نتائج الطرق السابقة .

## Introduction:

Wavelet networks were first proposed by Zang, et al., [1], Wavelet neural network (WNN) combine some of the useful classification properties of neural networks with the localization and feature extraction properties of wavelets. WNN`s replace the global sigmoidal activation function units of classical feed-forward NN with wavelets. Wavelet networks have been applied to variety of applications, including nonlinear functional approximation and nonparametric estimation; system identification; adaptive control; detection and classification and nonlinear modeling, image registration, time-series forecasting and optimization [2]. In recent years, much research has been done on applications of wavelet neural networks, which combine the capability of artificial neural networks in learning from processes and the capability of wavelet decomposition, for identification and control of dynamic systems.

The design of robot controller requires a precise knowledge of all parameters of the robot. Therefore modeling and identification are important steps in the design of control system. Generally, identification process is conducted to obtain a parametric model with the same dynamic behavior of the real plant or process. In this paper, WNN is used to obtain model with the same performance as the real plant. In this paper, WNN is used, in which the connection weights between the hidden layer units and output units are replaced by a local linear model. The usually used learning algorithm for WNN is gradient descent method.

A gradient descent method is used for training the WNN controller. Simulation results for control robot manipulator problems show the effectiveness of the used method Due to the crudeness of the local approximation, a large number of basis function units have to be employed to approximate a given system. A shortcoming of the wavelet neural network is that for higher dimensional problems many hidden layer units are needed[3].

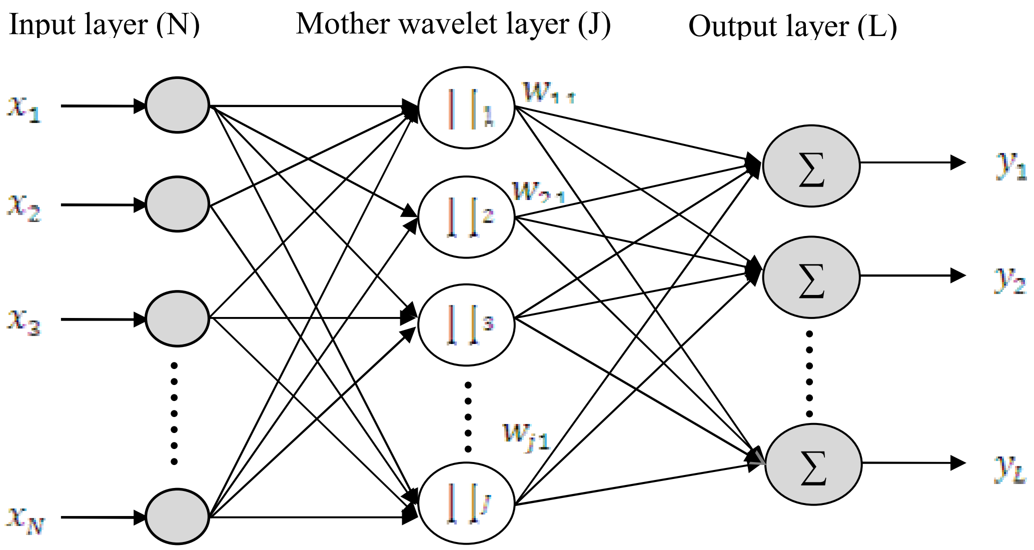
## Controller Model using Wavelet Neural Network (WNN):

Wavelet networks are a class of neural networks consisting of wavelets. It is a feed forward neural network based on wavelet transform and adaptive neural network and very wide used in identification and control. It is employing wavelet as activation functions recently have been researched as an alternative approach to the traditional neural network with sigmoid activation functions. According to the structure of Figure (1), the output of a wavelet neural network (WNN) is given by:

$$y_i = \sum_{j=1}^{J} w_{ji}\psi_j(x)$$

$$= \sum_{j=1}^{J} w_{ji}\psi_j\left(\frac{x-b_j}{a_j}\right) \qquad \ldots (1)$$

Where the activation functions of the hidden layer units are the wavelets $w_{ji}$, $\psi_j$




**Figure(1): General architecture WNN Network**

Is the straightforward weight connection the $jth$ hidden layer unit to the output layer unit $y_i$ and each activation function $\psi_j$ is represented in Eqn.(3). The numbers of the hidden layer units are $J$.

From Figure (1), the signal propagation and the basic function in each layer can be described as follows. For every node $n$ in the input layer, the net input and the net output can be described by the following equations:

$$net_n^1 = x_n^1, \quad y_n^1 = f_n^1(net_n^1) = net_n^1 \quad \ldots\ldots(2)$$

Where $n = 1,..,N$ is the number of inputs. The family of wavelets is usually constructed by translation and dilations performed on a single fixed function, called the mother wavelet. In each layer, each node performs a wavelet $\psi_j$ that is derived from its mother wavelet.

The first derivative of a Gaussian function is adopted as a mother wavelet in this study and expressed as in Eqn.(3)[4],

$$\psi(x) = -x exp\left(-\frac{x^2}{2}\right) \ldots\ldots(3)$$

$$net_{jn}^2 = \frac{x_n^2 - b_{jn}}{a_{jn}} \ldots\ldots(4)$$

Where $x_n^2 = net_n^1$

$$s_j^2 = \prod_{n=1}^{N} f_j^2(net_{jn}^2) = \prod_{n=1}^{N} \psi_j(net_{jn}^2) \ldots\ldots(5)$$

where $b_{jn}$ and $a_{jn}$ are the translation and dilation in the $jth$ term of the $nth$ input $x_n^2$ to the node of the mother wavelet layer, and $j$ is the number of wavelets with respect to each input nodes, $y_j^2$ represent the output for each activation function in mother wavelet layer. Each node $j$ in the wavelet layer is denoted by $\prod j$ where $\prod_i$ represent multiplies the inputs signals for $nth$ role inputs property wavelet multidimensional [5] and $j = 1,2,\ldots.J$ is the number of activation input. The output layer is computed as follows:

$$net_i^3 = \sum_{j=1}^{J} s_j^2 * w_{ji} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (6)$$

$$y_i^3 = f_i^3(net_i^3) = net_i^3 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (7)$$

See Figure (1) which used as controller. There are many these algorithms used to adjust parameters. We will collect some training algorithms to tune parameters of wavelet network identifier.

**Direct Adaptive Control of Robot Manipulator:**

It consists of two main parts, the important part is the controller because our goal is to satisfy stable system with minimum error between outputs of the robot and desired trajectories. . We must design controller that gives high performance to make the output of the robot track the desired trajectory signal. Wavelet neural network (WNN) will be used here as a controller.

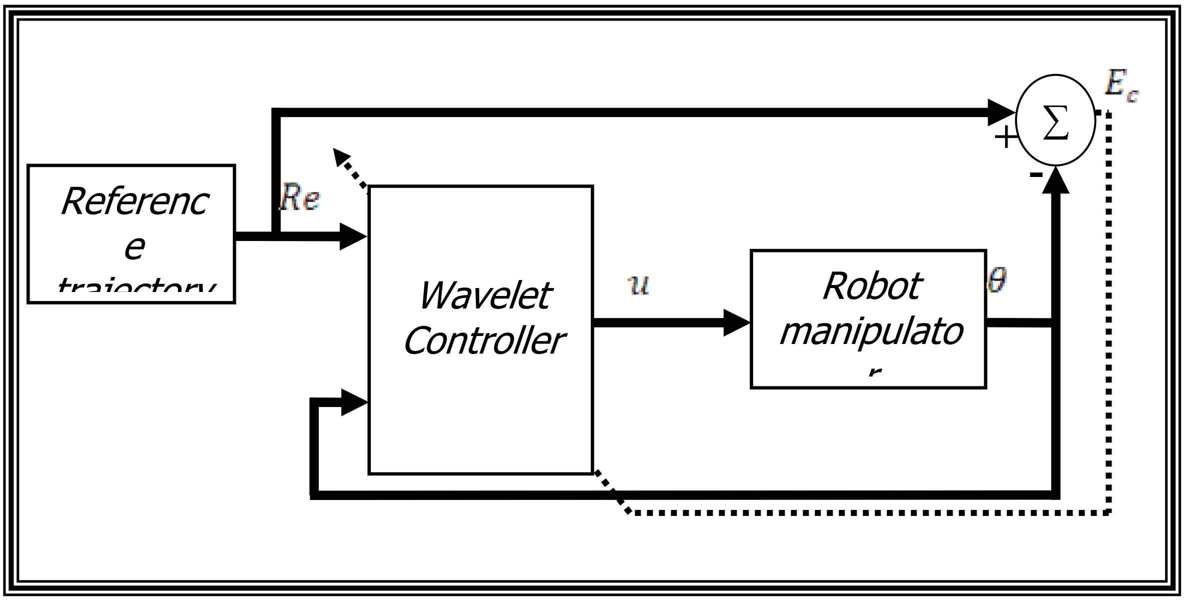



**Figure (2): Direct Control of Robot Manipulator**

**1- Robot model:**

**Consider** the 2-DOF robot manipulator, thus, the dynamic equations of the robot constitute a set of two nonlinear differential equations of the state variables

$$x = [\theta^T \quad \dot{\theta}^T]^T$$ .

The compact form of the robot is obtained in Eqn.(8)[6], i.e.

$$\tau = M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + G(\theta) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (8)$$

Where $\tau$ where $\tau_1$ and $\tau_2$ are the external torques delivered by the actuators at joints 1 and 2. And $\tau$ is an $n \times 1$ vector of joint actuator torques and $\theta, \dot{\theta}$ and $\ddot{\theta}$ are $n \times 1$ vectors of generalized joint position, velocities and acceleration respecti1vely. In Eqn.(8), $M(\theta)$ is $n \times n$ mass matrix that captures the configuration dependent inertial properties of the robot, $C(\theta, \dot{\theta})$ is accounts for Coriolis and centrifugal forces and $G(\theta)$ represent the vector of joint torques due to gravity, Where

$$M_{11}(\theta) = m_1 l_{c1}^2 + m_2 [l_1^2 + l_{c2}^2 + 2 l_1 l_{c2} \cos(\theta_2)] + I_1 + I_2$$
$$M_{12}(\theta) = m_2 [l_{c2}^2 + l_1 l_{c2} \cos(\theta_2)] + I_2$$
$$M_{21}(\theta) = m_2 [l_{c2}^2 + l_1 l_2 \cos(\theta_2)] + I_2$$
$$M_{22}(\theta) = m_2 l_{c2}^2 + I_2$$
$$C_{11}(\theta, \dot{\theta}) = -m_2 l_1 l_{c2} \sin(\theta_2) \dot{\theta}_2$$
$$C_{12}(\theta, \dot{\theta}) = -m_2 l_1 l_{c2} \sin(\theta_2) [\dot{\theta}_1 + \dot{\theta}_2]$$
$$C_{21}(\theta, \dot{\theta}) = m_2 l_1 l_{c2} \sin(\theta_2) \dot{\theta}_1$$
$$C_{22}(\theta, \dot{\theta}) = 0$$
$$g_1(\theta) = [m_1 l_{c1} + m_2 l_1] g \sin(\theta_1) + m_2 l_{c2} g \sin(\theta_1 + \theta_2)$$
$$g_2(\theta) = m_2 l_{c2} g \sin(\theta_1 + \theta_2)$$

**Particle Swarm Optimization**:

Particle Swarm Optimization (PSO), in its present form, has been in existence for roughly a decade, with formative research in related domains (such as social modeling, computer graphics, simulation and animation of natural swarms or flocks) for some years before that; a relatively short time compared with some of the other natural computing paradigms such as artificial neural networks and evolutionary computation.

However, in that short period, PSO has gained widespread appeal amongst researchers and has been shown to offer good performance in a variety of application domains, with potential for hybridization and specialization, and demonstration of some interesting emergent behavior [7].

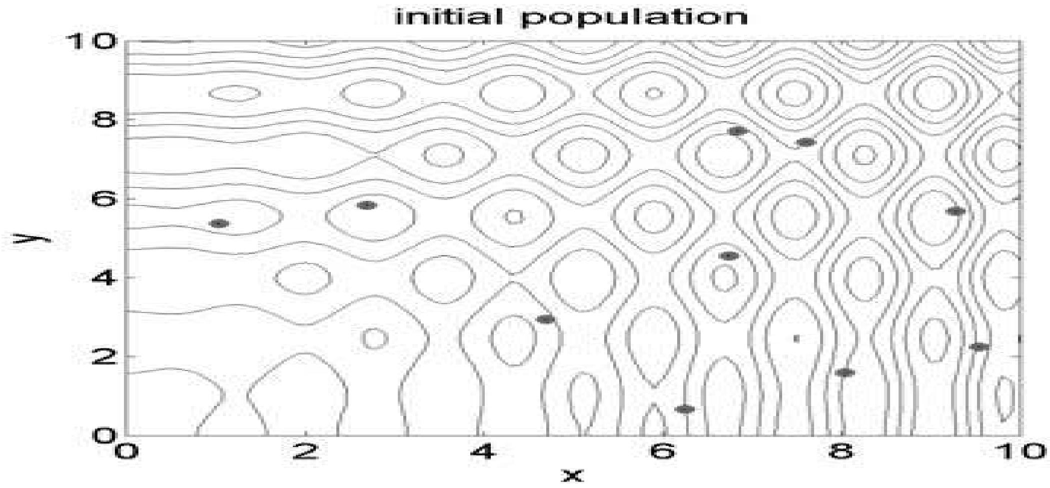



**Figure:( 3 ) Initial random swarm of 10 particles.**

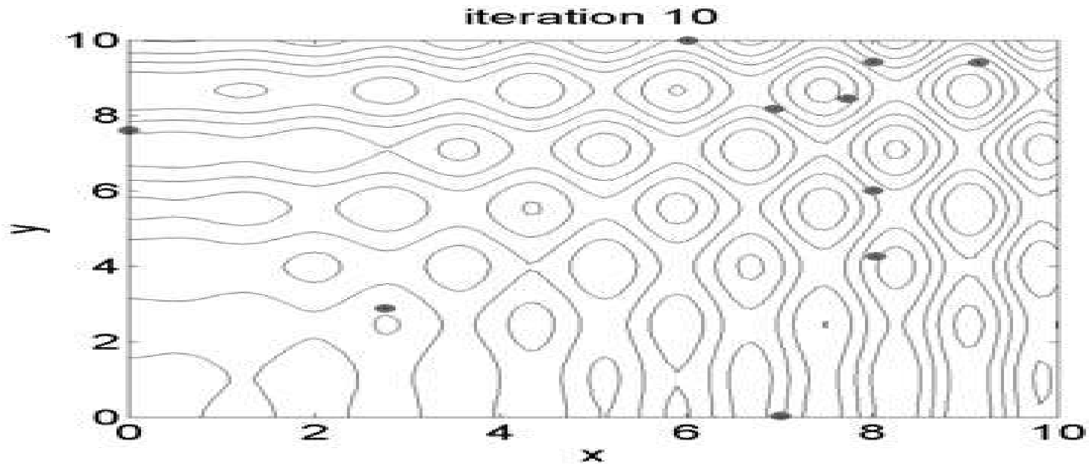



**Figure:( 4 )Swarm after 10 iterations.**

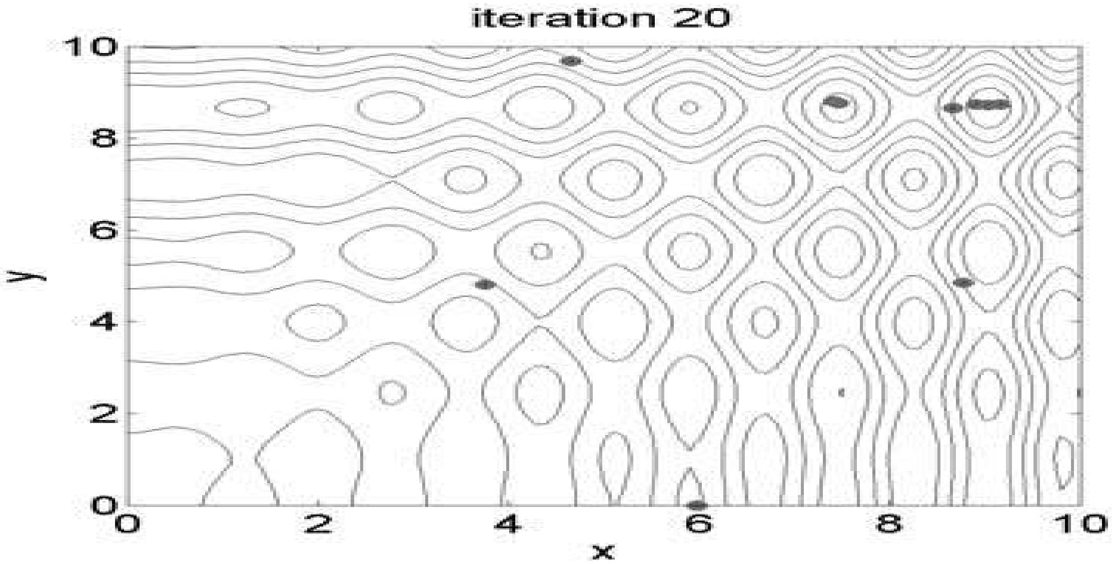



**Figure:( 5) Swarm after 20 iterations.**

**1- :Training Parameters by using Particle Swarm Optimization**:

We will use the particle swarm optimization (PSO) with constriction and inertia weight factors. Each particle $x^p(t) \in X(t)$ and $x^p(t)$ contain $\gamma$ elements $x_j^p(t) \epsilon\, x^p(t)$ at the *t-th* iteration,

where $p = 1,2, \dots \dots \dots po$ and $j = 1,2, \dots \dots, \gamma$; $po$ denote the number of the particle in swarm, and *k* is the dimension of the particle[8]

The purpose of the PSO is to minimize the objective function of particles, this objective function represent error between output of robot arms and desired output . The velocity $v_j^p(t)$ and the position $x_j^p(t)$ of the *j-th* element of the *p-th* particle at the *t-th* iteration can be calculated using the following formulae [9]:

$$v_j^p(t) = 2.rand_j^p().\left(pbest_j^p - x_j^p(t-1)\right) +2.rand_j^p().(gbest_j - x_j^p(t-1))) \quad \dots\dots\dots (9)$$

$$x_j^p(t) = x_j^p(t-1) + v_j^p(t)$$

Where

$$pbest^p = [pbest_1^p \;\; pbest_2^p \dots \dots .pbest_\gamma^p]$$

$$gbest = [gbest_1 \;\; gbest_2 \;\; \dots \dots \dots gbest_\gamma]$$

$j = 1,2, \dots \dots \dots, \gamma$: Where ($\gamma$) is the no. of element in each particle.

$p = 1,2, \dots \dots \dots po$: Where $po$ is population size.

The best previous position of a particle so far is recorded from the previous iteration and represented as $pbest^p$; the position of the best particle among all the particles is represented as $gbest$; $rand(\,)$ returns a uniform random number in the range of [0,1].

The PSO can be improved by using constriction and inertia weight factors, these parameters make PSO more stable and effective. Therefore, the Eq.(9) will become:

$$v(t) = \emptyset\,.\{\vartheta.v(t-1) +\varphi_1.rand().\left(pbest - x(t-1) + \varphi_2.rand().\left(gbest - x(t-1)\right)\right\} \quad (10)$$

Where $\vartheta$ is an inertia weight factor; $\varphi_1$ and $\varphi_2$ are acceleration constants; $\emptyset$ is a constriction factor derived from the stability analysis of (3) to ensure the system to be converged but not prematurely. Mathematically, $\emptyset$ is a function of $\varphi_1$ and $\varphi_2$ as reflected in the following equation:

$$\emptyset = \frac{2}{\left|2-\varphi-\sqrt{\varphi^2-4\varphi}\right|} \quad \dots\dots\dots (11)$$

Where

$\varphi = \varphi_1 + \varphi_2$ and $\varphi > 4$

A suitable selection of the inertia weight $\vartheta$ provides a balance between the global and local explorations. Generally, $\vartheta$ can be dynamically set with the following equation:

$$\vartheta = w_{max} - \frac{w_{max} - w_{min}}{T} * t \quad \dots\dots\dots (12)$$

Where $t$ is the current iteration number, $T$ is the total number of iterations, and $w_{max}$ and $w_{min}$ are the upper and lower limits of the inertia weight and are set to 1.2 and 0.1, respectively.

## Basics Genetic Algorithm :

Genetic algorithms are often viewed as function optimizers, although the range of problems to which genetic algorithms have been applied is quite broad.

**1- Initial Population:**

We mentioned previously that POS it must have population (group of particle) and each particle have group of elements, all details are obtained in section four. From this side GA are similar to PSO in population therefore will followed the same step that mentioned previously.

**2-The cost Function:**

A cost function generates an output from a set of input variables (a particle), we defined previously in chapter four what is particle which represent all parameters of wavelet controller $w$, $a$ and $b$. The cost function may be a mathematical function, an experiment, or a game. The object is to modify the output in some desirable fashion by finding the appropriate values for the input variables[6 ] .

**3- Points Crossover**:

The simplest methods choose one or more points in the chromosome to mark as the crossover points. Then the variables between these points are merely swapped between the two parents. For example purposes, consider the two parents to be:

$$parent1 = [P_{m1}, P_{m2}, P_{m3}, \dots \dots, P_m N_{var}]$$

$$parent2 = [P_{d1}, P_{d2}, P_{d3}, \dots \dots, P_d N_{var}]$$

Crossover points are randomly selected, and then the variables in between are

Exchanged:

$$offspring1 = [P_{m1}, P_{m2}, P_{d3}, P_{d4}, P_{m5}, P_{m6} \dots \dots, P_m N_{var}]$$

$$offspring2 = [P_{d1}, P_{d2}, P_{m3}, P_{m4}, P_{d5}, P_{d6}, \dots \dots, P_d N_{var}]$$

The extreme case is selecting *Nvar* points and randomly choosing which of the two parents will contribute its variable at each position. Thus one goes down the line of the chromosomes and, at each variable, randomly chooses whether or not to swap information between the two parents. This method is called uniform crossover:

$$offspring1 = [P_{m1}, P_{d2}, P_{d3}, P_{d4}, P_{d5}, P_{m6} \dots \dots, P_m N_{var}]$$

$$offspring2 = [P_{d1}, P_{m2}, P_{m3}, P_{m4}, P_{m5}, P_{d6}, \dots \dots, P_d N_{var}]$$

The problem with these point crossover methods is that no new information is introduced: each continuous value that was randomly initiated in the initial population is propagated to the next generation, only in different combinations. Although this strategy worked fine for binary representations, there is now a continuum of values, and in this continuum we are merely interchanging two data points. These approaches totally rely on mutation to introduce new genetic material.

## GA and PSO :

Since PSO and GA both work with a population of solutions, combining the searching abilities of both methods seems to be a good approach. Originally, PSO works based on social adaptation of knowledge, and all individuals are considered to be of the same generation . On the contrary, GA works based on evolution from generation to generation, so the changes of individuals in a single generation are not considered. Based on the compensatory property of GA and PSO.

Therefore, a new algorithm that combines the evolution ideas of both. In the reproduction and crossover operation of GAs, individuals are reproduced or selected as parents directly to the next generation without any enhancement.

However, in nature, individuals will grow up and become more suitable to the environment before producing offspring. To incorporate this phenomenon into GA, PSO that is inspired by social interaction of knowledge is adopted to enhance the top-ranking individuals on each generation. In a social science context, PSO enhances individuals by both sharing information between each other and their individually learned knowledge. Then, these enhanced individuals are reproduced and selected as parents for crossover operation. Offspring produced by the enhanced individuals are expected to perform better than some of those in original population, and the poor-performed individuals will be weeded out from generation to generation. The HGAPSO combines GA with PSO to form a hybrid GA . This hybrid of a GA with existing algorithms can always produce a better algorithm than either the GA or the existing algorithms alon. In this section, basic concepts of GAs and PSO are introduced, followed by a detailed introduction of HGAPSO in the next section.

**1- Hybrid Of GA AND PSO (HGAPSO):**

The detailed design algorithm of recurrent networks by HGAPSO consists of three major operators: enhancement, crossover, and mutation. The learning process of training HGAPSO will described as follows:

Coding**:** will concerns the way of coding parameters of WNN Controllers are represented by particles.

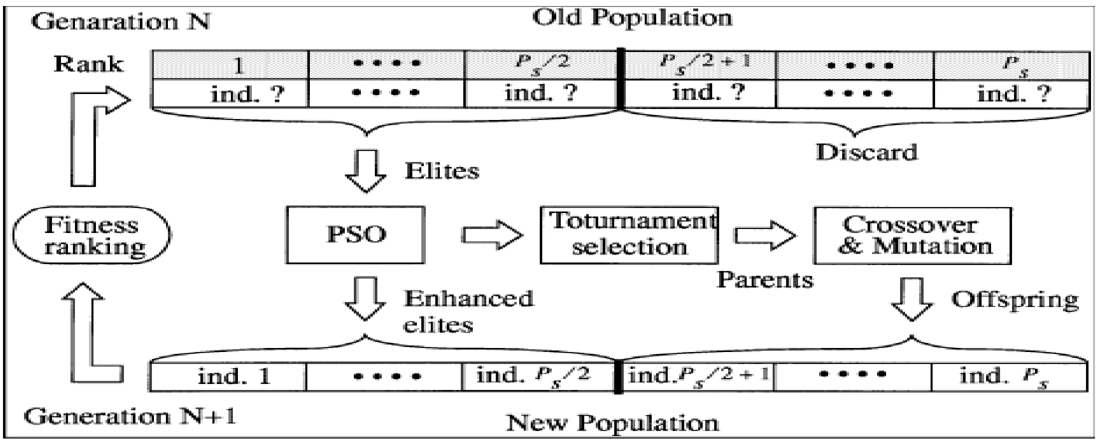



**Figure (6): Operation of HGAPSO**

In HGAPSO, GA and PSO both work with the same population. Initially, individuals forming the population should be randomly generated. After initialization, new individuals on the next generation are created by enhancement, crossover, and mutation operations we must be refer to we will use wavelet mutation in operation mutation that described in chapter to get best results. Above figure (6) show the operation of HGAPSO. Where in HGAPSO, GA and PSO both work with the same population. Initially, individuals forming the population should be randomly generated. After initialization, new individuals on the next generation are created by enhancement, crossover, and mutation operations we must be refer to we will use wavelet mutation in operation mutation that described in chapter to get best results. Above figure (6) show the operation of HGAPSO.

**Crossover**: We mentioned previously there are different way to perform crossover operation o produce well-performing particles, in the crossover operation parents are selected from the enhanced elites only as shown in figure(7). To select parents for the crossover operation, the Tournament selection scheme.

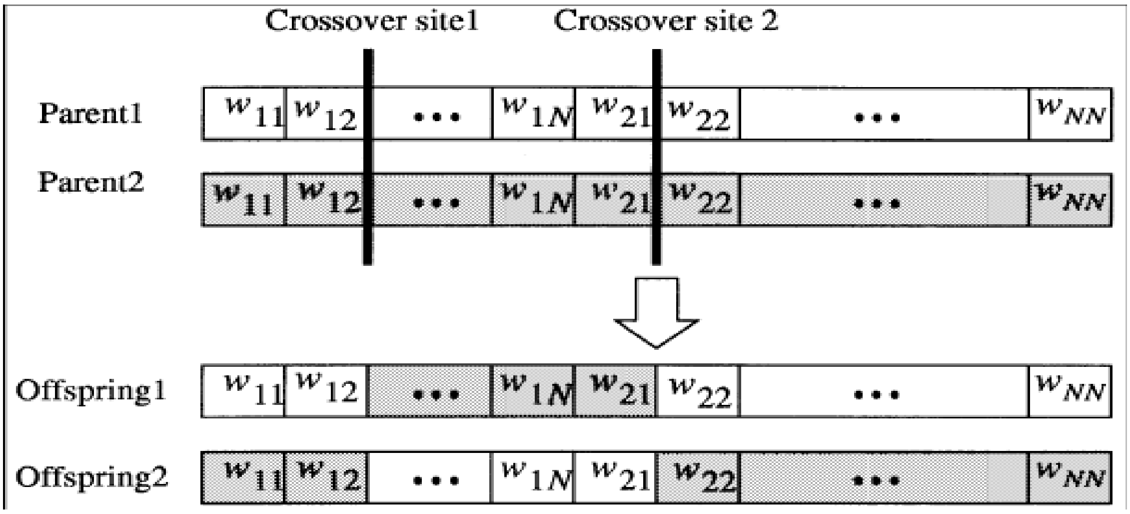



**Figure (7): Crossover operation on the weights particle encoding WNN controller.**

is used, in which two enhanced elites are selected at random, and their fitness values are compared to select the elite with better fitness value as one parent. Then the other parent is selected in the same way. Two offspring are created by performing crossover on the selected parents. Some of the commonly used crossover techniques are k-point crossover, where crossover sites are selected randomly In this study, two point crossover operation is used. WNN and LLAWNN, the two crossover sites are selected at random. Then, the parents are crossed an separated at the two sites as shown in figure (7) which obtain the crossover of weights $w$ in particle. These produced offspring occupy half of the population in the next generation.

**<u>Simulation Results By using WNN controller</u>:**

Figure (8) show the output Robot manipulator by using WNN controller at first iteration and MSE =0.672 and figure ( 9 ) show the output of Robot manipulator by using WNN controller at final iteration with MSE = 0. 000723

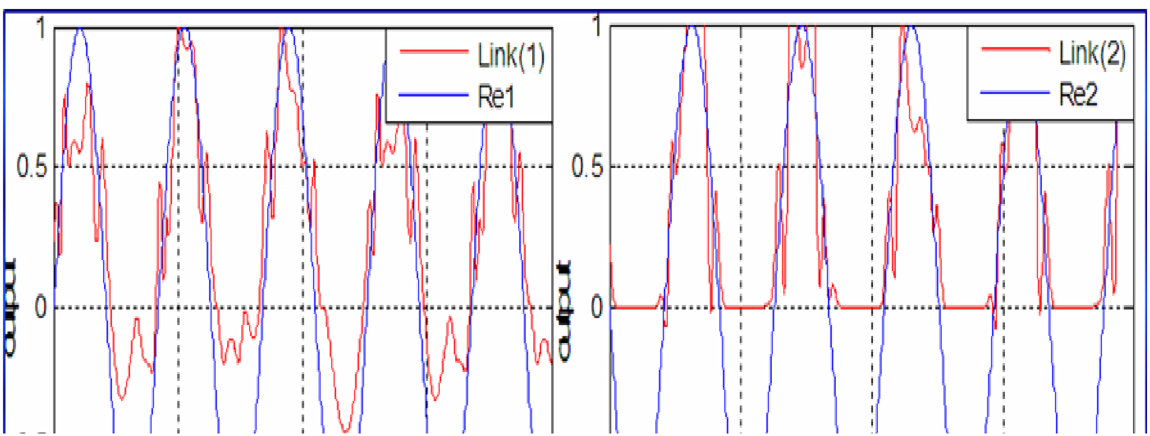


**Figure (8) Results by using WNN controller at first iteration (a). link1 (b). link2**

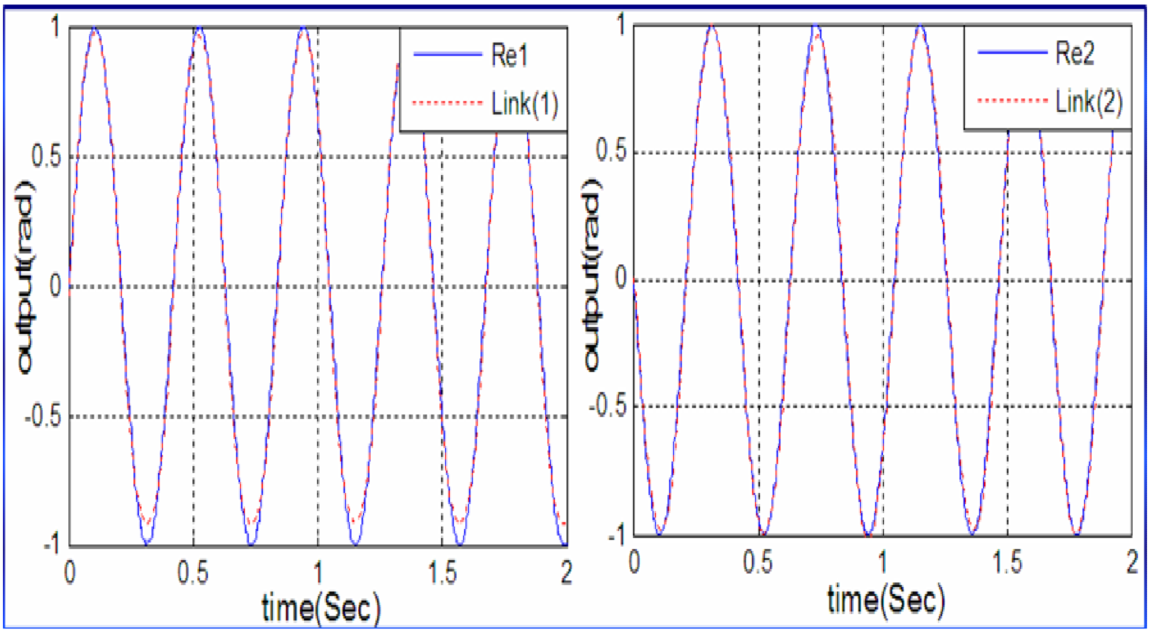



**Figure (9) Results by using WNN controller at final iteration (a). link1 (b). link2**

The final iteration is 700 and MSE=0.000723.

## <u>Conclusion</u>:

In this paper, wavelet neural network design by a new evolutionary algorithm, HGAPSO, is used. HGAPSO introduces the concept of the maturing phenomenon in nature into the evolution of individuals originally modeled by GA. The maturing phenomenon is mimicked by PSO, where individuals enhance themselves based on social interactions and their private cognition. From the perspective of PSO, HGAPSO introduces the crossover operation into the society. Thus, the evolution individuals is no longer restricted to be in the same generation, and better-performed individuals may produce offspring to replace those with poor performance. Thus HGAPSO combines the new individual generation function of both GA and PSO, which together mimics social behaviors of animals, breeding, and survival of the fittest. To demonstrate the performance of HGAPSO, designs of wavelet neural network for robot controller is simulated.

**References:**


1- Arne. E R.," *A Novel Focused Local Learning Wavelet Network with Application to In Situ Monitoring during Selective Silicon Epitaxy*", Ph.D. thesis, North Carolina State University, May 2001.

2- M. Wachowiak P., Smolikova R., Zheng Y., J. Zurada M, and A. Elmaghraby S.,"An Approach to Multimodal Biomedical Image Registration Utilizing Particle Swarm Optimization", IEEE Transaction on Evolutionary Computation", vol. 8, NO. 3, JUNE 2004.

3- H. habibiyan, A. Syadian and H. Ghfoori-Fard," *Wavelet network controller for nuclear steam generators*", *Journal of Physics: Conference* Series 23 (2005) 182–191.

4-Yuehui Chen, Bo Yang and Ajith Abraham," *Optimal Design of Hierarchical Wavelet Networks for Time-series Forecasting*", ESANN'2006 proceedings - European Symposium on Artificial Neural

5- T.Wang and Y.sugai, "*A Local Linear Adaptive Wavelet Neural Network*", T.IEE Japan, vol.122-c, No.2, 2002.[6] S. H. Ling, H. H. C. Iu, K. Y. Chan, H. K. Lam, Benny C. W. Yeung, and Frank H. Leung, " Hybrid Particle Swarm Optimization With Wavelet Mutation and Its Industrial Applications", *IEEE Transactions on System, Man, and Cybernetics ,* Vol. 38, NO. 3, June 2008.

6- S. H. Ling, H. H. C. Iu, K. Y. Chan, H. K. Lam, Benny C. W. Yeung, and Frank H. Leung, " Hybrid Particle Swarm Optimization With Wavelet Mutation and Its Industrial Applications", IEEE Transactions on System, Man, and Cybernetics , Vol. 38, NO. 3, June 2008.

7-G. Goos, J. Hartmanis, and J. van Leeuwen, "*Wavelet Analysis and Its Applications*",Springer, Vol. 2251 ISBN 3-540-43034-2 December 18-20, 2001.

8- Mohammed E. and Mohamed S. Kamel," A Taxonomy of Cooperative Particle Swarm Optimizers", *International Journal of Computational Intelligence Research,* ISSN 0973-1873 Vol.4, No.2 (2008), pp.137–144.

9- L. Liu, W. Liu and David A. Cartes," Permanent Magnet Synchronous Motor Parameter Identification using Particle Swarm Optimization", *International Journal of Computational Intelligence Research, ISSN* 0973-1873 Vol.4, No.2 (2008), pp.211–218.