Journal of Engineering and Development, Vol. 15, No. 1, March (2011) **ISSN 1813-7822**

Implementation of Artificial Neural Networks Trained by Particle Swarm Optimization using Multi-Phase Switched – Capacitor Circuits

Assist. Prof. Dr. Hanan A. R. Akkar
Department of Electrical and Electronic Engineering
University of Technology
Baghdad / Iraq
Email: - hnn_aaa@yahoo.com

Abstract

In this paper, a proposed design of Artificial Neural Networks Trained by Particle Swarm Optimization using multi-phase switched-capacitor circuits is presented. Swarm intelligence is based on collective behavior of self organized group of agents. Each agent follows a relatively simple set of rules and interacting with its local surrounding. Particle Swarm Optimization (PSO) has been an increasingly interesting topic in the field of computational intelligence. PSO is another optimization algorithm that falls under the soft computing address. One application of PSO has tremendous success is in the field of Artificial Neural Networks (ANNs) training. In this paper an adaption of the ANN weights using PSO is proposed as a mechanism to improve the performance of ANN.

For this purpose we have modified the MATLAB PSO toolbox to be suitable with neural application. In neural networks, the multiplier is needed to deal with the learning of weights, and the generation of associated outputs therefore, a proposed design of multiplier circuit using multi-phase switched-capacitor circuit that can be implemented in CMOS technology.

Generating multiple clock sources is a common requirement for the designing multi-phase switched-capacitor circuits so; a proposed design of multi-phase clock generator is presented which produces sequential non-overlapping clock pulses. The proposed design of multi-phase switched-capacitor neuron and its corresponding "synapses" also presented in details. Simulation results are presented using EWB package, which illustrates the validity of the proposed switched capacitor circuit's designs.

Key word: - Artificial Neural Networks (ANN), (PSO), Switched Capacitor Network (SCN).

الخلاصة

تم في هذا البحث, تصميم للشبكات العصبية المدربة بطريقة افضلية الحشد الجزيئ مستخدمة دوائر المتسعات المفتاحية المتعددة الأطوار. إن ذكاء الحشد (Swarm Intelligence) قائم على التصرف الجماعي لمجموعة جزيئات ذات نظام ذاتي. كل جزيئة تتبع مجموعة بسيطة نسبياً من القوانين وتتفاعل فقط مع الجزيئات الموقعية المحيطة بها. افضلية الحشد الجزيئي (Particle Swarm Optimization) قد اصبحت موضوع متزايد الاهمية في مجال الحسابات الذكية. حيث من اهم تطبيقات افضلية الحشد الجزيئي التي لاقت نجاح واسع هو في تدريب الشبكات العصبية الاصطناعية (Artificial Neural Network).

افترض في هذا العمل تكييف اوزان الشبكة العصبية الاصطناعية بأستخدام ألية افضلية الحشد الجزيئي وذلك لتحسين اداء الشبكات العصبية الاصطناعية. لهذا الغرض قمنا بتطوير صندوق ادوات افضلية الحشد الجزيئي في بيئة الماتلاب MATLAB لكي يكون مناسبا لتطبيقنا في تنفيذ دوائر رقمية بأستخدام الشبكات العصبية الاصطناعية. نحتاج في الشبكات العصبية دائما إلى دائرة ضرب لكي تتعامل مع تكييف الأوزان وكذلك للحصول على القيمة النهائية المتعلقة بها, لذلك يوجد تصميم مقترح لدائرة ضرب باستخدام دائرة المتسعات المفتاحية المتعددة الأطوار والتي ممكن تنفيذها بالكامل في تكنولوجية أشباه الموصلات الفلزية.

إن توليد عدة مصادر للتردد المتعدد الأطوار هو من المتطلبات الأساسية في تصميم دوائر المتسعات المفتاحية ذات الأطوار المتعددة, لذلك يوجد تصميم مقترح لتوليد التردد المتعدد الأطوار الذي يولد نبضات ترددية متعاقبة وغير متداخلة. يوجد تصميم مقترح للخلية العصبية باستخدام المتسعات المفتاحية المتعددة الأطوار مقدم في هذا البحث مع الشرح الوافي. إن نتائج المحاكاة للدوائر الالكترونية المقترحة باستخدام برنامج التحليل الالكتروني (EWB) قد أثبتت فعالية هذه الدوائر المقترحة.

1. Introduction

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous system, such as brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by example. ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons [1].

In practical terms Neural Networks (NN) are non-linear statistical data modeling tools. They can be used to model complex relationships between inputs and outputs or to find patterns in data. NN is an interconnected group of nodes, akin to the vast network of neurons in the human brain. More complex NN are often used in parallel distributed processing [2]. ANN models are specified by the net topology, node characteristic, and training or learning rules. These rules specify an initial set of weights and indicate how weights should be adapted during use to improve performance. Most neural net algorithms also adapt connection weights in time to improve performance based on current results[3]. Maundy and El-Masry designed a novel analog realization of a self-organizing neural network using switched-capacitor circuits [4]. The method for designing switched-capacitor four-quadrant analog multiplier is designed by Changyue, Peng and yizhong[5]. The designs of electronic circuits that perform several neural models and learning rules are proposed by Hanan [6].

There has been significant increase in research and development in the area of applying evolutionary computation techniques for the purpose of evolving one or more aspects of ANNs. These evolutionary techniques have usually been used to develop NN structure or the network learning algorithm. A new learning algorithm combined ANN was proposed to determined the optimal weights, these weights adjusted by PSO[7].

PSO has its own advantages and drawbacks over other computational algorithms, advantages like its probabilistic mechanism and multi starting points, hence PSO can avoid getting into the local optimal solution, but the most utilized property PSO in this paper is its free derivative activation function, which means that we will train feed forward ANNs using PSO as the learning algorithm with only hard limit activation function for all network layers. According to the hard limit activation function properties the output will be either one or zero, this property will be very helpful simplifying the network multiplication process. Numerous studies have further explored the power of PSO as a training algorithm, these studies shown that ANNs trained by PSO have more accurate results than other training algorithms, but in the same time its slowest than other algorithms like back-propagation [8-10].

2. Particle Swarm Optimization (Pso)

Particle swarm optimization is a population based evaluation optimization technique developed by J. Kennedy and R. Eberhart in 1995 motivated by the social behavior of bird flocking or fish schooling. PSO is a kind of random search algorithm that simulates nature evolutionary process and performs good characteristic in solving some difficulty optimization problems. The basic concept of PSO comes from a large number of birds flying randomly and looking for food together. Each bird is an individual and called a particle. As the birds looking for food, the particles fly in a multidimensional search space looking for the optimal solution. Here all the particles are composed of a family rather than the isolated individual for each other; they can remember their own flying experience and share theirs companion's flying experience [11].

The basic PSO model consists of swarm of particles moving in a D-dimensional search space the direction and distance of each particle in the hyper dimensional space is resolute by its fitness and velocity. In general the fitness is primarily related with the optimization objective and velocity is updated according to a sophisticated rule [12].

In PSO, populations starts with random initialization of individuals in the search space and then repeat the social behavior of the particles in the swarm till achieves the best possible result by iterative searching. At each iterative step the velocity (position change) is updated and the particle is moved towards a new position. The best previously visited position at the nth particle is denoted by the personal best position pbest, while the position of the best individuals of the whole swarms is denoted as the global best position gbest. In other words, the particle swarm optimization idea consists of at each time step, changing the velocity and location of each particle towards its pbest and gbest locations according to (2) and (3):

$$V_{id} = W*V_{I+}C_1*rand1*(P_{id}-X_{id}) + C_2*rand2*(P_{gd}-X_{id}) \quad \ldots (2)$$

$$X_{id} = X_{id} + V_{id}. \quad \ldots (3)$$

Where W is the inertia weight which bring stability between global exploration and local exploration, C_1 and C_2 are two constants called learning factors, rand1 and rand2 are two independent random numbers uniformly spread in the range of [0, 1], for (2) the first part represents the inertia of previous velocity; the second part is the cognition part which represents the private thinking by itself; the third part is the social part, which represents the assistance among the particles. P_{jd} represent personal best position recorded by particle I and P_{gd} is the global position and d is the index of dimension in the search space. During the past few years PSO has been shown successful for many applications[13-14] several papers discuss how to apply PSO in training NNs and their advantages [15- 18]. For the purpose of NNs learning the empirical error referred to as the objective function (mean square error) to be optimized by the optimization method (minimized to 0) is given by :

$$M.S.E. = \sqrt{\frac{\sum_{j=1}^{n} \sum_{k=1}^{m} (Tjk - Yjk)^2}{nm}} \qquad ...(4)$$

Where n is the number of training patterns, m is the number of outputs, T is the target and Y is the actual value.

2.1 The Success and Limitation of PSO

PSO is a computational intelligence based technique that is not largely affected by the size and nonlinearity of the problem, and can converge to the optimal solution in many problems where most analytical methods fail to converge. It can therefore, be effectively applied to different optimization problems. It's an optimization algorithm that using only primitive mathematic calculation. So the advantage of the PSO over many of the other optimization algorithms is its ability to handle optimization problems with multiple local optima reasonably well and relative simplicity of implementation the calculation speeds and fast. More over, PSO has advantages over other similar optimization techniques, namely the following:

1) PSO is easier to implement and there are fewer parameters to adjust.

2) In PSO, every particle remembers its own previous best value as well as the neighborhood best; therefore, it has a more effective memory capability than other optimization technique.

3) PSO is more efficient in maintaining the diversity of the swarm (more similar to ideal social interaction in a community), since all the particles use the information related to the most successful particle in order to improve themselves [18].

There are not many parameter need to be turned in PSO. Here is a list of the parameters and their typical values

- The number of particles, the typical range is about 20 to 40. Actually for most of the problems 10 particles is large enough to get good results. For some difficult or special problems, one can try 100 or 200 particles as well.
- Dimension of particles, it is determined by the problem to be optimized.
- Range of particles, it is also determined by the problem to be optimized, we can specify different ranges for different dimension of particles.
- Learning factors, c1 and c2 usually equal to 2.
- The stop condition, the maximum number of iterations the PSO execute and the minimum error requirement .

2.2 PSONN for the Optimal Training and Topology

Neural network is an artificial intelligence model originally designed to replicate the human brain's learning process. A typical network is composed of a series of interconnected nodes and the corresponding weights. It aims at simulating the complex mapping between the input and output. The training process is carried out on a set of data including input and output parameters. The learning procedure is based on the training samples and the testing samples are used to verify the performance of the trained network. During the training, the weights in the network are adjusted iteratively till a desired error is obtained [1].

This capability of learning from data without a prior knowledge makes neural networks particularly suitable for classification and regression tasks in practical situations. The most popular supervised learning technique in ANN is BP algorithm. The back-propagation algorithm involves the gradual reduction of the error between model output and the target output. It develops the input to output, by minimizing a Mean Square Error cost function measured over a set of training examples. The problem with neural networks is that a number of parameters have to be set before any training can begin. However, there are no clear rules as to how to set these parameters. Yet these parameters determine the success of the training. Therefore, PSO is used to find their optimum values [19].

A particle represents the weight vector of NN, including all biases. The dimension of the search space is therefore the total number of weights and biases. The fitness function is the Mean Squared Error (MSE) over the training set. Changing the position means updating the weight of the network in order to reduce the error. All the particles update their position by calculating the new velocity, which they use to move each particle to the new position. The new position is a set of new weights used to obtain the new error. For PSO, the new weights are adapted even though no improvement is observed. This process is repeated for all the particles. The particle with the lowest error is considered as the global best particle so far.

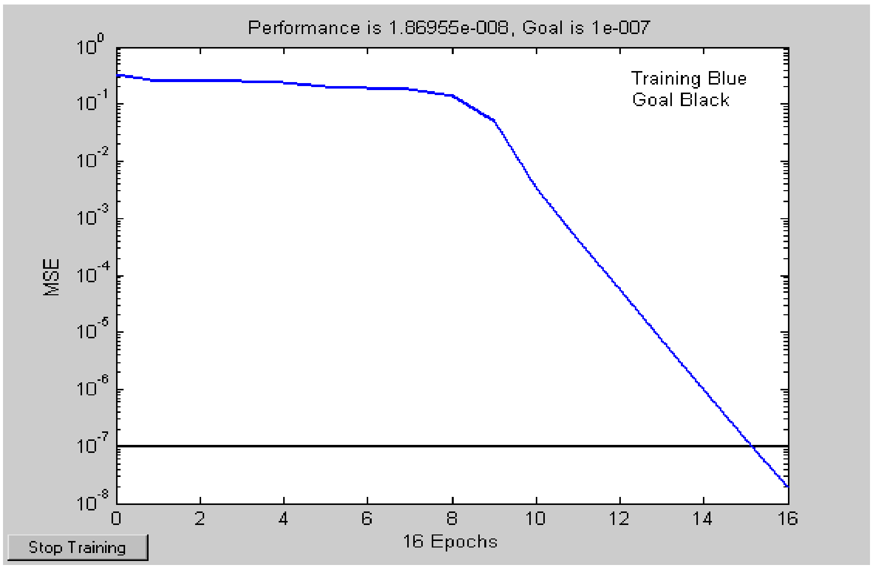
2.3 Computer Simulation Result of Training ANN by PSO

Before training process, the parameters of training algorithm (PSO) are set to swarm size=25 particles, $c_1 = c_2 = 2$, $r_1 = 0.8$, r_2=0.2 and Φ=0.9. The inertial weight (Φ) decreases from 0.9 to 0.2 linearly depending on the equation (2). The initial weights are randomly generated between [-22, 22] and biases between [-15,15] with maximum velocity [$V_{\max}$=18], these numbers represents the size of search space in X-Y dimensions.

Dimension = (input * hidden) + (hidden *output) + hidden$_{\text{bias}}$ +output$_{\text{bias}}$

The parameters are set to the momentum coefficient α=0.9 and the learning rate η=0.5. The initial weights and biases are randomly generated between [-0.5, 0.5]. The maximum number of iterations (epoch) =100 and Mean Square Error (MSE=10e-7).

Figure (1) and Figure (2) illustrate the performance of ANN which represents the relationship between desired and actual output. The accuracy of performance for ANNPSO on dataset is shown in Figure (3a and b), where the weights (particle positions) move towards G_{best} position. The performance of ANNPSO and the relationship between desired and actual output is shown in Figure (4 a and b).



Figure(1) The Mean Square Error ANN.

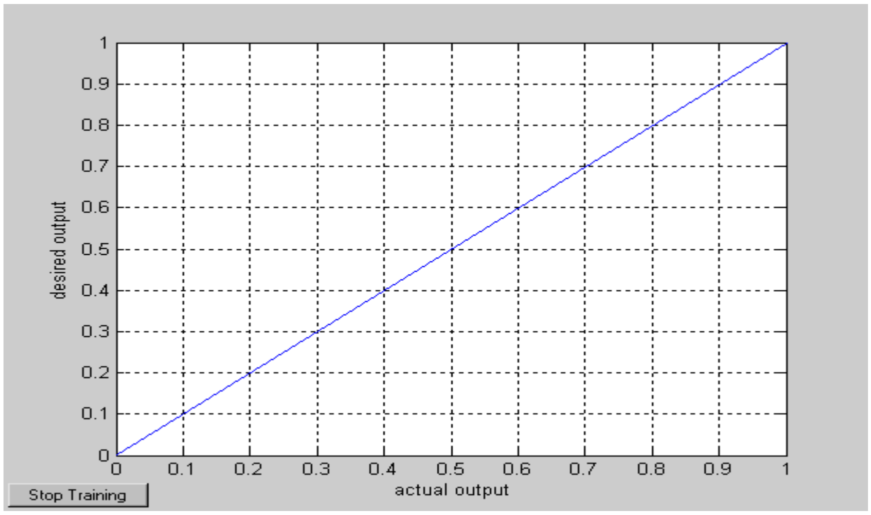
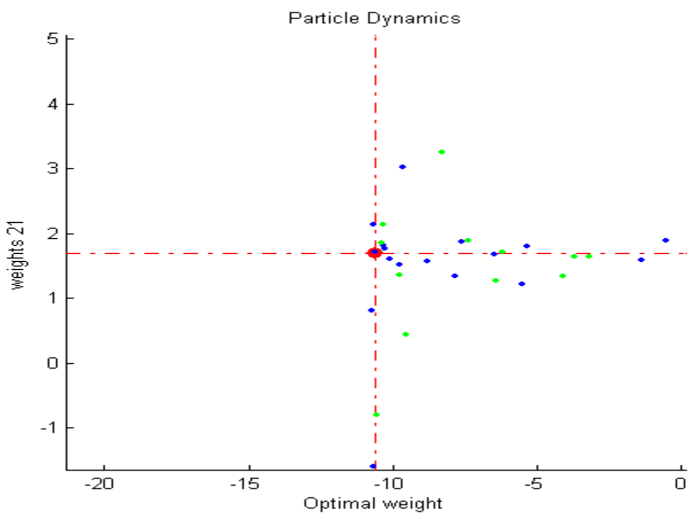
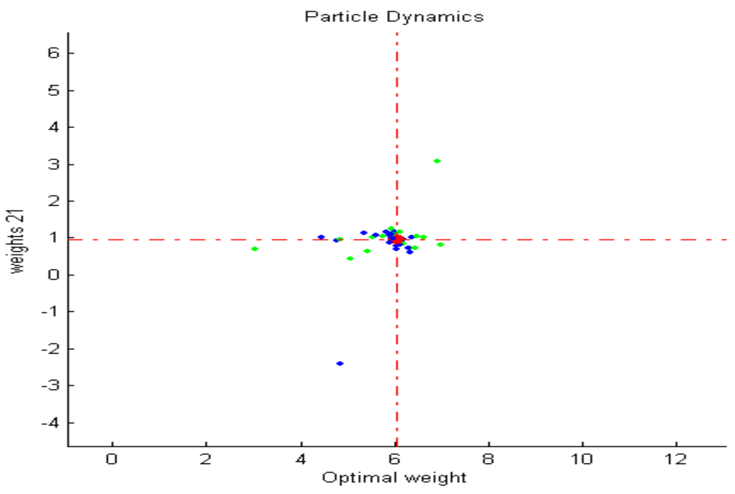


Figure (2)The performance of ANN (desired & actual output relationship).



a- After 2 iterations.



b- After 15 iterations.
Figure (3) The Accuracy Of Performance For ANNPSO On Dataset.

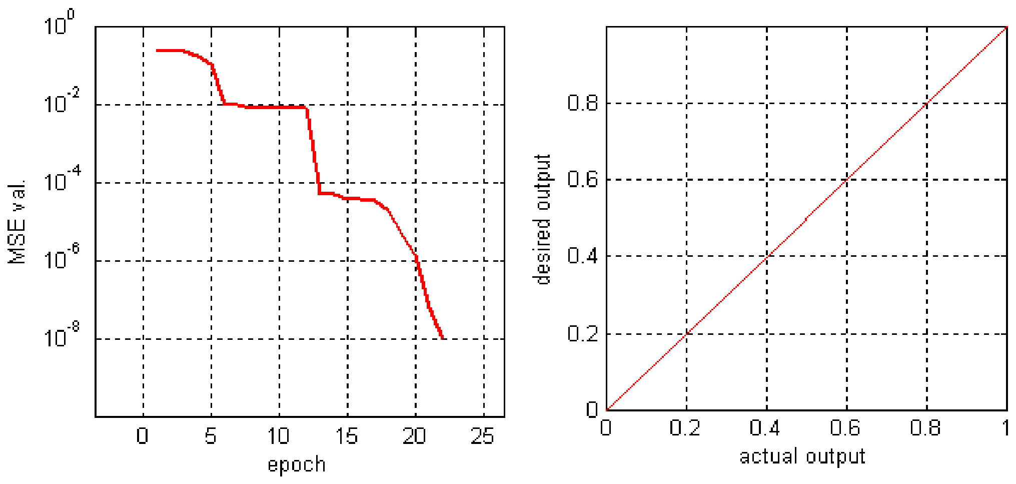


Figure (4) A- The Performance Of ANNPSO On Dataset. B- The Relationship Between Desired And Actual Output.

From table (1), the results show that ANNPSO convergence time depends on the swarm size and search space, so that, some times is fast and another times is slow depending on the initial weights, which are randomly generated in search space, these initial weights are changed over entire the search space when the run of program is repeated. Thus different convergence time is carried out for each run. The convergence time for ANN learning based on PSO algorithm is 1.5 seconds at 22 iterations. It shows that ANNPSO result is better with 99.1% accuracy.

Table (1) The Results of ANNPSO on dataset.

Items	ANNPSO
Learning Iterations	22
Error Convergence	1.25e-009
Convergence Time	1.5 sec
No. of initial weights	25 set
No. of derivatives/weight	Free
Accuracy (%)	99.1

3. Multi-Phase Switched – Capacitor Circuit

In MOS technology it is relatively easy to implement capacitors and MOS switches as well as op-amps, but difficult to construct resistors with the required accuracy. The recognition that a resistor could be approximated with two MOS transistors and one capacitor ((switched -capacitor)) was the key to solve this problem. MOS transistor in one specialized use of this transistor, the voltage between the source and the gate is either zero or a value larger than a threshold voltage Vth, typically 1 or 2 volt. In this mode of operation, the devise is known as the MOS switch. Switched – Capacitor (SC) circuits are based on the principle that a capacitor C is periodically switched between two circuit nodes at a sufficiently high rate clock frequency (f_c) that is approximately equivalent to a resistor $R = \frac{1}{C.f_c}$ connecting the two nodes as shown in Figure(5).

SC circuits are sampled–data analog systems, and as such they occupy an intermediate position between fully analog and fully digital. SC circuits are realized as integrated circuits, and hence are compacting, reliable and (for large–volume applications) inexpensive .The SC circuit realization usually requires a less complicated structure and hence often much less chip area on integrated circuits [20].

3.1 The Circuit Design of Multi-Phase Clock Generator

The Complexity of clock generator is one of the most important parameters in the design of SC circuits; therefore the clock generator must be designed precisely.

A multi-phase SC network must be driven by symmetrical non- overlapping 3-phase clock of frequency (f_c) denotes Φ1, Φ2 and Φ3 as shown in Figure(6). Switches state changes at the beginning of each period (τ) which is controlled by a common clock, so the time intervals are divided into three equal intervals, and the clock frequency must be much higher than the signal frequency (f_s) for accurate results.

A proposed clock generator design is represented by using a half of dual binary counter which is controlled by a simple multi-vibrator followed by simple logic gates in order to achieve three phases of clock frequency ($f_c = 10KHz$). So we generate several clocks from a single clock source as shown in Figure(7) .This circuit designs can easily be expanded by adding more logic gates in order to get the desired number of phases up to 2^8 with high precision, stability and non- overlapping pulses.

3.2 The Circuit Design of Multi-Phase SC Neuron Circuit

A proposed circuit design of multi-phase SC neuron circuit is shown in Figure(8).The switches are made of MOS transistors, and are controlled by 3-phase clock generator, so the switch is closed if the corresponding Φ is high otherwise it is open (i.e. Φ is low). When Φ1 is high, the values of the inputs X_1 and X_2 of the “neuron” are sampled on to the capacitors C1 and C2 respectively, while these capacitors hold these values, Φ2 goes high and the output of the neuron is zeroed. Then, when Φ3 goes high, the input capacitors are switched to virtual ground, transferring the charge of one of there plates on to the left – hand plate of capacitor C3. As long as the operational amplifier operates in its linear region, the charge conservation dictates that the output Y_1 is equal to the weighted sum of the inputs as shown below:

$$Y_1 = \frac{C1}{C2}X1 + \frac{C3}{C2}X2 \qquad \dots (5)$$

Where $X1$ and $X2$ are the input voltages, $C1$ and $C3$ are the input capacitors of the SC neuron circuit and $C2$ is the feedback capacitor.

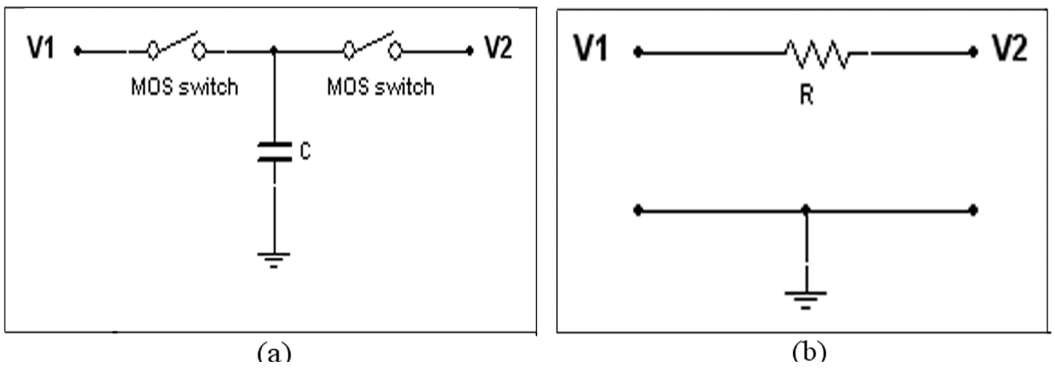
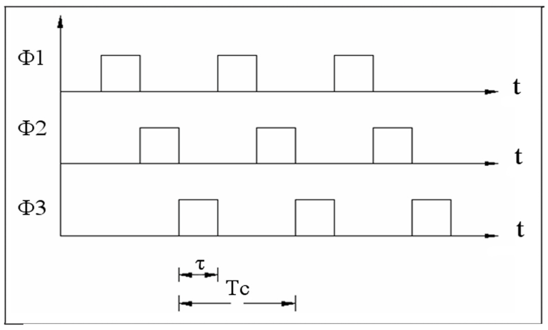


Figure (5) a) Switched-Capacitor circuit b) Equivalent resistor.



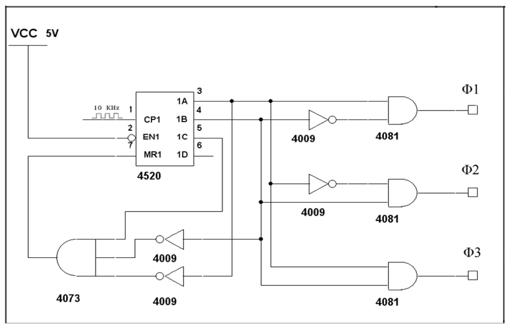


Figure (6) 3-Phase clock. **Figure(7) 3- Phase clock generator.**

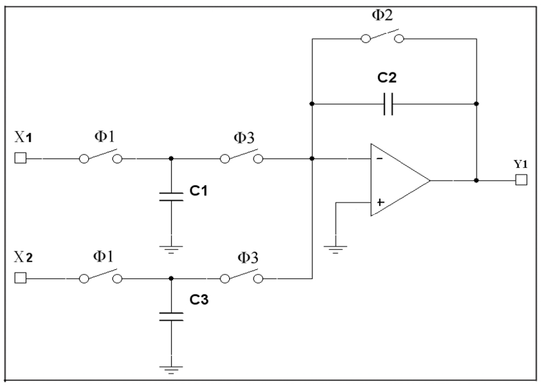


Figure (8) SC neuron circuit.

3.3 The 3-Phase SC Circuit Design Of ANN Trained By PSO Using Technique

A proposed circuit design of PSO using 3-phase SC technique is shown in Figure (9). The first stage is the proposed 3-phase multiplier represented by a general multiplier block which is shown in Figure (10), to compute the multiplication of the input voltage (x) by the weight (w). In Figure(10) the general multiplier contains n-multiplier block which is shown in Figure(11) and p-multiplier block which is shown in Figure(12) followed by two switches to obtain the correct multiplication in the case of positive weight value or negative weight value respectively. The output voltage of the multiplier is equal to:

$$V = \frac{C1}{C2}.K.w.x \qquad \text{... (6)}$$

Where $C1$ is the input capacitance of the SC multiplier circuit, $C2$ is the feed back capacitance, K is the process trans-conductance (typically of value $20\,\mu A/V^2$) and $(\frac{C1}{C2})$ is adjusted to make $(\frac{C1}{C2}K)=1$. The second stage in Figure(9) is the summing block which is shown in Figure(13) that verified:

$$net = -\frac{1}{C2}[w_{11}x_{11}C1 + w_{12}x_{21}C3] \qquad \text{... (7)}$$

Where C2 is the feedback capacitance, C1 and C3 are the input capacitors of the SC summation circuit and x_{11} and x_{21} are the input voltages. The third stage in Figure (9) is the inverter block which is shown in Figure (14) to invert the output voltage of the summation circuit. The fourth stage in Figure(9) is the hard limiter activation function block which is shown in Figure(15) where an analog extension based on a simple capacitive storage is proposed for enhancing the output resolution .The output voltage of this circuit is :

$$o_i = f_h(net) = \pm 1 \text{ Volt} \qquad \text{... (8)}$$

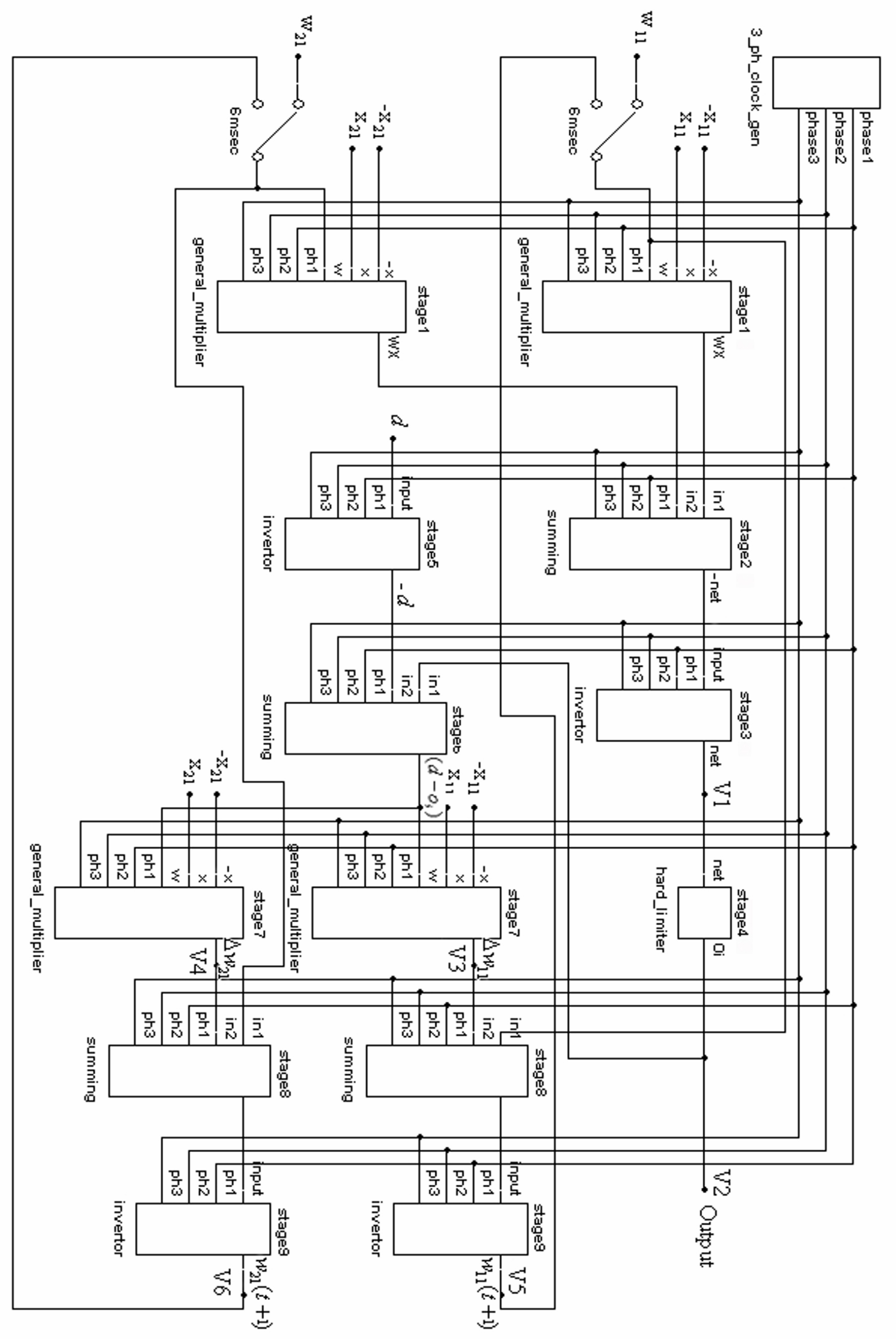
Where f_h is the hard limiter activation function. If $o_i \neq d$. The sixth stage is the summing block that computes $c(d_i - o_i)$. The seventh stage is the same general multiplier block that is shown in Figure (10) to compute the multiplication of the output voltage of the previous stage by the input voltage x. So the output voltages of the general multiplier verify eqn. (6). The next stage is the summing block which is shown in Figure (13) followed by the inverter block to verify eqn. (7). Then the update weight vector $(w(t+1))$ is fed back to the first stage and so on, until the output voltage of neuron is equal to the actual neuron's response (d_i).

The proposed circuit in Figure(9) is simulated using Electronic Work Bench (EWB) package. Let the clock frequency f_c = 10 KHz, the input x and the initial weight w vectors as follows: $x = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ Volt and $w^1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ Volt. The learning constant is assumed to be C1, $C2 = 1$ and the desired response d = -1 volt. $net = (w^{1t}x) = [1 \quad 1] \begin{bmatrix} 1 \\ 2 \end{bmatrix} = 3$ Volt. The output voltage of this circuit is:- $o_i = f_h(net) = f_h(3) = 1$ Volt. The second training for the same input vector x and desired response $d = -1$ volt is shown below:-

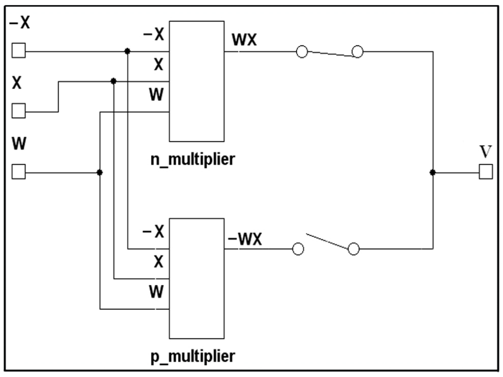
$$net^2 = w^{2t}.x = [-1 \quad -3] \begin{bmatrix} 1 \\ 2 \end{bmatrix} = -7 \text{ Volt.}$$

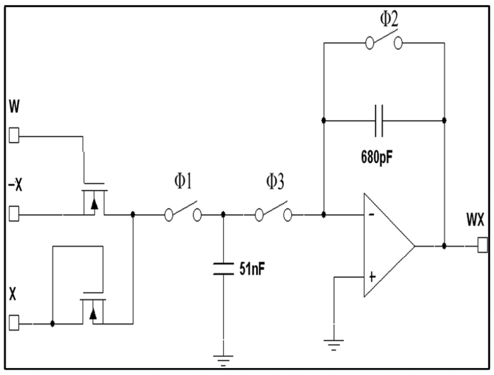
The output voltage of the proposed circuit in Figure (9) is equal to :- $o_i = f_h(net) = f_h(-7) = -1$ Volt. Since $o_i = d$, the weight vector is not modified by PSO.

Figure (16) shows the simulated output voltages of the proposed circuit illustrated in Figure(9). These results show that the proposed circuit is operated successfully and the outputs are equal to its corresponding target (d).

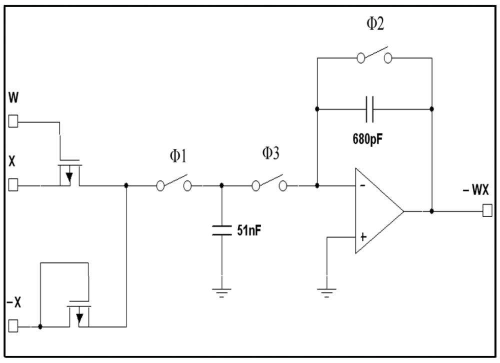


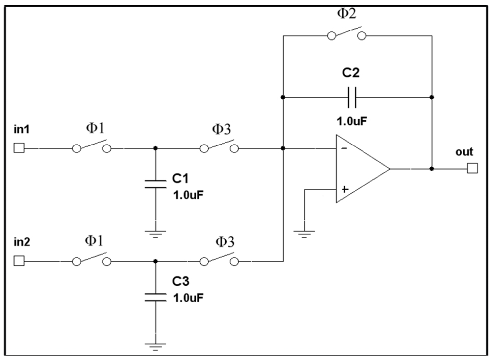
Figure(9) The Proposed SC Circuit Design Of ANNPSO Using Multi-Phase SC Technique.



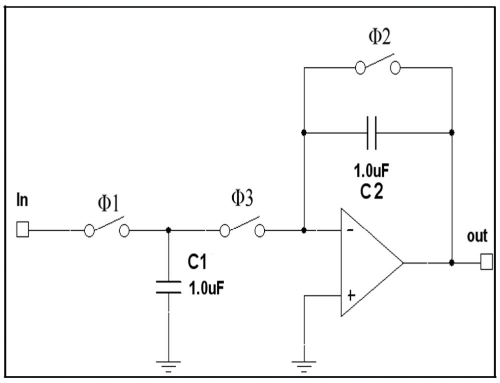


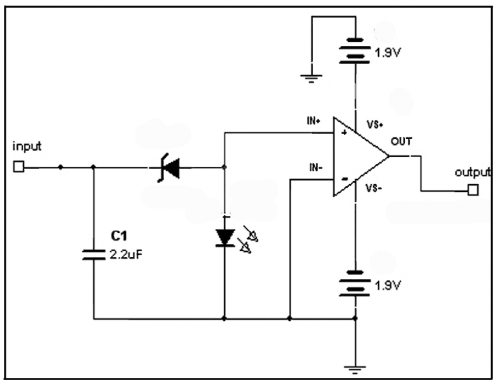
Figure(10) General multiplier sub-circuit. Figure(11) N-multiplier circuit.



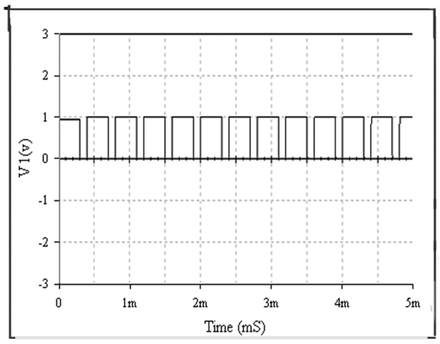


Figure(12) P-multiplier sub-circuit. Figure(13) Summing sub-circuit.

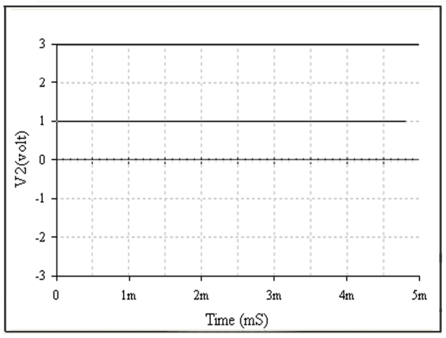




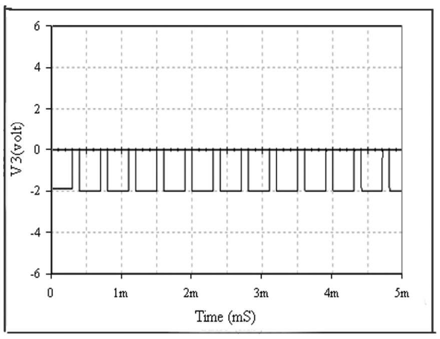
Figure(14) Inverter sub-circuit. Figure(15) Hard limiter sub-circuit.



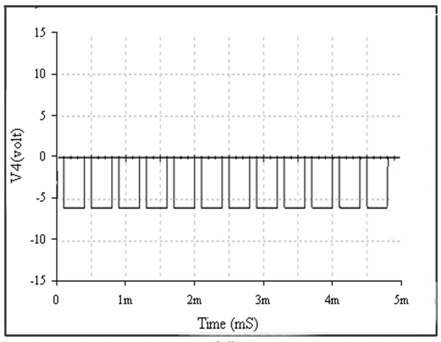
(a)



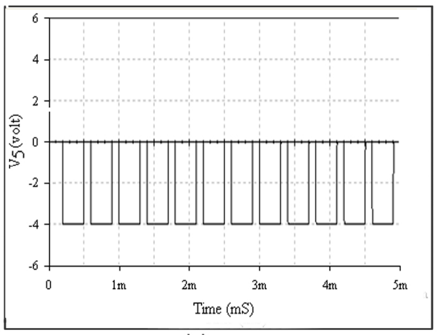
(b)



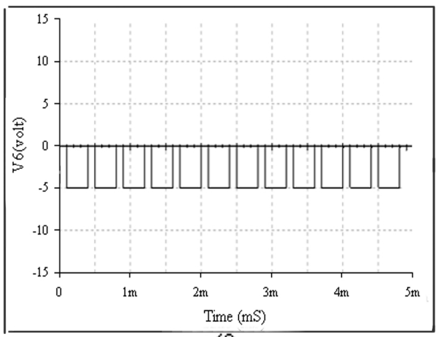
(c)



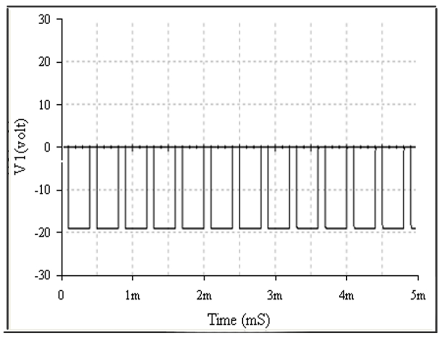
(d)



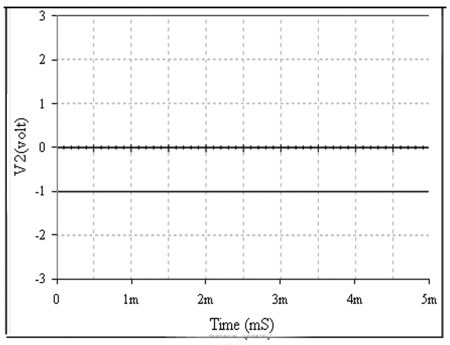
(e)



(f)



(g)



(h)

Figure(16) Simulation Output Voltages Of (A) V1 (B) V2 (C) V3 (D) V4 (E) V5 (F) V6.

4. Conclusions

In this paper, PSO algorithm achieved some merits as follows: Firstly, with parallel search strategy, it can locate the global optimization consistently. Secondly, its velocity displacement search model is simple and easy to implement. Thirdly, few parameters should be considered and set up. Moreover, the information flow with single direction can absolutely speed up the convergence process. Finally, variable inertial weight can effectively guarantee the compromise between global search and local search.

The proposed training PSO algorithm was very useful for reducing the neuron circuitry by lessining the multiplication process. PSO learning algorithm was advanced over other learning algorithms in decreasing the number of neurons needed for minimizing the mean squre error to zero which means 100% accuracy. The drawback of the PSO training algorithm was its slowness espicially for large number of particles over other training algorithm, but this is a normal matter because PSO is a multi-starting points algorithm unlike other algorithm like Backpropagation, however even this slowness is not big essue because the training of the network will be totally outside the Switched Capacitor circuits.

One of the most important problems in the design process of the SC circuits was the optimization of the clock waveforms, so a proposed design of multi-phase clock generator which produced sequential and non-overlapping clock pulses is presented to solve this problem. The successful design of the computational systems is often predicated on the realization of fast multiplication in digital or analog hardware. A key design issue is the trade-off between speed, complexity and chip area.

With this in mind, a proposed design of 3-phase multiplier circuit is presented, which is suitable for synapse analog VLSI implementation of ANN because of its small silicon area and its low power consumption with high precision and large linearity range. A proposed 3-phase SC neuron is presented. Nonlinear circuits are proposed for implementing neural networks trained by PSO using multi-phase SC integrated circuits technique. These proposed circuits are fully simulated using EWB package. The simulation results show that the proposed circuits are very suitable for neural networks and the simulated circuits have succeeded to learn each applied pattern.

5. References

1. Srowing," Artificial Neural Network", February, 2009. http://en.wikipedia.org/wiki/Artificial_neural_network.
2. R.P.Lippmann, "Introduction to Computing with Neural Nets", IEEE ASSP Magazine, April, 2005.
3. K.Meier,"Analog Neural Networks", Kirchhoff Institute for Physics, August, 2005.
4. B. J.Maundy and E. I. El-masry, "A Self-Organizing Switched-Capacitor Neural Network", IEEE Transactions on Circuits and Systems, Vol.38, No.12, December, 1991.
5. O. Changyue, Chen Peng and X. Yizhong, "Study of Switched Capacitor Multiplier", International Conference on Circuits and Systems, June, 1991.
6. Hanan A.R. Akkar, " Design of Analog Circuit Simulators for Artificial Neural Networks", Ph. D. thesis, University of Technology, October, 1998.
7. A. P. Engelbrecht, " Computational Intelligence: An Introduction", John Wiley & Sons Ltd, ISBN: 978-0-470-03561-0, South Aferica, 2007.
8. J. Kennedy and R. Eberhart, " Particle Swarm Optimization", IEEE International Conference on Neural Networks, Australia, PP. 1942-1948, 1995.
9. M. Michael Liu,"Demystifying Switched-Capacitor Circuits", Butterworth-Heinemann, May, 2006.
10. J. M. Zurada," Introduction to Artificial Neural Systems ", Jaico Publishing House, 1996.
11. L. Wang, X. Wang, J. Fu and L. Zhen, "A Novel Probability Binary Partical Swarm Optimization Algorithm and its Application", Academy publisher, Journal of software, China, Vol. 3, No. 9, December, 2008.
12. X. Xie, W. Zhang and Z. Yang, "Adaptive Particle Swarm Optimization on Individual Level", IEEE International Conference on Signal Processing (ICSP), China, PP. 1215-1218, 2002.
13. G. Kendall and Y. Su, " A Particle Swarm Optimization Approach in the Construction of Optimal Risky Portfolios", Proceedings of a 23rd IASTED International Multi-Conference Artificial Inteligence and Applications, Australia, PP. 140-145, February, 2005.
14. K. Chandramouli and E.Izquierdo, "Image Classification using Chaotic Particle Swarm Optimization", IEEE. ICIP, PP.3001-3004, 2006.

15. V.G. Gudise and G.K. Venayagamoorthy, “Comparison of Particle Swarm Optimization and Backpropagation as Training Algorithms for Neural Networks”, IEEE Symposium on Swarm Intelligence, USA, PP. 110-117, 2003.

16. F. Bergh and A. Engelbrecht, “ Cooperative Learning in Neural Networks using Particle Swarm Optimization”, SACJ/SART, 2000.

17. W. Zha, G. K. Venayagamoorthy, “Neural Networks Based Non Uniform Scalar Quantizer Design with Particle Swarm Optimization”, IEEE Transactions, June, 2005.

18. J. Zhang, J. Zhang, T. Lok and M. Lyu, “ A hybrid Particle Swarm Optimization Back-Propagation Algorithm for Feedforward Neural Network Training”, Applied Mathematics and Computation185, PP. 1026-1037, 2007.

19. A. Cheyad," Training of Artificial Neural Networks by using PSO", M.Sc. Thesis University of Technology, Augusts, 2009.

20. S. Monther," Design of Switched Capacitor Circuits to Implement ANN", M.Sc. Thesis University of Technology, June, 2007.