

## Advanced Logistic Belief Neural Network Algorithm for Robot Arm Control

Mbaitiga Zacharie  
Department of Media Information Engineering,  
Okinawa National College of Technology,  
905 Henoko, Nago, 905-2192, Okinawa, Japan

---

**Abstract: Problem statement:** This study discusses the implementation of advanced logistic belief Neural Network for robot arms control. **Approach:** Given the desired trajectory of the end-effectors in space, the logistic function is used to compute the conditional probability of the neurons being active in response to its induced field. The computations of conditional probabilities are performed under two different null conditions. (1) for all vectors not belonging to the parent of element node  $i$  and node  $j$ . (2) for node  $i$  greater than node  $j$ , which follows from the fact that the network is acyclic. **Results:** The test results proved the merit of the proposed method due to the fact that the robot arms move in the expected desired trajectory position within the allocated time set for each action. **Conclusion/Recommendation:** Our future work will be to improve this method for its use in the industrial robot arms.

---

**Key words:** Robot arm control, neural network, logistic belief

---

### INTRODUCTION

Artificial Neural Network (ANN) has emerged over the years and has made remarkable contribution to the advancement of various fields of endeavor and can be defined as a distributed computing system composed of a number of individual processing element operation largely in parallel. Interconnected according to same specific architecture and having the capability to self-modify connection strengths and processing element parameters. It can function at all to do useful things by being incorporated into systems containing more or less conventional elements so that they can solve real world problems economically. Rabelo and Avula (1991) used two different artificial neural networks systems associated with the prototype of a scheme which uses the integration of artificial neural networks and knowledge-based systems for motion control of a 2D arm robot. The system involved a plan generated by the higher order element which includes the kind of desired trajectory to follow. Velagic *et al.* (2010) introduced a recurrent neural network for controlling the mobile robot with nonholonomic constraints. The network is trained online uses back propagation optimization algorithm with an adaptive learning rate which is very effective for real-time control requirements. Killy *et al.* (1996) presented a Neuro-fuzzy control for planning the trajectory of a three link robot arm in the presence of an obstacle. The robot arm operates in two dimensions in an environment containing a randomly placed obstacle

and goal of which the controller should determine a series of joint angle that move the end-effector from a given starting position to a desired final position without colliding with the obstacle. Massera *et al.* (2006) proposed an evolutionary technique for developing a neural network based controller for anthropomorphic robot arm with four Degrees of Freedom (4-DOF). In the authors proposed method, the neural controller consists of a feed forward neural network with three sensory neurons directly connected to 4 motor neurons of which are updated on the basis of a standard logistic function. The genotype of evolving individuals encodes the connection weights of the neural controller. So where the neural network does come from?

The Artificial Neural Network is inspired by Human brain (highly information-processing system) that performs the formidable task of storing a continuous flood of sensory information received from the environment. From the deluge of trivia, it must extract vital information, act upon it and files it away in long memory in which large numbers of cells that individual functions faster, perfectly and collectively performs tasks that even the largest computer at our disposal today cannot be able to match. The following is the brief description of the human brain the neural network drives. The human nervous system can be viewed as a three-stage system shown in the block diagram of Fig. 1.

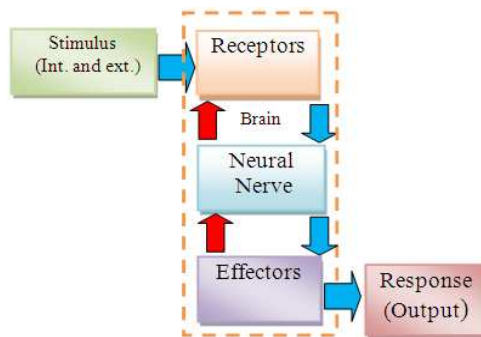


Fig. 1: Overview of human nervous system representation

Central to the system is the brain represented by the neural nerve, which continually receives information, perceives it and makes appropriate decisions. Two sets of arrows are shown in the figure. Those pointing from stimulus to response (blue) indicate the forward transmission of the information-bearing signals through the system. Those pointing from effectors to receptors (red) indicate the presence of feedback in the system. The receptors convert stimuli from the human body (internal stimulus) or external environmental (external stimulus) into electrical impulses that convey information to the neural nerve (brain). The effectors convert electrical impulses generated by the neural network in discernible response as a system outputs. But the various Artificial Neural Networks that are currently in fashion differ in their ability to make accurate distinctions, their ability to learn new things without erasing the previous information that has been learned and they're efficient (Tavoosi *et al.*, 2011). In this study the implementation of advanced logistic belief Neural Network for robot arms control is discussed. Given the desired trajectory of the end-effector in space, the logistic function is used to compute the conditional probability of the neurons being active in response to its induced field.

## MATERIALS AND METHODS

**Literature review:** The neural networks are one of the most control system used in the control engineering due to its efficiency and it is about to take the first place over the so far well-known control system, such as PID control, fuzzy logic, fuzzy control or genetic algorithm. The number of control topics involving neural network is so huge that someone can spend one week even one month reading books, articles without finding the best control method that fits his system. In spite of the number of the articles in the open literature, there has been so far no attempt to apply the belief logistic network to control the robot arm. But as far as it concerns robot arm control using neural networks, Raffaele and Stefano (Bianco and Nolfi, 2004) have

chosen the evolving neural network controller for a robotic arm that grasp objects on the basis of tactile sensors. By their method each individual in the neural controller is controlled by the fully connected neural network with 15 sensory neurons and 9 motor neurons. Neurons are updated by the logistic function. The sensory neurons encode the angular position of the nine Degrees of Freedom (9-DoF) of the joints and the state of the six contact sensors located in the arm and in the fingers. The motor neurons control actuators of 9 corresponding joints. The output of the neurons is normalized within the range of the movement of the corresponding joint. Oyama *et al.* (2005) applied the inverse kinematics learning for robotic arms by the modular neural network system. Their proposed method consists of a number of experts, with each expert approximates a continuous region of the inverse function. The forward model in the system approximates the forward kinematics of the robot arm and the performance index of each expert is the predicted end-effector. Position and orientation error are calculated by using the forward model. The expert selector chooses one approximate expert by using the expected performance of the experts. At this stage the system can learn a precise inverse kinematics model of the robot arm with equal or more degrees of freedom than that of its end-effector. However there are still some robot arm with few degrees of freedom that the system cannot lean at the present stage and to overcome this problem they adopted a modified Gauss-Newton method for finding the least-squares solution. Bouganis and Shanahan (2010) introduce a training spiking neural network to control a 4-DoF robotic arm based on spike training-dependent plasticity. The proposed neural network consists of spiking neurons which are organized into seven input layers and four outputs which used a population of 1200 neurons in each input layer and a population of 800 neurons for each output layer. Four of the input layers encode the information that is given by proprioception and the firing pattern at each one of them indicates the angle of the respective joint. The four joints of interest are located at the shoulder (roll, pitch and yaw) and the elbow of the arm. The network encodes these angles after discretizing them into bins with five degree resolution. The remaining three input layers represent the spatial direction that the end-effector should move at the next time step, with each layer encoding the projection of the 3D directional vector to one of the world axes.

**The target robot system:** The robot used in this experiment shown in Fig. 2 is a prestige robot with a rugged wheeled Wi-Fi equipped with two gripping arms that optionally provide the robot with one wrist-mounted Complementary Metal-Oxide Semiconductor (CMOS) camera installed on its right arm.

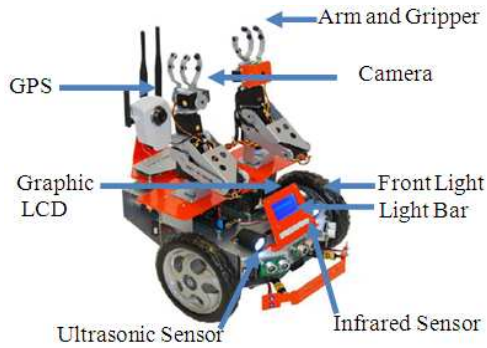


Fig. 2: Multi-DOF gripping arm robot

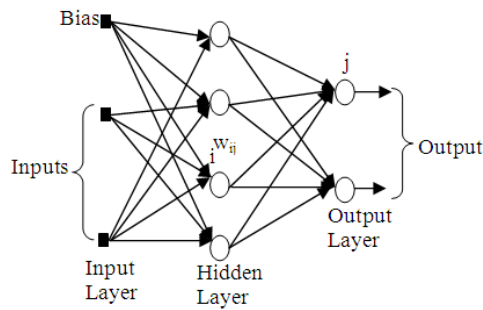


Fig. 3: Direct logistic belief neural network

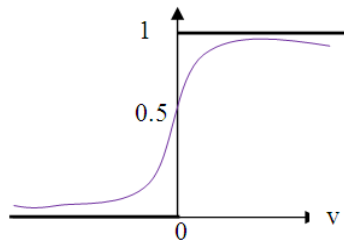


Fig. 4: Sigmoid function  $f(v)$

Combining mobility and a new ability to grasp and manipulate, the robot offers users broad versatility in its application. The wheels-based platform consists of 12V DC motors with integrated 800 counts per cycle optical encoder, yielding a top speed of  $0.75\text{ms}^{-1}$ . The robot is light as it weighs only 4kg with a capability to carry a maximum payload of 15 kg. Concerning the sensor types, the robot comes with ultrasonic range sensors and infrared range sensors including two-way audio capability. These range sensors are for environment detection and collision avoidance, while the two-way audio is for communication between the robot and the user. The collision avoidance and the sensing may not be corrected by information acquired from the only vision, therefore three ultrasonic sensors, with one located at the middle front bottom, one in the left front bottom hand side and one on the right front bottom

hand side of the robot are integrated. The middle front sensor is used for detecting obstacle, while those on each side are used for assisting the six infrared sensors of which one is located at the middle front upper just above the middle front bottom of the ultrasonic sensor, one in the upper front left, one in the upper front right, one in the rear middle, one in the rear left and one in the rear right of the robot respectively. Two quadrature encoders are also integrated in the robot, where the left one uses the channel-1 and the right one uses the channel-2. DC servomotor is used to steer and driving of the prestige robot.

**Methodology:** Robot arms, also known as robot manipulator are mechanical structures designed to carry loads from one point to another. They are commonly used in the industry in which the majors' applications are welding spray printing, palletizing and assembly. To perform such action a robust control system is required and one of the most robust controls is the neural network. Fig. 3 shows the multilayer network used.

The neural network can be used to model the input-output behavior of general and specific classes of the system, without detailed mathematical models. In other words, the neural network is universal approximators to the behavior of systems. This suggests that they can be used to approximate inverse kinematics without actually performing the matrix inversions associated with inverse kinematics. The inverse kinematics are defined as the computation of the joint coordinates that result in a desired special position of the end-effector (Zacharie, 2011). Given a desired trajectory of the end-effector in space, the goal of this control system is to compute the torques and forces at the joints needed to move the manipulator (arm) through this trajectory. The acyclic property of the neural network makes it easy to perform probabilistic computations as the neural network uses logistic function. Equation 1 is the logistic function used to compute the conditional probability of the neurons being active in response to its induced field:

$$P(X_i = x | X_i = x_i \{k_{i=1, i \neq j}\}) = \phi\left(\frac{x}{T} \sum_{i \neq j} w_{ji} x_i\right) \quad (1)$$

where,  $\phi(\cdot)$  is the logistic function Eq. 2:

$$\phi(\mu) = \frac{1}{1 + \exp(-\mu)} \quad (2)$$

Where:

x varies between 1 and -1.

$$\mu = \frac{x}{T} \sum_{i \neq j} w_{ji} x_i \text{ (The whole argument)}$$

k vary between  $-\infty$  and  $+\infty$  as illustrated in Fig. 4.

By differentiating Eq. 1, the need for the partition function Z has been eliminated. First, we have chosen the vector  $V$  consisting of the variable  $v_{a_1}, v_{a_2}, v_{a_3}, \dots, v_{a_n}$  that denote the behavior of the logistic belief network comprising N stochastic neurons. The parents of element of node i and node j in the vector V are defined by Eq. 3:

$$Pa(v_i) \leq \{v_{a_1}, v_{a_2}, v_{a_3}, \dots, v_{a_{j-1}}\} \quad (3)$$

where,  $\{v_{a_1}, v_{a_2}, v_{a_3}, \dots, v_{a_j}\}$  is the smallest subset of the vector that excites nodes for which the conditional probability is Eq. 4:

$$P(X_j = x_j / Pa(v_i)) \quad (4)$$

As there is a direct link from node i and node j. An important of the logistic belief network is its ability to exhibit the conditional dependencies of the underlying probabilistic model to the input data, with the probability that the neuron is active being defined by the logistic function where  $w_{ji}$  is the synaptic weight from neuron i to neuron j. The computations of conditional probabilities are performed under two different null conditions:

- $w_{ji} = 0$  for all  $v_i$  not belonging to  $pa(v_i)$ , which follows from the definition of a parent
- $w_{ji} = 0$  for node  $i > \text{node } j$ , which follows from the fact that the network is acyclic

As the system has five degrees of freedom, we have computed five desired joint positions for each instant of time, where the increment between these instants depends on the sampling rate of the arm movement. The sampling rate must be fast enough to compare to the highest frequencies present to avoid aliasing. To do so we used the sampling rate of five to ten times higher frequencies present in the system.

The sequence of operation of this system is as follows:

- The initial state of the system is known and the desired endpoint trajectory in Cartesian coordinates (KT) is computed and sampled at the desired update frequency (1/T). The index K is a counter for the sampling times
- The Computation unit computes the corresponding desired joint angle  $\theta(kT)$
- The desired joint angles are used as inputs for each of the five actuator servos that compute the actual joint position at each interval

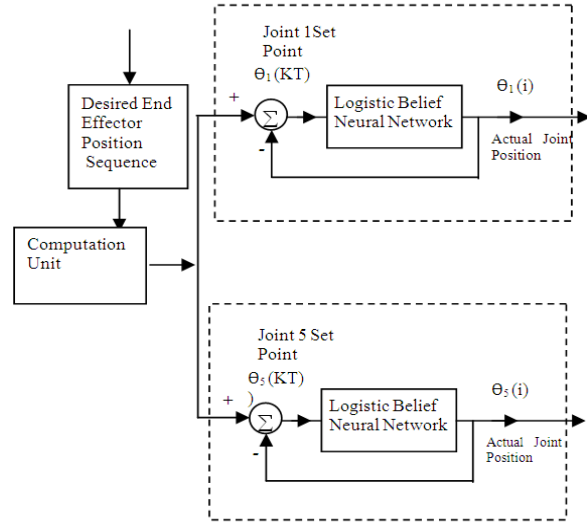


Fig. 5: Logistic Belief Neural Net Control System

The architecture of the joint position control system is shown in Fig. 5.

## RESULTS

**Experimental results:** The proposed Logistic Belief Neural Network (LBNN) has been tested on a robot with a rugged wheeled WI-fi equipped with two gripped arms that optimally provide the robot with one wrist-mounted complementary metal-oxide semiconductor camera installed on its right arm. Each arm has five degrees of freedom and the LBNN used is a multilayer network, which contains an input-layer, output-layer and a hidden-layer unit. The input layers are presented to the network and the network outputs are compared to the desired actual or outputs corresponding to the inputs. To determine how to allocate the error not just to the weights in the output layer, but to those in the hidden layer as well, we used the update rule back-propagation algorithm:

$$\Delta w_{ij} = \alpha z_i = (y_d - y_i) T' = \alpha z_j e_j T' \quad (5)$$

Where:

$z_i$  = is the output of the jth units in the hidden layer.

$\alpha$  = is the learning rate.

$T'$  = is the derivative of the function T(x).

Figure 6 and 7 are the plot results of each arm. During the training stage, a set of configurations of the arms is selected as initial position where the generator sends motor commands to the range of  $[-60^\circ, 60^\circ]$  at each joint and their effect on the special position of the end-effectors are computed based on the conditional probability.

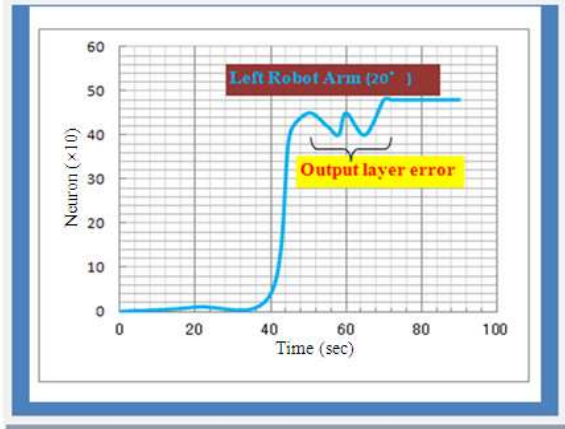


Fig. 6: Robot left arm movement at 20°

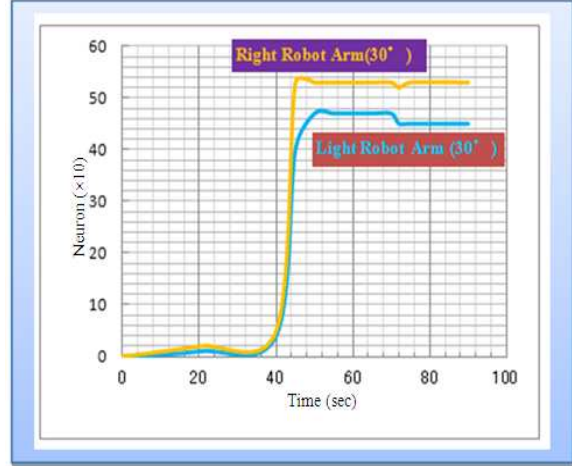


Fig. 9: Robot left and right arm movement at 30°

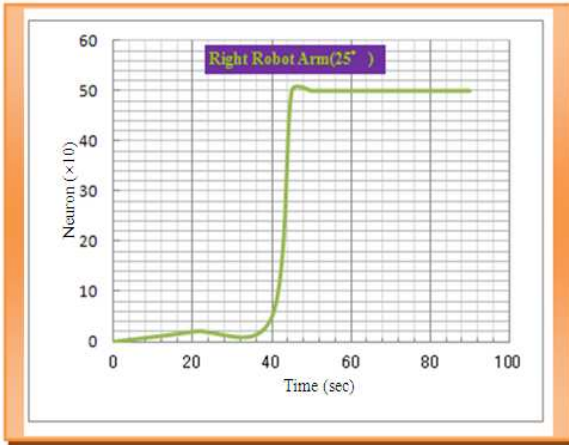


Fig. 7: Robot right arm movement at 25°

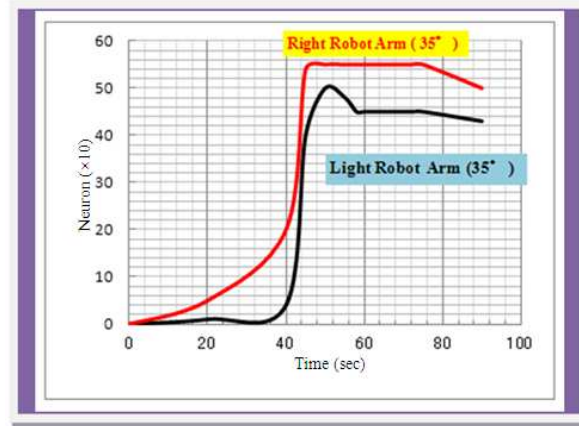


Fig. 10: Robot left and right arm movement at 35°

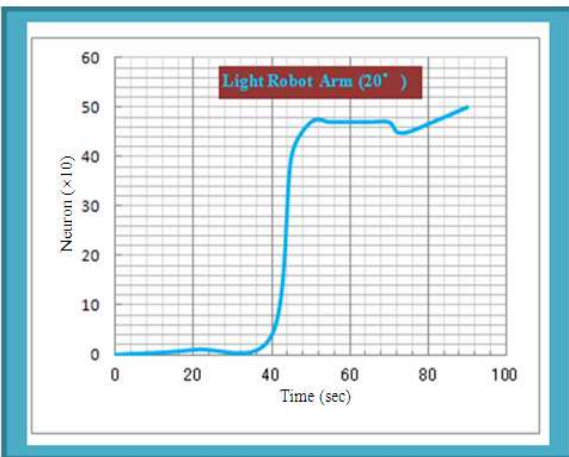


Fig. 8: Robot left arm movement at 20° without error

Two objects were placed on the left and the right side of each robot arm at 20° for left robot arm and 25° for the right arm. On the command go, both arms move at the same time towards each object then returned back to the initial position as expected and the data results were analyzed. The results show a small propagation error from the output layer of the left robot arm (Fig. 6). To propagate the error from the output layer back to weights in the hidden layer we used the chain rule differentiation by substituting the weighted inputs sums for the terms  $z_i$  of Eq. 5.

The resulting expression is Eq. 6:

$$\Delta^2 w_{ij} = \alpha x_i T'_j \sum_{i=1}^n (w_{ji} e_i T'_i) \quad (6)$$

And the test has been repeated three times at 20° of which the test result is shown in Fig. 8 where the error in

the output layer has been eliminated. The test has been repeated for 30°, 35°, 40°, 45°, 50°, 55°, -15°, -20°, -30° and -33° respectively of which the Fig. 9 and 10 illustrated the right and left movement results at 30° and 35°.

## DISCUSSION

The use of the Logistic Belief Neural Network has given us a new powerful control method to be used in many systems. Not only to control all mobile robot systems including speed, arms or industrial robot arms but also in other fields such as communication, network security control. What makes the LBNN powerful is that with the probability condition under two null conditions, it is easy to define and find the input that fits the system. But at this stage it is too early to claim the success of the present system as there are still some efforts to do to improve the way the propagation of the error from the output layer back to weights in the hidden layer without using the chain rule differentiation. Although the result we obtained prove the robustness of the method. To compare the present method to other existing methods used in robotic arm control, we have looked at (Vaezi and Nekoule, 2011) that proposed the adaptive control of a robot arm based neural network which is a combination of two controllers. The adaptive control and neural network and it were based on nonlinear autoregressive moving average where the simulation result was not convincing compared to our method. In the simulation test there was not the arm reaction to the go command but only the neural network training output, while in (Simmons and Demiris, 2005) the result is somewhat good and not so far from our results.

## CONCLUSION

In this study, we have presented the logistic belief neural network algorithm to control five degrees of freedom of the robot arm. The proposed algorithm has been tested on a robot with a rugged wheeled WI-fi equipped with two gripped arms that optimally provide the robot with one wrist-mounted complementary metal-oxide semi-conductor camera installed on its arm. The neural network consists of 600 and the acyclic property of the neural network makes it easy to perform probabilistic computations. The desired robot arm joint angles are used as inputs to the each of the five actuator servos that compute the actual joint position at each interval. The experimental evaluation test has been performed 12 times and validated the proposed algorithm. Our future work is to add extra neurons to the hidden layer unit and make the arm grasp objects at a specific location.

## REFERENCES

- Bianco, R. and S. Nolfi, 2004. Evolving the neural controller for a robotic arm able to grasp objects on the basis of tactile sensors. *Adaptive Behav.*, 12: 37-45. DOI: 10.1177/105971230401200102
- Bouganis, A. and M. Shanahan, 2010. Training a spiking neural network to control a 4-dof robotic arm based on spike timing-dependent plasticity. *Proceedings of the IEEE World Congress on computational Intelligence*, Jul. 18-23, CCIB, Barcelona, Spain, pp: 4104-4111.
- Killy, W.E., R. Chaloo, R. Mclauchlan and S.I. Omar. 1996. Neuro-fuzzy control of a robotic arm. *Proceedings of the Artificial Neural Networks in Engineering Conference*, Nov. 10-13, Louis, pp. 837-842.
- Massera, G., A. Cangelosi and S. Nolfi, 2006. Developing a reaching behaviour in an simulated anthropomorphic robotic arm through an evolutionary technique. *Frontiers Neuror.*
- Zacharie, M., 2011. Sensor fusion based discrete Kalman filter for outdoor robot navigation. *J. Worl Academy Sci., Eng. Technol.*, 80: 349-353.
- Oyama, E, T. Maeda, J.Q. Gan, E.M. Rosales and K.F. MacDorman, 2005. Inverse kinematics learning for robotic arms with fewer degrees of freedom by modular neural network systems. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Aug. 2-6, IEEE Xplore Press, Japan, pp: 1791-1798. DOI: 10.1109/IROS.2005.1545084
- Rabelo, L.C. and X.J.R. Avula, 1991. Intelligent Control of a robotic arm using hierarchical neural network systems. *Proceedings of the IJCNN-91-Seattle International Joint Conference on Neural Networks*, Jul. 8-14, IEEE Xplore Press, Rolla, MO, pp: 747-751. DOI: 10.1109/IJCNN.1991.155428
- Simmons, G. and Y. Demiris, 2005. Optimal robot arm control using the minimum variance model. *J. Robotics Syst.* 22: 677-690. DOI: 10.1002/rob.20092
- Tavoosi, J., M. Alaei and B. Jahan, 2011. Neuro-fuzzy for position control of robot arm. *Proceedings of the 5th SASTech, Khavaran Higher-education Institute*, May 12-14, Mashhad, Iran.
- Vaezi, M. and M.A. Nekoule, 2011. Adaptive control of a robotic arm using neural networks based approach. *Int. J. Robotics Autom.*, 1: 87-99.
- Velagic, J., N. Osmic and B. Lacevic, 2010. Design of neural network mobile robot motion controller. *J. New Trends Technol.*, DOI: 10.5772/7584