Proceedings of the 6th WSEAS Int. Conf. on NEURAL NETWORKS, Lisbon, Portugal, June 16-18, 2005 (pp211-215)

# Neural network path planning applied to PUMA560 robot arm

SAID AOUGHELLANET, TAYEB MOHAMMEDI, YOUCEF BOUTERFA
Departement of Electronic Engineering
University of Batna
Rue Boukhlouf Med El-Hadi, 05000 Batna
ALGERIA

*Abstract:* In this paper, we present a trajectory planning method using a recurrent neural network with strictly limited interconnections. The trajectory obtained is then applied to PUMA560 robot arm which moves in an environment with obstacles. Each neuron of the network is connected only to the nearer neighboring neurons. This makes possible to reduce considerably the number of interconnections and to thus decrease the complexity of the network. The neural network evolves from an initial state to a final state, thus delivering an optimal trajectory that the robot must follow to avoid the obstacles and to reach the desired configuration. In order to avoid local minima, we use an adaptive parameter in the neural network activity equation. The most significant feature in this method is that it can be established to have a real-time trajectory planning in a dynamic environment.

*Key-Words:* Recurrent neural network, Real-Time-Cost, Robot navigation, Collision avoidance.

## 1 Introduction

Motion planning with obstacle avoidance is a very important issue in robotics. There are a lot of studies on robot motion planning using various approaches. Most of the previous models [1] use global methods to search the possible paths in the workspace. Ong and Gilbert [2] proposed a path planning model using penetration growth distance, which searches over the collision paths instead of the free workspace. These models deal with static environment only and are computationally complicated when in a complex environment. Several neural network models were proposed for real-time robot motion planning through learning, e.g. Muniz, Zalama, Gaudiano and Lopez-Coronado [3] proposed a neural network model for dynamic navigation of a mobile robot with obstacle avoidance through unsupervised learning; Fujii, Arai, Asama and Endo [4] proposed a multilayered model for collision-free motion planning through reinforcement learning. However, the planned robot motions using learning based approaches are not optimal, particularly at the initial learning phase. Glasius, Komoda and Gielen [5] proposed a Hopfield-type neural network model for dynamic trajectory formation without learning, but it requires the robot to have faster dynamics than the environment.

In this paper we present a structure of recurrent neural network, which is applied to real-time trajectory planning in a dynamic environment. With the aim of decreasing the computing time, the structure of the neural network is selected in such that it reduces the possible maximum number of interconnections between the neurons. The trajectory planning is obtained from the progressive evolution of the neural network from an initial state towards a final state. This procedure is done without training.

## 2 Neural model

The robot configuration is determined by the articular positions $\{\theta_1, \theta_2, \theta_3, \dots \theta_k\}$ where $\theta_i$ is the articular position of the $i^{th}$ articulation and K the number of freedom degrees. The robot articular space is discretized in cells with a step of discretization $\delta\theta_i$ given by :

$$\delta\theta_i = (\theta_{max} - \theta_{min})/g_i$$

where $g_i$ is the discretization order of the $i^{th}$ articulation. $\theta_{max}$ and $\theta_{min}$ are the maximum and minimal acceptable values of $\theta_i$.

Thus for a robot with K freedom degrees the total number of cells N is given by:

$$N = \prod_{i=1}^{K} g_i$$

The problem arises in the following way, the robot is initially in a starting configuration which corresponds to the cell $C_0$ and wants to reach the cell $C_f$ which corresponds to the target configuration. The trajectory followed between the starting cell $C_0$ and the target cell $C_f$ is defined by cells $\{C_0, C_1\ C_2, \dots C_f\}$.

Proceedings of the 6th WSEAS Int. Conf. on NEURAL NETWORKS, Lisbon, Portugal, June 16-18, 2005 (pp211-215)

To this discretized articular space we assign a locally recurrent neural network, where each neuron $n_i$ is the image of a cell $C_i$. The number of neurons of the network is thus equal to the number N of cells of the robot articular space. Each neuron $n_i$ is connected only to the immediate nearer neurons, which represent the cells in direct contact with cell $C_i$. In a 3-D space the number of neighboring neurons is 26, as shown on Figure 1 (diagonal connections are not represented).

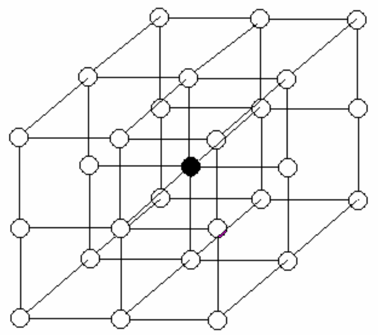

Fig.1 3-D neighboring interconnected neurons.

The weights $W_{ij}$ and $W_{ji}$ of the connections between neuron $n_i$ and neuron $n_j$ are equal to the inverse of Euclidean distance separating their images $C_i$ and $C_j$ in the articular space:

$$W_{ij} = W_{ji} = 1/|dij|$$

Where |dij| is the Euclidean distance between cell $C_i$ and cell $C_j$ in the articular space. In addition to the connections between neurons, each neuron $n_i$ receives a bias entry $I_i$ which informs it of the presence or not of a possible obstacle. If the neuron $n_i$ corresponds to a cell which contains an obstacle, its bias $I_i$= -E. Otherwise if the cell is free its bias is zero $I_i$=0. Only the neuron $n_f$ which corresponds to the target cell $C_f$ has a bias $I_f$=+E.

The initial state of the neural network is such that all outputs $x_i$ of neurons are zeros:

$$x_i=0 \quad \{i=1,2\ldots N\}$$

The network starts to evolve to a final state according to the following activity equation :

$$x_{i+1} = x_i - \mu_i.\{ax_i + (b - x_i)[\max(I_i,0) + \sum_{j=1}^{m} w_{ij} \max(x_j,0)] - (c + x_i)\max(-I_i,0)\} \quad (1)$$

Where a, b, and c are constants. m is the number of neighboring neurons of neuron $n_i$.

The manner of interconnecting neural network and the values set to the biases force the maximum activity to moves gradually from neuron $n_0$ image of initial cell $C_0$ through neighboring neurons and with each iteration until reaching the neuron $n_f$ which corresponds to the target cell $C_f$. The cells $\{C_0, C_1\ C_2,\ldots C_f\}$ which constitute the trajectory of the robot, are obtained by :

$$C_i \leftrightarrow \max(x_i\ , i=1,2,\ldots m) \quad (2)$$

If the next iteration corresponds to the same cell, the evolution of the network stops without achieving the goal. To avoid such a state we introduces an adaptive factor μ in the activity equation :

$$\mu_{i+1} = \mu_i + \mathrm{sgn}(\delta e)\frac{\delta.\mu}{abs(\delta e)+1} \quad (3)$$

Where sgn(μ)=1 if μ>0 and sgn(μ)= -1 if μ<0. δμ is constant. e is the mean quadratic error of articulation positions, and δe= $e_{i+1}$ - $e_i$.

This allows the neural network after a certain number of iterations to avoid this situation and to continue its evolution until reaching the final goal. It can be proved that equation (1) satisfies stability conditions [6]. The stability and convergence of this equation can be rigorously proved using the Lyapunov stability theory [7]. Therefore the proposed neural network system is stable.

## 3 Simulation Results

The trajectory planning previously developed is applied to PUMA560 robot arm. Knowing that the three last articulations concerning the movements of the end-effector are of reduced size, only the first three degrees of freedom are considered. Articular space is thus of three dimensions. The order of discretization $g_i$ is 64 for each articulation. Thus the number of neurons N is equal to (64x64x64)=262144 neurons. Therefore the steps of discretization of the various articulations are respectively: (5, 3.9, 3.28) degrees. Two obstacles are placed on the way of the robot. The first obstacle is a box open on two parallel faces. The second one is a plan placed at a minimal distance from the robot.

According to the simulation results, we notice that the second obstacle can be avoided by local planning methods when the plan is placed beyond a certain distance from the robot. On the other hand, the first obstacle is avoided only by the global or mixed planning methods.

The method developed in this paper proved its effectiveness of avoidance of the most complicated obstacles with a short computing time. We give an example on the Figure2 which represents the trajectories followed by the first three articulations in an environment without obstacles, then in an environment provided with the previous obstacles. In this example, the initial configuration corresponds to $[\theta_1\ \theta_2\ \theta_3]=[0\ 0\ 0]$ and the target configuration is $[\theta_1\ \theta_2\ \theta_3]= [170\ -45\ 80]$ in degrees.

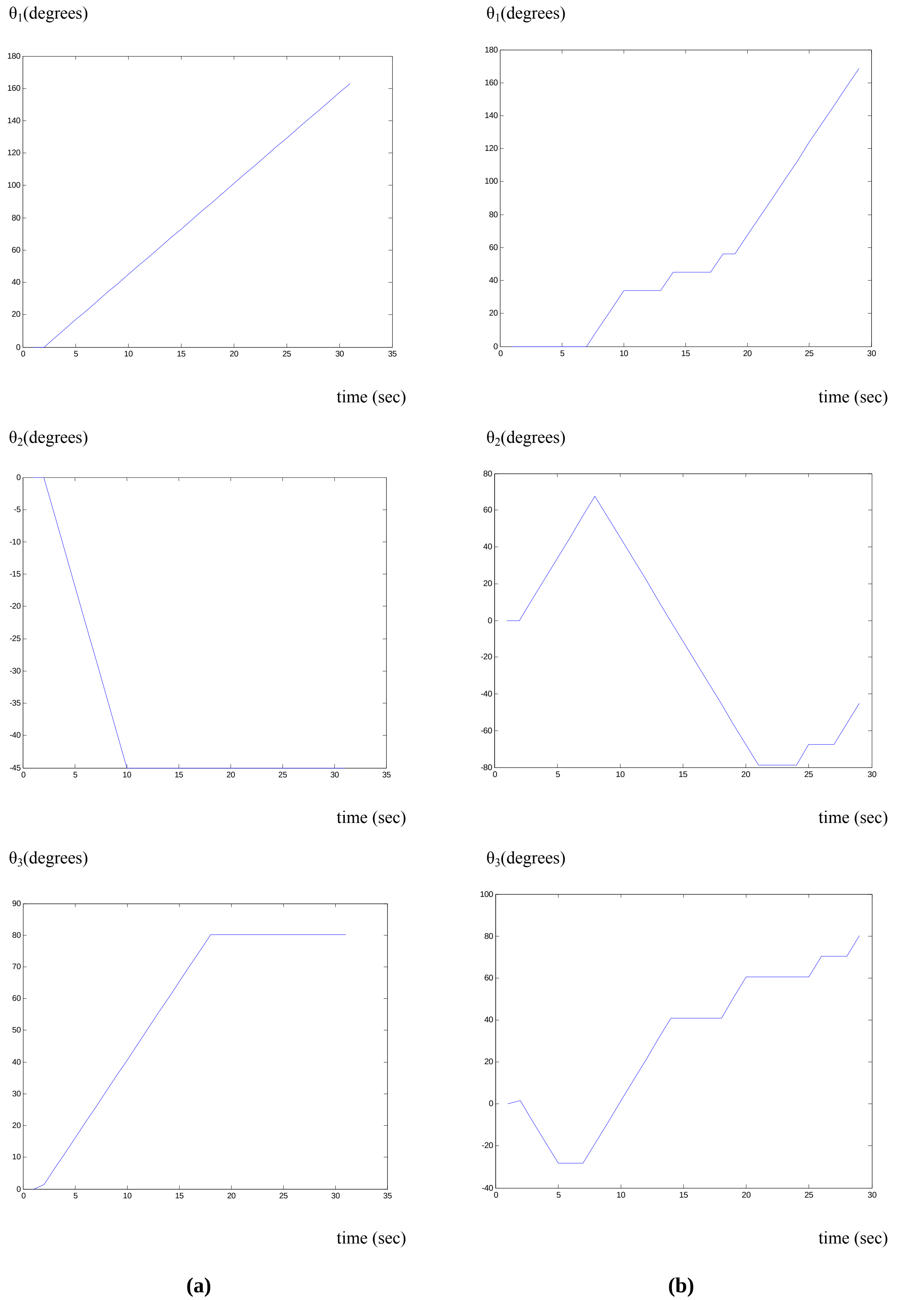


Fig.2 Trajectories: (a) Without obstacles (b) In the presence of obstacles.

The Figure3 represents the sequence of the movements of robot during its motion from the starting configuration (first image) to reach the final configuration (last image). This result is obtained for the following parameters values:
a=10, b=1, c=1, E=100, $\mu_1$=0.03, $\delta\mu$=0.001.

## 4 Discussion

In the proposed neural network structure, it could be established that a real-time trajectory planning, in a dynamic environment, using hardware or software implementation is possible. This method can also be applied to a mobile robot or a multi-robots system. The fact of introducing the adaptive factor $\mu$ into the neural network activity equation, allows avoiding local minima, but this can introduce instability of the network when its state approaches the final state. In order to avoid this problem it is required only to stop the neural network evolution process once the mean squared error of articular positions is smaller then the discretization step ($e<\max(\delta\theta_i)$ ).

## 5 Conclusion

In this paper a real-time trajectory planning method with obstacles avoidance is developed. The trajectory obtained is optimal in the sense of the shortest path. Simulation results are satisfactory and encouraging for further research work. The system stability is guaranteed by both qualitative analysis and the Lyapunov stability theory.

*References:*

[1] Zelinsky, A. (1994). Using path transforms to guide the search for findpath. 2nd International Journal of Robotics Research, 13 (4), 315-325.

[2] Ong,C.J., & Gilbert, E. G. (1998). Robot path planning with penetration grouth distance. Journal of Robotic Systems, 15 (2), 57-74.

[3] Muniz, .,Zalama, E.,Gaudlano, P & Lopez-Coronado,J. (1995). Neural controller for a mobile robot in a nonstationary environment. Proceedings of 2nd IFAC Conference on Intelligent Autonomous Vehicules. Helsinki, Finland (pp. 279-284).

[4] Fujii, T.Arai, Y.Asama, H.Endo, (1998), Multilayred reinforcement learning for complicated collision avoidance problems, Proceedings of IEEE International Conference on Robotics and Automation. Leuven, Belgium (pp. 2186-2188).

[5] Glasius, R.Komoda, S., & Gielen, S.C.A.M. (1995). Neural network dynamics for path planning and obstacle avoidance. Neural Network,8 (1), 125-133.

[6] Meng,M. & yang,X. (1998). A neural network approach to real-time trajectory generation. Proceedings of IEEE International Conference on Robotics and Automation. Leuven, Belgium (pp. 1725-1730).

[7] Grossberg, S. (1988). Non linear neural networks : principales, mechanisms, and architecture. Neural Networks, 1, 17-61.

[8] Hodkin, A. L., & Huxley, A.F. (1952). A quantitative description of membrane current and its application to conduction***

[9] Yang, X. & Meng, M. (1998). A Neural Network approach to real-time path planning with safety consideration. Proceedings of IEEE International Conference on systems, Man, and Cybernetics, San Diego, USA (pp.3412-3417).

[10] Yang, X. & Meng, M. (1999). Real-time motion planning of car-like robots. Proceedings of IEEE/RSJ, International Conference on Intelligent Robotic Systems, Kyungju. Korea (pp. 1298-1303).

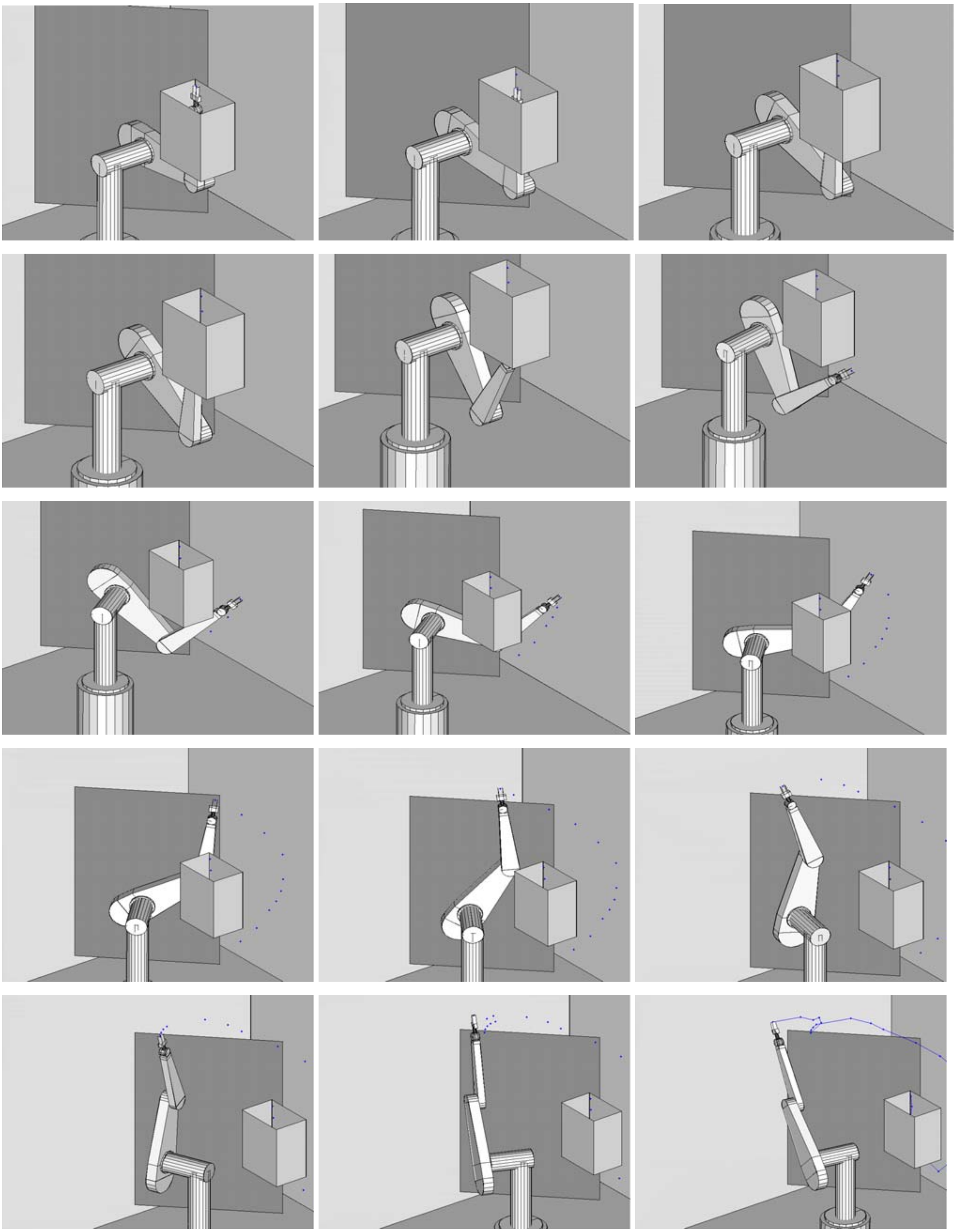

Fig.3 Movements sequence of PUMA560 arm.