IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE)
e-ISSN: 2278-1676,p-ISSN: 2320-3331, Volume 10, Issue 2 Ver. I (Mar – Apr. 2015), PP 81-88
www.iosrjournals.org

Redundant Three-link Robot Manipulator Local Approach

Ivan Isho Gorial
Mechatronics Engineering Branch, Control and Systems Engineering Department, University of Technology, Baghdad, Iraq

Abstract: *This paper proposed a fuzzy logic-based joint space path planning system of robot manipulator with 3-DOF joints. The proposed planning system was composed of several separate fuzzy units which control individually each manipulator joint. The Main inputs of the first fuzzy block were* $\Delta\theta_{1g}(i+$ *1, θ1 i, the main inputs for the second fuzzy block were Δθ2gi+1, θ2 i and Δθ3gi+1, θ3 i of the third fuzzy block. The outputs were*$\Delta\theta_1(i+1)$ *and* $\Delta\theta_2(i+1)$*and* $\Delta\theta_3(i+1)$*respectively. The objective was to move the arm from an initial configuration (start configuration) to a final configuration (goal configuration). Simulation results shows that the robot reached the goal configuration successfully.*

Keywords: *Joint space, path planning, fuzzy logic, robot manipulator, motion planning*.

I. Introduction

A robot manipulator is a movable chain of links interconnected by joints. One end is fixed to the ground, and a hand or end effector that can move freely in space is attached at the other end [1].

Most robotic manipulators are strong rigid with powerful motors, strong gearing systems, and very accurate models of the dynamic response [2].

Robotics is a relatively new field of modem technology that crosses traditional engineering boundaries. Understanding the complexity of robots and their applications requires knowledge of electrical engineering, mechanical engineering, industrial engineering, computer science, and mathematics. New disciplines of engineering, such as manufacturing engineering, applications engineering, and knowledge engineering, are beginning to emerge in order to solve more complex problem [3].

The accuracy and complexities of positions of robots may be better achieved by supplementing human capabilities with computer power in order to generate these complex trajectories and to control the robot manipulator, respectively [3].

The advantage of fuzzy logic not only fast response, low cost and good real-time ability also it is not necessary to know the exact model of the object or process to be controlled when apply the fuzzy logic control and it meet the real- time requirements for robot motion planning [4]. There are several robot motion planning methods such as artificial potential field method, configuration space method, and method based on the fuzzy logic. The artificial potentialfield method needs the accurate information of the obstacles, so it cannot be applied for moving objects or inaccurate information. The configuration space method maps the obstacles into the C-space. The computation burden is huge. It would fail to meet the real-time requirements for robot motion planning [5, 6, 7, 8].

Wei and Shimin [9] presented a paper on 3-D Path Planning using Neural networks for a Robot Manipulator. They reported that it is complex to find a good path when the robot is in a complex dynamic change condition. Algorithms for the device to perform path planning and trajectory prediction were described.

Kermiche et al [10] demonstrated fuzzy logic control of robot manipulator in the presence of fixed obstacle. They presented a solution for the problem of learning and controlling a 2R-plan robot manipulator in the presence of fixed obstacle. The objective was to move the arm from an initial position (source) to a final position (target) without collision. Potential field methods were rapidly gaining popularity in obstacle avoidance applications for mobile robots and manipulators. They propose an approach based on potential fields principle, and defined the target as an attractive pole (given as a vector directly calculated from the target position) and the obstacle as a repulsive pole (a vector derived by using fuzzy logic techniques). The linguistic rules, the linguistic variables and the membership functions were the parameters to be determined for the fuzzy controller conception. A learning method based on gradient descent for the self-tuning of these parameters was introduced. Thus, it was necessary to have an expert person for moving the arm manually. During this operation of teaching, the arm moves and memorizes the data (inputs and outputs). This operation was used to find the controller parameters in order to reach the desired outputs for given inputs.

Koker et al [11] presented a neural network based inverse kinematics solution of a robotic manipulator. Inverse kinematics problem is generally more complex for robotic manipulators. Many traditional solutions such as geometric, iterative and algebraic are inadequate if the joint structure of the manipulator is more complex. In that study, three-joint robotic manipulator simulation software, developed in their previous studies, is used. Firstly, they have generated many initial and final points in the work volume of the robotic manipulator by using cubic trajectory planning. Then, all of the angles according to the real-world coordinates (x, y, z) were recorded in a file named as training set of neural network. Lastly, they have used a designed neural network to solve the inverse kinematics problem. The designed neural network has given the correct angles according to the given (x, y, z) Cartesian coordinates. The online working feature of neural network made it very successful and popular in this solution.

Tejomurtula and Kak et al [12] presented inverse kinematics in robotics using neural networks. The inverse kinematics problem in robotics requires the determination of the joint angles for a desired position of the end-effector. For this under constrained and ill-conditioned problem we propose a solution based on structured neural networks that can be trained quickly. The proposed method yields multiple and precise solutions and it is suitable for real-time applications.

Fuzzy logic method simulates the human being's thinking, and analyses fuzzy information. Fuzzy control rules are concluded by fuzzy math method. It has a good real-time ability. It is not necessary to know the exact model of the object to be controlled. It can meet the real-time requirements for robot motion planning. Gerk and Hoyer applied the fuzzy method in multi-robot motion planning [13].Their method needs to program off-line beforehand, and cannot be used for unknown environments. Zavlangas applied the fuzzy method in the motion planning for a 3-DOF robot [14].

In this paper, a Joint space path planning using fuzzy logic approach is proposed for 3-DOF industrial robots operating. Effectiveness of the proposed approach is verified through simulations. This approach can meet the real-time requirements for robot motion planning.

II. Fuzzy Logic

Fuzzy set theory is ''a body of concepts and techniques that give a form of mathematical precision to human cognitive processes that are in many ways imprecise and ambiguous by the standards of classical mathematics''. In effect, this theory allows one to deal with fuzziness by grouping elements, which do not have clear boundaries, into different classes.

Fuzzy logic uses fuzzy set membership functions, whose value range from 0 to 1, and allows for capturing linguistic representations of knowledge. The impreciseness of knowledge is handled by the membership functions associated with each linguistic variable.

In a narrow sense, fuzzy logic is a logical system, which is a generalization of multivalued logic. Due to this reason, the fuzzy theory has been applied to various engineering problems that are too complex or ill-defined for the conventional mathematical analysis.

In a fuzzy system, knowledge is represented by rules associated with fuzzy variables. These rules along with the membership functions are processed through the so-called compositional rule of inference.

Unlike other reasoning processes where qualitative reasoning is used with pure linguistic rules, fuzzy inference logic involves numerical synthesis based on membership functions to form a fuzzy decision table.

Since the quantities synthesized in a fuzzy reasoning procedure are generally fuzzy, the final decision is also fuzzy. Fuzzy inferencing is, therefore, usually called approximate reasoning; that is, it matches process quantities with the rules in the rule base to perform fuzzy inferencing by using the compositional rule of inference, Figure 1 shows fuzzy rule based system.

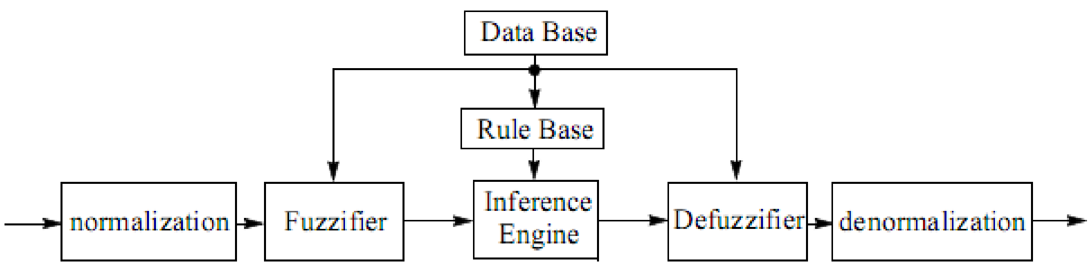


Figure 1. Fuzzy rule based system [16]

III. Proposed Method

The suggested fuzzy consists of three blocks for three-link robot planar as example. The first block for first link produces $\Delta\theta_1(i+1)$ depending on the first input $\Delta\theta_{1g}(i+1)$the error between the goal value and the current value of (θ_1) and on the current value $\theta_1(i)$which represents the second input to the first fuzzy block. Similarly the output of the second fuzzy block is $\Delta\theta_2(i+1)$ depending on the first input $\Delta\theta_{2g}(i+1)$the error between the goal value and the current value of θ_2 and on the current value $\theta_2(i)$which represents the second input to the second fuzzy block and the output of the third fuzzy block is $\Delta\theta_3(i+1)$ depending on the first input $\Delta\theta_{3g}(i+1)$the error between the goal value and the current value of $\theta_3(i)$ and on the current value $\theta_3(i)$which represents the second input to the third fuzzy block. Figure (2) shows system of the proposed motion planning for a three-link robot.

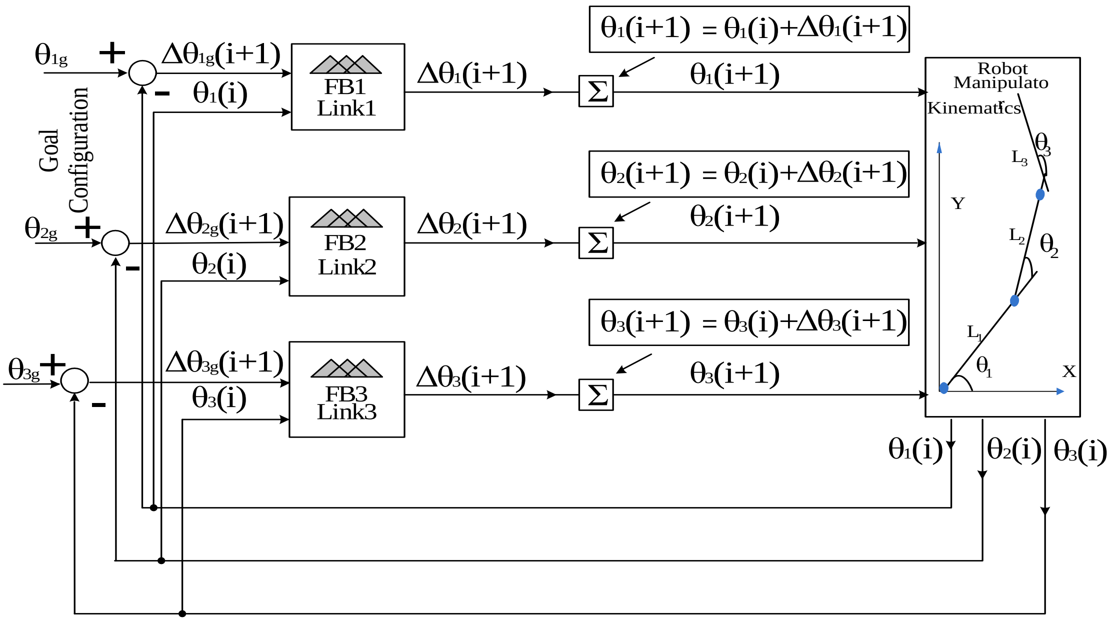


Figure (2). Fuzzy system for on-line planning of a three-link robot manipulator

Fuzzy membership function design for every fuzzy block is shown in the Figure. (3, 4,5).

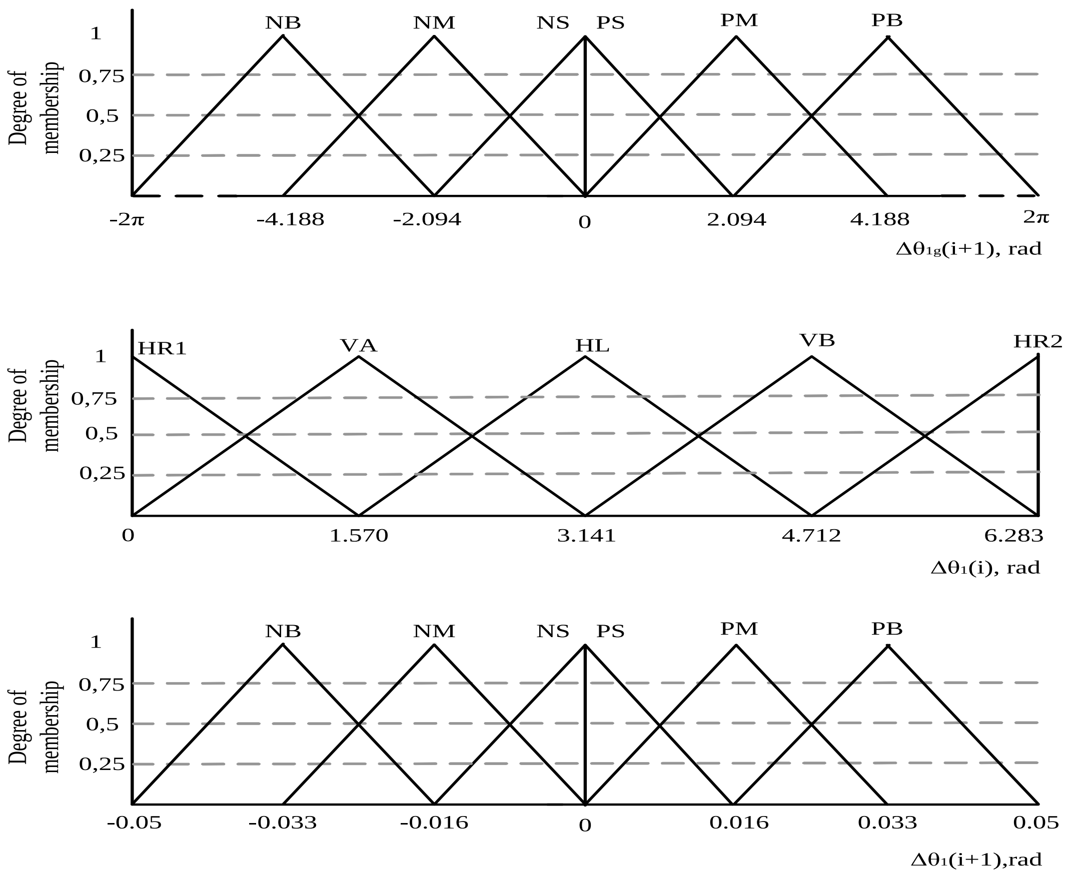


Figure 3. Fuzzy membership functions FB1 for the second robot link

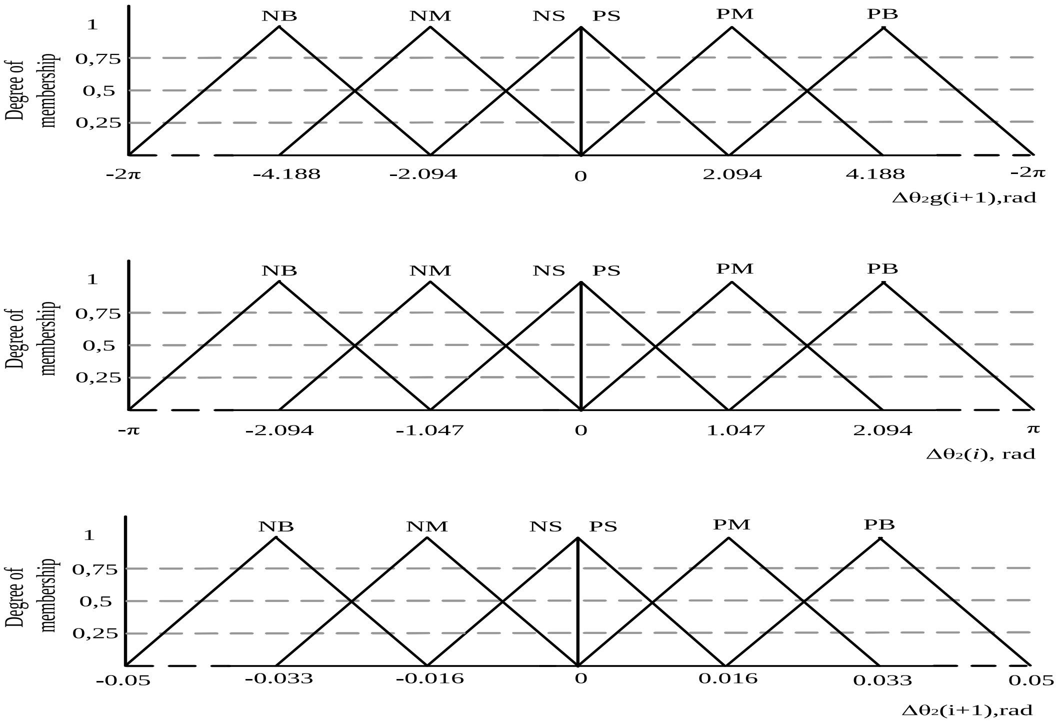


Figure 4. Fuzzy membership functions FB2 for the second robot link

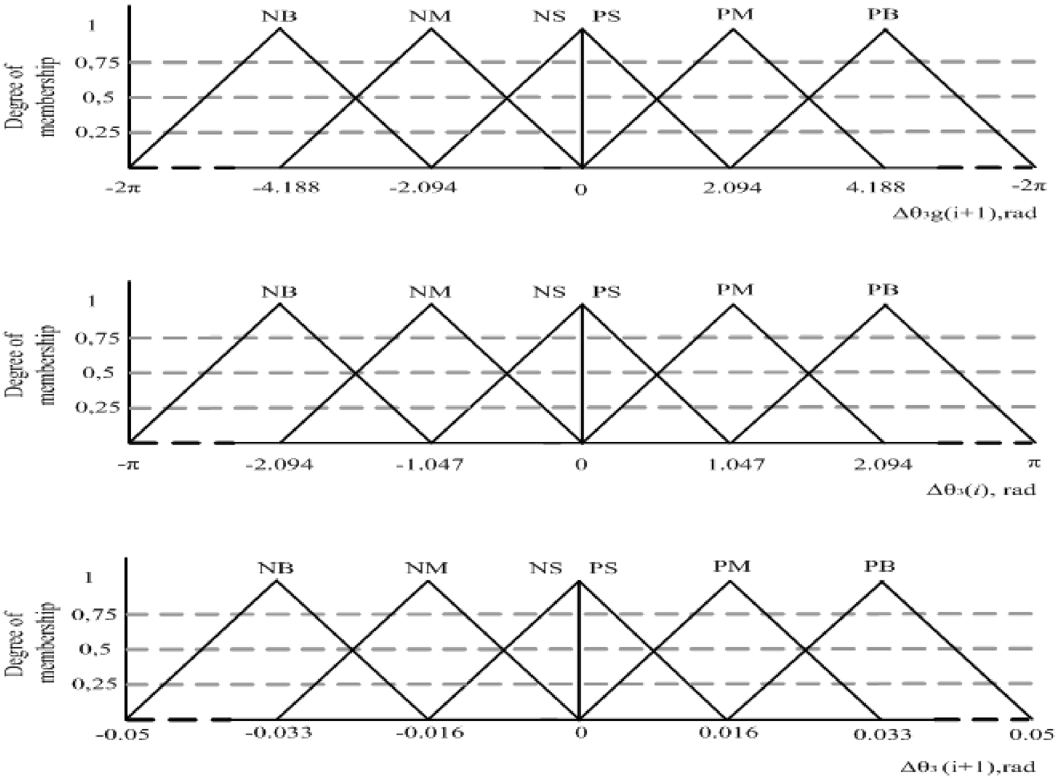


Figure 5. Fuzzy membership functions FB3 for the second robot link

The rules used for every fuzzy block to produce the proper output using the k Mamdani method. For example the rules of fuzzy block1 for first link will be as:

If ($\Delta\theta_{1g}$ is NS) and (θ_1 is HR1) then ($\Delta\theta_1$ is NB).

If ($\Delta\theta_{1g}$ is PS) and (θ_1 is HR1) then ($\Delta\theta_1$ is PS).

The most popular methods to calculate the fuzzy intersection (fuzzy-AND) operation according the fuzzy rules are the minimum and product operators. The final output of every fuzzy block can be computed using the center of gravity (COG) defuzzification method over all rules.

$$\Delta\theta^{crisp} = \frac{\sum_k b_k \int \mu(k)}{\sum_k \int \mu(k)}$$

Where b_k is the center of membership function of the consequent of rule (k). $\int \mu(k)$ denote the area under the membership function.

Where:

NB:	Negative Big	PB:	Positive Big
NM:	Negative Medium	HR1:	Horizontal Right1
NS:	Negative Small	VA:	Vertical Above
PS:	Positive Small	HL:	Horizontal Left
PM:	Positive Medium	VB:	Vertical Below
HR2	Horizontal Right2	FB:	Fuzzy Block

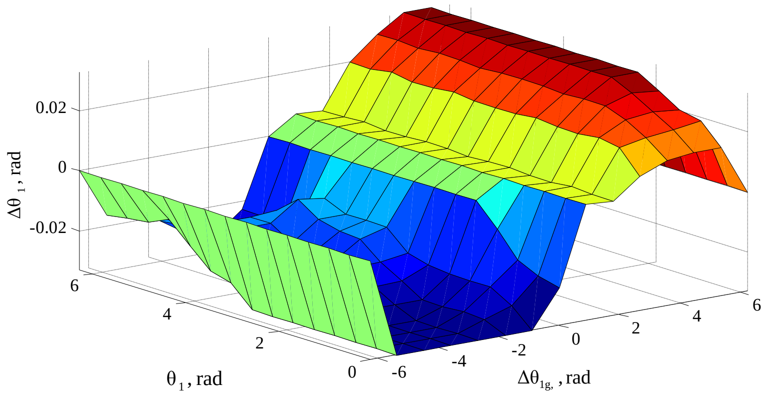
IV. Computer Modeling And Results

Computer modeling and simulation have been done to test the overall system of figure 2. The signals used to mode the robot motion were $(\Delta\theta_1(i+1), (\Delta\theta_2(i+1), (\Delta\theta_3(i+1))$. A three-link robot arm (1 meter, 1 meter, 1 meter) used for this model.

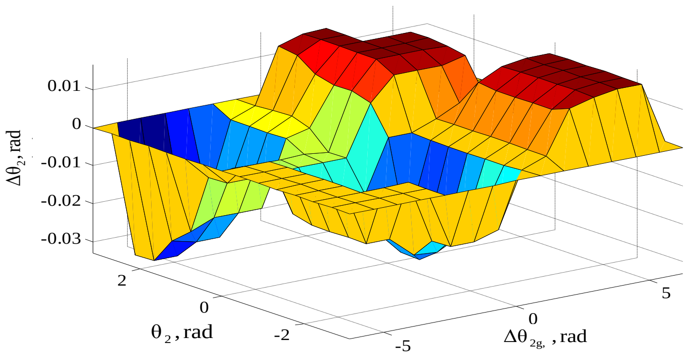
The results of the computer modeling are shown in figure6 where the robot has to move from the start configuration $(\theta_1 = 0^0, \theta_2 = 0^0, \theta_3 = 0^0)$to the goal configuration $(\theta_1 = 180^0, \theta_2 = 90^0, \theta_3 = 90^0)$ the error in reaching the goal after (454) program iterations was:

$$\Delta\theta_{1g}(i+1) = -2.467546496 \times 10^{-04} \text{ rad}$$
$$= -0.014138 \text{ degree}$$
$$\Delta\theta_{2g}(i+1) = -1.173873548 \times 10^{-03} \text{ rad}$$
$$= -0.067258 \text{ degree}$$
$$\Delta\theta_{3g}(i+1) = -1.173873548 \times 10^{-03} \text{ rad}$$
$$= -0.067258 \text{ degree}$$

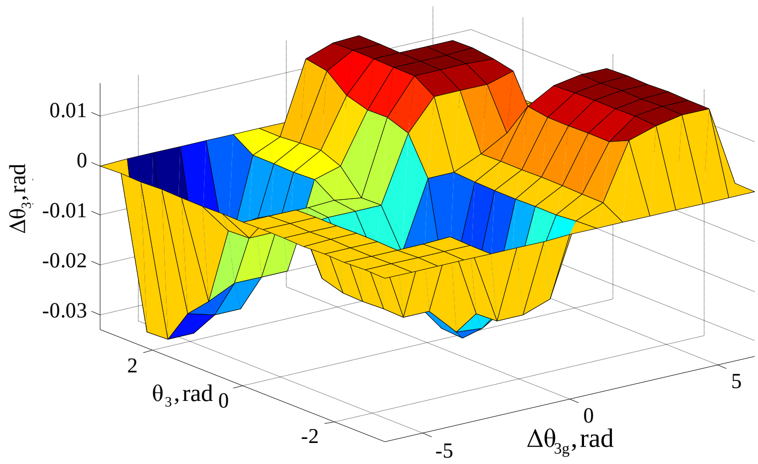
Figure 6 shows surface view of block1, block2 and block3, Figure 7 shows the graphs of $(\Delta\theta_{1g}, \Delta\theta_{2g}, \theta_1, \theta_2, \Delta\theta_1, \Delta\theta_2)$.



(a)



(b)



(c)

Figure 6. Surface View of (a) Block1 (b) Block2 (c) Block3

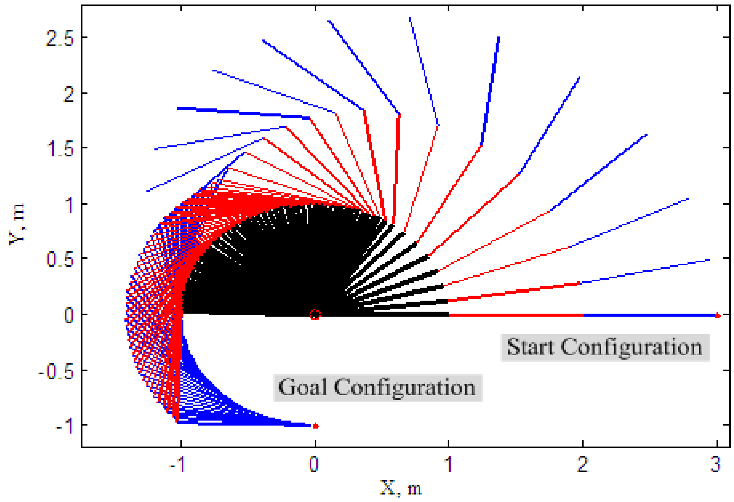


Figure 7.modeling results of run1

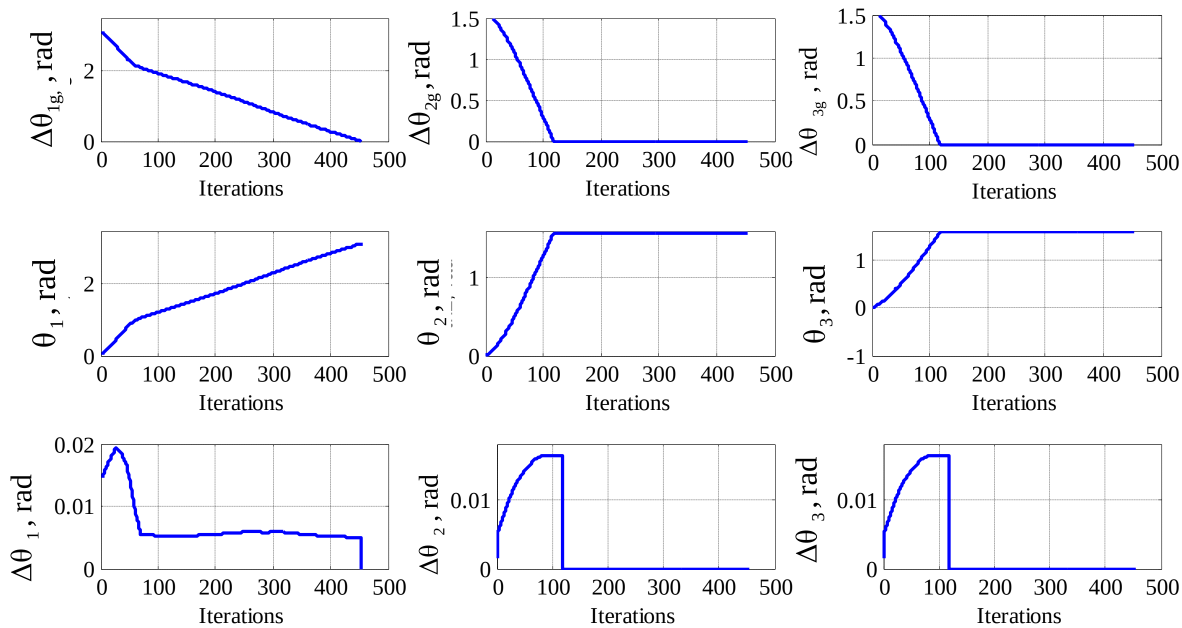


Figure 8. Change of Joint path planning parameters with program iteration index of run 1.

In run2, the results of the computer modeling are shown in figure 5 where the robot has to move from the start configuration ($\theta_1 = 20^0, \theta_2 = 0^0,, \theta_3 = 0^0$)to the goal configuration ($\theta_1 = 150^0, \theta_2 = 90^0, \theta_3 = 90^0$) the error in reaching the goal after (508) program iterations was:

$$\Delta\theta_{1g}(i+1) = -2.443810019 \times 10^{-04} \text{ rad}$$
$$= -0.014002 \text{ degree}$$
$$\Delta\theta_{2g}(i+1) = -1.173873548 \times 10^{-03} \text{ rad}$$
$$= -0.067258 \text{ degree}$$
$$\Delta\theta_{3g}(i+1) = -1.173873548 \times 10^{-03} \text{ rad}$$
$$= -0.067258 \text{ degree}$$

Figure 9 shows modeling results of run2, Figure 10 shows the graphs of ($\Delta\theta_{1g}, \Delta\theta_{2g}, \theta_1, \theta_2, \Delta\theta_1, \Delta\theta_2$).

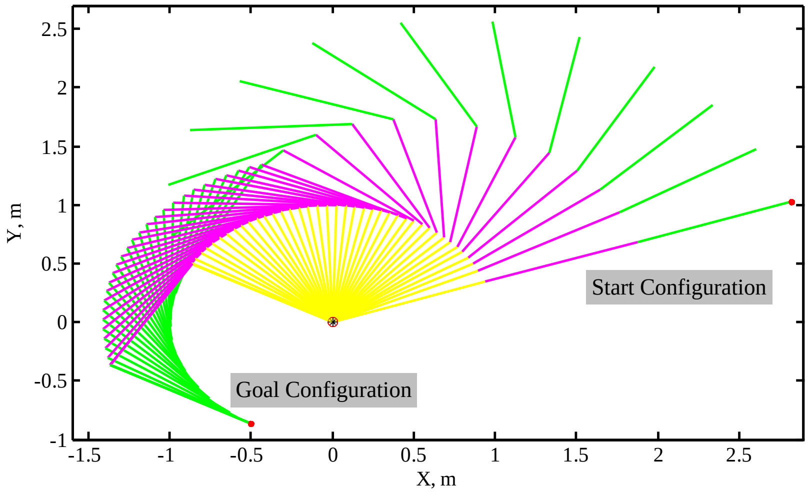


Fig.7 modeling results of run2

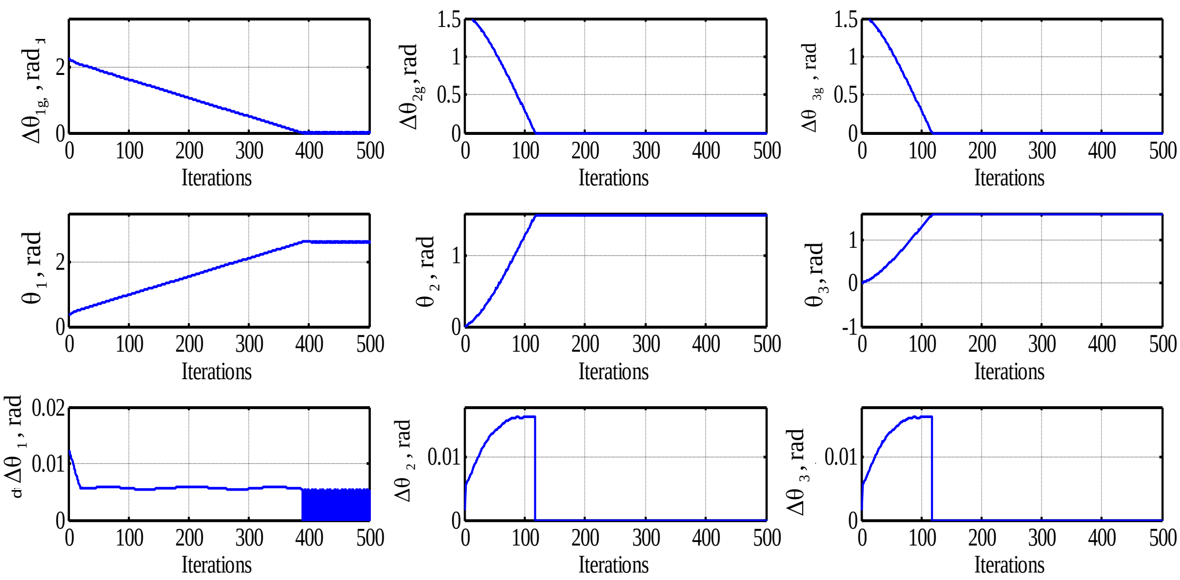


Figure 8. Change of Joint path planning parameters with program iteration index of run 2

V. Conclusion

Our planning system in joint space gives good results since the error of reaching the goal configuration after 454 program iterations was without motion oscillation of the robot gripper near the goal configuration.The error in reaching the goal after (454) program iterations was:

$$\Delta\theta_{1g}(i+1) = -2.467546496 \times 10^{-04} \text{ rad} = -0.014138 \text{ degree}$$
$$\Delta\theta_{2g}(i+1) = -1.173873548 \times 10^{-03} \text{ rad} = -0.067258 \text{ degree}$$
$$\Delta\theta_{3g}(i+1) = -1.173873548 \times 10^{-03} \text{ rad} = -0.067258 \text{ degree}$$

In run2the error in reaching the goal after (508) program iterations was:

$$\Delta\theta_{1g}(i+1) = -2.443810019 \times 10^{-04} \text{ rad} = -0.014002 \text{ degree}$$
$$\Delta\theta_{2g}(i+1) = -1.173873548 \times 10^{-03} \text{ rad} = -0.067258 \text{ degree}$$
$$\Delta\theta_{3g}(i+1) = -1.173873548 \times 10^{-03} \text{ rad} = -0.067258 \text{ degree}$$

References

[1]. Carl D, Crane III, Joseph D, (Eds). Kinematic Analysis of Robot Manipulators, Cambridge University Press, UK,USA, Australia, 2008.

[2]. Appin Knowledge Solutions, "Robotics: Appin Knowledge Solutions", Infinity Science Press Publications LCC, Hingham, Massachusetts, New Delhi, 2007.

[3]. Kermiche S., Larbi S. M., and Abbassi H. A., " Fuzzy Logic Control of Robot Manipulator in the Presence of Fixed Obstacle",The International Arab Journal of Information Technology, Vol. 4, No.1, pp. 26-32, January 2007.

[4]. Yi-liF. , Bao J., Han L., and Shu-guo W., "A Robot Fuzzy Motion Planning Approach in Unknown Environments for Three-Degree Industrial Robots", higer education press and springer-verlag, Vol. 3, pp. 336-340,2006.

[5]. Cheung E., and Lumelsky V., "Proximity Sensing in Robot Manipulator Motion Planning: System and Implementation Issues", IEEE Trans. Robotics and Automation, 1989, 5(6): 740–751. Downloaded from Iraqi Virtual science library at 8/2/2013.

[6]. Lumelsky V. J., Shur M. S., and Wagner S., "Sensitive Skin", IEEE Sensors Journal, 1(1): 41–51, 2001.

[7]. Cheung E., and Lumelsky V., "Development of Sensitive Skin for A 3D Robot Arm Operating in an Uncertain Environment", IEEE International Conference on Robotics and Automation, 12: 1056–1061, 1989.

[8]. Cheung E., and Lumelsky V., "Motion Planning for a Whole-Sensitive Robot Arm Manipulator", IEEE International Conference on Robotics and Automation, 1: 344–349, 1990.

[9]. Wei W. and Shimin W., "3-D Path Planning using Neural Networks for a Robot Manipulator", IEEE computer society pp. 3-6, 2011.Downloaded from Iraqi Virtual science library at 3/11/2011.

[10]. Kermiche S., Larbi S. M., and Abbassi H. A., " Fuzzy Logic Control of Robot Manipulator in the Presence of Fixed Obstacle",The International Arab Journal of Information Technology, Vol. 4, No.1, pp. 26-32, January 2007.

[11]. Koker R., Oz C., Cakar T., and Ekiz H., "A Study of Neural Network Based Inverse Kinematics Solution for a Three-Joint Robot", Robotics and Autonomous Systems 49, 227–234, 2004.

[12]. Tejomurtula S., and Kak S., "Inverse Kinematics in Robotics using Networks", Information Sciences, vol. 116, pp147-164, 1999.

[13]. Gerke M. and Hoyer H., "Fuzzy Collision Avoidance for Industrial Robots", IEEE/RSJ International Conference on Intelligent Robots and Systems, 2: 510–517, 1995.

[14]. Zavlangas P. G. and Tzafestas S. G., " Industrial Robot Navigation and Obstacle Avoidance Employing Fuzzy Logic",Journal of Intelligent and Robotic Systems Vol. 27: pp. 85–97, 2000.

[15]. Shin Y. C. and Xu C., "Intelligent Systems: Modeling, Optimization, and Control", CRC Press, Taylor & Francis Group, LLC, Boca Raton London New York, 2008.