SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ DERGİSİ

Sakarya University Journal of Science
SAUJS

e-ISSN 2147-835X | Period Bimonthly | Founded: 1997 | Publisher Sakarya University | http://www.saujs.sakarya.edu.tr/en/

Title: Real Time Control Application of the Robotic Arm Using Neural Network Based Inverse Kinematics Solution

Authors: Nurettin Gökhan ADAR

Recieved: 2021-04-04 21:57:17

Accepted: 2021-05-18 12:34:31

Article Type: Research Article

Volume: 25
Issue: 3
Month: June
Year: 2021
Pages: 849-857

How to cite
Nurettin Gökhan ADAR; (2021), Real Time Control Application of the Robotic Arm Using Neural Network Based Inverse Kinematics Solution. Sakarya University Journal of Science, 25(3), 849-857, DOI: https://doi.org/10.16984/saufenbilder.907312
Access link
http://www.saujs.sakarya.edu.tr/en/pub/issue/62736/907312

Sakarya University Journal of Science 25(3), 849-857, 2021

Real Time Control Application of the Robotic Arm Using Neural Network Based Inverse Kinematics Solution

Nurettin Gökhan ADAR*[1]

Abstract

Robotic arms are widely used in many industrial applications at present. The control of robotic arms involves position coordination Cartesian space by a forward/inverse kinematics solution method. The inverse kinematics is difficult for real-time control applications, computational requirements are intensive and the run-time is high. The traditional solution methods used geometric, algebraic, and numerical iterative techniques are inadequate and slow in the inverse kinematics solution. Recently, alternative solution methods based on artificial intelligence techniques have been developed to solve the inverse kinematics problem. In this study, a multi-layered feed-forward Artificial Neural Network model was developed to solve the inverse kinematics of the 5 degrees of freedom robotic arm. Using the Proportional-Integral control algorithm combined with this Artificial Neural Network model, the real-time position control of the robotic arm was accomplished. The obtained results were compared with the PI control supported by an analytical inverse kinematics solution in real-time. The results showed that the PI control combined with Artificial Neural Network has superior tracking ability, smaller control error, and better absolute fit to the reference.

Keywords: Inverse kinematics, Robotic arm, Artificial neural network, PI control, Real-time

1. INTRODUCTION

Robot manipulators are extensively used in the manufacturing industry and also have many other specialized applications such as health care, agriculture, military. Manipulator applications involve pick and place an object in a specific position, precision grasping, and manipulation of the objects. Motion control of a robot manipulator is a fundamental problem that must be addressed at the design stage. In recent years, researchers have aimed to develop intelligent and learning robots to further accelerate robotic systems by maintaining high precision and accuracy in robot control. Some of these techniques are the artificial neural network imitated the human brain [1-3], particle swarm optimization algorithm inspired by the bird and fish flocking [4,5], firefly algorithm inspired by the social behavior of fireflies in the tropical [6].

*Corresponding author: gokhan.adar@btu.edu.tr
[1] Bursa Technical University Faculty of Engineering and Natural Sciences, Department of Mechatronics Engineering.
ORCID: https://orcid.org/0000-0001-6888-5755

For the motion control of robot manipulators, there is a requirement for the robot kinematics to identify the relationship between Cartesian coordinates in which robot movements are explained and joint parameters in which the control is achieved. Forward kinematics calculate the position, orientation, and velocity of the end effector from the joint displacements and angles. The inverse kinematics, which is the more difficult and complex, calculate the joint displacements and angles from the end effectors position and velocity. In the inverse kinematics solution, traditional methods used geometric, iterative, and algebraic methods are inadequate and slow due to the nonlinear equations and the complexity of the robot manipulator increasing exponentially with the number of joints. For this reason, researchers have proposed alternative solution methods based on artificial intelligence techniques to solve the inverse kinematics problem, as in other problems related to robots. Robot manipulators are widely used in various industrial automation applications and also the other specialized fields as medical, military. Manipulator applications involve pick and place an object in a specific position, precision grasping, and manipulation of the objects. To accomplish these, the end-effector motion which should be efficiently controlled is defined. Robot kinematics defined mapping between joint space and Cartesian space (x, y, z) is needed for the position control of robot manipulators. Forward kinematics and inverse kinematics are used for the kinematic analysis of robot manipulators. Forward kinematics computes the position, orientation, and velocity of the end effector from the joint displacements and angles, whereas inverse kinematics computes the joint displacements and angles from the end effector's position and velocity.

Köker solved the inverse kinematics problem of the six-joint Stanford robot manipulator by combining characteristics of neural networks and genetic algorithms [7]. Karlik and Aydin investigated the best neural-network configurations to solve the inverse kinematics of a six-joint robot manipulator. To find the inverse kinematics solutions, placement and orientation angles of the robot were used [8]. Köker et al. proposed the neural network solution of inverse kinematics by using cubic trajectory planning for a three-joint robotic manipulator [9]. Mayorga and Sanongboon presented a neural network for fast calculation of the inverse kinematics and effective geometrically bounded singularities prevention of redundant manipulators [10]. Daya et al. presented a neural network architecture that consists of 6 sub-neural to control the position of robotic manipulators [11]. Duka used feed-forward neural network computed desired trajectories in the two-dimensional Cartesian coordinate system for the three-link planar manipulator [12]. Zacharie implemented Logistic Belief Neural Network (LBNN) on a mobile robot with two gripped arms. LBNN was designed to control five degrees of freedom robot arm in real-time [13]. El-Sherbiny et al. compared Artificial Neural Network (ANN), adaptive neuro fuzzy inference system (ANFIS), and genetic algorithm techniques used to solve the inverse kinematics problem of the 5-DOF robot arm [14]. Jiang and Ishita presented a control system consisting of a traditional controller and a neural network controller with a parallel structure for trajectory tracking control of industrial robot manipulators. Neural network controller played a major role in the generating of the actuating force/torque required by the dynamic trajectory [15]. Xu et al. suggested a recurrent neural network-based controller for redundant manipulators subject to kinematic uncertainties. The controller can provide robustness and adaptability even under uncertain conditions because it can learn uncertain model parameters online [16]. Sharma used a 2-DOF Proportional Integral Derivative (PID) for trajectory control of a two-joint planar robot arm [17].

The control of robotic manipulators has a wide research area because of difficult manipulation tasks. Even though many robots work with high accuracy, repeatability, and stability, research continues on the improvement of robotic manipulators to upward increase the precision in robot control. Numerous advanced control algorithms have been developed so far, but PID controllers are employed as the first choice in most current robots in industrial operations due to their simplicity and practicability. However, the linear PID controllers are not adequate for precise control of robot manipulators equipped with direct-drive actuators or perform high-speed motion despite all their advantages. It fails to provide effective control to nonlinear, uncertain, and coupled systems. The capability and utility of the conventional PID controller can be increased to a good extent with hybridization.

This paper describes the implementation of the PI control algorithm combined with ANN for inverse kinematics solving to real-time position control of robot manipulator. Obtained results of the proposed control algorithm are compared with the outcomes of PI control method used for analytical inverse kinematics solving in real-time.

2. KINEMATICS ANALYSIS OF ROBOTIC ARM

5 DOF robotic arm which is similar to human arm structure was studied. A forward and inverse kinematics equation is needed to control the robotic arm in Cartesian space. Forward kinematics equations are derived using the Denavit-Hartenberg (D-H) method. The coordinate systems of the robotic arm are shown in Figure 1 and D-H parameters of the robot arm are given in Table 1 where θ_i represents rotation about the Z-axis, a_i transition along the X-axis, α_i rotation about the X-axis and d_i transition along the Z-axis.

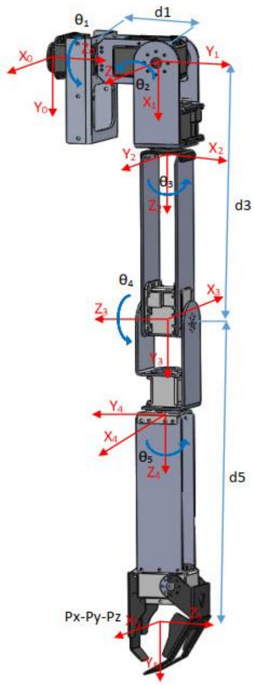


Figure 1 The robotic arm coordinate systems

Table 1
D-H parameters of the robotic arm

	d_i	α_i	a_i	θ_i
1	d_1=100	90°	0	θ_1+90°
2	0	90°	0	θ_2+90°
3	d3=250	90°	0	θ_3+90°
4	0	-90°	0	θ_4
5	d_5=100	90°	0	θ_5

Details of the forward kinematics equations of the robotic arm whose x-y-z position coordinates are given below are presented in [18]:

$$P_x = \cos\theta_1(d_3 + d_5\cos\theta_4)\sin\theta_2 - d_5(\cos\theta_1\cos\theta_2\cos\theta_3 + \sin\theta_1\sin\theta_3)\sin\theta_4$$

$$P_y = (d_3 + d_5\cos\theta_4)\sin\theta_1\sin\theta_2 + d_5(-\cos\theta_2\cos\theta_3\sin\theta_1 + \cos\theta_1\sin\theta_3)\sin\theta_4 \quad (1)$$

$$P_z = d_1 - \cos\theta_2(d_3 + d_5\cos\theta_4) - d_5\cos\theta_3\sin\theta_2\sin\theta_4$$

where P_x, P_y, and P_z represent the position of the end-effector according to the base frame respectively.

In this study, the Algebraic method, one of the closed-form methods, was used to obtain the robotic arm inverse kinematics equations. The target homogeneous transformation matrix can be selected as follow:

$$T_{target} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

where p (p_x, p_y, p_z) is the position of the end-effector and n (n_x,n_y,n_z), o(o_x,o_y,o_z) and a (a_x,a_y,a_z) are orthogonal unit vectors that define the orientation of end-effector frame.

Derivation of the inverse kinematics equations of the robotic arm whose θ_1, θ_2, θ_3, θ_4, θ_5 equations are given below is presented in [18]:

$$\theta_1 = atan2(p_x - o_x\, d_5, -o_y\, d_5 + p_y\,)$$

$$\theta_2 = acos(\,(o_z\, d_5 + d_1 - p_z)/d_3)$$

$$\theta_3 = atan2(o_x \cos(\theta_1)\cos(\theta_2) + o_y \cos(\theta_2)\sin(\theta_1) + o_z \sin(\theta_2)\,, o_y \cos(\theta_1) - o_x \sin\,(\theta_1) \quad (3)$$

$$\theta_4 = \text{acos}\,(((d_1 - p_z)\cos(\theta_2) + (p_x\, cos(\theta_1) + p_y\, sin(\theta_1))\sin(\theta_2) - d_3)/d_5)$$

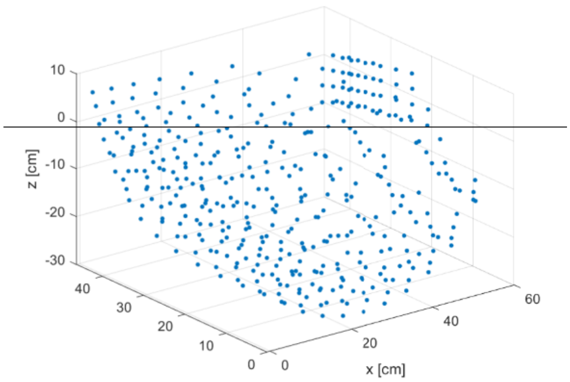
$$\theta_5 = atan2(-a_z \cos(\theta_2) + a_x \cos(\theta_1)\ \sin(\theta_2) + a_y \sin\,(\theta_1)\sin(\theta_2)\,, -n_z \cos(\theta_2) + a_z \cos(\theta_1)\ \sin(\theta_2) + n_z \sin\,(\theta_1)\sin(\theta_2)$$

where θ_1, θ_2, θ_3, θ_4, and θ_5 are presented joint angles respectively.

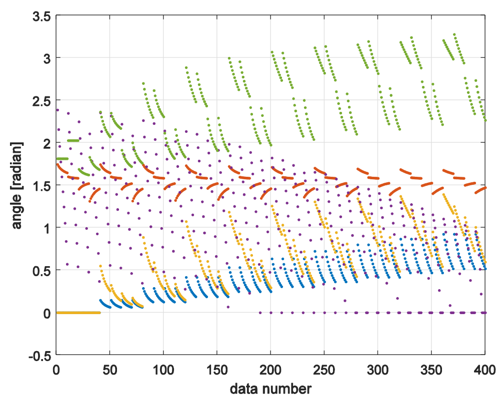
3. SOLUTION OF INVERSE KINEMATICS WITH ANN

ANN can be used to overcome the complexity and difficulty of solving robot inverse kinematics equations. The model can be characterized as a fully connected three-input which are the x-y-z Cartesian coordinate of end-effector, fifteen-hidden, five-output which are θ_1, θ_2, θ_3, θ_4 and θ_5 joint angles neural network. Levenberg-Marquardt is selected to implement Multilayer neural networks training and the sigmoid logistics transfer function is used in the output.

There are two main steps to develop ANN: training and testing. For this, a total of 520 data were collected from the experimental setup. 400 of the 520 data were for training while 120 data were for testing. The development of the neural network was carried out in the Matlab/Simulink Neural Network toolbox. Input and output data of the ANN are given in Figure 2.



(a)



(b)

Figure 2 ANN input and output training data set

4. THE CONTROL OF ROBOTIC ARM

The robotic arm was controlled to compare the performance of ANN and analytical inverse kinematics. PI control algorithm was applied to control the position of the robotic arm for both ANN model and analytical inverse kinematics solution in real-time. The structure of the proposed control system is shown in Figure 3.

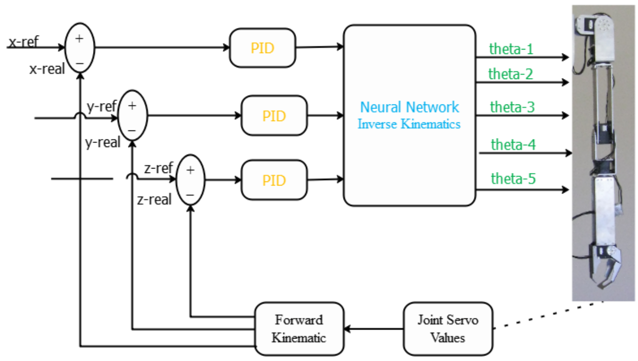


Figure 3 Structure of the proposed control system

General expressions of PI controller are given below:

$$u(t) = K_p\left((r(t) - y(t)) + \frac{1}{T_i}\int_0^t (r(t) - y(t))\,dt\right) \quad (4)$$

where *r(t)*, *y(t)* and *u(t)* are the reference, process output, and control signal, respectively. K_p is the PID gain, T_i is the integral time constants.

The gains of PI controller for 5 DOF robot arm automatically were tuned by MathWorks algorithm to achieve a balance between performance and robustness. K_p and K_i gain values of the PI controllers used for Cartesian coordinates (x, y, z) are tuned as shown in Table 2.

Table 2
The gain values of the PI controllers

PI Controller	K_p	K_i (K_p/T_i)
x-coordinate	0.5	0.1
y-coordinate	0.5	0.2
z-coordinate	0.3	0.15

In this control method, reference x-y-z values are provided externally by the user. Servo motors used in the robotic arm have embedded sensors. Each motor's angular position can be provided feedback in real-time. Forward kinematics is used to compute the actual position of the end effector from the feedback motor's angular position. Error is obtained by comparing the computed actual position (real x-y-z values) and reference x-y-z values. This error is the input of the PI controller to get the control signal. Control signals obtained from the controller are given as input to the ANN or Inverse kinematics to calculate joint angles. Using these joint angle values, it is provided to move to the desired position of the robotic arm with servo motors.

5. IMPLEMENTATION AND RESULTS

Experimental setup is given in Figure 6. A computer, power supply, USB converter, and robotic arm constitute the hardware part of the experimental setup. the robotic arm is driven with Dynamixel servo motors. Each motor communicates with a computer with RS-485 using a USB converter. Figure 4 demonstrates the fundamental configuration of the arm at its home position.

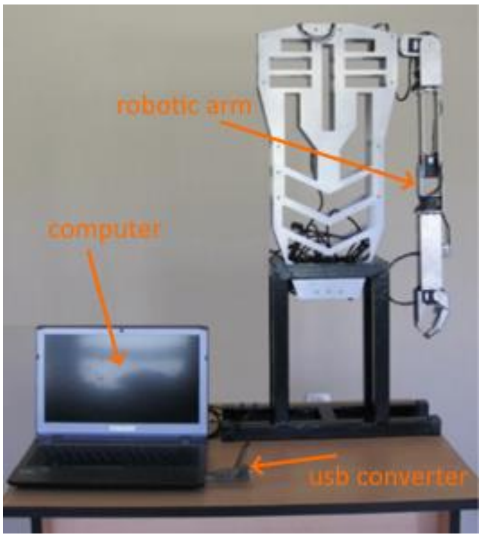
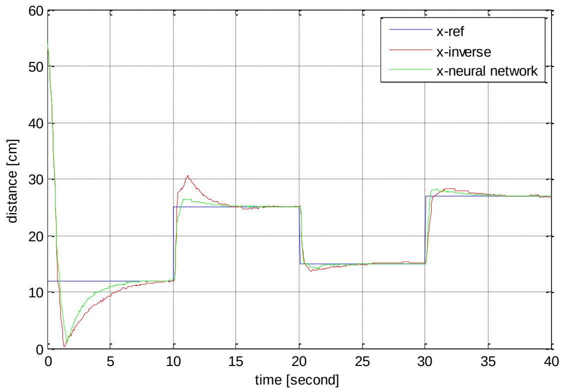


Figure 4 Fundamental configuration of the robotic arm

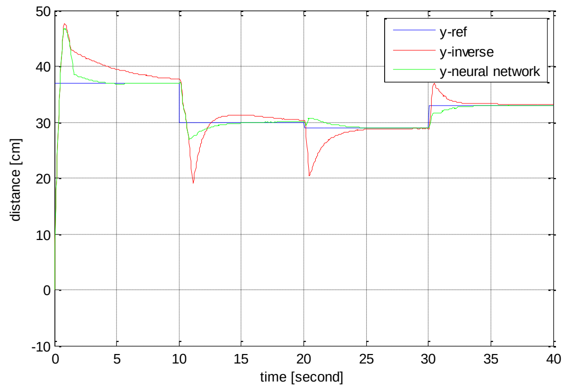
The algorithm was developed using Matlab-Simulink. Communication between Matlab-Simulink and servo motors is established using the RapidSTM32 library. RapidSTM32 is a Simulink device driver block set and tool suite for the STM32 microcontroller family.

In this study real-time applications of the 5-DOF robotic arm were implemented to compare the performance of ANN and analytical inverse kinematics.

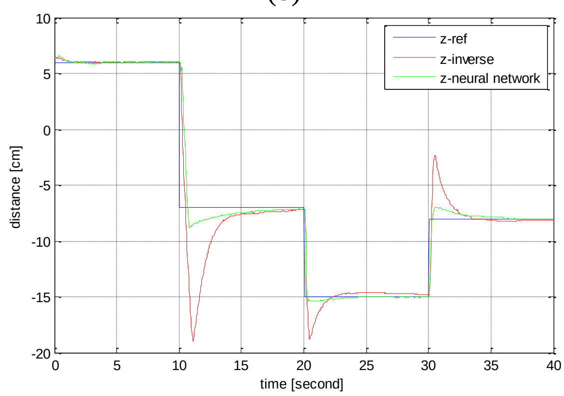
Figure 5 shows the tracking capabilities on the desired trajectory of the robotic arm used different inverse kinematics solution approaches (analytical and ANN). Also, it can be seen how the end-effector tracks the desired position on the x, y, and z coordinates in the Cartesian space.



(a)



(b)



(c)

Figure 5 Trajectory tracking capabilities of the analytical and ANN approach for the x coordinate, the y coordinate, the z coordinate.

The reference position was modified at 10, 20, and 30 seconds. PI control system combined with ANN solves responded to the changes in x-y-z coordinates of reference position in less time and with smaller error than the analytical solution. In the PI control with the analytical solution, large overshoot/undershoot are observed at the points of change of the reference position. In initial time (t=0), PI controller combined with ANN has values away from x and y reference positions. This is because the ANN solution initially uses random weights. However, it has faster control response and lower settling time.

The relative errors and Root Mean Square Errors (RMSE) in the x-y-z coordinates for each interval of the reference position are given in Table 3 and Table 4, respectively. Relative Error and RMSE are calculated as follows:

$$Relative\ Error\ (\%) = \frac{Ref.Position - Arm's\ Position\ in\ Real\ Time}{Ref.Position} \times 100$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(Ref.Position - Arm's\ Position\ in\ Real\ Time)^2} \quad (5)$$

Table 3. The relative errors in the x-y-z coordinates of PI control approaches used analytical and ANN-based inverse kinematics.

Coordinate	Time (sec)	Relative Error (%)	
		Analytical Inverse Kinematics	**ANN-Based Inverse Kinematics**
x	10	1.4655	-0.2799
	20	-0.4980	-0.2088
	30	-0.6612	0.3126
	40	0.5636	0.1269
y	10	-1.8490	-0.0589
	20	1.2316	-0.3517
	30	0.5013	0.1028
	40	0.4398	-0.0665
z	10	-0.3731	0.1854
	20	-2.8472	-1.6168
	30	1.3420	-0.0673
	40	-1.9753	-0.0342

Table 4 RMSE in the x-y-z coordinates of PI control approaches used analytical and ANN-based inverse kinematics.

Coordinate	Time İnterval (sec)	RMSE	
		Analytical Inverse Kinematics	ANN-Based Inverse Kinematics
x	0-10	9.5114	9.0954
	10-20	2.4381	1.7242
	20-30	1.3613	1.2985
	30-40	1.7884	1.5574
y	0-10	5.5952	4.9086
	10-20	2.7881	1.3991
	20-30	2.3009	0.6003
	30-40	1.0700	0.6950
z	0-10	0.1107	0.1350
	10-20	3.9429	2.4764
	20-30	1.2369	0.8968
	30-40	1.5826	0.8437

The PI control method used the ANN-based inverse kinematics solution has achieved smaller relative errors to the reference position. Also, the robotic arm controlled in real-time using ANN-based inverse kinematics has reached close to the reference position with lower RMSE values at all-time intervals, as can be seen from Tables 4. It is seen that the ANN method has also a better absolute fit to the reference.

6. CONCLUSION

This study aims to minimize the PI control error in real-time of the 5 DOF robotic arm by using the ANN model to solve inverse kinematics. For this aim, an ANN model with optimal topology was developed that solves the inverse kinematics problem and predicts joint angles. The training performance of the designed network was evaluated and a high correlation was obtained. Validation results showed that the ANN model solves this problem with high accuracy, provides a strong estimate and good generalization capability.

The position control of the robotic arm was accomplished the PI control algorithm combined with ANN and This proposed approach was compared with the PI control supported by an analytical inverse kinematics solution in real-time.

Tracking ability and control errors in the x-y-z coordinates of the robotic arm, which was controlled by two different methods for 40 seconds in real-time, were examined and the high accuracy of the results obtained from the PI control combined with ANN can be clearly observed.

Funding

The author has no received any financial support for the research, authorship or publication of this study.

The Declaration of Conflict of Interest/ Common Interest

No conflict of interest or common interest has been declared by the authors.

The Declaration of Ethics Committee Approval

This study does not require ethics committee permission or any special permission.

The Declaration of Research and Publication Ethics

The authors of the paper declare that they comply with the scientific, ethical and quotation rules of SAUJS in all processes of the paper and that they do not make any falsification on the data collected. In addition, they declare that Sakarya University Journal of Science and its editorial board have no responsibility for any ethical violations that may be encountered, and that this study has not been evaluated in any academic publication environment other than Sakarya University Journal of Science.

REFERENCES

[1] C. Song, S. Xie, Z. Zhou, and Y. Hu, “Modeling of pneumatic artificial muscle using a hybrid artificial neural network

approach. Mechatronics," vol. 31, pp. 124-131, 2015.

[2] F. Luan, J. Na, Y. Huang, and G. Gao, "Adaptive neural network control for robotic manipulators with guaranteed finite-time convergence," Neurocomputing, vol. 337, pp. 153-164, 2019.

[3] K. Zheng, Q. Zhang, Y. Hu and B. Wu, "Design of fuzzy system-fuzzy neural network-backstepping control for complex robot system," Information Sciences, vol. 546, pp. 1230-1255, 2021.

[4] Y. Wang, Y. Shi, D. Ding and X. Gu, "Double global optimum genetic algorithm–particle swarm optimization-based welding robot path planning," Engineering Optimization, vol. 48, pp. 299-316, 2016

[5] S. Dereli and R. Köker, "IW-PSO Approach to the Inverse Kinematics Problem Solution of a 7-DOF Serial Robot Manipulator," Sigma Journal of Engineering and Natural Sciences, vol. 36, pp. 77-85, 2018

[6] A.K. Sadhu, A. Konar, T. Bhattacharjee and S. Das, "Synergism of Firefly Algorithm and Q-Learning for Robot Arm Path Planning," Swarm and Evolutionary Computation, vol. 43, pp. 50-68, 2018

[7] R. Köker, "A genetic algorithm approach to a neural-network-based inverse kinematics solution of robotic manipulators based on error minimization," Information Sciences, vol. 222, pp. 528-543, 2013.

[8] B. Karlik and S. Aydin, "An improved approach to the solution of inverse kinematics problems for robot manipuşators," Engineering Applications of Artificial Intelligence, vol. 13, pp. 159-164, 2000.

[9] R. Köker, C. Öz, T. Çakar and H. Ekiz, "A study of neural network based inverse kinematics solution for a three-joint robot," Robotics and Autonomous Systems, vol. 49, pp. 227-234, 2004.

[10] R.V. Mayorga and P. Sanongboon, "Inverse kinematics and geometrically bounded singularities prevention of redundant manipulators: An artificial neural network approach," Robotics and Autonomous Systems, vol. 53, pp. 164-176, 2005.

[11] B. Daya, S. Khawandi and M. Akoum, "Applying Neural Network Architecture for Inverse Kinematics Problem in Robotics," J. Software Engineering & Applications, vol. 3, pp. 230-239, 2010.

[12] A.V. Duka, "Neural network based inverse kinematics solution for trajectory tracking of a robotic arm," Procedia Technology, vol. 12, pp. 20-27, 2014.

[13] Zacharie M., "Advanced Logistic Belief Neural Network Algorithm for Robot Arm Control", Journal of Computer Science, vol. 8, no. 6, pp. 965-970, 2012.

[14] A. El-Sherbiny, M.A. Elhosseini and A.Y. Haikal, "A comparative study of soft computing methods to solve inverse kinematics problem," Ain Shams Engineering Journal, vol. 9, no. 4, pp. 2535-2548, 2018.

[15] Z.H. Jiang and T. Ishita, "A Neural Network Controller for Trajectory Control of Industrial Robot Manipulators," vol. 3, no. 8, pp. 1-8, August 2008.

[16] Z. Xu, S. Li, X. Zhou, W. Yan, T. Cheng and D. Huang, "Dynamic neural networks based kinematic control for redundant manipulators with model uncertainties," vol. 329, pp. 255-266, February, 2019.

[17] R. Sharma, P. Gaur and A.P. Mittal, "Performance analysis of two-degree of freedom fractional order PID controllers for robotic manipulator with payload," ISA Transactions, vol. 58, pp. 279-91, 2015.

[18] N. G. Adar, "Desing, Manufacturing and Control of Mobile Robot" Ph.D. dissertation, Dept. Mech. Eng., Sakarya Univ., Sakarya, 2016.