Sigma J Eng & Nat Sci 36 (1), 2018, 77-85

Sigma Journal of Engineering and Natural Sciences
Sigma Mühendislik ve Fen Bilimleri Dergisi

Research Article

IW-PSO APPROACH TO THE INVERSE KINEMATICS PROBLEM SOLUTION OF A 7-DOF SERIAL ROBOT MANIPULATOR

Serkan DERELİ*[1], Raşit KÖKER[2]

[1]*Sakarya Vocational High School, Sakarya University, SAKARYA;* ORCID:0000-0002-1856-6083
[2]*Faculty of Technology, Sakarya University, SAKARYA;* ORCID:0000-0002-3811-2310

Received: 16.08.2017 Accepted: 03.10.2017

ABSTRACT

In this paper, two variants of particle swarm optimization (PSO) are used to calculate the inverse kinematics of a new 7-revolute jointed robot arm. This robot arm is required to move from a position to the desired position with a minimum error in the workspace. A scenario has been set for this purpose. According to this scenario, it is desired that the end effector of the robot arm reach a predetermined position with the minimum error. The results obtained with Random Inertia Weight and Global Local Best Inertia Weight, are compared with the standard PSO. Moreover, the path of the robot arm obtained by cubic trajectory planning is depicted with graphs. Results that computer simulated based, reveal that PSO can be efficiently used for inverse kinematics solution. However, for the inverse kinematic solution, the PSO variables are much more effective than the standard PSO.
Keywords: Inverse kinematics, particle swarm optimization, inertia weight, 7-Dof.

1. INTRODUCTION

Especially today, because of the kinematic and dynamic performance, redundant robots (Degree of freedom) attract both industry and researchers. Despite the flexible working structure, the control of these robots is incredibly difficult owing to the large number of joint [1]. The kinematic calculation that is studied in two parts as forward and inverse kinematics, is a scientific discipline that examines robot movements. Forward kinematics that is to determine the position of the end effector from the joint angles, is relatively easy [2]. On the other hand, inverse kinematics that is to obtain the joint angles from the position of the end effector, is a non-linear equations, more complex and impossible to solution by conventional methods such as algebraic, geometric and iterative [3]. Moreover, a redundant robot has infinite number of inverse kinematics solutions. Hence, the most suitable method for this solution is intelligent optimization algorithms such as artificial bee colony, particle swarm optimization, firefly algorithm and artificial neural network [4]. Redundant structure makes impossible to solve with analytical methods, because this robot manipulator has too much the number of joint [5]. This is why researchers have focused on solving the inverse kinematics problem using artificial intelligence algorithms.

* Corresponding Author: e-mail: dereli@sakarya.edu.tr, tel: (264) 295 54 54

Tejomurtula and Kak used an artificial neural network to perform an inverse kinematic solution of a 3-jointed robot arm. They have achieved a more effective result by eliminating some of the disadvantages such as training time and accuracy, of the BP algorithm [6]. Dash *et al*. have implemented the inverse kinematic solution of a 6-articulated robot arm with artificial neural networks [7]. Huang et al. have calculated the inverse kinematic solution of a seven articulated robot manipulator quickly and accurately using classical particle swarm optimization [8]. Ayyıldız and Çetinkaya have designed a new manipulator with 4-DOF and have obtained the inverse kinematic solution of this manipulator with four different intelligent algorithms which are PSO, QPSO, GSA and GA. they demonstrated comparatively results obtained [9]. Rokbani and Alimi have studied the contribution to the of inverse kinematics solution using variants of PSO, such as inertia weight, constriction factor and linear decreasing weight [10]. In addition, they have used a two-jointed robot arm for simulation test. Rokbani *et al.* have used the firefly method which defined as the newest swarm algorithm, and simulated the results in a three-articulated robot arm [11]. Köker has proposed a new hybrid model consisting of artificial neural network and genetic algorithm for the inverse kinematic solution of a six-axis manipulator [12]. Similarly, Pam et al. have proposed an inverse kinematic solution using bee algorithm and artificial neural network in their work and simulated their work in a three-jointed manipulator. The bee algorithm was used to train the neural network which has a multilayer perceptron structure [13].

The purpose of this work is to perform inverse kinematics calculations of a 7-joint redundant robot arm using the PSO variables which are called Inertia Weight and are very effective in convergence [14], with minimal error and to test the simulation base on a newly designed manipulator.

2. KINEMATIC ANALYSIS OF A 7-DOF REDUNDANT ROBOT MANIPULATOR

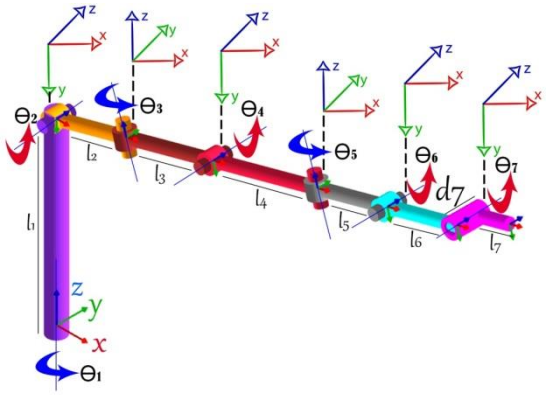


Figure 1. The structure of 7-DOF robot manipulator

Robotic manipulators which are two ways including prismatic and revolution, consist of links sequentially connected to each other with joints which perform a movement mechanism taking certain angles or by a certain percentage of elongation and shortening by actuators [15]. Designed robot manipulator for this study has 7-DOF with seven revolute joints and is shown in Figure 1. A 7-DOF robotic manipulator not only performs the movement from one position to another position in a comfortably but also has infinite inverse kinematic solutions. Of course, objective is to provide the end effector to be positioned correctly [16]. Today, kinematic calculations are done by homogeneous transformation matrices which are created with the help of four parameters that is called Denavit-Hartenberg parameters [17]. In Table 1, "i" represents the number of joints. The length values are used in meters and the angle values are used in degree. The homogeneous

transformation matrix whi ch is formed by DH parameters in Eq. (1) is used to achieve the robot's kinematic equations [18], [19].

Table 1. DH parameters of 7-Dof redundant manipülatör

i	ai (m)	αi (Deg)	di (m)	Θi (Degree)
1	0	-90	l1=0,5	-180<Θ_1<180
2	l2=0,2	90	0	-90<Θ_2<30
3	l3=0,25	-90	0	-90<Θ_3<120
4	l4=0,3	90	0	-90<Θ_4<90
5	l5=0,2	-90	0	-90<Θ_5<90
6	l6=0,2	0	0	-90<Θ_6<90
7	l7=0,1	0	0	-30<Θ_7<90

$$ {}^{i}_{i-1}T = \begin{bmatrix} cos\theta\mathrm{i} & -cos\alpha i.sin\theta\mathrm{i} & sin\alpha i.sin\theta\mathrm{i} & ai.cos\theta\mathrm{i} \\ sin\theta\mathrm{i} & cos\alpha i.cos\theta\mathrm{i} & -cos\theta\mathrm{i}.\sin\alpha\mathrm{i} & ai.sin\theta\mathrm{i} \\ 0 & sin\alpha i & cos\alpha i & di \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1) $$

$$ {}^{7}_{0}T = {}^{1}_{0}T.{}^{2}_{1}T.{}^{3}_{2}T.{}^{4}_{3}T.{}^{5}_{4}T.{}^{6}_{5}T.{}^{7}_{6}T = \begin{bmatrix} n_x & s_x & a_x & P_x \\ n_y & s_y & a_y & P_y \\ n_z & s_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2) $$

Where ${}^{i}T_{i+1}$ is the transfer matrix of link i. ${}^{0}T_7$ matrix produces a Cartesian coordinate for any seven joint angles. Because the cost function of the proposed approach is the Euclidian distance in Cartesian space between the obtained and the target points. ${}^{0}T_7$ can be used to calculate the Cartesian coordinate of the obtained point in the cost function.

In Eq. (2), Px, Py, and Pz denotes the elements of the position vector whereas nx, ny, nz , sx, sy, sz, ax, ay, az denote the rotational elements of the transformation matrix [20]. In this study, only the position vector will be used to calculate the position error. The position vector equation is as follows (where s and c represent the sine and cosine functions):

$$ P_x = (c\theta_1c\theta_2c\theta_3c\theta_4 - s\theta_1s\theta_3s\theta_4 - c\theta_1s\theta_2s\theta_4)(c\theta_5c\theta_6l_7c\theta_7 - c\theta_5s\theta_6l_7s\theta_7 - s\theta_5d_7 + c\theta_5l_6c\theta_6 + l_5c\theta_5) + (-c\theta_1c\theta_2s\theta_3 - s\theta_1c\theta_3)(s\theta_5c\theta_6l_7c\theta_7 - s\theta_5s\theta_6l_7s\theta_7 + c\theta_5d_7 + s\theta_5c\theta_6l_6 + l_5s\theta_5) + (c\theta_1c\theta_2c\theta_3s\theta_4 - s\theta_1s\theta_3s\theta_4 + c\theta_1c\theta_4s\theta_2)(-s\theta_6l_7c\theta_7 - c\theta_6l_7s\theta_7 - l_6s\theta_6) + c\theta_1c\theta_2(c\theta_3c\theta_4l_4 + l_3c\theta_3) - s\theta_1(s\theta_3c\theta_4l_4 + l_3s\theta_3) - c\theta_1s\theta_2l_4s\theta_4 + c\theta_1c\theta_2l_2 \quad (3) $$

$$ P_y = s\theta_1c\theta_2c\theta_3c\theta_4 + c\theta_1s\theta_3c\theta_4 - s\theta_1s\theta_2s\theta_4)(c\theta_5c\theta_6l_7c\theta_7 - c\theta_5s\theta_6l_7s\theta_7 - s\theta_5d_7 + c\theta_5c\theta_6l_6 + l_5c\theta_5) + (-s\theta_1c\theta_2s\theta_3 + c\theta_1c\theta_3)(s\theta_5c\theta_6l_7c\theta_7 - s\theta_5s\theta_6l_7s\theta_7 + c\theta_5d_7 + s\theta_5c\theta_6l_6 + l_5s\theta_5) + (s\theta_1c\theta_2c\theta_3s\theta_4 + c\theta_1s\theta_3s\theta_4 + s\theta_1s\theta_2c\theta_4)(-s\theta_6l_7c\theta_7 - c\theta_6l_7s\theta_7 - l_6s\theta_6) + s\theta_1c\theta_2(c\theta_3c\theta_4l_4 + l_3c\theta_3) + c\theta_1(s\theta_3c\theta_4l_4 + l_3s\theta_3) - s\theta_1s\theta_2s\theta_4l_4 + s\theta_1c\theta_2l_2 \quad (4) $$

$$ P_z = (-s\theta_2c\theta_3c\theta_4 - c\theta_2s\theta_4)(c\theta_5c\theta_6l_7c\theta_7 - c\theta_5s\theta_6l_7s\theta_7 - s\theta_5d_7 + c\theta_5c\theta_6l_6 + l_5c\theta_5) + s\theta_2s\theta_3(s\theta_5c\theta_6l_7c\theta_7 - s\theta_5s\theta_6l_7s\theta_7 + c\theta_5d_7 + s\theta_5c\theta_6l_6 + s\theta_5l_5) + (-s\theta_2c\theta_3s\theta_4 + c\theta_2c\theta_4)(-s\theta_6l_7c\theta_7 - c\theta_6l_7s\theta_7 - s\theta_6l_6) - s\theta_2(c\theta_3c\theta_4l_4 + l_3c\theta_3) - c\theta_2s\theta_4l_4 - s\theta_2l_2 + l_1 \quad (5) $$

In the equation (3)(4)(5), if the angle values are known, the position of the end effector can be obtained very easily and this process is referred to as forward kinematics. However, the process of obtaining angle values from position information is called inverse kinematics and it is known as a process that cannot be conducted by conventional methods [21]. In addition, there are infinite inverse kinematics solutions for the 7-DOF robot arm [22]. In this paper, the optimal joint values of the 7-DOF robotic manipulator were obtained using particle swarm optimization.

3. PARTICLE SWARM OPTIMIZATION AND VARIANTS

3.1. Conventional PSO

The particle swam optimization which was developed inspired by a flock of birds, has a powerful search algorithm. Firstly, it is a technique used by Kennedy and Eberhart [23]. This algorithm has some advantages such as its easily applied and having powerful control parameters, compared with other heuristic techniques [24].

The particle swarm optimization is preferred intensively for nonlinear problems which are one of the major problems with the search space and give better solution. In PSO, particles reach optimal solution together using experience and the neighborhood [25].

Only two equations are used to move on the target of particles in PSO, position update equations (Eq. 7) and velocity update equation (Eq. 6). These equations are modified again at each iteration [26].

$$v_{id} = v_{id} + c_1.r_1.(p_{best} - x_{id}) + c_2.r_2.(g_{best} - x_{id}) \quad (6)$$

$$x_{id} = x_{id} + v_{id} \quad (7)$$

Where d=1, 2, …, D is the dimension and i=1, 2, …, S is the swarm size; c1 and c2 are weight of personal best and weight of global best, respectively; r1 and r2 are random numbers distributed uniformly in [0,1]. The algorithm used in this study is as follows:

- Initialization PSO
- Randomly assign angle values in the appropriate range
- **Loop Iteration**
  - **Loop particle**
    - Perform inverse kinematics calculation (Eq. 3, 4, 5)
    - Calculate error value (Eq. 12)
    - Find the value of Pbest
    - Update particle velocity value
    - Update particle angle values
  - **End Loop Particle**
  - Compare Gbest with Pbest
- **End Iteration**
- Return Gbest value

3.2. Inertia Weight Strategies in PSO

Sometimes, although the PSO has shown very good convergence in complex problems, it can get stuck near its optimal solution and cannot go any further [27]. Therefore, based on the idea that the PSO algorithm can be developed with some variables, Inertia Weight has been used in many researches and has produced effective results [28]. Inertia weight provides a way to stabilize the particles close to optimal results of search space. So, it has an important role to play in making the correct position of the particle in the search space. It makes this work by contributing to the velocity of the particles [29]. In 1998, Shi and Eberhart have been using inertia weight first time. According to their opinions; although a large inertia weight is very effective in

the global search, a small inertia weight is much more successfully in the local search. In addition, today, some researchers have done studies on the inertia weight dynamically adjustable [30]. Eberhart and Shi [31] have achieved effective results in increasing the velocity of particles at random in each iteration and they call this technique as the random inertia weight (Eq. 8).

$$w = 0.5 + \frac{rand()}{2} \tag{8}$$

They have achieved much more effective results increasing the velocity of particles based on global best and local best in each generation. In this way, global-local inertia weight will not have to take both fixed and linearly decreasing values (Eq. 9).

$$w = 1.1 - \frac{gbest}{pbest} \tag{9}$$

The resulting velocity update equation becomes:

$$v_{id} = w * v_{id} + c_1.r_1.(p_{best} - x_{id}) + c_2.r_2.(g_{best} - x_{id}) \tag{10}$$

Inertia weight is used to control the particle velocity. Because, the velocity of particles are a great influence for the convergence of its particles [13]. In this study, the inverse kinematics solution of having redundant feature the 7-Revolute articulated robot arm was performed with both standard PSO and inertia weight strategies and Results are presented as comparative.

3.3. The Method for Solving the Inverse Kinematics Problem

In this study, the aim of the optimization problem is to find the optimum angle value (θ) for each joint with the given initial Cartesian coordinate and the target coordinate, so the end-effector of robot arm is transferred to desired location by θ. Obviously, the accurate calculations of θ values are very important.

$$Error = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \tag{12}$$

The main goal of this study is to solve this optimization problem by implementing the PSO. For this purpose, we designed a fitness function that is based on Euclidian distance given equation (12) between the desired location (x_2, y_2, z_2) and the current location (x_1, y_1, z_1) described. This cost can be used to calculate fitness function.

4. SIMULATION RESULT AND DISCUSSION

The purpose of this paper is to examine how effective and successful of the PSO algorithm to solve the inverse kinematics solution of the 7-Dof serial robot arm. These simulations are performed with the parameters: iteration=5000, particle=300 and each simulation is the best of ten runs between them. The scenario for this study is as follows: The initial position of the manipulator [$\theta_{1i},\theta_{2i},\theta_{3i},\theta_{4i},\theta_{5i},\theta_{6i},\theta_{7i}$]=[0,0,0,0,0,0,0], and determined final position (fig. 2) [$\theta_{1f},\theta_{2f},\theta_{3f},\theta_{4f},\theta_{5f},\theta_{6f},\theta_{7f}$]=[45,0,45,0,45,0,0].

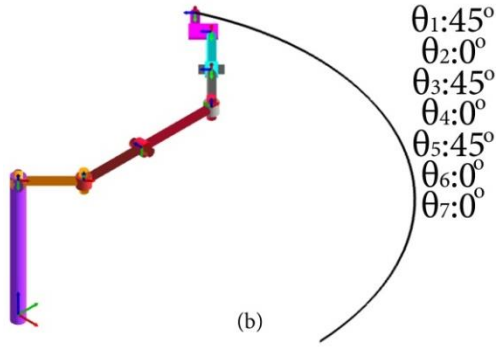
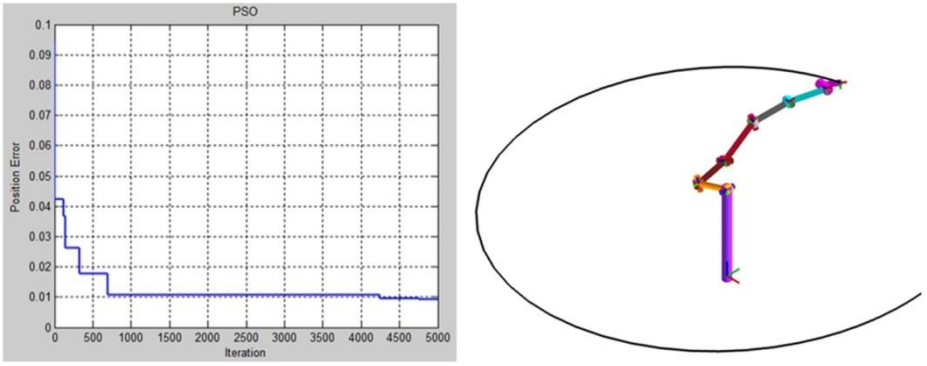


Figure 2. Final position of the robot arm

Table 2. The results of simulation

Algorithm	C1	C2	Error	Iteration	Time (sec)
Conventional PSO	1,4	1,4	9,13e-03	4773	1,9
Random IW	1,2	1,2	6,20e-03	3789	1,6
Global-Local Best IW	1,2	1,2	3,64e-03	2846	1,2

A cubic trajectory planning is preferred for locating of the robot arm from initial position to final position. The purpose of this paper is to obtain the joint angles that are optimal values where the end effector is closest to the determined final position, of the end effector of the robot arm. In this study, obtained the results and used the parameters appear in Table 2. According to given results in Table 2, distance between computed position and desired position by global-local best inertia weight smaller than the other two.

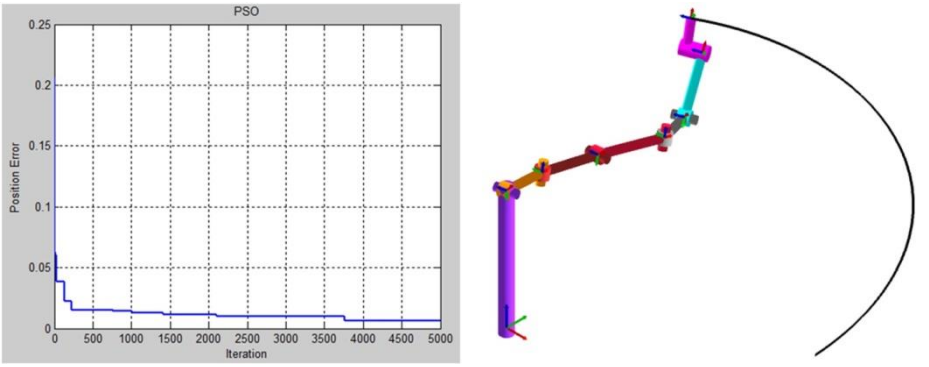


(a) (b)

Figure 3. Standart PSO position error value (a) and followed by the robot arm trajectory (b)

As shown in Fig. 3a, the error rate of the end effector is reduced by PSO successfully. However, a long road of robot trajectory tracking is possible (Fig. 3b). This reveals the high energy costs. Additional precautions should be taken in the algorithm to avoid this situation. The

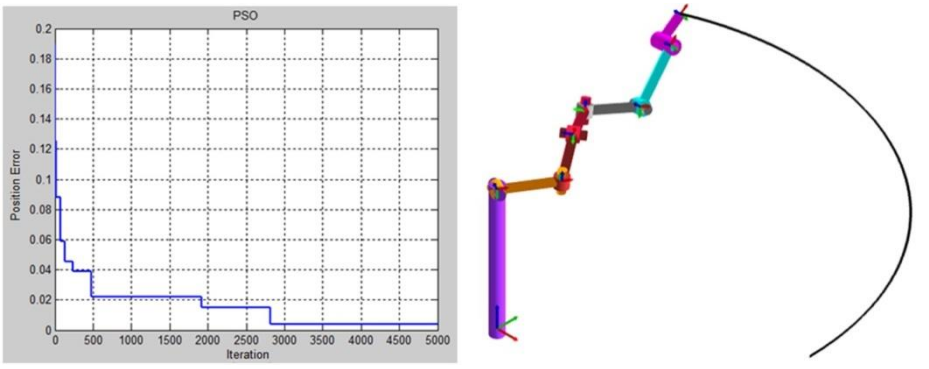
evolved optimal solution of joint angle is [$\theta_{1f},\theta_{2f},\theta_{3f},\theta_{4f},\theta_{5f},\theta_{6f},\theta_{7f}$]=[-180,17.58,-74.28,-11.72,-34.77,11.99,12.76].



(a) (b)

Figure 4. Random inertia weight position error (a) and followed by the robot arm trajectory (b)

In Fig. 4, the results seem to be obtained by the PSO algorithm using Random Inertia Weight strategy. This result is again an optimal result. This result better than standard PSO, but it is a worse result than the Global-Local Best Inertia Weight strategy. The evolved optimal solution is [$\theta_{1f},\theta_{2f},\theta_{3f},\theta_{4f},\theta_{5f},\theta_{6f},\theta_{7f}$]=[102,14.17,-10.07,2.47,24.4,-65.87,-30].



(a) (b)

Figure 5. Global-local best inertia weight position error and followed by the robot arm trajectory

Fig. 5 represents the best result of this study. This simulation has the best value of the error position and has also come to the fore with the approach results in smaller iterations. According to Fig. 5b, the final angle of the joint is as follows [θ1f,θ2f,θ3f,θ4f,θ5f,θ6f,θ7f] = [62.76,3.45,64.26,20.18,-49.34,-64.68,11.39].

5. CONCLUSION

In this paper, inverse kinematics problem have transformed into an optimization problem and this problem is solved with intelligent optimization techniques by a comparative study. The study was carried out as a computer based simulation. The inverse kinematics solutions based on

particle swarm optimization has been performed avoiding time consuming methods and numerical problems. This paper shows that averaged distance by Global-Local Best Inertia PSO smaller than the other two. These simulations clearly indicate that the particle swarm optimization is extremely successful in obtaining the optimal solution of the 7-revolute robotic manipulator. Moreover, the simulation results clearly show that the importance of the inertia weight strategies was highlighted in the article. It can be concluded that contribute to the solution of PSO variable is indisputable.

REFERENCES

[1] Dereli S, Köker R, (2016) In a research on how to use inverse kinematics solution of actual intelligent optimization method. International Symposium on Innovative Technologies in Engineering and Science, 3-5 Nov 2016, Antalya, Turkey.

[2] Dereli S, Köker R, (2017) Design and Analysis of Multi-Layer Artificial Neural Network Used for Training in Inverse Kinematic Solution of 7-DOF Serial Robot, *Gaziosmanpasa Journal of Scientific Research* 6, 60-71.

[3] Köker R, Çakar T, Sarı Y, (2014) A neural-network committee machine approach to the inverse kinematics problem solution of robotic manipulators. *Engineering with Computers* 30, 641-649.

[4] Poon JK, Lawrence PD, (1988) Manipulator Inverse Kinematics based on Joint Functions. *IEEE on Robotics and Automation* 2, 669-674.

[5] Martin JAH, Lope L, Santos M, (2009) A method to learn the inverse kinematics of multi-link robots by evolving neuro-controllers. *Neurocomputing* 72, 2806-2814.

[6] Tejomurtula S, Kak S, (1999) Inverse kinematics in robotics using neural network. *Information Sciences* 116, 147-164.

[7] Dash KK, Choudhury BB, Khuntia AK, Biswal BB, (2011) A Neural Network Based Inverse Kinematic Problem. *Recent Advances in Intelligent Computational Systems (RAICS)*, 471-476, Trivandrum, India.

[8] Huang H, Chen C, Wang P, (2012) Particle Swarm Optimization for Solving the Inverse Kinematics of 7-DOF Robotic Manipulators. *IEEE Systems, Man, and Cybernetics*, 3105-3110, Seoul, Korea.

[9] Ayyıldız M, Çetinkaya K, (2015) Comparison of four different heuristic optimization algorithms for the inverse kinematics solution of a real 4-DOF serial robot manipulator. *Neural Computing and Applications* 27, 825-836.

[10] Rokbani N, Alimi AM, (2013) Inverse Kinematics Using Particle Swarm Optimization, A Statistical Analysis, *Procedia Engineering* 64, 1602-1611.

[11] Rokbani N, Casals A, Alimi AM, (2015) IK-FA, a New Heuristic Inverse Kinematics Solver Using Firefly Algorithm, *Studies in Computational Intelligence* 575, 369-395.

[12] Köker R, (2013) A genetic algorithm approach to a neural-network-based inverse kinematics solution of robotic manipulators based on error minimization, *Information Sciences* 222, 528-543.

[13] Pham DT, Castellini M, Fahmy AA, (2008) Learning the Inverse Kinematics of a Robot Manipulator using the Bees Algorithm, *IEEE International Conference on Industrial Informatics*, 493-498, Daejeon, South Korea.

[14] Bansal JC, Singh PK, Saraswat M, Verma A, Jadon SS, Abraham A, (2011) Inertia Weight Strategies in Particle Swarm Optimization, *Third World Congress on Nature and Biologically Inspired Computing*, 640-647.

[15] Rokbani N, Alimi AM, (2012) IK-PSO, PSO Inverse Kinematics Solver with Application to Biped Gait Generation, *International Journal of Computer Applications* 58, 33-39.

[16] Wang X, Ming-Lin H, Yu-Hu C, (2008) On the use of differential evolution for forward kinematics of parallel manipulators, *Applied Mathematics and Computation* 205, 760-769.

[17] Rocha CR, Tonetto CP, Dias A, (2011) A comparison between the Denavit–Hartenberg and the screw-based methods used in kinematic modeling of robot manipulators, *Robotics and Computer-Integrated Manufacturing* 27, 723-728.

[18] Köker R, Çakar T, (2016) A neuro-genetic-simulated annealing approach to the inverse kinematics solution of robots: a simulation based study, *Engineering with Computers* 32 1-13.

[19] Huang HC, Xu SD, Hsu HS, (2014) Hybrid Taguchi DNA Swarm Intelligence for Optimal Inverse Kinematics Redundancy Resolution of Six-DOF Humanoid Robot Arms, *Mathematical Problems in Engineering*, 1, 1-9.

[20] Kung Y, Shu G, (2005) Design and implementation of a control IC for vertical articulated robot arm using SOPC technology, *IEEE International Conference on Mechatronics*, 532-536, Taipei, China.

[21] Gan JQ, Oyama E, Resales EM, Hu H, (2005) A complete analytical solution to the inverse kinematics of the Pioneer 2 robotic arm, *Robotica* 23, 123-129.

[22] Huang H, Yan J, (2007) A fast and smooth walking pattern generator of biped robot using Jacobian inverse kinematics, *IEEE Workshop on Advanced Robotics and Its Social Impacts*, 1-6, Hsinchu, Chinese.

[23] Kennedy J, Eberhart RC, (1995) Particle Swarm Optimization. *EEE international conference on neural networks*, 1942-1948, Perth, Australia.

[24] Sun J, Fang W, Palade V, Wu X, Xu W, (2011) Quantum-behaved particle swarm optimization with Gaussian distributed local attractor point. *Applied Mathematics and Computation* 218, 3763-3775.

[25] Poli R, Kennedy J, Blackwell T, (2007) Particle Swarm Optimization. *Swarm Intell* 1, 33-57.

[26] Bansal JC, Singh PK, Saraswat M, Verna A, Jadon SS, Abraham A, (2011) Inertia Weight Strategies in Particle Swarm Optimization. *Third World Congress on Nature and Biologically Inspired Computing (NABIC)*, 640-647, Salamanca, Spain.

[27] Chatterjee A, Siarry P, (2006) Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization, *Computers & Operations Research* 33, 859-871.

[28] Jiao B, Lian Z, Gu X, (2008) A dynamic inertia weight particle swarm optimization algorithm *Chaos, Solitons & Fractals* 37, 698-705.

[29] Harrison KR, Engelbrecht AP, Ombuki-Berman BM, (2016) Inertia Weight Control Strategies for Particle Swarm Optimization. *Swarm Intelligence* 10, 267-305.

[30] Shi Y, Eberhart RC, (1998) A modified particle swarm optimizer, *IEEE World Congress on Computational Intelligence*, 69-73.

[31] Eberhart RC, Shi Y, (2001) Tracking and optimizing dynamic systems with particle swarms *Evolutionary Computation* 1, 94-100.