

Iterative Technique for Solving the Inverse Kinematics Problem of Serial Manipulator

Abdelrady Okasha Elnady

Mechatronics Department, Faculty of Engineering, O6University, Egypt

Abstract The inverse Kinematics problem and obtaining its solution is one of the most important problems in robotics. It is quite complex, due to its non-linear formulations and having multiple solutions. It is not possible to formulate the solution of inverse kinematics problem in closed form, for all kind of robot configuration or for robots with high degrees of freedom. In the present work, an iterative technique to the IK problem solution of the serial manipulator is proposed. Iterative formulae for IK of 2-link and 3-link planar manipulators are presented. Joint variables for 2-link and 3-link manipulators are calculated using the present technique and compared with the exact one. The relative error is calculated for different iteration numbers. The results prove that the proposed algorithm is simple and efficient. The present technique can be applied to the serial manipulator with high degrees of freedom.

Keywords Inverse Kinematics, Serial manipulator, Iterative Techniques

1. Introduction

Kinematic problem solution is the first step for controlling the robot to carry out the desired task. Robotic manipulators are of many types; here only open chain serial manipulator is considered. Firstly, forward kinematics is evaluated, which can be easily analyzed using the Denavit-Hartenberg parameter. Next step is to evaluate inverse kinematics for obtaining joint variable for the defined pose of the robot. Closed form or analytical solution is the simplest one for inverse kinematics. But, it is not possible to obtain the analytical relation between joint variables and robot end effector position and orientation for all robot configurations, especially in case of high degrees of freedom manipulator due to its non-linear behavior. So, it is required to solve the inverse kinematics problem using another method.

There are several techniques for solving IK problems. These include; cyclic coordinate descent methods [1], quasi-Newton and conjugate gradient methods [1,2,3], and neural net and artificial intelligence methods. Utilization of Neural network (NN) and Fuzzy logic to solve the inverse kinematics problems is much reported in [4-9]. Li-Xin Wei *et al.* [10] and Rasit Koker *et al.* [11] proposed neural network based inverse kinematics solution of a robotic manipulator.

The cyclic coordinate descent method (CCD) is so popular because it is fast computationally, simple algorithmically,

and straight-forward technique for generating IK solutions that can run at interactive frame rates.

S R Buss [12] presented various Jacobian based methods demonstrating Pseudoinverse method being computationally fast but quality of approximation is poor. Pseudoinverse method gives very great end effector velocity near the singularity and also when the target is out of work space. M. R. de Gier [13] used Jacobian Pseudoinverse method for controlling robot having 6 DOF for application of 3D printing. It is noticed that, complexity in getting closed form solution increase with increase in DOF. In Jacobian Pseudoinverse method, the position is obtained by integrating the velocity as a result there would be a small drift in the solution, which can be reduced by reducing step size.

Jacobian transpose method is the simplest one because it utilizes Jacobian transpose instead of Jacobian inverse. This reduces the computation of matrix inversion. The main advantage of this method is faster convergence, minimum computation, and less complexity. The Jacobian inverse method requires large computation for matrix inversion but it has fast convergence property [13]. This method is not very popular due to limitation as stated before. It has another limitation during singularity, when Jacobian loss its full-rank and Jacobian inversion becomes impossible or gives large velocity near to singular configuration. Jacobian matrix is of size $(6 \times n)$, where n is the number of degree of freedom. So, when the degree of freedom is less than 6 the Jacobian matrix becomes rectangular or not of full rank. As a result, calculation of Jacobian inverse becomes impossible. So, Moore Penrose inversion is the best possible approximate solution also known as pseudo inversion.

* Corresponding author:

radynady@yahoo.com (Abdelrady Okasha Elnady)

Received: May 3, 2021; Accepted: May 12, 2021; Published: May 26, 2021

Published online at <http://journal.sapub.org/jmea>

The major common problem in the previous methods is a poor performance during singularity, which can be efficiently handled by Damped Least Square (DLS) method, by introducing a damping factor [14]. The damping factor damps the large velocity near singularity which makes the system robust against singularity. Also, the poor performance during singularity can be avoided by using a numerically stable technique for approximation.

Ignacy Dule *et al.* [15], had made a comparative study for various Jacobian based methods i.e. Jacobian transpose, Jacobian pseudo-inverse, Damped least square method (DLS), Modified Damped least square method and approximation methods. The comparison is done on the basis of a number of elementary operations to compute iteration. It is shown that Pseudoinverse requires large operation followed by damped least square method and followed by Jacobian transpose method. This method is computationally cheapest and damped least square method is more better with bit more computational cost. Similarly, in comparison based on time consumed for computation, Pseudoinverse takes maximum computation time followed by DLS and Jacobian transpose method for movement of the end effector on a specific path.

Dereli *et al.* [16] used two variants of particle swarm optimization (PSO) to calculate the inverse kinematics of a new 7-revolute jointed robot arm. The results obtained with Random Inertia Weight and Global Local Best Inertia Weight, are compared with the standard PSO. The PSO variables are much more effective than the standard PSO.

Dereli *et al.* [17], proposed a quantum behaved particle swarm algorithm for inverse kinematic solution of a 7-degree-of-freedom serial manipulator and the results have been compared with other swarm techniques such as firefly algorithm (FA), particle swarm optimization (PSO) and artificial bee colony (ABC). The Quantum behaved PSO has yielded results that are much more efficient than standard PSO, ABC and FA. The advantages of the improved algorithm are the short computation time, fewer iterations and the number of particles.

In the present work, an algorithm to the IK problem solution of the serial manipulator is proposed. Iterative formulae for IK of 2-link and 3-link planar manipulators are presented. Joint variables for 2-link and 3-link manipulators are calculated using the present technique and compared with the exact one. The relative error is calculated for different iteration numbers. The results prove that the proposed algorithm is simple and efficient. The present technique can be applied to the serial manipulator with high degrees of freedom.

2. Forward Kinematics (FK)

2.1. FK of 2-link Planar Manipulator

The forward kinematics of 2-link planar manipulator can be obtained directly from geometry Fig. 1. The position coordinates (x, y) of the end effector is given in equations (1)

and (2).

$$x = l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2) \quad (1)$$

$$y = l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2) \quad (2)$$

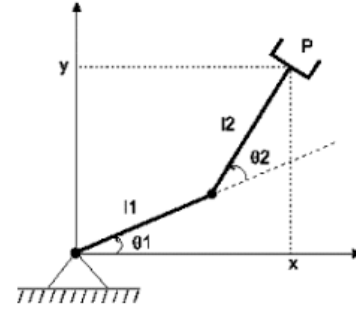


Figure 1. 2-link planar manipulator

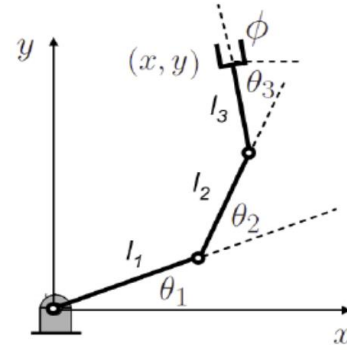


Figure 2. 3-link planar manipulator

2.2. FK of 3-link Planar Manipulator

The forward kinematics of 3-link planar manipulator can be obtained directly from geometry Fig. 2. The position coordinates (x, y) of the end effector is given in equations (3) and (4). The orientation of the end effector is given by equation (5)

$$x = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3) \quad (3)$$

$$y = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3) \quad (4)$$

$$\phi = \theta_1 + \theta_2 + \theta_3 \quad (5)$$

3. Inverse Kinematics

3.1. IK of 2-link Planar Manipulator

The iteration formula for joint variables of the 2-link planar manipulator can be written in terms of the position coordinates (x, y) of the end effector as:

$$\theta_1^k = \tan^{-1} \frac{y - l_2 \sin(\theta_1^k + \theta_2^k)}{x - l_2 \cos(\theta_1^k + \theta_2^k)} \quad (6)$$

$$\theta_2^k = -\theta_1^k + \tan^{-1} \frac{y - l_1 \sin \theta_1^k}{x - l_1 \cos \theta_1^k} \quad (7)$$

3.2. IK of 3-link Planar Manipulator

The iteration formula for joint variables of the 3-link planar manipulator can be written in terms of the position

coordinates (x, y) of the end effector as:

$$\theta_1^k = \tan^{-1} \frac{y - l_2 \sin(\theta_1^{k-1} + \theta_2^{k-1}) - l_3 \sin(\theta_1^{k-1} + \theta_2^{k-1} + \theta_3^{k-1})}{x - l_2 \cos(\theta_1^{k-1} + \theta_2^{k-1}) - l_3 \cos(\theta_1^{k-1} + \theta_2^{k-1} + \theta_3^{k-1})} \quad (8)$$

$$\theta_2^k = -\theta_1^{k-1} + \tan^{-1} \frac{y - l_1 \sin \theta_1^{k-1} - l_3 \sin(\theta_1^{k-1} + \theta_2^{k-1} + \theta_3^{k-1})}{x - l_1 \cos \theta_1^{k-1} - l_3 \cos(\theta_1^{k-1} + \theta_2^{k-1} + \theta_3^{k-1})} \quad (9)$$

$$\theta_3^k = \theta_1^{k-1} - \theta_2^{k-1} + \tan^{-1} \frac{y - l_1 \sin \theta_1^{k-1} - l_2 \sin(\theta_1^{k-1} + \theta_2^{k-1})}{x - l_1 \cos \theta_1^{k-1} - l_2 \cos(\theta_1^{k-1} + \theta_2^{k-1})} \quad (10)$$

If the orientation of the end effectors is specified, the iteration formula for joint variable θ_3 is given by eq. (11).

$$\theta_3^k = \phi - \theta_1^k - \theta_2^k \quad (11)$$

The iteration formulae for joint variables of 2-link planar manipulator given in equations (6) and (7) are applied to obtain the inverse kinematics of 2-link planar manipulator. Given an initial value to joint variable θ_1 , the joint variable θ_2 is calculated. Using the obtained θ_2 to calculate θ_1 and repeat until certain acceptable error. The configuration of 2-link planar manipulator with $l_1 = 5$ and $l_2 = 3$ are shown in Fig. 3. The coordinates are $x = 6$ and $y = 4$. The absolute error in x and y coordinates, is specified to be less than 10^{-3} . The initial value of θ_1 is zero.

The absolute error in the x and y coordinates with the number of iterations is presented in Table 1. It is obvious from the results that the method is highly convergent. The error is exponentially decreasing as shown in Fig. 4.

4. Results and Discussion

4.1. 2-link Planar Manipulator

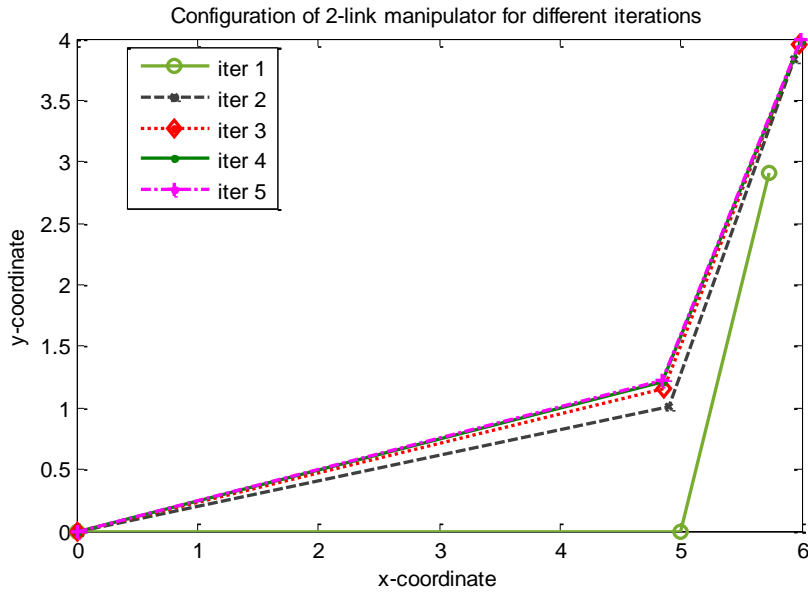


Figure 3. Configurations of 2-link planar manipulator for different iterations

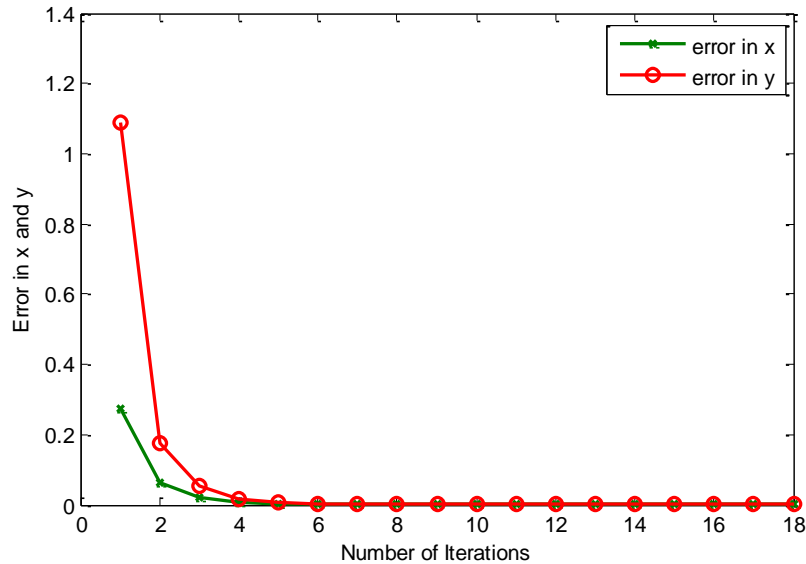


Figure 4. Variation of error in x and y coordinates with the number of iterations for 2-link planar manipulator

Table 1. Error in the x and y coordinates with the number of iterations for 2-link manipulator

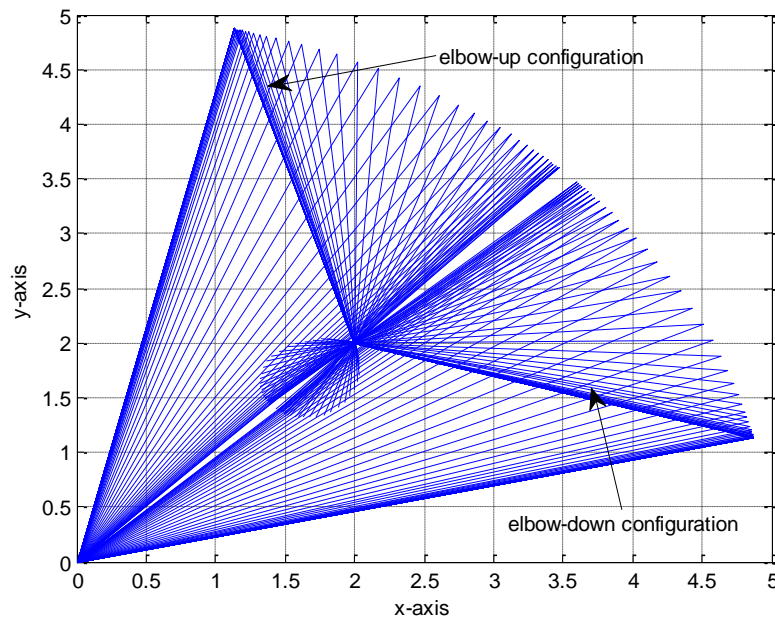
No of iteration	6	9	11	13	15	18
ϵ_x	9.4661e-04	4.4081e-05	5.7124e-06	7.4032e-07	9.5945e-08	4.4764e-09
ϵ_y	0.0023	1.0580e-04	1.3710e-05	1.7768e-06	2.3027e-07	1.0743e-08

Table 2. Values of θ_1 and θ_2 with iteration number for 2-link manipulator

Iter #	1	2	3	4	5	6	7	8
θ_1	0	11.6762	13.4431	13.9721	14.1515	14.2147	14.2374	14.2455
% ϵ_r	100	18.06194	5.66267	1.95040	0.691456	0.247948	0.08865	0.031808
θ_2	75.9638	58.0553	54.7215	53.6834	53.3269	53.2006	53.1555	53.1392
% ϵ_r	42.97695	9.27007	2.99528	1.04140	3.70E-01	1.33E-01	4.78E-02	1.71E-02

Table 2. Values of θ_1 and θ_2 with iteration number for 2-link manipulator. Cont.

Iter #	9	10	11	12	13	14	15	16
θ_1	14.2484	14.2494	14.2498	14.2500	14.2500	14.2500	14.2500	14.2500
% ϵ_r	1.15E-02	4.44E-03	1.63E-03	2.29E-04	2.29E-04	2.29E-04	2.29E-04	2.29E-04
θ_2	53.1334	53.1313	53.1305	53.1303	53.1302	53.1301	53.1301	53.1301
% ϵ_r	6.21E-03	2.25E-03	7.48E-04	3.72E-04	1.84E-04	4.42E-06	4.42E-06	4.42E-06

**Figure 5.** Effect of initial value on the 2-link manipulator configuration

The values of joint variables θ_1 and θ_2 with iteration number are presented in Table 2. The relative error (ϵ_r) for each iteration is presented. The exact values of θ_1 and θ_2 are $\theta_1 = 14.2500327^\circ$ and $\theta_2 = 53.13010235^\circ$.

The initial value of θ_1 is arbitrary, and the solution is highly convergent to the nearest configuration. The only exception is if the target position near the singular position and the initial value of θ_1 equals $\tan^{-1}(y/x)$. If the target position is $(2, 2)^T$ say, and the initial value of θ_1 is 45° the robot can't reach to the target position. Slight changes in the initial value of θ_1 avoid this status. Fig. 5 shows the configurations of the manipulator if the initial value is 44° and 46° . The manipulator goes to the elbow-down configuration for 44° initial value and goes to elbow-up configuration for 46° initial value. The error is 10^{-6} .

4.2. 3-link Planar Manipulator

The iteration formulae for joint variables of 3-link planar manipulator given in equations (8), (9) and (10) are applied to obtain the inverse kinematics of 3-link planar manipulator. Given an initial value to joint variable θ_1 and θ_2 , the joint variable θ_3 is calculated. Using the obtained θ_3 to calculate θ_1 and θ_2 , then repeat until certain acceptable error. The configuration of 3-link planar manipulator with $l_1 = 5$, $l_2 = 3$ and $l_3 = 2$ are shown in Fig. 6. The coordinates are $x = 7$, and $y = 6$. The absolute error in x and y coordinates, is specified to be less than 10^{-6} . The initial value of θ_1 is zero.

The absolute error in the x and y coordinates with the number of iterations is presented in Table 3. It is obvious from the results that the method is highly convergent. The error is exponentially decreasing as shown in Fig. 7.

The values of joint variables θ_1 , θ_2 and θ_3 with iteration errors are presented. number are presented in Table 4. The absolute and relative

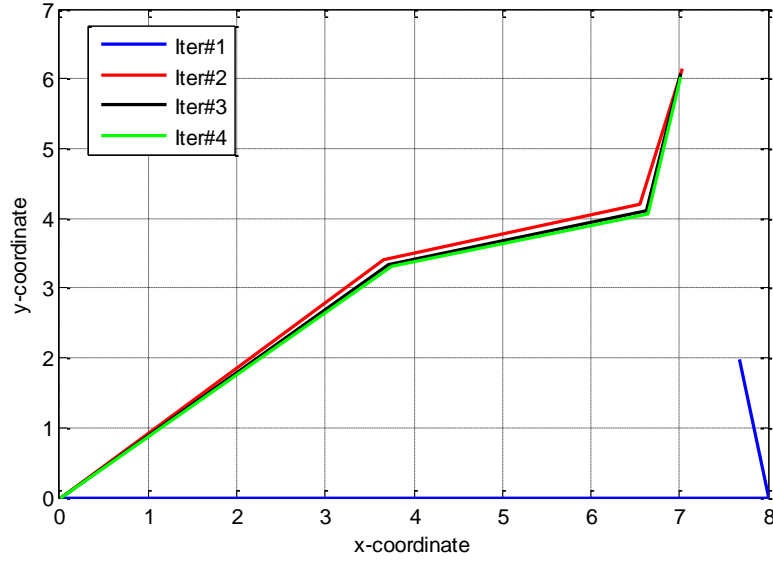


Figure 6. Configurations of 3-link planar manipulator for different iterations

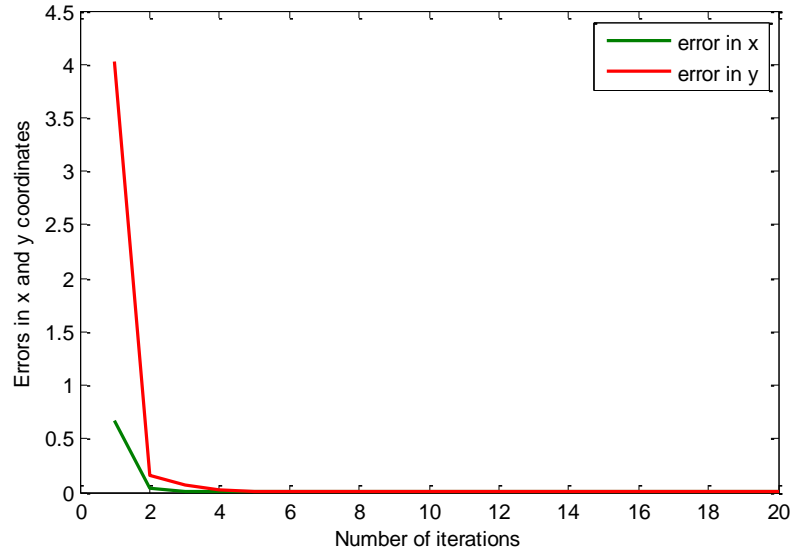


Figure 7. Variation of error in x and y coordinates with the number of iterations for 3-link planar manipulator

Table 3. Error in the x and y coordinates with the number of iterations for 3-link manipulator

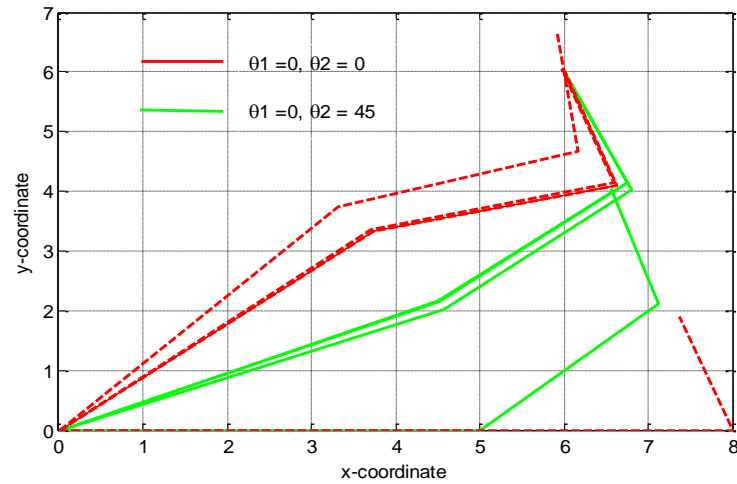
No of iteration	6	8	11	13	15	18
Error in x coord. ε_x	6.4655e-04	8.7826e-05	4.4175e-06	6.0212e-07	8.2074e-08	4.1304e-09
Error in y coord. ε_y	0.3900e-03	5.3019e-04	2.6707e-05	3.6406e-06	4.9625e-07	2.4973e-08

Table 4. Values of θ_1 , θ_2 and θ_3 with iteration number for 3-link manipulator

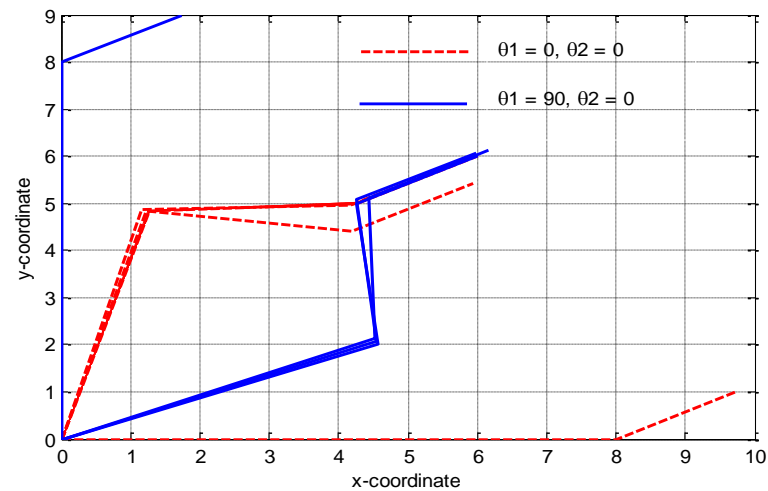
Iter #	1	2	3	4	5	6	7	8
θ_1	0.0000	42.9317	41.9348	41.4535	41.2587	41.1842	41.1556	41.1441
% error ε_r	N/A	100	2.3862	1.1530	0.47716	0.18425	6.9188E-2	2.570E-2
θ_2	0.0000	-27.3415	-27.1525	-27.0092	-26.9462	-26.9233	-26.9118	-26.9061
% error ε_r	N/A	100	0.7039	0.5173	0.2407	9.676E-2	3.686E-2	1.377E-2
θ_3	99.4597	60.2923	63.8791	65.4146	66.0162	66.2397	66.3256	66.3600
% error ε_r	N/A	64.9625	5.6161	2.3411	0.9116	0.34378	0.12795	4.738E-2

Table 4. Values of θ_1 and θ_2 with iteration number for 3-link manipulator. Cont.

Iter #	9	10	11	12	13	14	15	16
θ_1	41.1384	41.1384	41.1384	41.1384	41.1384	41.1384	41.1384	41.1384
%error ε_r	9.513E-3	3.515E-3	1.298E-3	4.794E-4	1.770E-4	6.534E-5	2.413E-5	8.907E-6
θ_2	-26.9061	-26.9061	-26.9061	-26.9061	-26.9061	-26.9061	-26.9061	-26.9061
% error ε_r	5.105E-3	1.887E-3	6.973E-4	2.575E-4	9.508E-5	3.510E-5	1.296E-5	4.785E-6
θ_3	66.3714	66.3772	66.3772	66.3772	66.3772	66.3772	66.3772	66.3772
% error ε_r	1.751E-2	6.468E-3	2.388E-3	8.818E-4	3.256E-4	1.202E-4	4.438E-5	1.638E-5

**Figure 8.** Effect of initial values of θ_1 and θ_2 on the 3-link manipulator configurations**Table 5.** Values of θ_1 , θ_2 and θ_3 of 3-link planar manipulator for exact and present solutions

Joint Variables			θ_1	θ_2	θ_3
Manipulator Configuration	Elbow-up	Present Technique	75.256399565186499	-72.110653339522145	26.854253774335646
		Excat Solution	75.2563995651971	-72.1106533395283	26.8542537743312
		Relative error ε_r	1.42191E-13	8.59226E-14	1.63783E-13
	Elbow-down	Present Technique	23.776131258239133	72.110653339292313	-65.88678459753144
		Excat Solution	23.7761312580947	72.1106533395283	-65.886784597623
		Relative error ε_r	6.07332E-12	3.27274E-12	1.39031E-12

**Figure 9.** Effect of initial values of θ_1 and θ_2 on the 3-link manipulator configurations

The initial value of θ_1 and θ_2 are arbitrary, and the solution is highly convergent to the nearest configuration. The configurations depend on the initial values of θ_1 and θ_2 .

Fig. 8 shows two configurations for two initial values of θ_1 and θ_2 . The target position is (6, 6). Elbow down-elbow down configuration corresponds to $\theta_1 = 0^\circ$ and $\theta_2 = 45^\circ$

which is the green configuration. Elbow up-elbow down configuration corresponds to $\theta_1 = 0^\circ$ and $\theta_2 = 0^\circ$ which is the red configuration. The error in each case is less than 10^{-8} .

If the orientation of the end effector is specified, the inverse kinematics equations are (8), (9) and (11). The inverse kinematics results for 3-link planar manipulator, knowing the end-effector orientation is shown in Fig. 9. The values of θ_1 , θ_2 and θ_3 are presented in Table 5. The target position is $(x, y) = (6, 6)$. The orientation angle ϕ is 30° . The joint variables θ_1 , θ_2 and θ_3 are obtained using the present technique with error adapted to be less than or equal 10^{-12} and compared with the exact one. The number of required iterations is 18. It is clear that the present technique is highly convergent and power full tool to find the inverse kinematics parameters of 3-link planar manipulator.

The initial value of θ_1 is 90° and θ_2 is 0° for elbow-down configuration. The initial value of θ_1 is 0° and θ_2 is 0° for elbow-up configuration as shown in Fig. 9.

5. Conclusions

The mapping from the Cartesian space to the joint space is done by the iteration technique. An iteration algorithm to the IK problem solution of the serial manipulator is proposed. The iteration formulae for the IK of 2-link and 3-link planar manipulator are presented. The benefit of this method is that it is very simple and it can be used with any other serial robotic arm. Joint variables for 2-link and 3-link manipulators are calculated using the present technique and compared with the exact one. The relative error is calculated for different iteration numbers. Results show that the method is highly convergent and stable. Its convergence rate is exponentially as shown in the figures. The advantage of the method is that it easily tractable close to the singular configurations of the manipulator. Results of the application of the method on 2-link and 3-link manipulators are shown and clearly show that the technique is simple and accurate.

REFERENCES

- [1] Wang, L-CT, and Chih-Cheng Chen. "A combined optimization method for solving the inverse kinematics problems of mechanical manipulators." *IEEE Transactions on Robotics and Automation* 7.4 (1991): 489-499.
- [2] Deo, Arati S., and Ian D. Walker. "Adaptive non-linear least squares for inverse kinematics." [1993] *Proceedings IEEE International Conference on Robotics and Automation*. IEEE, 1993.
- [3] Zhao, Jianmin, and Norman I. Badler. "Inverse kinematics positioning using nonlinear programming for highly articulated figures." *ACM Transactions on Graphics (TOG)* 13.4 (1994): 313-336.
- [4] Nedungadi, A, "Application of fuzzy logic to solve the robot inverse kinematics problem," *Proceeding of 4th World Conf. on Robotics Research*, 13, pp. 1-14, 1991.
- [5] David W. Howard and Ali Zilouchian, "Application of Fuzzy Logic for the Solution of Inverse Kinematics and Hierarchical Controls of Robotic Manipulators," *Journal of Intelligent and Robotic Systems*, 23, 217-247, 1998.
- [6] Sreenivas Tejomurtula, Subhash Kak, "Inverse kinematics in robotics using neural networks," *Information Sciences*, 116, 147-164, 1999.
- [7] Yang Ming Lu, Lu Guizhang, Li Jiangeng, "An Inverse Kinematics Solution for Manipulators," *Proceedings of IEEE*, Vol.4, 400-404, 2001.
- [8] Tiberiu Vesselenyi, Simona Dzitac, Ioan Dzitac, Misu-Jan Manolescu, "Fuzzy and Neural Controllers for a Pneumatic Actuator," *International Journal of Computers, Communications and Control*, Vol. II, No. 4, pp. 375-387, 2007.
- [9] Alavandar, Srinivasan, and M. J. Nigam. "Inverse kinematics solution of 3DOF planar robot using ANFIS." *Int. J. of Computers, Communications & Control* 3 (2008): 150-155.
- [10] Li-Xin Wei, Hong-Rui Wang, Ying Li, "A new solution for inverse kinematics of manipulator based on neural network," *Proceedings of the Second International Conference on Machine Learning and Cybernetics*, Xian, 3(5), 1201-1203, November 2003.
- [11] Rasit Koker, Cemil Oz, Tark Cakar, Huseyin Ekiz, "A study of neural network based inverse kinematics solution for a three-joint robot," *Robotics and Autonomous Systems*, 49, 227-234, 2004.
- [12] Buss, Samule R, "Introduction to Inverse Kinematics with Jacobian Transpose, Pseudoinverse and Damped Least Squares methods," *University of California, San Diego*, 2009.
- [13] Gier, M. R. de, "Control of a robotic arm: Application to on-surface 3Dprinting," *Delft University of Technology*, 2015.
- [14] Bruno Siciliano, Stefan Chilverini, "Review of the Damped Least- Squares Inverse Kinematics with Experiments on an Industrial Robot Manipulator," *IEEE Transactions On Control Systems Technology*, vol. 2, no. 2, 1994.
- [15] Duleba, Ignacy, and Michał Opalka. "A comparison of Jacobian-based methods of inverse kinematics for serial robot manipulators." *International Journal of Applied Mathematics and Computer Science* 23.2 (2013).
- [16] Dereli, Serkan, and Raşit Köker. "IW-PSO approach to the inverse kinematics problem solution of a 7-DOF serial robot manipulator." *Sigma J Eng Nat Sci* 36.1 (2018): 77-85.
- [17] Dereli, Serkan, and Raşit Köker. "A meta-heuristic proposal for inverse kinematics solution of 7-DOF serial robotic manipulator: quantum behaved particle swarm algorithm." *Artificial Intelligence Review* 53.2 (2020): 949-964.