MULTISCIA JOURNALS PUBLISHERS

FRONTIERS IN CHEMICAL AND LIFE SCIENCES

ISSN: (3065- 4238)

https://multisciajournals.com/journals/index.php/fcls
editor.fcls@gmail.com

Frontiers in Chemical and Life Sciences
Volume 2 Issue 1 2026

Genetic Algorithms for the Resolution of the Inverse Kinematics Problem

Shaherhji Momani, S. Abo-Hammoursdc*
Department of Chemical and Life Sciences

Article Info

***Received:* 27-12-2024** ***Revised:*03-01-2026** ***Accepted:*14-01-2026** ***Published:*25-01-2026**

Abstract:

The paper presents a genetic algorithm (GA) solution to the inverse kinematics problem of robot manipulators. Two versions of GA are used, the conventional GA and the continuous GA. The inverse kinematics problem is defined as an optimization problem based on the idea of minimizing the accumulative path deviation in a workspace without obstacles. The simulation results show that the continuous GA performs better than the conventional GA in every aspect. The continuous GA is superior because it always provides faster and smoother solutions than the conventional GA.

Keywords: Inverse kinematics problem; Robot manipulators; genetic algorithms

1 Introduction

On the basis of the Darwinian principles of biological evolution, GAs are essentially generate-and-test AI optimization algorithms. Researchers have focused a lot of emphasis on GAs [4,5, 6] despite the fact that there are alternative AI approaches [1, 2, 3]. A fitness assessment function, genetic operators, and an initial population of solutions are the three main components of a genetic algorithm that may solve any optimization problem [7]. As a consequence of these causes, there are a plethora of GA variations documented in the literature. The smoothness of the solution curve in robot manipulators is an example of a situation where precise solutions are required. Later on, it will become clear that the suggested CGA resolves this issue in a way that the traditional CA cannot. The inverse kinematics problem has grown into a basic issue in robotics, with several suggested solutions, due to its well-known use in controlling the posture of an articulated body [8]. It has been found that most of these strategies, which include geometric, iterative, and algebraic approaches, are insufficient for duplicate robots [9]. A solution to the robotics inverse kinematics issue using neural networks has recently attracted a lot of interest [9, 10, 11]. When the kinematics equations are complicated, highly nonlinear, coupled, and have multiple solutions in terms of these robot manipulators, solving the inverse kinematics problem can be challenging [12]. This has motivated the GA approach to investigate whether this kind of problem can be solved for robot motion planning. There are two primary approaches to the planning of robot manipulators' movements: continuous/Cartesian and point-to-point. The goal of continuous/Cartesian motion planning is to achieve the required motion in Cartesian space by determining the set of joint angles or velocities using the manipulator's inverse kinematics equations [9, 10, 11, 13, 14].

The majority of studies have been on either continuously planning the motion of redundant manipulators or on generating point-to-point trajectories for redundant and non-redundant manipulators. Given the boundless variety of possible outcomes in joint space for the two classes described above, an optimization strategy is necessary to fully exploit this fact in order to improve robot motion performance through the use of some minimization or maximization criteria such as minimizing the time of motion [3, 15, 16], minimizing the jerk [17, 18], minimizing the torque [19], or minimizing the consumed energy [20]. The non-redundant manipulators continuous path planning has received a little attention among the researchers community. This includes the solution of the inverse kinematics of the non-redundant manipulator which has unique or multiple feasible solutions for the problem depending on the manipulator’s configuration and the joints limits. The solution strategies of the manipulators inverse kinematics problem are divided into two main classes; closed-form solutions and numerical solutions [21]. The closed-form solution for the inverse kinematics problem is generally difficult to derive for general serial manipulators.

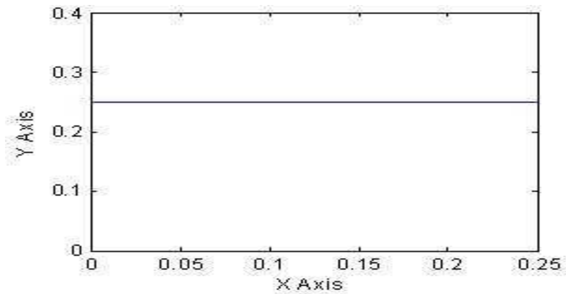


Fig. 1: Cartesian geometric path for the 3R planar manipulator.

L1 = L2 = L3 = 0.5 meter are the link parameters of the 3R redundant manipulator. If N = 2, M = 3, and θlower(h) = −180 and θupper(h) = 180 for all h = 1, 2, and 3, then this is the case. For manipulators, the forward kinematics model is provided using traditional crossover methods. Most of the time, these findings are the result of the fact that traditional genetic algorithms use path-level operators that are local in nature, such as those for initialization, crossover, mutation, extinction, and immigration. Because of this, the routes of the joints experience breaks or oscillating values between successive places in the paths. In contrast, the CGA operators are applied at the level of the joints' paths and are global in nature. The consequence is that the joint values never leap in a way reminiscent of a step function since the transitions between them are always smooth.

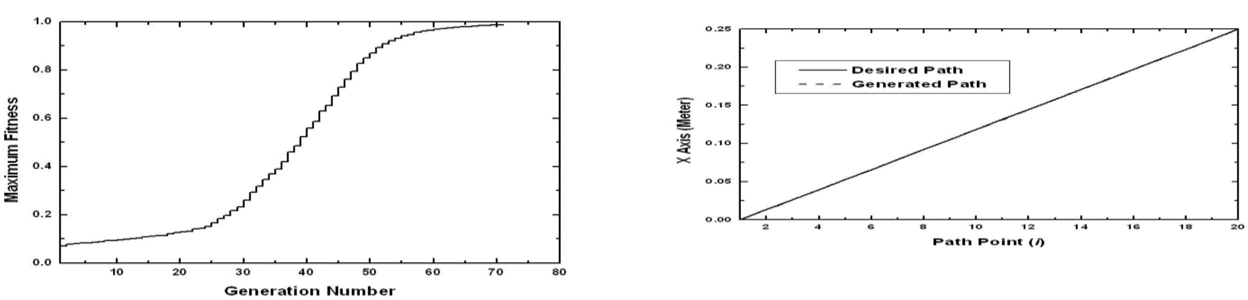
The next step is to examine how the hybrid algorithm's convergence time and the character of the produced joint pathways are affected by the conventional and continuous versions of the initialization phase, crossover operator, and mutation operator. Table 1 provides the necessary information for the 3R manipulator. The table clearly shows that the smoothness or nonsmoothness of the solution curves is most affected by the initialization phase. So, for conventional initialization, the paths of the joints are oscillatory with large or medium magnitude oscillations, while for continuous initialization, the paths of the joints are either smooth or oscillatory with small magnitude oscillations. The CGA (i.e., continuous initialization types, crossover, and mutation) achieves the best convergence speed and least execution time.

Table 1: Step-by-step switching to CGA for the 3R manipulator.

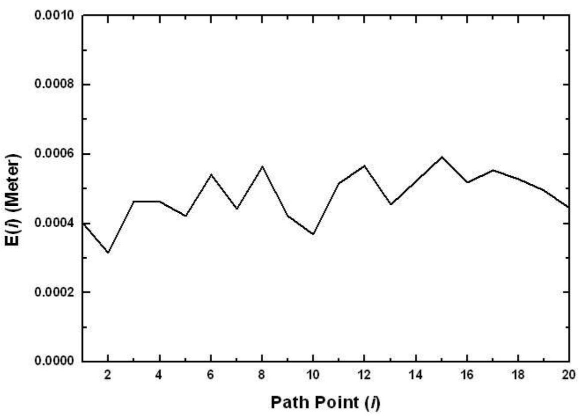
Initialization Type	Crossover Type	Mutation Type	Average Execution Time	Average Number of Generations	Nature of Joints Paths
Conventional	Conventional	Conventional	487.72	124	Oscillations with Large Magnitude
Conventional	Conventional	Continuous	390.06	105	Oscillations with Medium Magnitude
Conventional	Continuous	Conventional	295.14	83	Oscillations with Large Magnitude
Conventional	Continuous	Continuous	486.1	144	Oscillations with Medium Magnitude
Continuous	Conventional	Conventional	188.89	53	Oscillations with small Magnitude
Continuous	Conventional	Continuous	191.2	56	Oscillations with small Magnitude
Continuous	Continuous	Conventional	181.57	55	Oscillations with small Magnitude
Continuous	Continuous	Continuous	148.58	49	Smooth Solution Curves

Table 2: Effect of the degree of redundancy on the convergence speed of the conventional genetic algorithm.

Number of Manipulator's Links	Average Execution Time(Seconds)	Average Number of Generation	Average Time per Generation(Seconds)
4	242.46	46	5.25
6	459.43	57	8.05
8	677.43	63	10.74
10	981.23	75	13.08



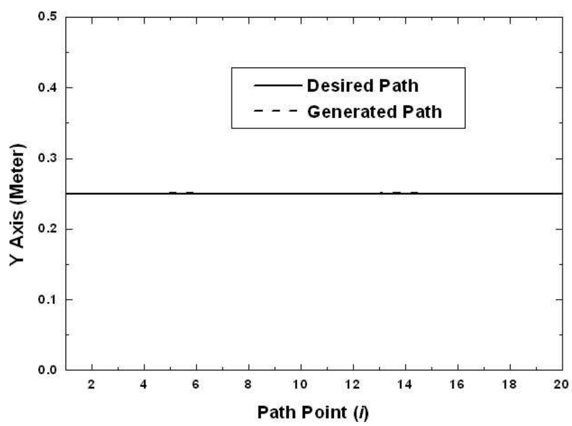
(a)



(b)

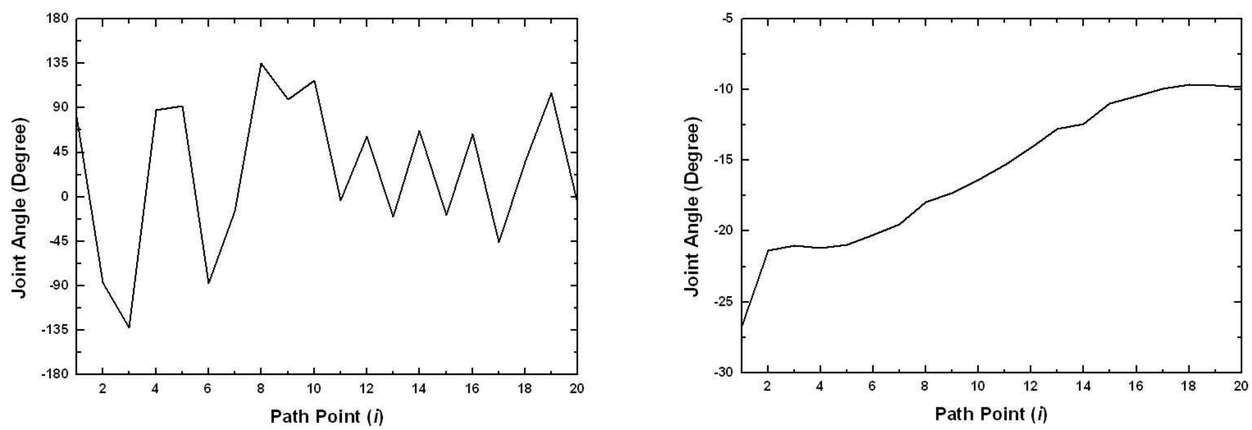
Fig. 2: (a) Evolutionary progress plot for the best-of-generation individual for the 3R planar manipulator, (b) corresponding path point deviation.

(a)

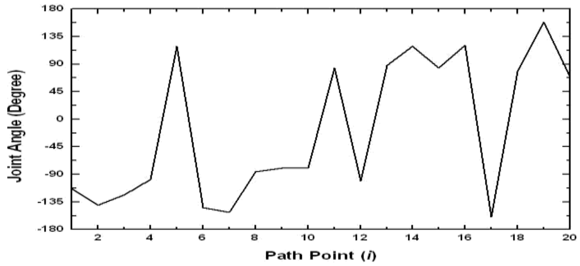


(b)

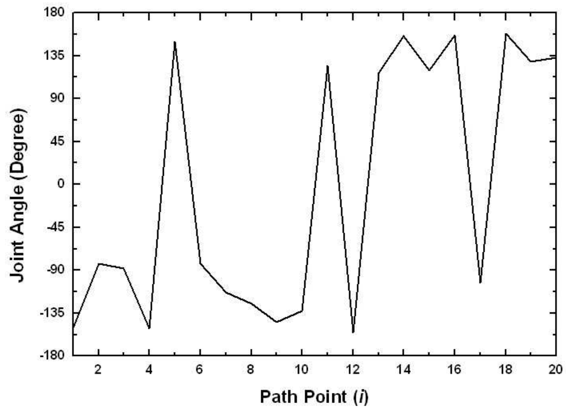
Fig. 3: Desired and generated Cartesian path for the 3R planar manipulator in (a) X-plane, (b) Y-plane.



(a)



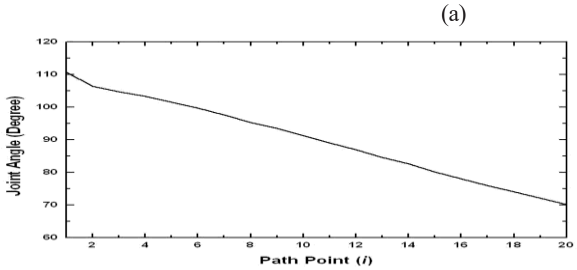
(b)



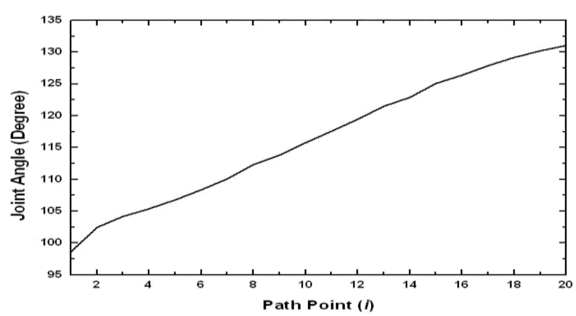
(b)

Fig. 4: Joints paths of the 3R manipulator using conventional GA: (a) 1st Joint, (b) 2nd Joint, (c) 3rd Joint.

(a)



(b)



(b)

Fig. 5: Joints paths of the 3R manipulator using CGA for (a) 1st Joint, (b) 2nd Joint, (c) 3rd Joint.

Table 3: Effect of the degree of redundancy on the convergence speed of the CGA.

Number of Manipulator's Links	Average Execution Time(Seconds)	Average Number of Generation	Average Time per Generation(Seconds)
4	101.39	48	2.10
6	122.6	46	2.65
8	164.27	48	3.41
10	201.06	47	4.27

Table 4: Number of knots effect on the convergence speed of the conventional GA for the 3R manipulator.

Number of Knot	Average Execution Time(Seconds)	Average Number of Generation	Average Time per Generation(Seconds)
20	326	76	4.28
40	1856	217	8.55
60	3981	317	12.55
80	6567	434	15.13
100	15563	840	18.52

Table 5: Number of knots effect on the convergence speed of the CGA for the 3R manipulator.

Number of Knot	Average Execution Time(Seconds)	Average Number of Generation	Average Time per Generation(Seconds)
20	69	50	1.38
40	170	69	2.46
60	275	75	3.66
80	349	74	4.71
100	469	78	6.01

The typical initialization, continuous crossover, and continuous mutation situation produces the highest number of generations needed for convergence for the 3R manipulators. Table 1 shows that these hybrid schemes still produce large-magnitude oscillations when conventional initiation, continuous crossover, and conventional mutation are applied. Given that the typical mutation process disrupts the smoothness attained by the continuous crossover procedure, this outcome is not surprising. Figure 5 shows the joint pathways for the 3R manipulator's first, second, and third joints as they are represented by CGA. Clearly, the net displacement of the joints is minimized since the resultant solution curves in joint space are smooth and do not transition between the two feasible solutions. The next thing that is explored is how the convergence speed of the conventional and CGAs are affected by the degree of redundancy (number of connections) of the planar redundant manipulator. In both methods, the number of planar manipulator connections, M, is raised from 4 to 10 in 2 stages, with the link length set as Li = 1/Mmeter f ori = 1, 2, up to M. Both Table 2 and Table 3 provide pertinent data obtained from the CGA and traditional genetic algorithms, respectively, for the issue with route generation before. It is evident from these tables that traditional genetic algorithms need an increasing average number of generations to reach convergence as the number of linkages rises, but CGAs are parameter insensitive, meaning that the number of generations remains relatively constant. Furthermore, traditional genetic algorithms often take twice as long as CGAs do every generation on average. This demonstrates that CGA not only produces pathways with smooth joints, but also reduces the number of generations needed for convergence and, on average, uses almost half as much time per generation as standard GA. Lastly, the study examines how the convergence speed of the conventional and CGAs for both manipulators is affected by the number of knots along the provided route generating issue. Both methods increase the number of knots from 20 to 100 in increments of 20. Table 4 shows the pertinent data using the normal genetic algorithm for the 3R manipulator, whereas Table 5 shows the data using the CGA. There is a clear correlation between the number of knots along the Cartesian path and the average number of generations needed for convergence using conventional genetic algorithms, but CGAs show a nearly constant average generational requirement independent of the number of knots along the Cartesian path. In other words, the standard genetic algorithm requires an additional 76–840 generations when the number of knots is raised from 20–100, but the CGA only requires 50–78 generations.

4 Conclusion

Robot manipulators' inverse kinematics issue is solved using CGA in this research. A comparison between the CGA and the conventional GA revealed that the former produced joint paths with numerous switching points in solutions involving non-redundant manipulators, whereas the latter produced paths with highly oscillatory dynamics, leading to extremely large net displacements in both systems. Giving thought to the problems with the

traditional GA, the CGA operators (initialization phase, crossover, mutation) were developed to maintain exceptional accuracy throughout the Cartesian route and produce pathways with smooth joints. The results show that the smoothness of the joint pathways is most affected during the initiation phase. Compared to the traditional GA, the CGA accomplishes convergence much more quickly, both in terms of the average execution time and the number of generations needed for convergence.

Potential Forgeries Regarding this paper's publishing, the authors state that they have no conflicts of interest.

References

[1] Zhixiang Chen and Feilong Cao published an article in the Journal of Computational Analysis and Applications on the logistic function's characteristics and their use in neural network approximation. 15–6 (2013), pages 1046–1056, pp. Journal of Computational Analysis and Applications, [2] Kun Liu and Zeng-Tai Gong, Numerical solution of completely fuzzy linear matrix equations. pp. 1026-1035, volume 15, issue 6, 2013.

[3] K. Demirli, M. Balazinski, L. Baron, and A. Khoukhi
Engineering Applications of Artificial Intelligence, vol. 21, no. 8, 2008, pp. 974–984, presents a hierarchical neuro-fuzzy system for near-optimal-time trajectory planning of redundant manipulators. [4] In the International Journal of Computational Intelligence Systems, volume 1, issue 1, pages 353-260, O. Liu, Y. Yu, S. Liu, and M. Zhang discuss a strategy based on genetic algorithms for obstacle avoidance using a class of underactuated robot manipulators.
[5] In Communications in Nonlinear Science and Numerical Simulation, M.D. Marcos, J.A.T. Machado, and T.P. Azevedo-Perdicoulis discuss the use of evolutionary algorithms for trajectory planning of duplicate manipulators. pages 2858–2869, volume 14, 2009).
[6] In the Journal of Mechatronics, L.F. Tian and C. Collins presented a genetic algorithm-based approach to robot trajectory planning in 2004 (volume 14, pages 455-770).
The book "Genetic Algorithms in Search, Optimization, and Machine Learning" was written by D.E. Goldberg and published in 1989 by Addison-Wesley in New York.
[8] Samuel R. Buss, "Introduction to Inverse Kinematics with Jacobian Transpose, Pseudoinverse and Damped Least Squares methods," 2009, http://www.math.ucsd.edu/ sbuss/ResearchWeb/ikmethods/ iksurvey.pdf. [9] Rait Koker, Tark akar, Yavuz Sari, A neural-network committee machine approach to the inverse kinematics problem solution of robotic manipulators. Computer Engineering, 2013, DOI:10.1007/s00366-013-0313-2.
A new three-degrees-of-freedom (DOF) actuator parallel manipulator was analyzed kinematically using an AI technique [10] by Zhang and Lei. Robotics Comput Integr Manuf 27:157163, 2011.
[11] Martin JAH, Lope J, Santos M, A method to learn the inverse kinematics of multi-link robots by evolving neuro-controllers. Neurocomputing 72:28062814, 2009.
[12] Fu Z, Yang W, Yang Z, Solution of Inverse Kinematics for 6R Robot Manipulators With Offset Wrist Based on Geometric Algebra. J Mech Robot. 5(3) pp. 310081-310087, 2013.

[13] Z.S. Abo-Hammour, Advanced continuous genetic algorithms and their applications in the motion planning of robotic manipulators and the numerical solution of boundary value problems, Ph.D. Thesis, Quiad-Azam University, 2002. [14] Hasan AT, Hamouda AMS, Ismail N, Al-Assadi HMAA, An adaptive-learning algorithm to solve the inverse kinematics problem of a 6 D.O.F. serial robot manipulator. Adv Eng Softw 37:432438, 2006. [15] B. Li, X.J. Yang, J.G. Zhao, and P. Yan, Minimum time trajectory generation for a novel robotic manipulator, International Journal of Innovative Computing Information and Control. 5 (2009), pp.369-378 [16] J.Y. Kim, D.H. Kim, and S.R. Kim, On-line minimum-time trajectory planning for industrial manipulators. International Conference on Control, Automation and Systems, Seoul, Korea, 2007 [17] J. Yang, J. Kim, E.P. Pitarch, and K. Abdel-Malek, Optimal trajectory planning for redundant manipulators based on minimum jerk, 32nd Annual Conference on Mechanisms and Robotics, New York, NY, 2008. [18] P.F. Huang, Y.S. Xu, and B. Liang, Global minimum-jerk trajectory planning of space manipulator, International Journal of Control Automation and Systems. 4 (2006), pp. 405-413 [19] C.J. Wu, Minimum-Torque trajectory planning of redundant manipulators between two joint configurations, Journal of the Chinese Institute of Engineers. 22 (1999), pp. 159-169.

[20] A.A. Ata, W.F. Faris, and M.Y. Sasdeh, Optimal Trajectory Selection for a Three DOF Manipulators with Minimum Energy Consumption, Journal of Applied Engineering Research. 2 (2007), pp. 45-63.
[21] J.J. Craig, Introduction to Robotics: Mechanics and Control, Addison-Wesley, New York, 1989.
[22] Y. Davidor, A genetic algorithm applied to robot trajectory generation, in L. Davis, (ed.), Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York, 1991, pp.144-165.
[23] Y.G. Peng and W. Wei, A new trajectory planning method of redundant manipulator based on adaptive simulated annealing genetic algorithm, International Conference on Computational-Intelligence and Security, Guangzhou, China, 2006.
[24] L. Yao and W. Sethares, Nonlinear parameter estimation via the genetic algorithm, IEEE Transactions on Signal Processing. 42 (1994), pp. 927-935.
[25] Z.S. Abo-Hammour, N.M. Mirza, S.M. Mirza, and M. Arif, Cartesian path generation of robot manipulators using continuous genetic algorithms, Robotics and Autonomous Systems. 41 (2002),179223.