Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/100122

Solution of Inverse Kinematics Problem using Genetic Algorithms

Shaher Momani [1,2,*], *Zaer S. Abo-Hammour*[3] *and Othman MK. Alsmadi*[4]

[1] Department of Mathematics, Faculty of Science, The University of Jordan, Amman 11942, Jordan
[2] Nonlinear Analysis and Applied Mathematics (NAAM) Research Group, Faculty of Science, King Abdulaziz University, Jeddah 21589, Saudi Arabia
[3] Department of Mechatronics Engineering, The University of Jordan, Amman 11942, Jordan
[4] Department of Electrical Engineering, The University of Jordan, Amman 11942, Jordan

Received: 2 Mar. 2015, Revised: 5 Jul. 2015, Accepted: 15 Jul. 2015
Published online: 1 Jan. 2016

Abstract: In this paper, the solution of inverse kinematics problem of robot manipulators using genetic algorithms (GA) is presented. Two versions of genetic algorithms are used which include the conventional GA and the continuous GA. The inverse kinematics problem is formulated as an optimization problem based on the concept of minimizing the accumulative path deviation in the absence of any obstacles in the workspace. Simulation results show that the continuous GA outperforms the conventional GA from all aspects. The superiority of the continuous GA is seen in that it will always provide smooth and faster solutions as compared with the conventional GA.

Keywords: Inverse kinematics problem; Robot manipulators; genetic algorithms

1 Introduction

GAs are basically generate-and-test artificial intelligent optimization methods that are based on the Darwinian principles of biological evolution. Even with the existence of other artificial intelligent methods [1,2,3], GAs have also received much of the researchers attentions [4,5,6]. The construction of a genetic algorithm for the solution of any optimization problem depends on different tasks [7]: an initial population of solutions, genetic operators, and a fitness evaluation function. These factors resulted in the availability of numerous variants of GAs reported in literature. However, sometimes specific solutions are desired, such as smoothness of the solution curve in robot manipulators. As will be seen later, this problem is solved using the proposed CGA which will not be achieved using the conventional CA.

As it is well known that the inverse kinematics problem is used to control the posture of an articulated body, it has become a fundamental problem in robotics where several methods have been proposed [8]. Most of these methods include the approaches of geometric, iterative, and algebraic, which as has been reported, they are inadequate for redundant robots [9]. Recently, much attention has been focused on a neural-network-based inverse kinematics problem solution in robotics [9,10, 11]. However, Due to the fact that there exist some difficulties to solve the inverse kinematics problem when the kinematics equations are complex, highly nonlinear, coupled and multiple solutions in terms of these robot manipulators [12], the GA approach has been motivated to investigate the possibility of solving this kind of problem for robot motion planning. The motion planning of robot manipulators can be classified into two main categories; continuous/Cartesian motion planning, and point-to-point motion planning. The continuous/Cartesian motion planning process involves the use of inverse kinematics equations of the manipulator to obtain the set of joint angles or velocities corresponding to the desired motion in Cartesian space [9,10,11,13,14].

Much of the conducted research has been focused on either the point-to-point trajectory generation of redundant and non-redundant manipulators, or continuous motion planning of redundant manipulators. The infinite number of solutions in joint space for the two previously mentioned categories requires an optimization method to

* Corresponding author e-mail: s.momani@ju.edu.jo

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. In 1991, Davidor [22] proposed a special GA for path generation problem of redundant manipulators. However, his proposed GA has drawbacks and could not fully exploit the abilities of GAs. The recent applications of genetic algorithms mainly focused on the motion planning of redundant manipulators [5,23]. In this paper, we will focus on the continuous path generation of manipulators rather than the point-to-point planning. As mentioned, the previous applications of the genetic algorithms were limited to redundant manipulators, while this algorithm maybe applied to both redundant and non-redundant manipulators.

2 Problem Formulation with CGA Approach

Consider a robot manipulator with M degrees of freedom and N task space coordinates. Assume that a desired Cartesian path, P_{dc}, is given, the problem is to find the set of joint paths,P_θ, such that the accumulative deviation between the generated Cartesian path, P_{gc}, and the desired Cartesian path, P_{dc}, is minimum. In other words, we are interested in the determination of a set of feasible joint angles, which corresponds to a set of desired spatial coordinates of the end effector in the task space. In the proposed approach, the desired geometric Cartesian path is uniformly sampled. The number of sampling points (path points or knots) is specified by the programmer and depends on the desired accuracy of the generated path. The accuracy of the generated path increases as the number of path points increases. After the sampling process (with N_k samples), P_{dc} and P_{gc} are matrices of dimension N by N_k while P_θ is a matrix of M by N_k dimension. After sampling the geometric path, at the path update rate for best accuracy, the generated values of the joint angles using the genetic algorithm, P_θ, are used by the direct (forward) kinematics model of the robot to obtain the generated Cartesian path given by

$$P_{gc} = F_k(P_\theta) \tag{1}$$

where F_k represents the forward kinematics model of the manipulator.

The deviation between the desired Cartesian path, P_{dc}, and the generated Cartesian path, P_{gc}, at some general path point, i, is given by

$$E_{(i)} = \sum_{k=1}^{N} \left| P_{dc}(k,i) - P_{gc}(k,i) \right| \tag{2}$$

The accumulative deviation between the two paths (desired and generated) depends on whether the initial and final joint angles corresponding to the initial and final configurations of the end effector are given in advance using the inverse kinematics model of the manipulator or through other numerical technique (fixed end points) or the case in which the initial and final joint angles are not given (free end points). For the fixed end points case, the accumulative deviation between the two paths is given by

$$E = \sum_{i=2}^{N_k-1} \sum_{k=1}^{N} \left| P_{dc}(k,i) - P_{gc}(k,i) \right| = \sum_{i=2}^{N_k-1} E(i) \tag{3}$$

while for the free end points case, the accumulative deviation between the two paths is given by

$$E = \sum_{i=1}^{N_k} \sum_{k=1}^{N} \left| P_{dc}(k,i) - P_{gc}(k,i) \right| = \sum_{i=1}^{N_k} E(i) \tag{4}$$

The fitness function, a nonnegative measure of the quality of individuals, is defined as follows

$$F = \frac{1}{1+E} \tag{5}$$

The optimal solution of the problem is obtained when the deviation function, E, approaches zero and correspondingly the fitness function, F, approaches unity. As a result, the path generation problem is formulated as a minimization problem of the deviation functions or as a maximization problem of the fitness function.

In order to obtain the solution of this problem, the CGAs, which were developed by Abo-Hammour [13], are used. The CGAs were developed for the solution of optimization problems in which the parameters to be optimized are correlated with each other or the smoothness of the solution curve must be achieved. It has been successfully applied in the motion planning of robot manipulators [25,26,27], numerical solution of two-point boundary value problems [28,29], solution of differential-algebraic equations [30], solution of fuzzy differential equations [31], solution of Laplace equation

[32], and the solution of nonlinear partial differential equations [32].

The novel development of CGAs has opened the doors for wide applications of the algorithm in the field of engineering and mathematics. Recently, the algorithm has been applied for the Solution of Troeschs and Bratus Problems [34], nonlinear system of second-order boundary value problems [35], Systems of Singular Boundary Value Problems [36]; chemical reactor problem [37], singular two-point boundary value problems [38], and optimal control problems [39].

In the CGAs evolution process, an individual is a candidate solution of the joints angles; that is, each individual consists of M joints paths each consisting of N_k path points, this results in a two-dimensional array of the size MxN_k. The population undergoes the selection process, which results in a mating pool among which pairs of individuals are crossed with probability P_{ci}. Within that pair of parents, individual joints are crossed over with probability P_{cj}. This process results in an offsprings generation where every individual child undergoes mutation with probability P_{mi}. Within that child, individual joints are mutated with probability P_{mj}. After that, the next generation is produced according to the replacement strategy applied. This process is repeated till the convergence criterion is met where the MxN_k parameters of the best individual are the required joints angles. The final goal of discovering the required joints paths is translated into finding the fittest individual in genetic terms.

The conventional genetic algorithm used in our work, on the other hand, consists of the steps given previously. The evaluation step, selection step, replacement step and the termination step are identical in both algorithms. However, the differences between both algorithms lie in the initialization phase, the crossover operator, the mutation operator and the extinction and immigration operator. These operators have the same goal in both algorithms; the difference lies in the way in which each operator is applied in the corresponding algorithm. These operators are applied at the joints path level in case of the CGA while they are applied at the path point level in case of conventional genetic algorithm. That is, the operators of the CGA are of global nature while those of conventional genetic algorithm are of local nature. In addition to that, it is to be noted that the conventional genetic algorithm uses the genotype and phenotype data presentations while the CGA uses only the phenotype data presentation. This fact requires a coding process in conventional genetic algorithm, which is not the case in CGA.

In the conventional genetic algorithm, each joint angle of every joints path has to be encoded into a finite-length substring over some finite alphabet, which is normally 2 (binary coding). If we assume that each substring consists of N_s characters or genes, then the chromosomes or the individuals are formed by cascading the genes of M joints each of N_k path points forming a longer string of length $L = M * N_k * N_s$ genes. In this way, the population may be viewed as a vector of N_p elements where each element consists of L genes.

3 Conventional and Continuous GA

The CGA and the conventional GA were used to solve the Cartesian path generation problem of 3R planar manipulator. The input data to both algorithms is divided into two parts; the genetic-algorithm related parameters and the manipulator related parameters. The genetic-algorithm related parameters for the CGA include the population size, N_p, the individual crossover probability, P_{ci}, the joint crossover probability, P_{cj}, the individual mutation probability, P_{mi}, the joint mutation probability, P_{mj}, the immigration threshold value, the corresponding number of generations, and finally the termination criterion.

The GA related parameters for the conventional genetic algorithm include the population size, N_p, the crossover probability, P_c, the mutation probability, P_m, the required accuracy of the phenotype values, the immigration threshold value, the corresponding number of generations, and finally the termination criterion. While on the other hand, the robot related parameters include the link parameters, the number of joints in the manipulator, M, the robots degrees of freedom, N, the number of path points, N_k, the joints limits $(\theta_{lower}(h) and \theta_{upper}(h) for h = 1,,M)$, and the desired Cartesian path $(P_{dc}(k,i) for k = 1,,N and i = 1,,N_k)$. Regarding the initial and final joints angles $(\theta_{initial}(h) and \theta_{final}(h) for h = 1,,M)$, there are two cases, the fixed end points case and free end points case. In the fixed end points case, these values are fed to the algorithm as input parameters using closed-form inverse kinematics formulas or any numerical technique, while in the free end points case, the end points are not considered as input parameters to the algorithm since they are not given.

The initial settings of the CGA parameters are as follows: the population size is set to 500 individuals. The rank-based selection strategy is used where the rank-based ratio is set to 0.1. The individual crossover probability is kept at 0.9; the joint crossover probability is also set to 0.9. The individual mutation probability and the joint mutation probability are kept at 0.9. Generational replacement scheme is applied where the number

of elite parents that are passed to the next generation is one-tenth of the population. Extinction and immigration operator is applied when the improvement in the fitness value of the best individual over 400 generations is less than 0.01. The genetic algorithm is stopped when one of the following conditions is met. First, the fitness of the best individual of the population reaches a value of 0.99; that is the accumulative deviation of the end effector, E, of the best individual is less than or equal to 0.01. Second, the maximum deviation at any path point of the best individual is less than or equal to 0.001.

Third, a maximum number of 10000 generations is reached. Fourth, the improvement in the fitness value of the best individual in the population over 1000 generations is less than 0.01. It is to be noted that the first two conditions indicate to a successful termination process (optimal solution is found), while the last two conditions point to a partially successful end depending on the fitness of the best individual in the population (near-optimal solution is reached).

The initial settings of the conventional genetic algorithm parameters are similar to those of the CGA except those related to crossover, mutation and coding process which are as following: the crossover probability is kept at 0.7, the mutation probability is kept at 0.01. The uniform crossover method is used as the algorithms default crossover method. The required accuracy of the phenotype values is set to 0.001 and binary coding scheme is used. Due to the stochastic nature of GAs, twelve different runs were made for every result obtained in this work using a different random number generator seed; results are the average values whenever possible. The selected Cartesian path generation problem, shown in Figure 1, is of a straight line shape and is given by:

$$x_{initial} = 0.0, x_{final} = 0.25$$

$$P_{dc}(1,i) = X_{dc}(i) = x_{initial} + \frac{x_{final} - x_{initial}}{N_k - 1} * (i-1) \quad (6)$$

$$P_{dc}(2,i) = Y_{dc}(i) = 0.25$$

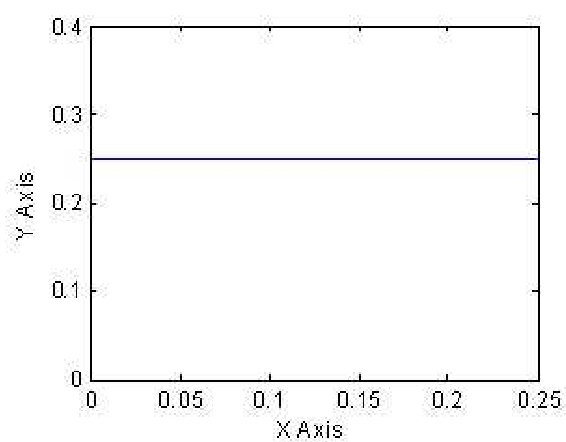


Fig. 1: Cartesian geometric path for the 3R planar manipulator.

The link parameters of the 3R redundant manipulator are given as $L1 = L2 = L3 = 0.5$ meter. For this case, $N = 2, M = 3, \theta_{lower}(h) = -180 and \theta_{upper}(h) = 180 for h = 1,2,3$. The forward kinematics model of the manipulator is given by

$$X_{gc}(i) = \sum_{j=1}^{M} \left(L_j * \cos\left[\sum_{k=1}^{j} \theta_{k,i}\right]\right) = \sum_{j=1}^{M} \left(L_j * \cos\left[\sum_{k=1}^{j} \P_{\theta}(k,i)\right]\right) \quad (7)$$

$$X_{gc}(i) = \sum_{j=1}^{M} \left(L_j * \sin\left[\sum_{k=1}^{j} \theta_{k,i}\right]\right) = \sum_{j=1}^{M} \left(L_j * sin\left[\sum_{k=1}^{j} \P_{\theta}(k,i)\right]\right) \quad (8)$$

where $1 \leq i \leq N_k$. The number of path points along the Cartesian path, N_k, is set to 20 points. The initial and final joints angles corresponding to the initial and final configurations of the end effector along the Cartesian path are not given (i.e., free end points case). The evolutionary progress plot of the best-fitness individual and the path point deviations for the 3R planar redundant manipulator are shown in Figure 2, which shows that the algorithm reaches a fitness value of 0.99 within 72 generations and the average path point deviation is almost 0.0005 meter. The desired and generated Cartesian paths are given in Figure 3. It is clear that the desired and the generated Cartesian paths are almost the same.

The joints paths for the first, second, and third joints of the 3R manipulator are shown in Figure 4. It is obvious that the resulting joints paths are highly oscillatory within the range of the joints limits, which results in large net displacements of the joints. The oscillatory behaviour of the joints paths, as shown in Figure 4, is the key reason behind the introduction of the analogous crossover operator by Davidor [22] in an attempt to avoid the discontinuities in the joints paths resulting from conventional crossover approaches. These observations, in general, are due to the fact that the initialization, crossover, mutation and extinction and immigration operators of the conventional genetic algorithm are of local nature and applied at the path point level. This fact results in discontinuities in the joints paths or oscillatory values among consecutive path points. The operators of the CGA, on the other hand, are of global nature and applied at the joints path level. As a result, the step-function-like jump in the joint values along the joints path is totally avoided due to the smooth transitions in the joint values.

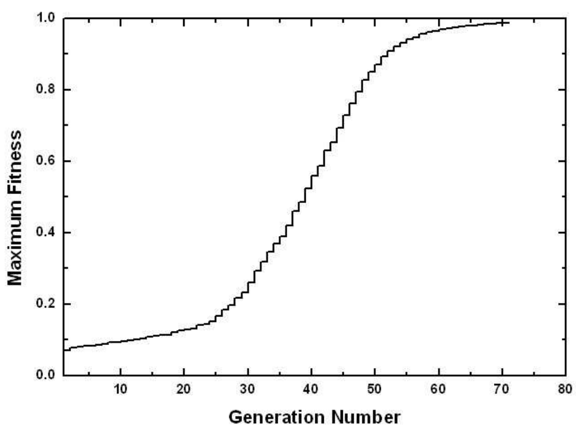
After that, the effect of both versions (conventional and continuous) of the initialization phase, crossover operator and mutation operator on the nature of the joints paths obtained and the convergence speed of the hybrid algorithm is studied. Table 1 gives the relevant data for 3R manipulator. As seen in this table, it is clear that the initialization phase has the greatest effect on the smoothness/nonsmoothness of the solution curves; that is, in case of conventional initialization, the joints paths are of oscillatory nature with large or medium magnitude oscillations while in case of continuous initialization, the joints paths are either smooth or of oscillatory nature with small magnitude oscillations. The minimum execution time and the best convergence speed are achieved using the CGA (i.e., continuous types of initialization, crossover and mutation).

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

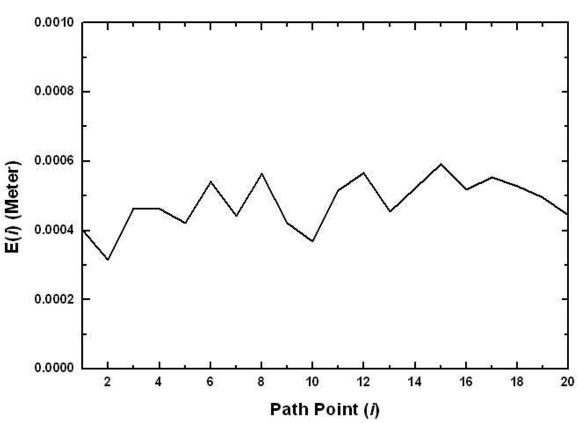
Initialization Type	Crossover Type	Mutation Type	Average Execution Time	Average Number of Generations	Nature of Joints Paths
Conventional	Conventional	Conventional	487.72	124	Oscillations with Large Magnitude
Conventional	Conventional	Continuous	390.06	105	Oscillations with Medium Magnitude
Conventional	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

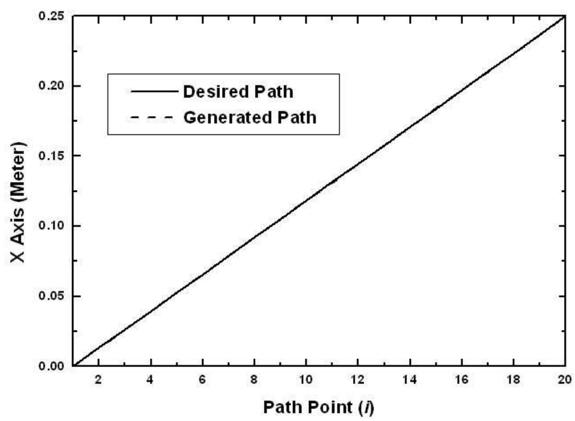


(a)

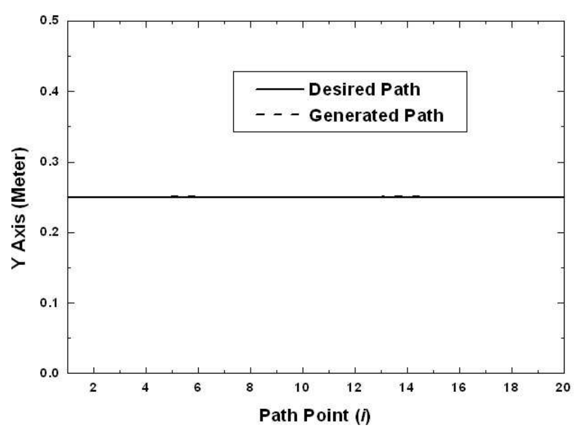


(b)

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

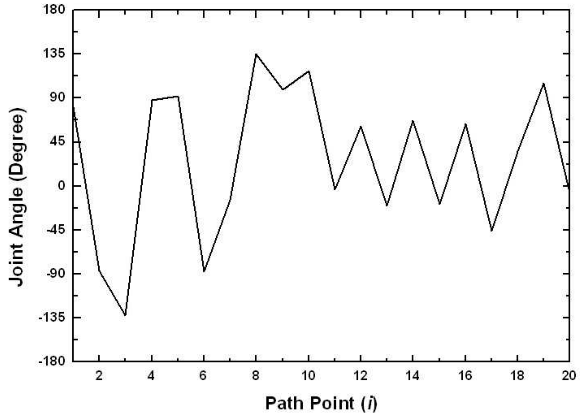


(a)

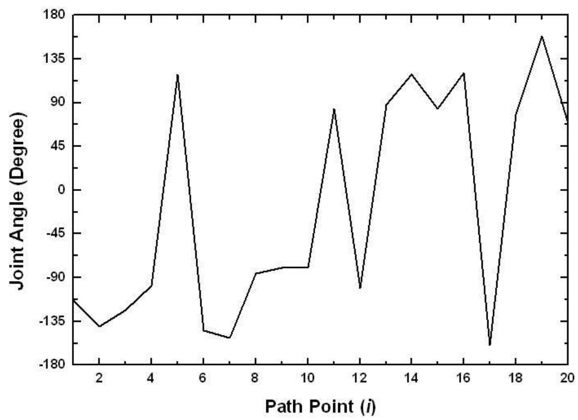


(b)

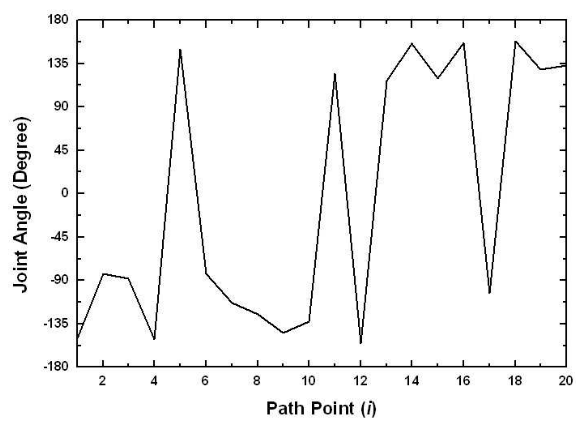
Fig. 3: Desired and generated Cartesian path for the 3R planar manipulator in (a) X-plane, (b) Y-plane.



(a)

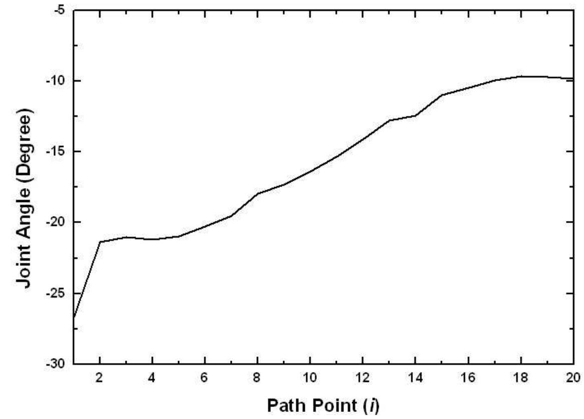


(b)

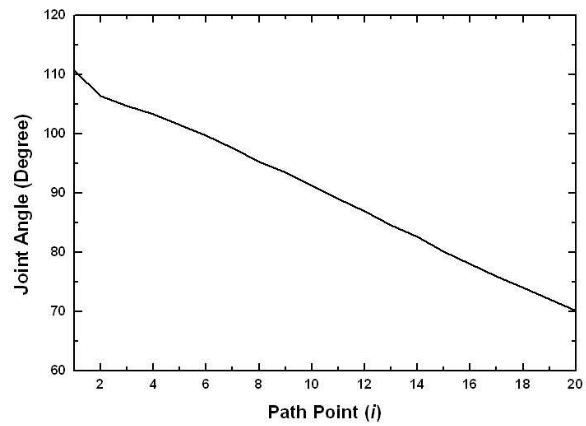


(b)

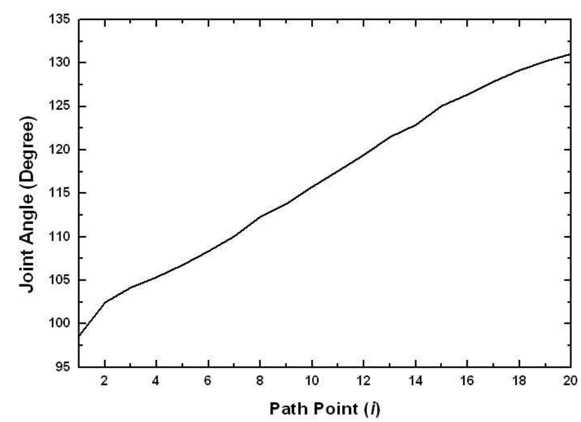
Fig. 4: Joints paths of the 3R manipulator using conventional GA: (a) 1st Joint, (b) 2nd Joint, (c) 3rd Joint.



(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

For the 3R manipulators, the conventional initialization, continuous crossover and continuous mutation case results in the largest number of generations required for convergence. For the case in which the conventional initialization, continuous crossover and conventional mutation are used, it is observed that this hybrid scheme still results in oscillations with large magnitude as shown in Table 1. This is an expected result since the smoothness achieved by the continuous crossover process is disturbed by the conventional mutation process.

The joints paths for the first, second and third joints of the 3R manipulator using CGA are shown in Figure 5. It is obvious that the resulting solution curves in joint space are smooth and do not have any switching between the two possible solutions, which results in minimizing the net displacement of the joints. The effect of the degree of redundancy (number of links) of the planar redundant manipulator on the convergence speed of the conventional and the CGAs is studied next. The number of links of the planar manipulator, M, is increased from 4 to 10 in steps of 2 for both algorithms where the link length is set as $Li = 1/Mmeter fori = 1,2,,M$. Table 2 shows the relevant data using the conventional genetic algorithm while Table 3 shows the relevant data using the CGA for the previous path generation problem. From these tables, it is clear that as the number of links increases, the average number of generations required for convergence increases in case of conventional genetic algorithm while the CGA is insensitive to this parameter (i.e., the number of generations is almost fixed). In addition to that, the average time per generation in the conventional genetic algorithm is two to three times that in the CGA. This shows that CGA not only results in smooth joints paths, but also results in smaller number of generations for convergence and the average time per generation is about half of that of the conventional GA. Finally, the effect of the number of knots along the given path generation problem on the convergence speed of the conventional and the CGAs for both manipulators is studied. The number of knots is increased from 20 to 100 in steps of 20 for both algorithms. For the 3R manipulator, Tables 4 and 5 show the relevant data using the conventional genetic algorithm and the CGA respectively. It is observed that the average number of generations required for convergence using conventional genetic algorithm increases sharply as the number of knots along the Cartesian path is increased while the average number of generations required for convergence using CGA is almost constant regardless the number of knots along the Cartesian path. That is, when the number of knots is increased from 20 to 100, the number of generations increases from 76 to 840 for the conventional genetic algorithm while the number of generations increases from 50 to 78 for the CGA.

4 Conclusion

In this paper, the inverse kinematics problem solution of robot manipulators was achieved using the CGA. As a CGA and a conventional GA comparison, it was observed that the resulting joints paths using the conventional GA have multiple switching points in some of the non-redundant manipulators solutions while they were of highly oscillatory nature for the redundant manipulators resulting in very large net displacements for both systems. Taking in consideration the shortcomings of the conventional GA, the CGA operators (initialization phase, crossover, mutation) were designed such that they result in smooth joints paths while they maintain an excellent accuracy along the Cartesian path. It was found that the initialization phase has the greatest effect on the smoothness of the joints paths. The convergence speed of the CGA in terms of both the number of generations for convergence and the average execution time is much superior to that of the conventional GA.

Conflicts of Interests The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] Chen, Zhixiang; Cao, Feilong, The properties of logistic function and applications to neural network approximation, Journal of Computational Analysis and Applications. 15, 6 (2013), pp. 1046-1056.

[2] Liu, Kun; Gong, Zeng-Tai, Numerical solution of fully fuzzy linear matrix equations, Journal of Computational Analysis and Applications. 15 (6) (2013), pp. 1026-1035.

[3] A. Khoukhi, L. Baron, M. Balazinski, and K. Demirli, *A hierarchical neuro-fuzzy system to near optimal-time trajectory planning of redundant manipulators*, Engineering Applications of Artificial Intelligence. 21 (2008), pp. 974-984.

[4] O. Liu, Y. Yu, S. Liu, and M. Zhang, *Obstacle avoidance of a class of underactuated robot manipulators: GA based approach*, International Journal of Computational Intelligence Systems. 1 (2008), pp. 353-360.

[5] M.D. Marcos, J.A.T. Machado, and T.P. Azevedo-Perdicoulis, *Trajectory planning of redundant manipulators using genetic algorithms*, Communications in Nonlinear Science and Numerical Simulation. 14 (2009), pp. 2858-2869.

[6] L.F. Tian and C. Collins, *An effective robot trajectory planning method using Genetic Algorithms*, Journal of Mechatronics. 14 (2004), pp. 455-470.

[7] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, New York, 1989.

[8] Samuel R. Buss, Introduction to Inverse Kinematics with Jacobian Transpose, Pseudoinverse and Damped Least Squares methods, " *http://www.math.ucsd.edu/ sbuss/ResearchWeb/ikmethods/ iksurvey.pdf*", 2009.

[9] Rait Koker, Tark akar, Yavuz Sari, A neural-network committee machine approach to the inverse kinematics problem solution of robotic manipulators. Engineering with Computers, DOI:10.1007/s00366-013-0313-2, 2013.

[10] Zhang D, Lei J, Kinematic analysis of a novel 3-DOF actuation parallel manipulator using artificial intelligence approach. 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.

[26] Z.S. Abo-Hammour, *A novel continuous genetic algorithms for the solution of the cartesian path generation problem of robot manipulators*, In Robot Manipulators: New Research, Lui JX (ed.). Nova Publishers: New York, 2005, pp. 133190.

[27] Za'er S. Abo-Hammour, Othman M. K. Alsmadi, Sofian I. Bataineh, *Continuous genetic algorithms for collision-free Cartesian path planning of robot manipulators*, International Journal of Advanced Robotic Systems. 8(6) (2011), pp. 14-36.

[28] Z.S. Abo-Hammour, M. Yusuf, N.M. Mirza, S.M. Mirza, and M. Arif, *Numerical solution of second-order, two-point boundary value problems using continuous genetic*

algorithms, International Journal for Numerical Methods in engineering. 61 (2004), pp. 1219-1242.

[29] H. Jaradat, *Numerical solution of temporal two-point boundary value problems using continuous genetic algorithms*, Ph.D. Thesis, University of Jordan, Amman, Jordan, 2006.

[30] O.F. Al-sayyed, *Numerical solution of differential-algebraic equations using continuous genetic algorithms*, Ph.D. Thesis, University of Jordan, Amman, Jordan, 2006.

[31] O.M. Abo-Arqoub, *Numerical solution of fuzzy differential equations using continuous genetic algorithms*, Ph.D. Thesis, University of Jordan, Amman, Jordan, 2008.

[32] Z.S. Abo-Hammour, R.B, Albadarneh, and M.S. Saraireh, *Solution of Laplace equation using continuous genetic algorithms*, Kuwait Journal of Science & Engineering. Academic Publication Council University of Kuwait, Kuwait. 37 (2010), pp. 115.

[33] R.B. Albadarneh, *Solution of Partial Differential Equations using continuous genetic algorithms*, Ph.D. Thesis, University of Jordan, Amman, Jordan, 2009.

[34] Abo-Hammour, Zaer; Abu Arqub, Omar; Momani, Shaher; Shawagfeh, Nabil; Optimization Solution of Troeschs and Bratus Problems of Ordinary Type Using Novel Continuous Genetic Algorithm; Discrete Dynamics in Nature and Society; Volume 2014, Article ID 401696, 2014.

[35] Abu Arqub, Omar; Abo-Hammour, Zaer; Momani, Shaher; Application of continuous genetic algorithm for nonlinear system of second-order boundary value problems; Applied Mathematics and Information Systems; 8(1), 235-248, 2014.

[36] Abo-Hammour, Zaer; Abu Arqub, Omar; Alsmadi, Othman; Momani, Shaher; An Optimization Algorithm for Solving Systems of Singular Boundary Value Problems; Applied Mathematics and Information Systems; in press, 2014.

[37] Alsmadi, Othman M. K.; Abo-Hammour, Zaer S.; Al-Smadi, Adnan M.A.; Robust and efficient genetic algorithm for solving a chemical reactor problem: theory, application and convergence analysis; Transactions of the Institute of Measurement and Control; 34(5), 594-603, 2012.

[38] Abu Arqub, Omar; Abo-Hammour, Zaer; Momani, Shaher; Solving singular two-point boundary value problems using continuous genetic algorithm; Abstract and Applied Analysis; volume 2012; Article ID 205391; 2012.

[39] Abo-Hammour, Zaer S.; Asasfeh, Ali Ghaleb; Al-Smadi, Adnan M.; A novel continuous genetic algorithm for the solution of optimal control problems; Optimal Control Applications & Methods; 32(4), 414-432, 2011.

Shaher Momani received his Ph.D from the university of Wales (UK) in 1991. He then began work at Mutah university in 1991 as assistant professor of applied mathematics and promoted to full Professor in 2006. He left Mutah university to the university of Jordan in 2009 until now. His research interests focus on the numerical solution of fractional differential equations in fluid mechanics, non-Newtonian fluid mechanics and numerical analysis. Prof. Momani has written over 250 research papers and warded several national and international prizes. Also, he was classified as one of the top ten scientists in the world in fractional differential equations according to ISI web of knowledge.

Zaer Salem Abo-Hammour is an associate professor at the Mechatronics Engineering Department, Faculty of Engineering, University of Jordan. He received his master and Ph.D. degrees in Systems Engineering from Pakistan Institute of Engineering and Applied Sciences, Pakistan Atomic Energy Commission in 1998 and 2002 respectively. His research interests are focused on genetic algorithms, control systems, robotic systems, numerical optimization, model order reduction, and numerical fields. He is also a specialist in solar energy systems and fuel efficiency improvements. He is ranked as the top researcher in Jordan for the period 2010-2015 in engineering discipline according to ISI publications.

Othman Alsmadi graduated from Tennessee State University in 1993 with a B.S. in Electrical Engineering; and an M.S. degree in Electrical Engineering in 1995. In 1999, he received his Ph.D. from Wichita State University, Kansas. Dr. Alsmadi is now an associate professor at The University of Jordan and is currently serving as a visiting professor at Yarmouk University, Department of Electrical Power Engineering. He is ranked as one of the top researchers at the University of Jordan while his research interests are focused on optimal control, model order reduction, system identification, neural networks and artificial intelligence techniques.