

Manipulator Redundancy Reduction as a Tool for Reinforcing Motion Planning Using Genetic Algorithms

Amir Homayoun Javadi A.

Abstract—A novel approach to plan an optimum motion of redundant robot manipulators for a predefined end-effector trajectory using genetic algorithms (GA) is presented. The efficiency of the proposed approach, without loss of generality, is demonstrated through a simulation carried out on a planar 6-DOF robot manipulator. The approach benefits from two key features. First, the method of data representation which guarantees the satisfaction of joints angle limits, and second the conversion of considered model's 6-DOF construction to 4-DOF construction along with an additional binary value which guarantees the exact placement of the end-effector on the predefined trajectory. Comparison with three other approaches shows that the result of the presented solution is substantially better. In addition the difference of two kinds of Random Number Generator (RNG) is addressed. It is shown that using RNG with normal distribution leads to faster convergence of the proposed algorithm than RNG with uniform distribution.

Index Terms—Genetic Algorithms, Manipulator, Motion Planning, Trajectory Tracking

I. INTRODUCTION

Recently, there has been an increased research interest in the development of efficient procedures to solve the inverse kinematics problem of redundant robot manipulators. In general, solving the inverse kinematics problem of redundant robots is not trivial since the necessary mapping from the task coordinates to the joint coordinates is not one to one, and yields an infinite of solutions [1, 2]. Redundant robot manipulators can provide a better ability to avoid singular configuration and the excessive velocities and accelerations encountered at singularities [3].

Generally, there are three main approaches for trajectory planning for redundant manipulators, pseudo-inverse of Jacobian matrix, variational approach, and optimization techniques based on the direct kinematics [4].

Davidor [5] applied a GA to generate the robot trajectory by finding the inverse kinematics for predefined end-effector robot paths. A trajectory of a 3-link planar redundant robot is simulated by minimizing the sum of the position errors at each of the knot points along the path. Yun and Xi [6] presented a new method for optimum motion planning based on an improved genetic algorithm. This approach incorporates kinematics constraints, dynamics constraints as

well as control constraints. Simulation results for two and three DOF robots were presented. Hirakawa and Kawamura [7] proposed a combination of B-spline trajectory generation and steepest gradient optimization to design an optimal motion planning for redundant manipulators. However, the proposed optimization approach needs to determine the gradients of the objective function. McAvoy et al. [2] proposed an approach utilizing genetic algorithms for optimal point-to-point motion planning for kinematically redundant manipulators to satisfy both the initial conditions and some other specified criteria. Their approach combines B-spline curves for the generation of smooth trajectories with genetic algorithms for optimal solution. Ata and Myo [4] has proposed an optimal point-to-point trajectory planning for planar redundant manipulator. Their main objective was to minimize the sum of the position error of the end-effector at each intermediate point along the trajectory so that the end-effector can track the prescribed trajectory accurately. They introduced an algorithm combining Genetic Algorithm and Pattern Search as a Generalized Pattern Search GPS to design the optimal trajectory. To verify the proposed algorithm, simulations for a 3-DOF planar manipulator with different end-effector trajectories have been carried out.

Our proposed GA optimization mechanism differs from the mentioned methods in a number of ways. The first difference lies on robot's model alternation in order to computationally track the end-effector trajectory without any evolution, so the end-effector would be exactly placed on the predefined trajectory. Therefore, there would be no tracking error. The second difference is that the joints angle are not manipulated directly, rather they are accessed through a mapping function, (8), which maps real values $]-\infty, +\infty[$ to lower and upper boundaries of joints angle. Thus the joints angle limits are satisfied without any conditioning statement. Moreover it is shown that in this special type of problem, selecting a special kind of Random Number Generator (RNG) can improve the optimization process.

The remainder of the paper is organized as following: Section 0II gives the robot's model and the method of model manipulation. Section III gives a brief overview of the concept of GA in each building block of the algorithm and presents the proposed GA solution. Section IV demonstrates and discusses the efficiency of the proposed GA through an experiment. In the same section, the performance of the proposed solution is compared to that of three previous approaches. Finally, Section V summarizes the contribution of the paper.

Manuscript received March 31, 2008.

Amir Homayoun Javadi A. is with Robotics and Automation Lab, Noise Company and National Olympic and Paralympics Academy of Iran, Tehran, Iran (phone: +98 912 182 0929; e-mail: javadi@noise.ir).

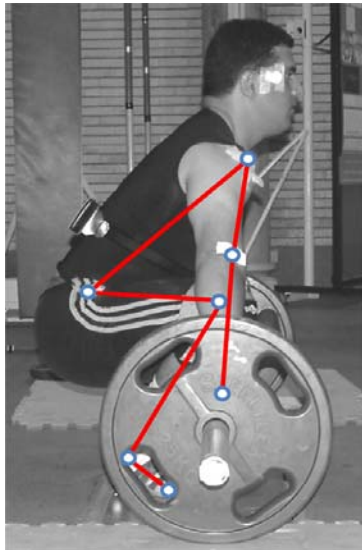


Fig. 1. Weightlifter at the initial state. Links are shown in red lines and joints are shown in blue circles.

II. THE MANIPULATOR'S MODEL

A 6-DOF robot manipulator is considered throughout the simulation. The model and the end-effector trajectory is adopted from human body during weightlifting task, Fig. 1, in which the end-effector trajectory is the trajectory of barbell in a sample snatch lift. Table I shows the links length along with the links mass. It is supposed that center of mass of each link is placed on the geometric center of itself.

Table I. Links name along with Links length and mass

Link	Length (cm)	Mass (kg)
l_1 , foot	13.20	3.04
l_2 , shank	38.57	9.76
l_3 , thigh	31.10	21
l_4 , trunk	50.10	59.85
l_5 , upper-arm	26.89	5.88
l_6 , lower-arm	42.01	4.62

The proposed model is shown in Fig. 2. $\theta_j, j = 1..6$, are joints angle based on Denavit-Hartenberg (DH) representation. Table II shows the joints angle limits based on DH representation for the sample proposed manipulator. Values exceeding 360° indicate the portion of the unit circle between $\theta_{j,min}$ and $\theta_{j,max}$, positive horizontal axis, as shown in Fig. 3.

Table II. Joints angle limits based on Denavit-Hartenberg representation

	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
	toe	ankle	knee	hip	shoulder	elbow
Min	90°	287°	5°	224°	191°	280°
Max	150°	348°	134°	378°	377°	390°

As mentioned, one of the key features of our approach lies on the method of model conversion, in which, to ensure the trajectory tracking task, two degrees of freedom of the model are replaced with a single binary alternation and as a consequence the problem is reduced to define four real value

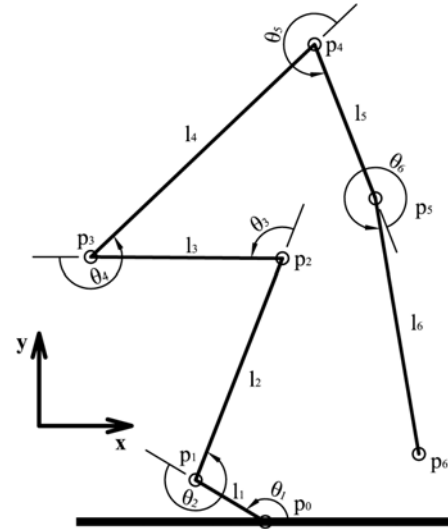


Fig. 2. Manipulator's model; Joints angle are based on DH representation.

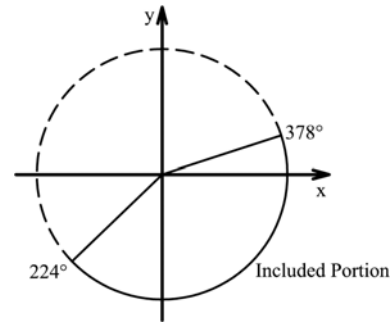


Fig. 3. Unit circle showing (378 – 224) degrees of freedom of a sample joint (θ_4)

joints angle along with a single binary value. The definition of model's control parameters is converted from 6 rotary parameters to 4 rotary parameters and one binary value to fulfill the trajectory tracking task. The other two undefined joints angle are calculated geometrically in the sense to place the end-effector on the desired position. Fig. 4 shows the robot with two different configurations as below

$$\text{Configuration}_1 = \begin{cases} \{p_0, p_1, p_2, p_3, p_4, p_{5,1}, p_6\} \\ \{\theta_1, \theta_2, \theta_3, \theta_4, \theta_{5,1}, \theta_{6,1}\} \end{cases} \quad (1a)$$

$$\text{Configuration}_2 = \begin{cases} \{p_0, p_1, p_2, p_3, p_4, p_{5,2}, p_6\} \\ \{\theta_1, \theta_2, \theta_3, \theta_4, \theta_{5,2}, \theta_{6,2}\} \end{cases} \quad (1b)$$

in which the points $p_i, i = 1..6$, are defined as below,

$$p_0 = [0,0]^T \quad (2)$$

$$p_i = p_{i-1} + l_i \left[\cos\left(\sum_{j=1}^i \theta_j\right), \sin\left(\sum_{j=1}^i \theta_j\right) \right]^T, i = 1..4 \quad (3)$$

$$p_6 = T|_{t=\kappa} \quad (4)$$

in which θ_j are joints angle based on DH representation as shown in Fig. 2. p_0 is the robot's base point, positioned at the position $[0,0]^T$. $T|_{t=\kappa}$ shows the desired position of end-effector on the trajectory in the instance of time κ . Predefined trajectory is broken into discrete intervals of time denoted by $\kappa, \kappa = 0..n$, which n shows the total number of

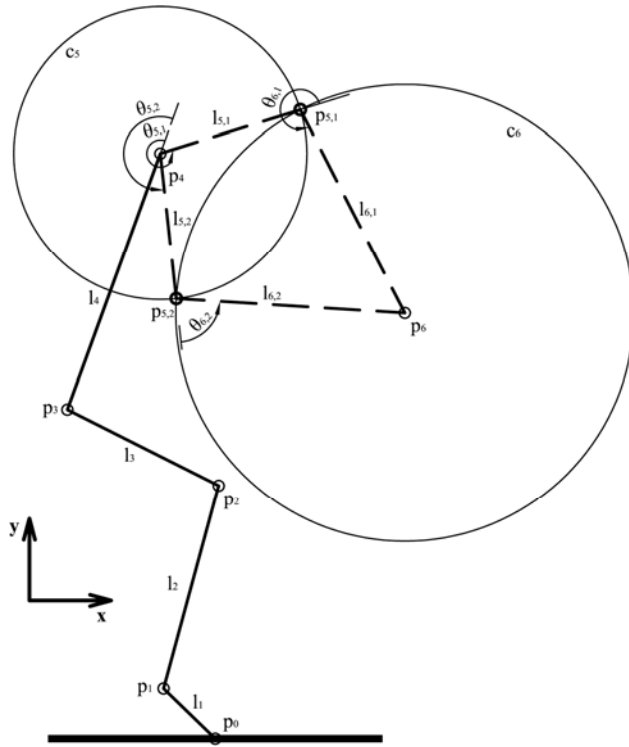


Fig. 4. Two possible configurations (1a) and (1b) using predefined positions $\{p_0, p_1, p_2, p_3, p_4, p_6\}$

discrete time steps. $\theta_j, j = 1..4$, are known in each frame of time, using the GA solution, p_0 is a fixed point and p_6 is placed on a predefined position on the trajectory in each instance of time, τ . p_5, θ_5 and θ_6 can be calculated using the system of equations below,

$$\begin{cases} (x_5 - x_4)^2 + (y_5 - y_4)^2 = l_5^2 \\ (x_5 - x_6)^2 + (y_5 - y_6)^2 = l_6^2 \end{cases} \quad (5)$$

which x_i and y_i shows the horizontal and vertical position of joint position p_i , respectively. System of equations 5 has two pairs of solutions if $\langle p_4, p_6 \rangle < l_5 + l_6$, two configurations 0 and 1, has one pair of solution if $\langle p_4, p_6 \rangle = l_5 + l_6$ and has no real value solution if $\langle p_4, p_6 \rangle > l_5 + l_6$, where $\langle p_n, p_m \rangle$ stands for the Euclidean distance between points p_n and p_m . Joints angle θ_5 and θ_6 must be calculated according to joints position p_4, p_5 and p_6 to validate the joints angle limits due to Table I. The condition to be met is as following,

$$\begin{cases} (\theta_{j,\min} \leq \theta_{j,c}) \wedge (\theta_{j,c} \leq \theta_{j,\max}) & \text{for } \theta_{j,\max} \leq 360^\circ \\ (\theta_{j,\min} < \theta_{j,c}) \vee (\theta_{j,c} < (\theta_{j,\max} \bmod 360)) & \text{for } \theta_{j,\max} > 360^\circ \end{cases} \quad (6)$$

in which $j = 5..6$ and $c = 1..2$ and $\theta_{j,c}$ is the calculated angle. $(a \bmod b)$ means the remainder, on division of a by b . For the sample configurations shown in Fig. 4, $\theta_{6,2}$ is obviously out of the range $\theta_{6,\min} = 280^\circ$ and $\theta_{6,\max} = 390^\circ$, so the configuration shown in (1b) is not valid. $\theta_{5,1} = 307^\circ$ and $\theta_{6,1} = 280^\circ$ are both in the valid range defined in Table

II, so for the predefined trajectory and the at hand values $\{\theta_1, \theta_2, \theta_3, \theta_4\}$ or respectively $\{p_0, p_1, p_2, p_3, p_4\}$, the only feasible configuration is (1a).

In the case of $\langle p_4, p_6 \rangle < l_5 + l_6$, two possible positions of p_5 are named as $p_{5,1}$ and $p_{5,2}$ and joints angle θ_5 and θ_6 are denoted as $\theta_{5,1}, \theta_{6,1}$ and $\theta_{5,2}, \theta_{6,2}$. For $\langle p_4, p_6 \rangle = l_5 + l_6$, indeed, we have two positions $p_{5,1}$ and $p_{5,2}$ which $p_{5,1} = p_{5,2}$.

III. THE GENETIC ALGORITHMS SOLUTION

Given the manipulator's model and the predefined end-effector trajectory, the GA plans an optimum sequence of configurations in the sense of minimizing the total consumed energy. In the following the proposed GA solution is presented.

A. The Representation Mechanism

The first, and perhaps the most critical aspect in designing a GA for a specific optimization problem is the basic mechanism that links the GA to the solution space of the problem [1]. This mechanism consists of choosing a method to represent a solution to the real problem as a finite-length string over a specific alphabet, the chromosome. The second key feature of our approach is the method of data representation, the content of each chromosome.

In our approach, each member of the GA consists of one initial configuration, $\tau = 0$, along with a sequence of configurations in which the sequence of movements of the manipulator is stored. The mentioned sequence contains the configurations from the frame next to the beginning to the end of the trajectory, $\tau = 1..n$.

The chromosome encoding the initial configuration consists of a single binary value, $v_0|_{\tau=0}$ indicating the configuration and four elements $v_r|_{\tau=0}, r = 1..4$, which each element indicates the quantity of $\sigma_{j,0}$ in the function $S_j(\sigma_{j,0}), j = 1..4$, (8). The function $S_j(\sigma_{j,k})$ incorporates a sigmoid function, $1/(1 + \exp(-\sigma_{j,k}))$, in order to map the random values, $\sigma_{j,k}$, to degree values, $\theta_j|_{\tau=k}$. The configuration here denotes the selection of the position of joint number 5, $p_{5,1}$ or $p_{5,2}$.

The sequence of chromosomes encoding the sequence of configurations throughout the rest of the trajectory contains a single binary value indicating the configuration type, $v_0|_{\tau=k}$, and four elements $v_r|_{\tau=k}, r = 1..4$, which each element contributes in the calculation of the value $\sigma_{j,k}$ which is used in the function $S_j(\sigma_{j,k}), j = 1..4$, as shown below,

$$\sigma_{j,k} = \sum_{r=0}^k (v_r|_{\tau=k}), \quad (7)$$

$$S_j(\sigma_{j,k}) = \theta_{j,\min} + (\theta_{j,\max} - \theta_{j,\min}) \cdot \frac{1}{1 + \exp(-\sigma_{j,k})} \quad (8)$$

$$\theta_j|_{\tau=k} = S_j(\sigma_{j,k}). \quad (9)$$

Using the above formulation has two key benefits. The first benefit lies on the manner the joints angle limits are served. Using the sigmoid function, (8), guarantees the placement of the joint angle value in the boundary $[\theta_{j,\min} \dots \theta_{j,\max}]$. At extremes, $\sigma_{j,\kappa} \rightarrow +\infty$ and $\sigma_{j,\kappa} \rightarrow -\infty$, the value of $S_j(\sigma_{j,\kappa})$ would be $\theta_{j,\max}$ and $\theta_{j,\min}$, respectively. In this way, the need for evaluating several *if* statements, validating the joints angle limits is eliminated.

The following summarizes the notations defined above. A member is defined as,

$$\mathbf{M}_\mu = \begin{bmatrix} \mathbf{v}|_{\tau=0} \\ \mathbf{v}|_{\tau=1} \\ \vdots \\ \mathbf{v}|_{\tau=n} \end{bmatrix} \quad (10)$$

in which \mathbf{M}_μ , $\mu = 1..m$, stands for the member μ within a generation, and m is the total number of members in each generation. $\mathbf{v}|_{\tau=\kappa}$ is a chromosome concerning a single instance of time $\tau = \kappa$, defining the configuration of the robot as shown below,

$$\mathbf{v}|_{\tau=\kappa} = [v_0|_{\tau=\kappa} \quad v_1|_{\tau=\kappa} \quad v_2|_{\tau=\kappa} \quad v_3|_{\tau=\kappa} \quad v_4|_{\tau=\kappa}] \quad (11)$$

where $v_0|_{\tau=\kappa}$ is a binary value, either 0 or 1, selecting one of the configurations (1a) or (1b), and $v_r|_{\tau=\kappa}$, $r = 1..4$, defines a randomly generated value used in the calculation of $\sigma_{j,\kappa}$, (7). Finally the values $\sigma_{j,\kappa}$ are applied to the function $S_j(\sigma_{j,\kappa})$ to calculate the joints angle in each instance of time $\tau = 0..n$, (8).

B. Random Number Generation

Chromosomes, as noted in (11), consist of 5 random numbers defining the random binary configuration, for v_0 , and random joint angle variations inputted to equations (7)-(9), for v_r , $r = 1..4$. The Random Number Generator (RNG), \mathfrak{R} , must satisfy the condition below,

$$\int_{-\infty}^{+\infty} \mathfrak{R}(p) dp = 1 \quad (12)$$

which $\mathfrak{R}(p)$ is the probability density function (pdf) of \mathfrak{R} at point p .

A threshold value for the assignment of binary configuration value v_0 must be assigned due to $\mathfrak{R}(p)$ in order to satisfy the equation below,

$$\int_{-\infty}^T \mathfrak{R}(p) dp = \int_T^{+\infty} \mathfrak{R}(p) dp = 1/2 \quad (13)$$

to give an equal chance to both sides of binary value. T is the threshold value.

Two different kinds of \mathfrak{R} are proposed. \mathfrak{R} are supposed to be with uniform pdf and with normal pdf, \mathfrak{R}_u and \mathfrak{R}_n respectively. The two proposed RNGs, are shown below,

$$\mathfrak{R}_{u,r} = \begin{cases} \begin{cases} 0 & \text{if } \mathfrak{R}_u \leq 0.5 \\ 1 & \text{if } \mathfrak{R}_u > 0.5 \end{cases} & \text{for } r = 0 \\ 2\rho_u \cdot \mathfrak{R}_u - \rho_u & \text{for } r = 1..4 \end{cases} \quad (14)$$

$$\mathfrak{R}_{n,r} = \begin{cases} \begin{cases} 0 & \text{if } \mathfrak{R}_n \leq 0 \\ 1 & \text{if } \mathfrak{R}_n > 0 \end{cases} & \text{for } r = 0 \\ \rho_n \cdot \mathfrak{R}_n & \text{for } r = 1..4 \end{cases} \quad (15)$$

where ρ_n and ρ_u are constant scaling values. Threshold values for $\mathfrak{R}_{u,0}$ and $\mathfrak{R}_{n,0}$ are $T_u = 0.5$ and $T_n = 0$, respectively. $\mathfrak{R}_u(p)$ and $\mathfrak{R}_n(p)$ are defined as below,

$$\mathfrak{R}_u(p) = \begin{cases} 1/(b-a) & \text{for } a < p < b \\ 0 & \text{for } p \leq a \text{ or } p \geq b \end{cases} \quad (16)$$

$$\mathfrak{R}_n(p) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad (17)$$

where a and b , $a < b$, are the minimum and maximum values of $\mathfrak{R}_u(p)$, respectively and σ^2 and μ are variance and mean values of $\mathfrak{R}_n(p)$, respectively. Two mentioned RNGs are compared within a simulation. It is shown that the type and the values of the parameters of function \mathfrak{R} have a critical role in the performance of the GA. Several configurations are proposed and compared in chapter III section F.

C. The Initial Population

The initial population and new members in each generation are generated randomly. Generating members with some criteria applied to the initial configuration can drastically reduce the computation time required by the GA. It is shown that minimizing the static torque applied to each joint at the initial configuration, $\tau = 0$, improves the overall performance of the GA.

The initial population and new members in each generation are generated randomly. \mathbf{M}_μ is generated using a proposed \mathfrak{R} and the feasibility of \mathbf{M}_μ is checked against two criteria, first $\langle p_4, p_6 \rangle \leq l_5 + l_6$ and second, joint angle limits for θ_5 and θ_6 . Both the criteria must be satisfied in order to keep the generated member, \mathbf{M}_μ , in the generation.

D. The Evaluation Mechanism

In an optimization problem, the fitness function corresponds to the objective function which must be optimized. Fitness function plays the role of the environment in which during the evolution of the GA, the chromosomes must be adapted. Fitness function in our case, $F(\mathbf{M}_\mu)$, is applied to the member itself, rather than individual chromosomes, in order to optimize whole the sequence of configurations. Our fitness function is as following,

$$F(\mathbf{M}_\mu) = \sum_{j=1}^6 \left(1 / \sum_{\kappa=2}^n T(\theta_j)|_{\tau=\kappa} \right) \quad (18)$$

which $T(\theta_j)|_{\tau=\kappa}$ is the value of torque applied to joint j at instance of time $\tau = \kappa$. Torque, $T(\theta)$, is calculated using recursive Newton-Euler method [8].

E. The Genetic Operators

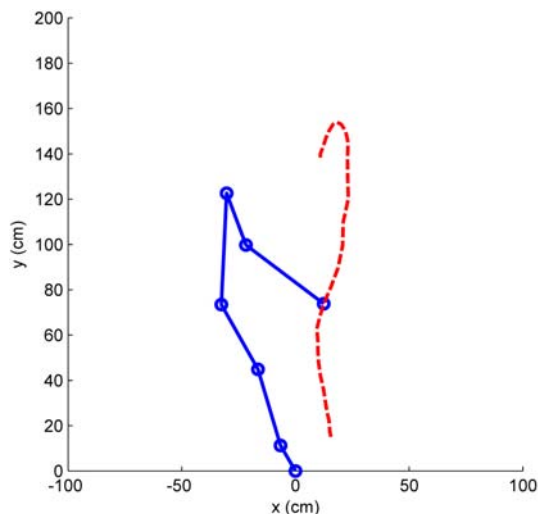


Fig. 5. Proposed trajectory along with a sample configuration.

Three different genetic operators are used in our algorithm, selection, crossover and mutation.

Reproduction or selection operator selects some of the current members to be passed to the next generation. The traditional method, proportionate selection or roulette wheel, is used in our approach [9]. The selection probability for each \mathbf{M}_μ , $R(\mathbf{M}_\mu)$, proportionate to its fitness function, $F(\mathbf{M}_\mu)$, in the population of m individuals is as below,

$$R(\mathbf{M}_\mu) = \frac{F(\mathbf{M}_\mu)}{\sum_{\mu=1}^m F(\mathbf{M}_\mu)}. \quad (19)$$

The next operator used in our approach is crossover which is a recombination operator that works on a pair of old chromosomes. We've used single point crossover which is introduced originally by Holland [10]. Under this type of crossover, each member in a pair is cut at one instance of time, τ , and two new members are formed. The first offspring receives the first part of the first parent along with the second part of the second parent whereas the other offspring receives the first part of the second parent and the second part of the first parent.

The last operator is mutation. Mutation is applied to each individual member. It randomly selects a member and alerts the chromosome at random instance of time $\tau = \kappa$, $\mathbf{v}|_{\tau=\kappa}$.

F. The Control Parameters

The most important control parameters are population size, crossover rate, mutation rate, generation gap, elitism [11]. Generation gap specifies how many of the members of the population will be replaced by the new offspring in each generation. Elitism is a selection strategy which guarantees the survival of the best member of one generation to the next. Without such a guarantee, it is possible for the best chromosome of the current generation to be lost due to mutation, crossover, or reproduction [12]. These parameters are listed in To evaluate the effect of different types of RNG \mathfrak{R} , two different setups are proposed. Table IV shows the parameters of the each setup.

Table III.

To evaluate the effect of different types of RNG \mathfrak{R} , two different setups are proposed. Table IV shows the parameters of the each setup.

Table III. Control parameters

Parameter	Quantity
Population Size, m	200
Generation Gap	50
Proportionate Selection	35
Elitism	15
Crossover Rate	70
Mutation Rate	30

Table IV. Parameters of the two simulated setups.

Random Number Generator	Parameters
Normal Distribution, $\mathfrak{R}_n(p)$	$\rho_n = 1, \sigma^2 = 1, \mu = 0$
Uniform Distribution, $\mathfrak{R}_u(p)$	$\rho_u = 1, a = 0, b = 1$

IV. RESULTS

A sample trajectory is proposed. Fig. 5 shows the proposed trajectory in dashed line along with a sample robot configuration on $\tau = 14$. The proposed trajectory is broken into 33 instances of time, $n = 33$. Each step is passed within 1/25s. The robot must traverse the predefined trajectory of length 1.61m in 1.32s.

Three other approaches in addition to our approach are applied to our sample manipulator and the predefined trajectory, [2, 4, 6]. The performance of the four approaches are compared, despite that the approaches mentioned in [2, 4, 6] have drift from the exact predefined trajectory. Fig. 6 shows the performance of the four approaches in a single task on vertical axis versus the generations on horizontal axis. Elitism operator is applied in all the 4 approaches. It is shown that within an equal number of generations, our approach reaches to a better fitness value. RNG with uniform distribution is used throughout this simulation.

To study the role of type of RNG, \mathfrak{R} , in the overall performance of our approach, two setups were considered, Table IV. Each setup ran for 100 generations. Fig. 7 shows the performance of the algorithm related to each setup. The fitness values are normalized for the best fitness value.

V. CONCLUSION

A motion planning and trajectory tracking optimization problem is solved. Genetic Algorithm (GA) is proposed as the optimization method. The introduced approach benefits from two key features. One lies on alternating the model of the manipulator from 6 degrees of rotary freedom to 4 degrees of rotary freedom and a single binary value. The other advantage of the proposed approach is the manner of data representation in each chromosome. Sigmoid function is used to infer the joints angle value. The proposed algorithm is compared with 3 other approaches in a single optimization task. It is shown that the introduced algorithm has a better performance. In addition the difference of using Random Number Generator (RNG) with uniform distribution and with normal distribution is addressed. It is shown that RNG with

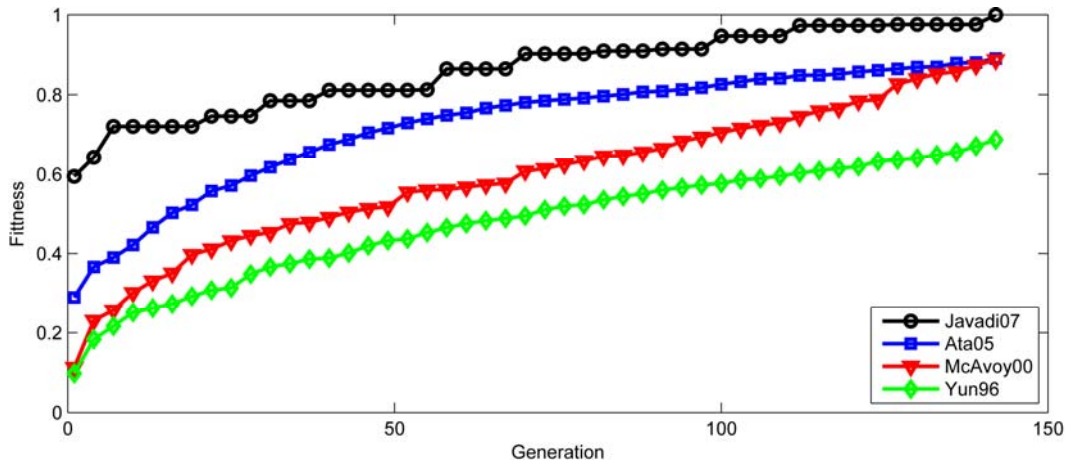


Fig. 6. Comparison between 4 different approaches.

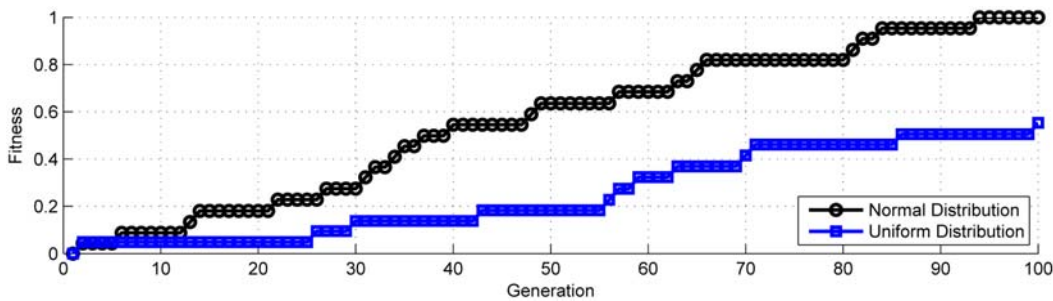


Fig. 7. Comparison between two different types of random number generators.

normal distribution can achieve better results.

ACKNOWLEDGMENT

The author thanks Dr. Ahmedreza Arshi and Dr. Elham Shirzad for their suggestions and technical help.

REFERENCES

- [1] A. C. Nearchou, "Solving the inverse kinematics problem of redundant robots operating in complex environments via a modified genetic algorithm," *Mechanism and Machine Theory*, vol. 33, pp. 273-292, 1998.
- [2] B. McAvoy, B. Sangolola, and Z. Szabad, "Optimal trajectory generation for redundant planar manipulator," in *IEEE International Conference on Systems, Man, and Cybernetics*, Nashville, 2000, pp. 3241-3246.
- [3] L. Tian and C. Collins, "Motion Planning for Redundant Manipulators Using a Floating Point Genetic Algorithm," *Journal of Intelligent and Robotic Systems*, vol. 38, pp. 297-312, 2003.
- [4] A. A. Ata and T. R. Myo, "Optimal Point-to-Point Trajectory Tracking of Redundant Manipulators using Generalized Pattern Search," *International Journal of Advanced Robotic Systems*, vol. 2, pp. 239-244, 2005.
- [5] Y. Davidor, "Genetic Algorithms and Robotics: A Heuristic Strategy for Optimization," World Scientific, 1991, pp. 163-168.
- [6] W. M. Yun and Y. G. Xi, "Optimum motion planning in joint space for robots using genetic algorithms," *Robotics and Autonomous Systems*, vol. 18, pp. 373-393, 1996.
- [7] A. R. Hiraakawa and A. Kawamura, "Trajectory Generation for Redundant Manipulators under Optimization of Consumed Electrical Energy," in *The 31st IAS Annual Meeting IAS*, 1996, pp. 1626-1632.
- [8] Y. Nakamura, *Advanced Robotics, Redundancy and Optimization*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1990.
- [9] D. E. Goldberg, *Genetic Algorithm in Search, Optimization and Machine Learning*. MA: Addison Wesley, 1989.

- [10] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: University of Michigan Press, 1975.
- [11] J. J. Grefenstette, "Optimization of control parameters for genetic algorithm," *IEEE Transaction on Systems, Man, Cybernetics*, vol. 16, pp. 122-128, 1986.
- [12] G. Rudolph, "Convergence Analysis of Canonical Genetic Algorithms," *IEEE Transactions on Neural Networks*, vol. 5, pp. 94-101, 1994.