Robotica (2005) volume 23, pp. 169–176. © 2005 Cambridge University Press
doi:10.1017/S0263574705001694 Printed in the United Kingdom

Optimal placement of a two-link planar manipulator using a genetic algorithm

Lianfang Tian and Curtis Collins

Department of Mechanical Engineering, University of California at Riverside, Riverside, CA 92521 (USA)

(Received in Final Form: August 4, 2003)

SUMMARY

In this paper, we study the base placement problem for a two-degree-freedom-robot. The feasible area for the robot base is described, and its manipulability measure is formulated. To find the optimal base location of the robot, we develop a genetic algorithm (GA) to search for the valid solution in the reachable area of the robot base. To speed up the evolutionary process, we keep the elitists from one generation to the next generation according to certain elitist rate. The genetic algorithm approach is beneficial because it can also be extended to trajectory planning for robots with more degrees-of-freedom. We demonstrate the algorithm with some examples.

KEYWORDS: Robot base placement; Optimization; Genetic algorithm.

1. INTRODUCTION

Industrial and medical robots are often used in high volume manufacturing processes to improve the productivity or fulfill some specified tasks. Such examples include loading a press, inserting electronic components, spot welding a workpiece, and conducting a robot-assisted surgery where the end-effector must reach a number of specified points and where repositioning of the patient is difficult to achieve. In these cases, the motion profile of the robot rarely changes throughout the whole operation. Therefore, searching for an optimal location for the robot base is a favorable solution to those problems.

Several researchers have investigated the problems of base placement of robots. Chirikjian *et al.*[1] developed several new metrics, which are particularly easy to compute and applicable to path generation, to be used to generate interpolated sequences of rigid and/or deformable motions of solid body. Martinez *et al.*[2] presented a critical analysis of the metric of rigid displacement obtained from so-called kinematic mapping and proposed some metrics for the set of spatial and planar displacements for a finite rigid body undergoing a finite displacement. Park[3] addressed the basic mathematical question in characterizing a set of desired goal frames and a particular left-invariant distance metric parametrized by length scale that is useful for kinematic applications is given. Tabarah *et al.*[4] illustrated an optimization procedure for determining the optimal base for one manipulator, or two cooperating manipulators, following a continuous path with constant velocity. They considered the constant velocity as constraint condition. Feddema[5] proposed an algorithm using a simple principle of coordinated motion to estimate the time of a joint-interpolated motion in determining the optimal placement of a robotic manipulator within a workcell for minimum time. Abdel-Malek *et al.*[6–8] described the criteria and implementation for the placement of robot manipulator with the objective of reaching a specified set of target points. The placement of a serial manipulator in a working environment is characterized by defining the position and orientation of the manipulator's base with respect to a fixed reference frame. Their approach is based on characterizing the placement by forcing a cost function to impel the work space envelope in terms of surface patches towards the target points subject to functionality constraints. Seraji[9] addressed a simple geometric approach to determine the appropriate base location from which a robot can reach a stationary or moving target point. Zeghloul and Pamanes[10,11] presented several kinematic performance criteria for the optimal placement of robots, and prescribed a general optimization method to automatically determine the placement of manipulators.

In the research mentioned above, the methods employed by the researchers are mainly concentrated on the conventional methods, such as numerical, geometrical and optimization methods in which the accurate knowledge of the system to be optimized is needed. Recently, intelligent methods (Fuzzy logic, neural network and genetic algorithm) have been developed that are widely used in different areas, especially in control and optimization problems. For these intelligent methods, even when prior complete knowledge about a system is not available, they can be used to conveniently control or optimize a complicated system Intelligent methods are also used in optimization for the trajectory planning of manipulators and for solving redundance resolution problems.

Mbeds *et al.*[12] proposed a fuzzy system to approximate the gradient of the harmonic function for continuous motion planning of manipulators in the known environment with unknown obstacles. Glacius and Yang[13,14] presented a model of a topologically organized neural network of a Hopfield type with nonlinear analog neurons for the robot path planning system both in a static and dynamic environment with a guaranteed solution. Shitaku[15] described a genetic algorithm (GA) for trajectory planning of an underwater manipulator, in which a polynomial is used to approximates time histories of the trajectory in joint space. The GA is used to search the parameters of the polynomial to minimize the energy consumption. Yano[16] applied a genetic algorithm

https://doi.org/10.1017/S0263574705001694 Published online by Cambridge University Press

(GA) to solve the position and movement of an end-effectors on the tip of a multi-joint robot arm. He defined objective functions in both Cartesian space and joint space, and combined them to optimize the robot trajectory.

In this paper, we first outline the problem of the placement of a robot base for a given path of its end-effector. We analyze the reachable workspace and derive a condition for singularity avoidance. Then, we apply a GA method to search for the optimal location of the robot base. An example simulation of a two degree-of-freedom robot is provided to demonstrate the effectiveness of the proposed method.

2. PROBLEM PRESENTATION

2.1. Feasible area of the robot base

In this work, we consider a robot which has two joints and two links as shown in Figure 1. The end-effector moves on the XY plane. Let the length of Link 1 and 2 be L_1 and L_2, respectively. Let the angles between Link 1 and positive X axis, and between Link 2 and extended Link 1 be θ_1 and θ_2, respectively. $P_s(x_s, y_s)$ and $P_t(x_t, y_t)$ are the starting and terminal points of the end-effector of the robot which moves along $\overrightarrow{P_s P_t}$. Let $P_b(x_b, y_b)$ be coordinate of the robot base, and the starting point $P_s(x_s, y_s)$ be located at the origin of XY frame. Our objective is to search for the optimal location of the robot base, so that the end-effector of the manipulator moves from the starting point P_s to the terminal point P_t in a minimum joint distance subject to additional constraints.

Trabia *et al.*[17] analyzed the feasible area of the robot base when the end-effector of the robot is located at one point within the boundary of robot workspace. Following Trabia, the feasible area of robot base, when the end-effector is positioned at a random point P, is the area between two concentric circles whose radii are expressed as follows:

$$R_1 = \sqrt{L_1^2 + L_t^2 + 2L_1L_2\min(\cos(\theta_2))} \tag{1a}$$

$$R_2 = \sqrt{L_1^2 + L_2^2 + 2L_1L_2\max(\cos(\theta_2))} \tag{1b}$$

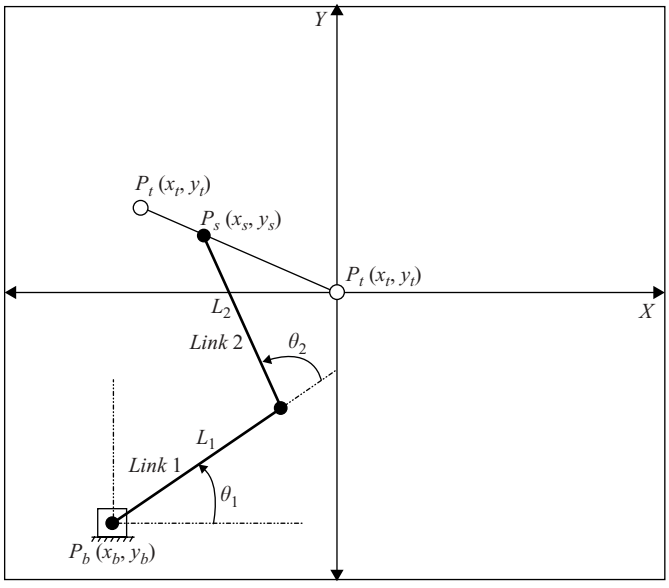


Fig. 1. Two-degree-freedom planar robot in Cartesian space.

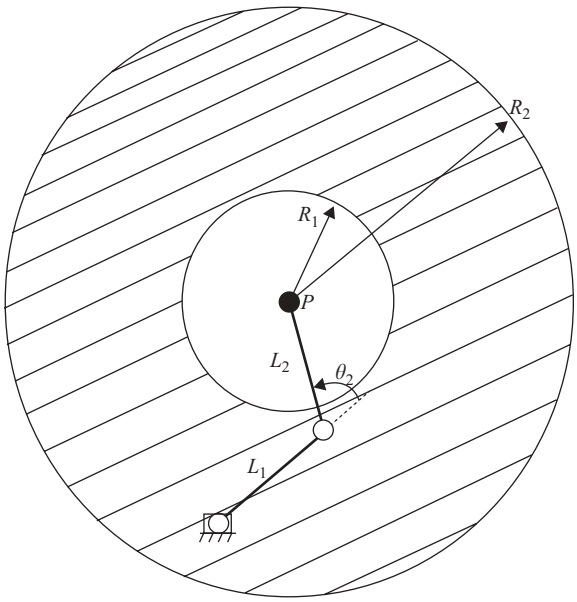


Fig. 2. Feasible area of the robot base for a desired end-effector position.

where, θ_2 is the angle between extended Link 1 and Link 2, and R_1 and R_2 are the inner and outer radius of feasible area of the robot base, respectively. Note that if $\theta_2 = 0$, then $R^2 = L_1 + L_2$. The locus of possible base locations is shown in the shaded region in Figure 2.

When the end-effector moves along a path, the outer boundary of the feasible area can be obtained by intersecting the two outer circles for the two end points of the path, and the inter boundary can be obtained by sweeping the inner circle along the path. There are three cases depending on the path length $|\overrightarrow{P_s P_t}|$.

$$\text{Case (a) for } |\overrightarrow{P_s P_t}| < R_2 - R_1, \tag{2a}$$

$$\text{Case (b) for } R_2 - R_1 < |\overrightarrow{P_s P_t}| < \sqrt{R_2^2 - R_1^2}, \tag{2b}$$

$$\text{Case (c) for } |\overrightarrow{P_s P_t}| > \sqrt{R_2^2 - R_1^2}. \tag{2c}$$

The shaded region in Figure 3 shows the feasible area for the base location of the robot.

Additional constraints may limit the robot base locus. For example, the robot base cannot interfere with fixed elements, such as a part feeder or a conveyer belt. The robot base also cannot be located along the prescribed path of the end-effector. In addition, the robot should avoid all possible obstacles in the workspace.

2.2. Path manipulability

Several kinematic performance criteria have been proposed for the optimal placement problem. Pamanes and Zeghloul[10,11] briefly described these criteria. For a serially connected nonredundant manipulator, when the end-effector passes close to a singular position, the determinant of the Jacobian matrix approaches zero, which results in excessive joint velocities and torques. To prevent this, it is natural to use the manipulability measure as a criterion for the optimal placement of the robot base.

https://doi.org/10.1017/S0263574705001694 Published online by Cambridge University Press

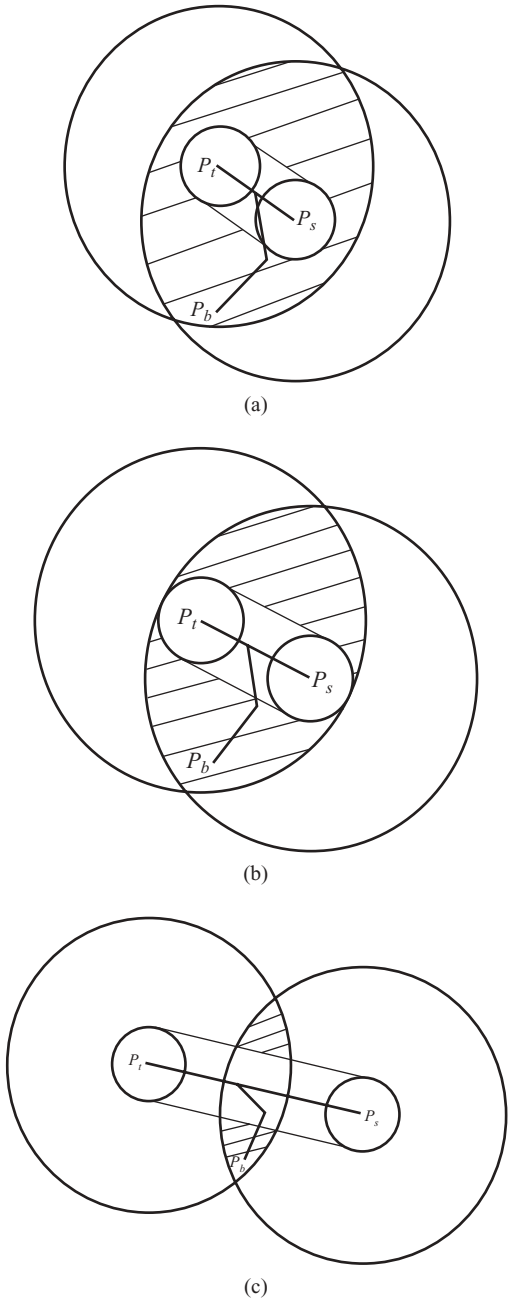


Fig. 3. Feasible area of the robot base when the end-effector moves along a path. For case (a), For case (b), For case (c).

As shown in Figure 1, the kinematic formulation for the two degree-of-freedom (dof) revolute-joint robot is as follows. The x and y coordinates of the end-effector can be expressed in terms of the input joint angles and link lengths as:

$$x_e = x_b + L_1 \cos\theta_1 + L_2 \cos(\theta_1 + \theta_2) \tag{3a}$$

$$y_e = y_b + L_1 \sin\theta_1 + L_2 \sin(\theta_1 + \theta_2) \tag{3b}$$

where x_b and y_b refer to the location of the base frame, L_1 and L_2 denote the link lengths, and θ_1 and θ_2 the joint angles.

The Jacobian matrix for the robot is

$$J(q) = \begin{bmatrix} -L_1 \sin\theta_1 - L_2 \sin\theta_2 & -L_2 \sin(\theta_1 + \theta_2) \\ L_1 \cos\theta_1 + L_2 \cos\theta_2 & L_2 \cos(\theta_1 + \theta_2) \end{bmatrix} \tag{4}$$

where q is the vector of joint coordinates, $q = \{\theta_1, \theta_2\}$, and

$$\theta_2 = \cos^{-1}\left(\frac{[(x_e - x_b)^2 + (y_e - y_b)^2] - \left(L_1^2 + L_2^2\right)}{2L_1L_2}\right)$$

or

$$\theta_2 = -\cos^{-1}\left(\frac{[(x_e - x_b)^2 + (y_e - y_b)^2] - \left(L_1^2 + L_2^2\right)}{2L_1L_2}\right)$$

θ_1 can be obtained from Equation (3).

The manipulability measure can then be written using J as:

$$w = \sqrt{\det JJ^T} \tag{5}$$

The path manipulability measure indicates the closeness of the Jacobian matrix manipulator to degeneracy while traversing a path. Therefore, it can be a constraint for the optimal placement of the robot base.

3. GENETIC ALGORITHM (GA)

In conventional methods for optimization problems, a precise methematical model is often required. When implementing a practical algorithm for the base placement problem, especially a multi-degree-freedom robot system, any modeling error in the parameters or uncertainties in the model will greatly affect the final results. Moreover, when the model of the manipulator is changed, the formulation of the optimization problem needs to be changed accordingly. Therefore, an intelligent optimization method known as a Genetic Algorithm, or (GA), is proposed to search for the optimal location of a robot base as it traverses a path. It is well known that the GA is an optimization algorithm based on the mechanics of natural genetics. The GA possesses the excellent ability for global searching. In addition, it does not require a precise mathematical model, and can be widely used to solve different optimization problems.[18–19]

The principle of the GA is as follows: To simulate the natural evolutionary process, the GA creates a population of randomly generated binary strings, Each binary string (called a chromosome) encodes the parameters to be solved. To create a new population, three genetic operators are needed. They are *reproduction, crossover* and *mutation*, respectively. The step from one population to the next is called a generation. The process for GA can be described as follows:

```
Begin
  Initialization
  Repeat
    Reproduction
    Crossover
    Mutation
  until termination condition is satisfied
End
```

Reproduction: Reproduction is a process in which individual strings are copied according to their objective values

https://doi.org/10.1017/S0263574705001694 Published online by Cambridge University Press

(Fitness Function). It means that strings with a higher fitness value have a higher probability of contributing one or more offspring in the next generation. This operator is an artificial version of natural selection.

Crossover: Crossover may proceed after reproduction. First, members of the newly reproduced strings (Chromosomes) in the mating pool are mated randomly. Second, each pair of strings undergoes crossing over as follows. An integer position k along the string is selected uniformly at random between 1 and the string length. Then, the new strings are created by swapping the characters between position $k + 1$ and the length of the strings inclusively.

Mutation: Although the crossover is a very potent means of exploring search spaces, it does have a disadvantage. Since it proceeds by recombining the information from parents, the children it produces contain only information that was already present in one parent or the other (or both). It is the mutation that is in charge of reintroducing some missing information back into the genetic pool.

As we proceed from one generation to the next, we need to decode a binary chromosome into the parameter values to define a potential solution. Given a potential solution, the GA evaluates how well the chromosome solves the problem by calculating a fitness function and assigning a fitness value to it. For a trajectory planning problem, the trajectory length or velocity of end-effector of the robot in joint or world space are often selected as fitness function. A higher fitness value gives a chromosome a better chance of being selected for mating. Once two chromosomes are selected, the GA will perform a crossover operation by swapping portions of their binary strings according to given crossover rate. The final step is that the GA generates random mutations by flipping a small percentage of bits in the whole population. After these operations, a new generation will be created, and the GA will be ready to evaluate it and proceed to the next generation. The evolutionary process will continue until it satisfies some conditions, such as the given number of generations.

4. OPTIMIZATION FOR THE PLACEMENT OF A ROBOT BASE

A workcell of a 2 dof planar robot is shown in Figure 1. In our calculations, the length for Link 1 and Link 2 are 0.5 m and 0.5 m, respectively. The first joint can rotate freely while the second joint is limited to $-150^\circ < \theta_2 < 150^\circ$ without loss of generality. The desired path is a straight line that starts at the origin $P_s(0.0, 0.0)$, and terminates at $P_t(-0.5, 0.15)$. The feasible area for the placement of the robot base is shown in Figure 4.

We select the minimum total rotation of joint 1 and 2 as the fitness function. When the velocities of joint 1 and 2 are constant, the smaller the total rotation angle the less time it takes for one cycle. When the end-effector of the robot moves from its starting point to the terminal point, the total joint displacement for joint i can be expressed as follows.

$$S_i = \sum_{j=0}^{n} \left({}^{(j+1)}\theta_i - {}^{(j)}\theta_i \right) \tag{6}$$

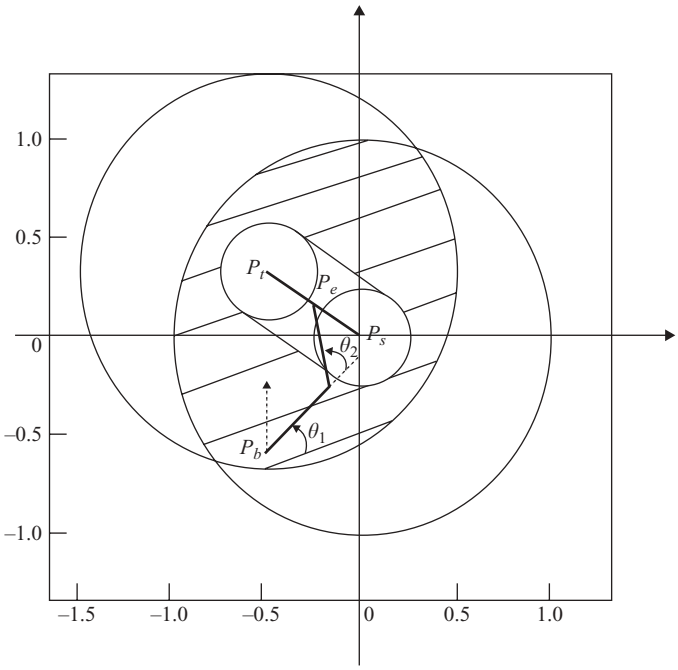


Fig. 4. Feasible base locations of 2 dof planar robot.

where, S_i is the total displacement of joint i, n is the number of steps of joint i, and ${}^{(j)}\theta_i$ is the position of joint i at jth step in joint space.

The objective is to search for an optimal location of the robot base, so that $S_i \rightarrow \min$. For the two-joint planar robot, the fitness function is obtained using a simple linear combination of S_1 and S_2. Then we can say

$$S(k) = \lambda S_1^{(k)} + (1 - \lambda) S_2^{(k)} \quad (0 < \lambda < 1) \tag{7}$$

where k is the kth chromosome in a generation.

In addition, once a base location is obtained, we require that it has at least a minimum manipulability value. Therefore, the evolutionary process has the following constraint for when the point P_e on the end-effector moves along the line $|\overrightarrow{P_s P_t}|$. We require that

$$\begin{cases} R_1 < |\overrightarrow{P_b P_e}| < R_2 \\ w(P_e) > w_{\min} \end{cases} \tag{8}$$

where, R_1 and R_2 are expressed in Equations (1a) and (1b), and

$w(P_e)$ is the manipulability measure for any point P_e along line $|\overrightarrow{P_s P_t}|$

$w_{\min}$ is the minimum manipulability measure the robot configuration should satisfy.

5. SIMULATIONS

5.1. Genetic encoding

We encode the coordinates (x_b, y_b) of the location of the robot base into a chromosome, which is a character string consisting of 0 or 1. The chromosome is comprised of two

https://doi.org/10.1017/S0263574705001694 Published online by Cambridge University Press

parts (two genes), one is for coordinate x_b, the other is y_b. The number of bits for the chromosome can be determined by the changing range of coordinates of robot base according to the desired precision. The higher the precision, the more bits needed. In general there is a tradeoff between the desired precision, and the computing time.

5.2. Evaluation of fitness

A smaller total displacement in joint space corresponds to a more desirable solution. Because of this,we use the inverse of objective function defined in Equation (7) as the fitness function. That is we let.

$$F(k) = 1/S(k) \tag{9}$$

The chromosome can survive to the next generation if it has a better fitness value than others.

5.3. Reproduction

In our proposed GA, the general reproduction strategy for the new generation is that parents with better fitness have a better chance of selection. For this purpose, the roulette wheel method, which is sized according to fitness, is used to generate the offspring. However, in the whole evolutionary process, each new generation can be generated by the combination of parents in the mating pool in an huge number of ways. Finding an effective generation among those combinations without falling into local solution is important. The preferable solution to this problem is that we move some of the fitter ones (elitists) into next generation. The fitter ones obtained in the previous generation are called elitists, and the number of elitists can be determined by an elitists rate. If the elitist rate is larger, it will move more elitists to next generation. On the other hand, it will affect the variety of the offspring. If the elitist rate is too large, it could cause the next generation to fall into a local optimization and converge to an undesirable location. Therefore, in order to avoid the loss of variety, it is important to select the elitist rate p_e carefully.

5.4. Crossover

A pair of parent chromosomes to be crossed over are randomly chosen according to the given crossover rate, p_c, with the crossover position selected randomly. Once the crossover position is determined, the genes are exchanged as follows. Assume the two chromosomes to be crossed over are

$$a_1 a_2 \cdots a_i a_{i+1} \cdots a_{n-1} a_n$$
$$b_1 b_2 \cdots b_i b_{i+1} \cdots b_{n-1} b_n$$

If the crossover position is selected at i, then after this operation, the offspring generated from their parents are

$$a_1 a_2 \cdots a_i b_{i+1} \cdots b_{n-1} b_n$$
$$b_1 b_2 \cdots b_i a_{i+1} \cdots a_{n-1} a_n$$

5.5. Mutation

Mutation is performed on a bit-by-bit basis. Given the mutation rate p_m, every bit in the whole population has an equal chance of undergoing a mutation. Once a bit is identified, it is changed from 1 to 0, or vice versa.

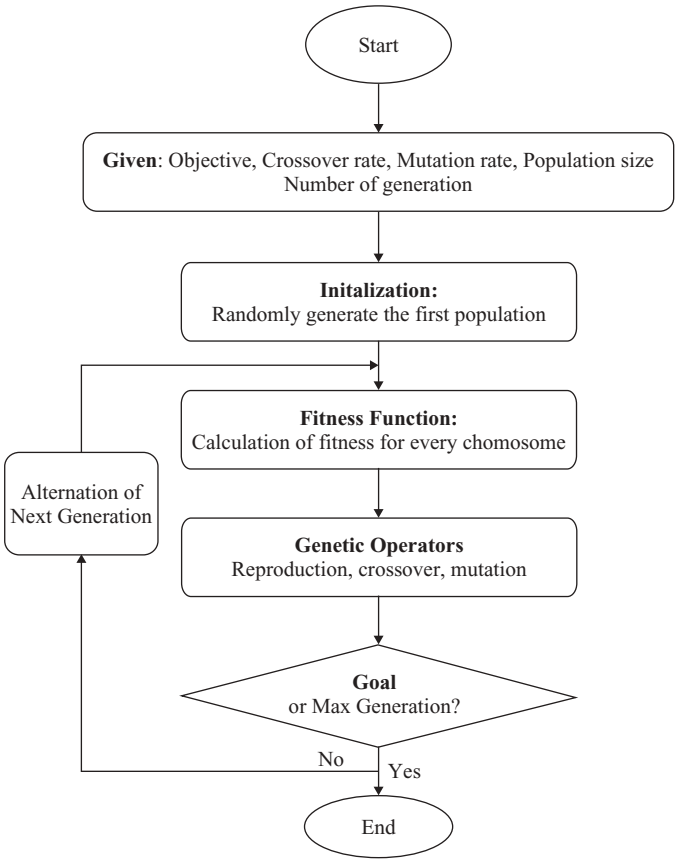


Fig. 5. Flow chart for the GA.

5.6. Simulations

The flow chart for the evolutionary process is shown in Figure 5. In the simulations, the population is $Popul = 60$, the bits for every gene in a chromosome is given as 12, the elitist rate is $p_e = 0.05$, the crossover rate is $p_c = 0.3$, and the mutation rate is $p_m = 0.02$.

Figure 6(a) demonstrates the evolutionary result without the elitists kept to the next generation, while Figure 6(b) demonstrates the evolutionary result keeping the elitists in the current generation to next generation. After 200 generations, the searching locations for (a) and (b) are located at $(-0.608, -0.686)$ and $(-0.605, -0.693)$, respectively. From Figure 6, it is shown that there is some undulation in the evolutionary process in the case without elitist, though it gradually converges to the optimal location of robot base. On the other hand, the evolutionary process quickly converges to the optimal location in the case with elitists. It can be seen that the initial values for both cases are different. This is because ${}^{(j+1)}\theta_i$ and ${}^{(j)}\theta_i$ in Equation (6) are generated randomly at each run, which results in different initial values expressed by Equation (9) for the two cases. For the latter case, the elitist rate is important for the whole evolutionary process, higher elitist rate may lead to local convergence. Therefore, in order to get global optimal result, a lower elitist rate is preferable. Figure 7 shows the trajectories of joint 1 and 2 at the optimal location of the robot

https://doi.org/10.1017/S0263574705001694 Published online by Cambridge University Press

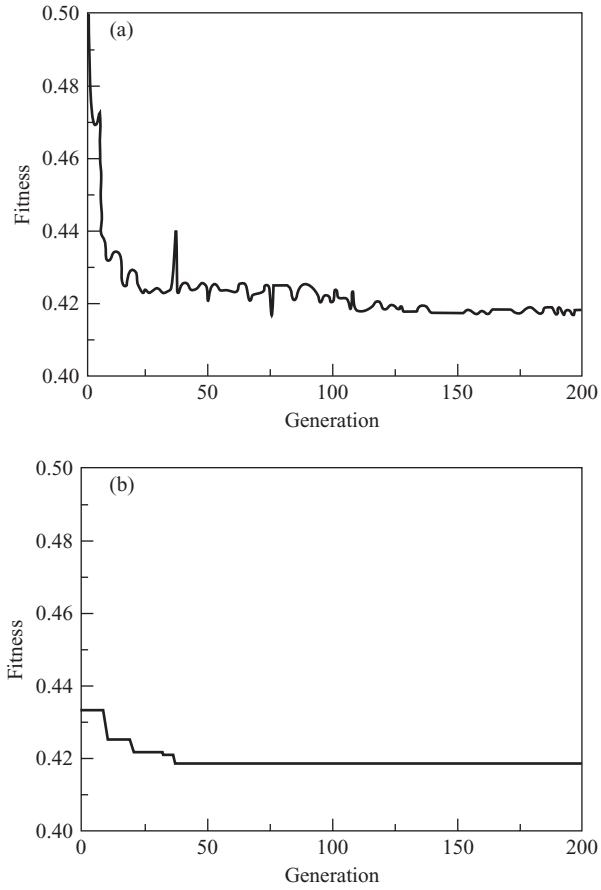


Fig. 6. The comparison of evolutionary process without and with elitists (a) Without elitists, (b) With elitists.

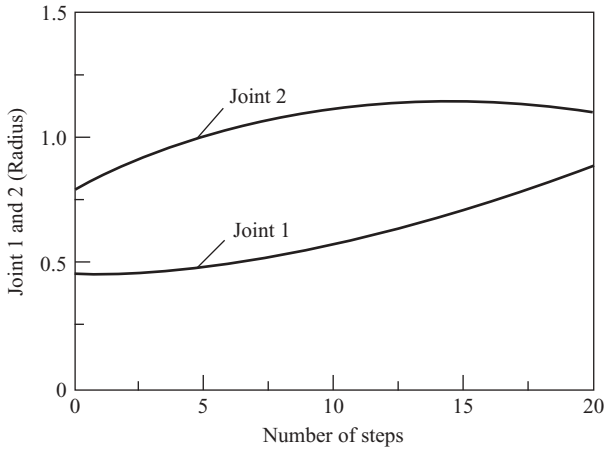


Fig. 7. Angles of joint 1 and 2 for the elbow-down configuration.

base under the condition of elbow-down. Figure 8 shows the elbow-down configurations in the workspace.

Simulations under the condition of elbow-up are also carried out. Figure 9 is the evolutionary process under the condition of elbow-up, resulting in the optimal location (0.374, 0.783). As shown in Figure 9, the evolutionary process quickly converges to the optimal location of robot base after 20 generations. Figure 10 and Figure 11 are the trajectories for joint 1 and 2 in the workspace and joint space, respectively.

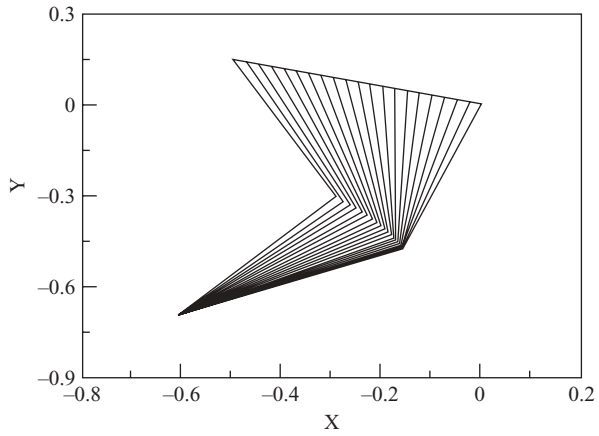


Fig. 8. Elbow-down configuration of the robot at optimal base position.

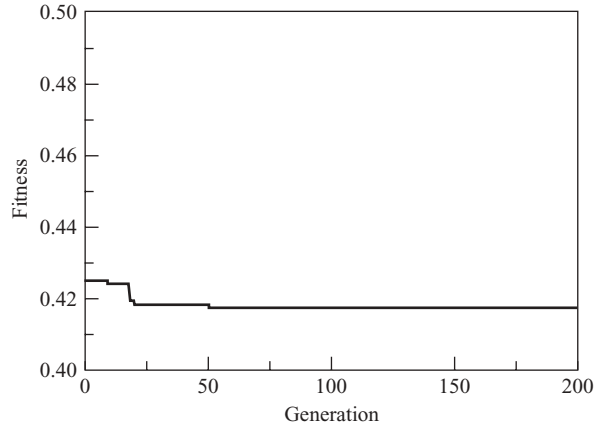


Fig. 9. The evolutionary process for elbow-up configuration.

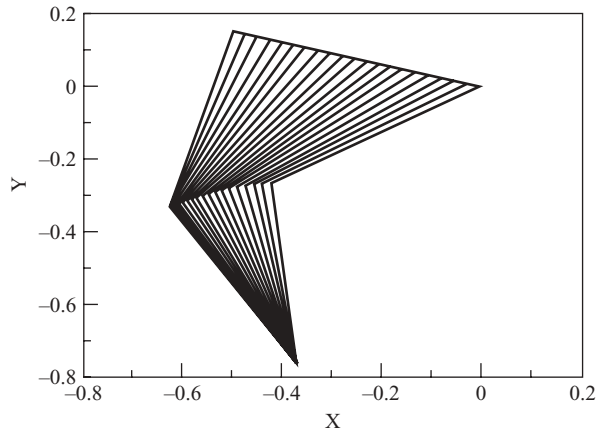


Fig. 10. Elbow-up configuration of the robot at optimal base position.

Figure 12 is the comparison of different populations in a generation. After the same number of generations, the larger the population the better the result. For a fixed population, more generations are needed to obtain better results.

https://doi.org/10.1017/S0263574705001694 Published online by Cambridge University Press

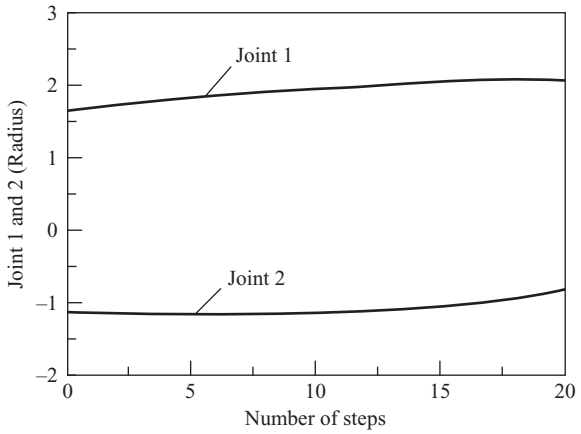


Fig. 11. Angles of joint 1 and 2 for the elbow-up configuration.

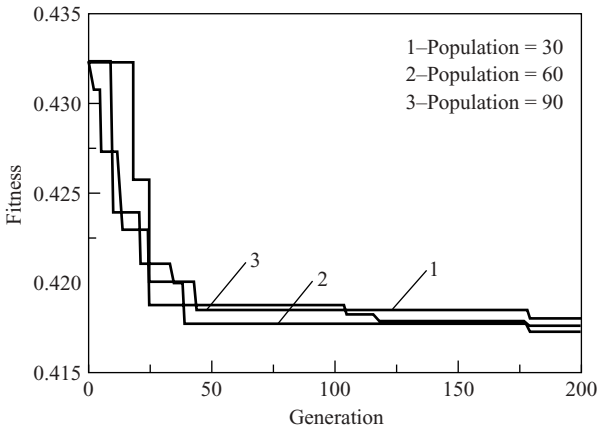


Fig. 12. The evolutionary process for different populations.

6. CONCLUSION

We studied the optimal placement problem for a robot base. T feasible area of a 2 Dof robot base placement is described when its end-effector moves from the starting point to the terminal point. An intelligent optimization method based on the genetic algorithm is proposed. In the proposed GA, in order to obtain the optimal location for the robot in the whole workspace, we keep the elitists in the current generation to next generation. Simulations and comparisons are carried out. Here, the following points are clarified:

(a) If the end-effector is fixed at one point, the feasible area for a two Dof robot base placement with joint limit is two concentric circles. If the end-effector moves along a path, the feasible area is shown in Figure 3.
(b) The proposed GA is practical for the optimization of robot base placement. It does not need complete knowledge of a system, and it is an effective optimization method, especially for a complicated system.
(c) The proposed GA is better than the simple GA without the elitists kept to next generation. The evolutionary process is quicker to converge to the optimal location as shown in Figure 6.
(d) We also studied the Convergence of different population sizes in a fixed number of generations as shown in Figure 12. It can be seen that the evolutionary process with large population is quicker to converge. While the result is nearer to the optimal location, it comes at a higher computational price.

In this study, it can be seen that the proposed GA is practical and effective for the optimization of robot base placement. The method can be extended to optimize the trajectory for the end-effector of a robot, and it also can be applied to solve the resolution problem for redundant manipulators.

Acknowledgement

This research was supported by Natural Science Foundation of South China University of Technology, grant No: B8-106-016 and Doctor Start Foundation of Guangdong, grant No: B9-108-061. The authors would thank the anonymous reviewers for their appreciated suggestions for this paper.

References

1. G. S. Chirikjian and S. Zhou, "Metrics on motion and deformation of solid models," *ASME J. of Mechanical Design* **120**, 252–261 (1998).
2. J. M. R. Martinez and J. Duffy, "On the metrics of rigid body displacement for infinite and finite bodies," *ASME J. of Mechanical Design* **117**, 41–47 (1995).
3. F. C. Park, "Distance metrics on the rigid-body motions with applications to mechanicsm design," *ASME J. of Mechanical Design* **117**, 48–54 (1995).
4. B. Tabarah, B. Benhabib, R. Fenton and R. Cohen, "Cycle-time optimization for single-arm and two-arm robots performing continuous path operation," *21st Biennial Mechanisms Conference*, Chicago IL (1990) pp. 401–406.
5. J. T. Feddema, "Kinematically optimal robot placement for minimum time coordinated motion," *Proc. of the 1996 IEEE Int. Conf. of Robotics and Automation*, Minneapolis, MN (1996) pp. 3395–3400.
6. K. Abdel-Malek and W. Yu, "On the placement of serial manipulator," *Proceedings of DETC00 2000 ASME Design Engineering Technical Conferences*, Baltimore, MD (2000) pp. 1–8.
7. K. Abdel-Malek, H.-J. YeH and N. Khariallah, "Workspace, void, and volume determination of the general 5 DOF manipulator," *Mechanics of Structure and Machines* **27**, No. 1, 91–117 (1999).
8. K. Abdel-Malk, F. Adkins, H.-J. Yeh and E. J. Haug, "On the determination of boundaries to manipulator workspaces," *Robotics and Computer-Integrated Manufacturing* **13**, No. 1, 63–72 (1997).
9. H. Seraji, "Reachability analysis for base placement in mobile manipulator," *J. of Robotic Systems* **12**, No. 1, 29–43 (1995).
10. S. Zeghloul and J. A. Pamanes, "Multi-criteria optimal placement of robots in constrained environments," *Robitica* **11**, Part 1, 105–110 (1993).
11. J. A. Pamanes and S. Zeghloul, "Optimal placement of robotic manipulators using multiple kinematic criteria," *Proc. of the 1991 IEEE Int. Conf. On Robotics and Automation*, Sacramento, CA (1991) pp. 933–938.
12. J. B. Mbede, X. Huang and M. Wang, "Fuzzy motion planning among dynamic obstacles using artificial potential fields for robot manipulators," *Robotics and Autonomous System* **32**, 61–72 (2000).
13. R. Glasius, A. Komoda and S. C. A. M. Gielen, "Neural network dynamics for path planning and obstacles avoidance," *Neural Networks* **8**, No. 1, 125–133 (1995).
14. S. X. Yang and M. Meng, "An efficient neural network approach to dynamic robot motion planning," *Neural Networks* **13**, 143–148 (2000).
15. E. Shitaku, "Minimum energy trajectory for an underwater manipulator and its simple planning method by using a genetic algorithm," *Advanced Robotics* **13**, No. 2, 115–138 (1999).

https://doi.org/10.1017/S0263574705001694 Published online by Cambridge University Press

16. F. Yano and Y. Toyoda, "Preferable movement of a multi-joint robot arm using genetic algorithm," *SPIE* **3837**, 80–88 (1999).
17. M. B. Tabia and M. Kathari, "Placement of a manipulator for minimum cycly time," *J. of Robotic Systems* **16**, No. 8, 419–431 (1999).
18. Y. Davidor, *Genetic Algorithms and Robotics, a heuristic strategy for optimization* (World Scientific Publishing Co. Pte. Ltd., Bangalore, India, 1991).
19. Z. Michalewicz, *Genetic Algorithms + Data structures = Evolution programs* (Springer-Verlag Berlin Heidelberg, Germany, 1992).

https://doi.org/10.1017/S0263574705001694 Published online by Cambridge University Press