

Optimal trajectory planning and obstacle avoidance for flexible manipulators using generalized pattern search

Atef Ata^{1*}, Thi Myo²

¹ Department of Engineering Mathematics and Physics, Faculty of Engineering, Alexandria University, Alexandria (21544), Egypt

² NMechatronics Division, University Industry Selangor (UNISEL), Malaysia

(Received March 14 2008, Accepted June 1 2008)

Abstract. Flexible-link manipulators have high demand in many industrial and space applications. This is simply because they have many advantages over the rigid manipulators such as light weight, fast response time and large payload capacity. Optimal trajectories for a two-link rigid-flexible manipulator based on trajectory tracking and obstacle avoidance criteria are designed using Generalized Pattern Search algorithm (GPS). The GPS algorithm is a hybrid optimization technique combining the Genetic Algorithm and the Pattern Search techniques. The kinematics analysis of flexible link is briefly discussed where the model of flexible link is derived using the Euler-Bernoulli's beam theory. Simulation analysis for trajectory tracking as well as obstacle avoidance is carried out and the obtained results verified the validity of the proposed approach.

Keywords: trajectory planning, optimal, Genetic Algorithm, Pattern Search, flexible manipulator, obstacle avoidance

1 Introduction

The problem of optimal trajectory planning for flexible manipulators is much more difficult compared to rigid robots. The main difficulty in the control of flexible manipulators is to provide precise tip tracking in the operational space. Even if joint angles are controlled successfully, the end-effector of the manipulator deviates from the desired position because of link deflections and vibrations^[6]. That deflection of the manipulator arms due to flexibility and mass load causes positioning error. The magnitude of the error depends on the amount of mass load and arm positions and the stiffness characteristics of arms^[7]. Also the kinematic equations of the flexible manipulators are highly nonlinear and difficult to solve using the ordinary inverse kinematics procedure. The utilization of the Genetic Algorithm as a stochastic optimization tool that depends on the principles of natural selection and natural evolution for nonlinear problems solves a lot of optimization problems^[3].

The problem of designing an optimal trajectory for flexible manipulator has attracted many researches in the past two decades. Wahyudi and Dissanayake^[10] introduced a computational technique, based on radial basis functions, for deriving trajectories for a single link flexible robot arm. Using an approximation based on radial basis function and point collocation method, the trajectory planning problem is transformed into a linear least-squares problem.

Kojima and Kibe^[4] constructed the trajectory planning algorithm for a two link rigid-flexible robot manipulator using Genetic Algorithm for residual vibration reduction. The first and second joint velocities are described using four cubic polynomial functions. The fitness function of the Genetic Algorithm for the residual vibration reduction of the flexible link is defined using the four unknown parameters as the genes. The numerical calculations and the experiments have been carried out to ensure the residual vibration can be reduced remarkably. Rouvinen et al.^[7] presented a Neural Network (NN) based trajectory planning method to

* Corresponding author. E-mail address: atefa@iee.org.

calculate modifications in the command values of joint angles for tracking control problem, so that the position error of the tip of the manipulator in the operational space is minimized. A Proportional Derivative (PD) controller is applied for the joint angle control, where the reference inputs are the modified values calculated by the neural network. Simulations are performed to evaluate the performance of the trajectory planning method and the control procedure.

Shaheed et al.^[8] presented an open-loop control strategy for vibration suppression of a single-link flexible manipulator system using Genetic Algorithm. The filtered torque inputs thus developed are applied to the system in an open-loop configuration and their performances in suppressing structural vibrations of the system are assessed in comparison to a bang-bang torque input. A comparative study of the low-pass and band-stop filtered torque inputs in suppressing the system vibrations are also presented and discussed. Yue et al.^[12] focused on the problem of point-to-point trajectory planning for flexible redundant manipulators in joint space. They presented a trajectory planning method to minimize vibration and/or executing time based on Genetic Algorithm. Kinematics redundancy is integrated into the presented method and Quadrinomial and Quintic polynomial were used to describe the segments. They simulated the proposed algorithm on a planar three-links flexible manipulator.

In this paper, an optimal trajectory for a rigid-flexible manipulator is presented using the Generalized Pattern Search (GPS). The GPS is a hybrid optimization technique combining the Genetic Algorithm and the Pattern Search and it was applied successfully to design an optimal trajectory for redundant rigid manipulators including the obstacle avoidance caseu [1, 2]. The kinematics of the rigid-flexible manipulator is presented and the elastic displacement of the flexible link is obtained. Simulation analysis for two cases are carried out to verify the possibility of extended this technique for flexible manipulators.

2 Problem formulation

Consider the rigid-flexible manipulator shown in Fig. 1. The first link is rigid while the second link is flexible and the mathematical model of the flexible link is derived using the Euler-Bernoulli's beam theory where shear deformation and rotary inertia can be neglected. The tangential coordinate system is employed to describe the elastic behavior of the second link where the maximum elastic displacement $w(x, t)$ is expected at the end-effector. X_0Y_0 frame represents reference coordinate frame while X_1Y_1 and X_2Y_2 are moving coordinate frames with origin at the hubs of link 1 and link 2 respectively. The angles θ_1 and θ_2 are the revolving angles of the two links with respect to their frames. This paper focuses on the optimal trajectory planning for two applications. The first one is the optimal trajectory tracking of flexible manipulator that minimizes the tracking error and joint movements and the other one is obstacle avoidance trajectory which avoids collision with any predefined obstacle in the course of motion. The robot joints are assumed to rotate 360 degree without any physical limit.

3 Kinematics analysis

To describe the kinematics of the flexible robot, reference frames have been assigned at each joint as shown in Fig. 1. The coordinates of any points on the flexible link can be given as:

$$R_x = L_1 \cos \theta_1 + x \cos \theta - w(x, t) \sin \theta \quad (1)$$

$$R_y = L_1 \sin \theta_1 + x \sin \theta + w(x, t) \cos \theta \quad (2)$$

Where, L_1 is the length of the first link, x is the location of the undeformed position on the flexible link, $w(x, t)$ is the transverse elastic displacement measured from the x_2 axis and θ_1 and θ_2 represent the joint angular motions. The elastic deflection of the flexible link can be obtained by solving the beam equation^[5]:

$$\rho \frac{\partial^2 w}{\partial t^2} + EI \frac{\partial^4 w}{\partial x^4} = 0 \quad (3)$$

Subject to the fixed-free homogeneous boundary conditions

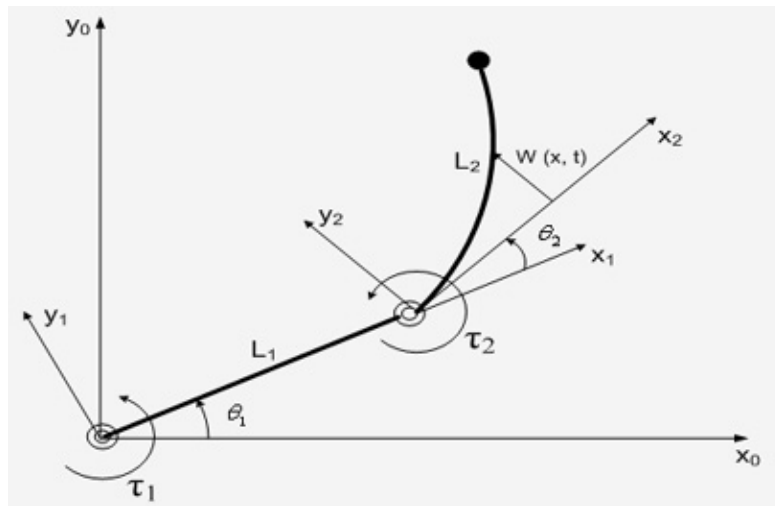


Fig. 1. Schematic diagram of rigid-flexible manipulator

$$w(0, t) = 0 \tag{4}$$

$$\frac{\partial}{\partial x}(w(0, t)) = 0 \tag{5}$$

$$\frac{\partial^2}{\partial x^2}(w(L_2, t)) = 0 \tag{6}$$

$$\frac{\partial^3}{\partial x^3}(w(L_2, t)) = 0 \tag{7}$$

Since the main objective is to design an optimal trajectory by minimizing the tracking error as well as the joint angle movements, the elastic deflection $w(x, t)$ should be obtained first. Using the separation of variables technique, $w(x, t)$ can be expressed as:

$$w(x, t) = \sum_{i=1}^m \varphi_i(x)\eta_i(t) \tag{8}$$

Where, the link bending deflection, $\varphi(x)$ can be described by the shape function in the spatial coordinate alone, $\eta(t)$ is the time function, and m is the number of mode shapes taking into considerations. Using modal analysis, the mode shape for fixed-mass boundary conditions is given by Meirovitch^[5].

$$\varphi(x) = \sin \beta_i x - \sinh \beta_i x - \sigma(\cos \beta_i x - \cosh \beta_i x) \tag{9}$$

where $\sigma = \frac{\sin \beta_i L_2 + \sinh \beta_i L_2}{\cos \beta_i L_2 - \cosh \beta_i L_2}$ and $\beta^4 = \frac{\rho \omega_n^2}{EI}$ in which E = Young's modulus, I = area moment of inertia, ρ = its mass per unit length of the flexible link.

On the other hand, the time function $\eta(t)$ can be described as a simple Harmonic motion and can be written as:

$$m\ddot{\eta}(t) + \omega_i^2 \eta(t) = 0 \tag{10}$$

The solution can be given in the form

$$\eta_t(t) = \eta_i(0) \cos \omega_i t \tag{11}$$

Where $\eta_i(0) = \int_0^{L_2} \rho w_i(x)w_0(x)dx$, $i = 1, 2, \dots$ and $w_0(x)$ is the initial displacement function.

4 Generalized pattern search

Generalized Pattern Search is a hybrid optimization algorithm used to find optimized angles according to some desired objectives. In GPS, Genetic Algorithm is used for the global search and Pattern Search Algorithm is used for the local search. GA has the advantage not to be trapped in local optima and proceeds toward global optima because GA searches from a population not a single point. Adding the Pattern Search optimization method refines the obtained result at each stage. As with any optimization routine, the evaluation function provides the mechanism for determining the direction of the search. The evaluation or fitness function is defined based on end-effector positioning error and joint angle displacements. Since a hybrid optimization approach is applied, the evaluation function is defined separately for Genetic Algorithm and Pattern Search methods. The objective function for GA is defined as follows:

$$F_{obj} = C_1 E_e + C_2 D_j \quad (12)$$

Where, E_e is the error between desired position and generated position of end-effector, D_j is the joint displacements between successive points and C_1 and C_2 are weighting factors to control the desired configuration which satisfy the constraint $C_1 + C_2 = 1$ used also by Tian and Collins^[9]. Since the objective is to minimize the error between the desired and generated position of end-effector, E_e will be defined as

$$E_e = \sum_{i=1}^n \sqrt{(x^i - x_g^i)^2 + (y^i - y_g^i)^2} \quad (13)$$

Where, (x^i, y^i) are desired end-effector positions and x_g^j, y_g^j are generated end-effector positions. The joint displacements between successive points are considered in evaluation function in order to minimize actuator motions. To minimize the joint movements along the trajectory, the function can be written as:

$$D_j = \sum_{i=1, k=1}^{i=m, k=2} (\theta_k^{i+1} - \theta_k^i)^2 \quad (14)$$

Where θ_k^i is the current joint angle for joint number k and θ_k^{i+1} is the next joint angle for joint number k . Pattern Search method is used only to reduce the tracking error of end-effector, so the evaluation function will be defined as:

$$F_{eval} = \sum_{i=1}^n \sqrt{(x^i - x_g^i)^2 + (y^i - y_g^i)^2} \quad (15)$$

The parameters used for GA and Pattern Search in this study are listed in Tab. 1 and 2. The termination criterion applied here is the number of generation. Through the simulation process it is found that there is no improvement in the solution after 200 generations.

Table 1. Genetic Algorithm parameters

Population	100
Fitness scaling	Rank
Selection	Stochastic Uniform
Reproduction	Elite count=2, crossover rate=1.2
Mutation	Gaussian(scale=1, shrink=1)
Crossover	Scattered function
Migration	Forward (fraction=0.2, Interval=20)
Generation	200

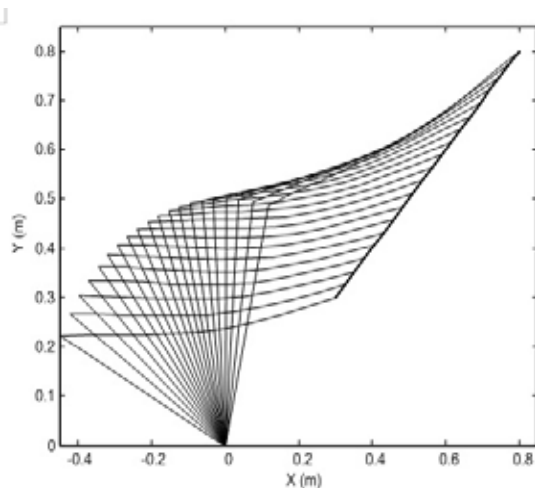
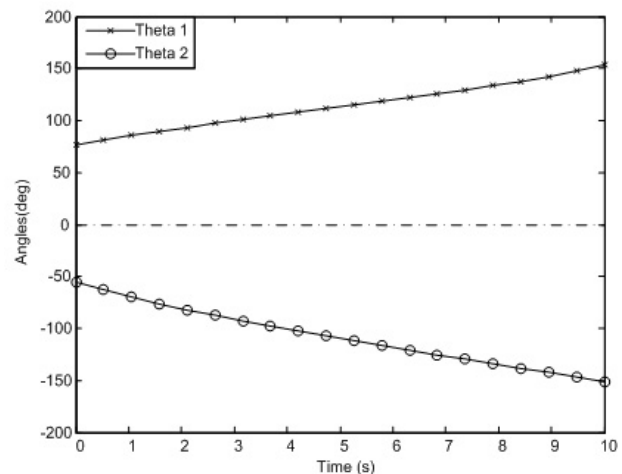
Table 2. Pattern Search parameters

Poll	Complete
Search	Complete
Polling Order	Consecutive

5 Point-to-point trajectory tracking

Simulations are carried out using MATLAB Toolbox Genetic Algorithm. The straight line and circle trajectories for the end-effector are designed and the Generalized Pattern Search is utilized to track these trajectories with minimum tracking error and joint displacements. The weighting factors are defined as $C_1 = 0.3$, $C_2 = 0.7$ and the traveling time taken is 10 seconds. The line trajectory starts from $(0.8, 0.8)m$ and ends at $(0.3, 0.3)m$ and the initial joint space configurations are defined as $\theta_1^1 = 60^\circ$ and $\theta_2^1 = 90^\circ$. For the circle trajectory, the circle is centered at $(0.3, 0.7)m$ with radius $0.2m$. The initial joint space configurations are defined as $\theta_1^1 = 60^\circ$ and $\theta_2^1 = -90^\circ$. The parameters used for this study are $L_1 = 0.5m$, $L_2 = 0.75m$, $EI = 2.4507Nm^2$, and $\rho = 0.5kg/m$.

The simulation results for both trajectories are illustrated in the following Figures. Fig. 2 and 5 illustrate the manipulator configuration, while Fig. 3 and 6 demonstrate the optimized joint angles. Fig. 4 and 7 illustrate the tracking error between the desired and actual trajectory obtained.

**Fig. 2.** Robot configuration**Fig. 3.** Optimized joint angles

6 Obstacle avoidance algorithm

In many robotic applications, there exist some obstacles in the workspace of the manipulator. The end-effector is required to follow the prescribed trajectory and avoid collision with any obstacle in the course of motion as well. The Generalized pattern search algorithm is applied here to design optimal collision-free trajectories for the end-effector. The flowchart of proposed obstacle avoidance algorithm using Generalized Pattern Search method is shown in Fig. 8.

In this approach, the optimal trajectory for manipulators, with end-effectors moving from a specified starting point to the goal point is generated. Specified number of intermediate points is selected and interpolated using cubic-spline interpolation to obtain a smooth. In order to avoid manipulator's end-effector collision with surrounding obstacles, interpolated points are checked either inside or outside of obstacles. Two types of obstacles are introduced here, circular and triangular. The triangular obstacle will be approximated as a circle passing through its three vertices. This approach enables modeling of many obstacles and ensures that there is a safety distance between the end-effector and the obstacles.

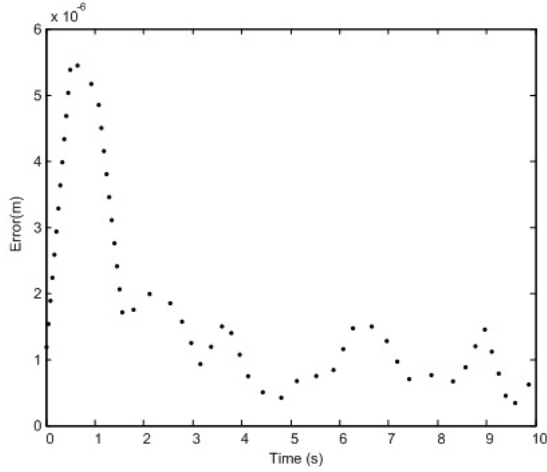


Fig. 4. Tracking error

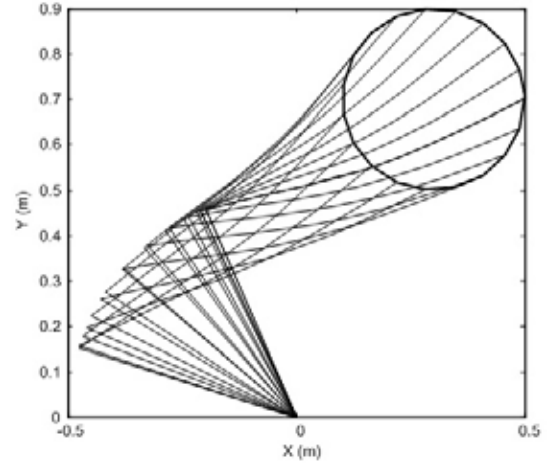


Fig. 5. Robot configuration

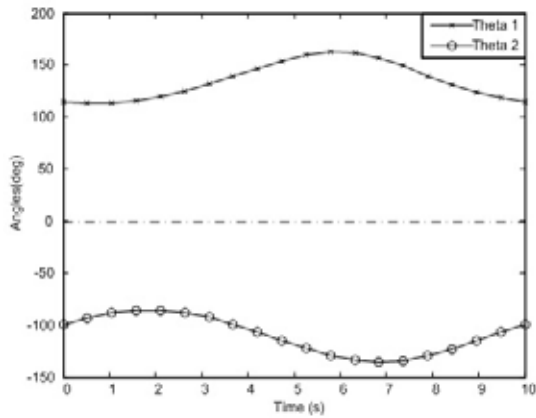


Fig. 6. Optimized joint angles

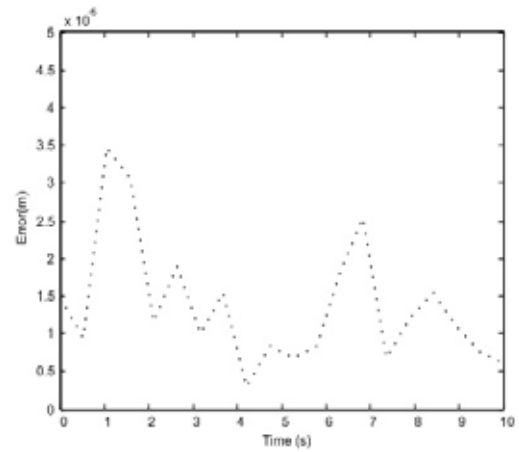


Fig. 7. Tracking Error

The algorithm will compare the distance between interpolated points and center of circle and radius of circle for avoiding obstacles. In the case of triangle obstacle, the longest distance between the cancroids and vertices of triangle will be used instead of radius as shown in Fig. 9. Points which are inside obstacles will be replaced by new points as follows:

$$\theta = A \tan 2(x_c - x(i), y_c - y(i)) \quad (16)$$

Where, (x_c, y_c) is the center of obstacle and (x, y) is the interpolated point which lies inside the obstacle and new points will be obtained using:

$$x_n = x - ((R - d) \times \cos(\alpha)) \quad (17)$$

$$y_n = y - ((R - d) \times \sin(\alpha)) \quad (18)$$

Where, $\alpha = 90 - \theta$, R is the radius of obstacle plus distance required away from obstacle and d is the distance between interpolated points located inside obstacle and center of obstacle.

The algorithm will check again to avoid obstacles collision. After ensuring all trajectory points are not trapped inside obstacles, the GPS finds optimized joint angles of each link of manipulator to track the resulted trajectory quite exactly.

The weighting factors are defined as $C_1 = 0.3$, $C_2 = 0.7$ and the time interval is assumed to be 10 seconds. Two cases are considered here for verification of the proposed algorithm. In the first case study, a circle is placed inside the manipulator workspace. It is centered at $(0.5, 0.4)m$ with radius $0.1m$. The trajectory starts from point $(0.7, 0.7)m$ to point $(0.3, 0.3)m$ and the initial joint configurations are $\theta_1^1 = 60^\circ$ and $\theta_2^1 =$

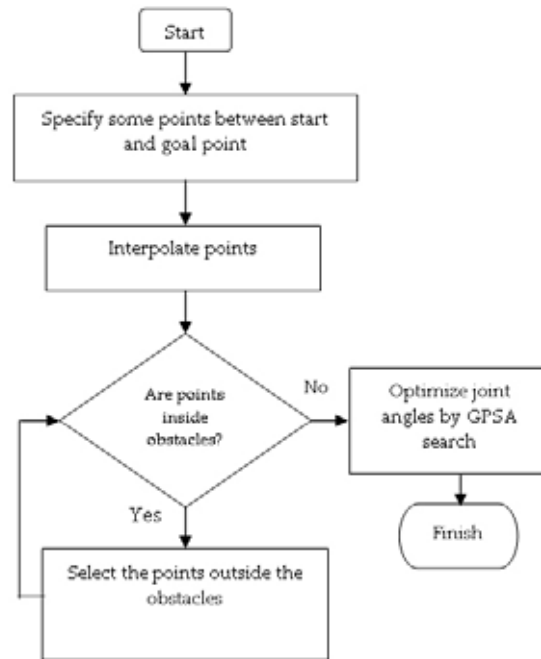


Fig. 8. Flowchart of the proposed obstacle avoidance algorithm

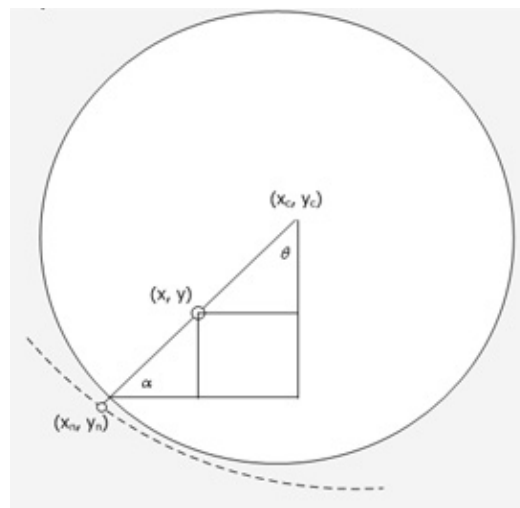


Fig. 9. Workspace with obstacle

-90° . The simulation results for this case such as robot configurations, optimized joint angles and tracking error are illustrated in Fig. 10, 11, and 12 respectively.

The second case includes circular and triangular obstacles. The center of circle is located at $(0.6, 0.5)m$ with its radius equals $0.08m$. The three vertices of triangle are $(0.3, 0.3)m$, $(0.4, 0.2)m$ and $(0.4, 0.4)m$. The starting point is $(0.7, 0.7)m$ and the end point is $(0.3, 0.3)m$. $\theta_1^1 = 60^\circ$ and $\theta_2^1 = -90^\circ$ are initial angles for link 1 and link 2. The simulation results for the second case are illustrated Fig. 13, 14, and 15 respectively.

7 Discussion

It is clear from the illustrated simulation results that the Generalized Pattern Search can be applied effectively to design an optimal trajectory for flexible manipulators. For the point-to-point trajectory tracking, the end-effector of the flexible manipulator follows the desired trajectory with negligible error. This can be seen clearly for the line and circle trajectories from Fig. 4 and 7. On the other hand the optimized joint angles are

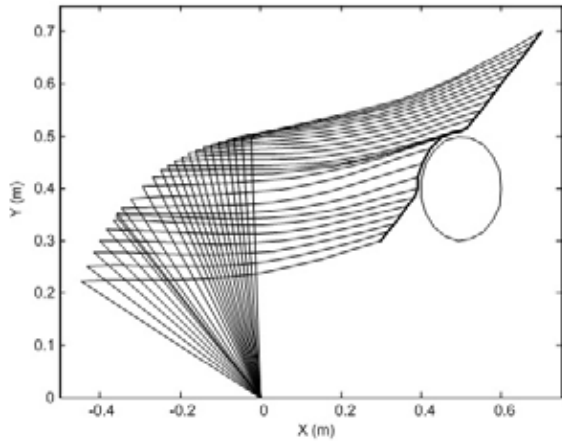


Fig. 10. Robot configuration with one obstacle

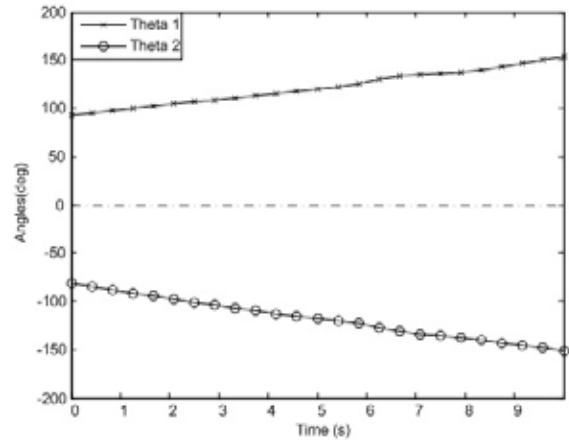


Fig. 11. Optimized joint angles

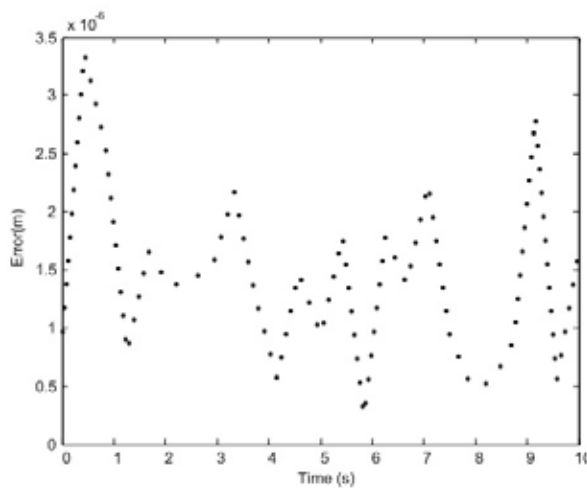


Fig. 12. Tracking error

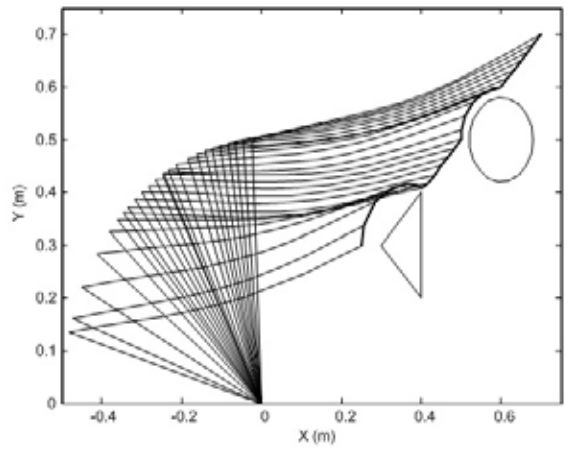


Fig. 13. Robot configuration with two obstacles

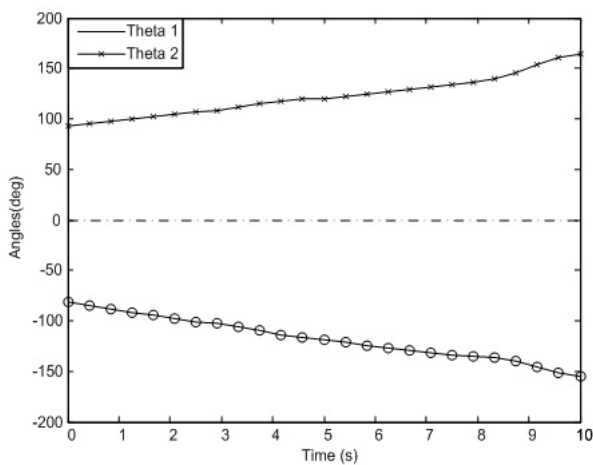


Fig. 14. Optimized joint angles

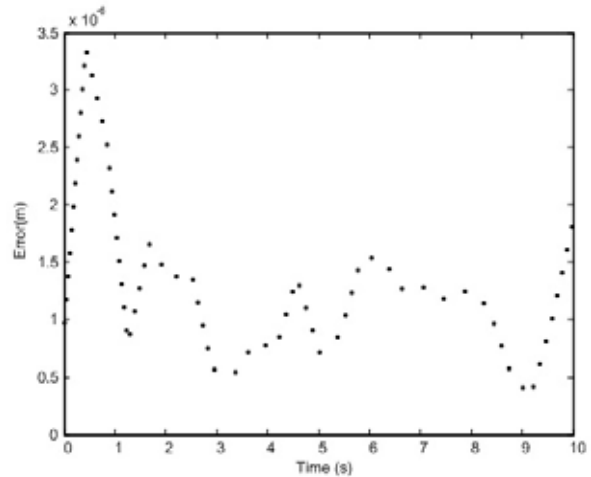


Fig. 15. Tracking error

smooth without fluctuation which adds another advantage to the GPS method. The algorithm is flexible and it can be applied to any other trajectory. As a matter of fact, the execution time for the rigid-flexible manipulator is higher and this is expected since we have to solve for the flexible deflection of the elastic link.

For the obstacle avoidance trajectory, as soon as the coordinates of the obstacle are identified, the algorithm can design the trajectory automatically. Based on the excellent tracking performance of the GPS, the end-effector can track the proposed trajectory without colliding with the obstacle at all. This can be observed clearly from Fig. 10 and 13. This algorithm is very flexible and can be applied to avoid any geometrical shapes without any difficulties. The number of interpolated point affects trajectory and operation time. However, more interpolated points will give smoother trajectory but will take more operation time.

References

- [1] A. Ata, T. Myo. Optimal point-to point trajectory tracking of redundant manipulators using generalized pattern search. *International Journal of Advanced Robotic Systems*, 2005, **2**: 239–244.
- [2] A. Ata, T. Myo. Collision-free trajectory planning for manipulators using generalized pattern search. *International Journal of Simulation Modelling*, 2006, **5**: 145–154.
- [3] D. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. MA: Addison-Wesley, 1989.
- [4] H. Kojima, T. Kibe. Optimal trajectory planning of a two-link flexible robot arm based on genetic algorithm for residual vibration reduction. **in:** *Proceeding of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Maui, Hawaii, USA, 2001, 2276–2281.
- [5] L. Meirovitch. *Fundamentals of Vibrations*. McGraw-Hill, New York, 2001.
- [6] G. Oke, Y. I Stefanopulos. Neural network based trajectory planning for tip tracking of a two-link flexible robot manipulator. **in:** *Proceeding of the 11th Mediterranean Conference on Control and Automation*, Rhodes, Greece, 2003.
- [7] A. Rouvinen, H. Handross. Robot positioning of a flexible hydraulic manipulator utilizing genetic algorithm and neural networks. **in:** *Proceeding of the Fourth Annual Conference on Mechatronics and Machine Vision in Practice*, Toowoomba, Australia, 1997, 182–187.
- [8] M. Shaheed, M. Tokhi, etc. Modeling and open-loop control of a single-link flexible manipulator with genetic algorithms. *Journal of Low Frequency Noise, Vibration and Active Control*, 2001, **20**: 39–55.
- [9] L. Tian, C. Collins. Motion planning for redundant manipulators using a floating point genetic algorithm. *Journal of Intelligent and Robotic Systems*, 2003, **38**: 297–312.
- [10] H. Wahyudi, M. Dissanayake. On trajectory planning for flexible robots. **in:** *Proceeding of the 12th IFAC World Congress*, Sydney, Australia, 1993, 128–134.
- [11] M. Wetter, J. Wright. Comparison of a generalized pattern search and a genetic algorithm optimization method. **in:** *Proceeding of the Eighth International IBPSA Conference*, Eindhoven, Netherlands, 2004, 1401–1408.
- [12] S. Yue, D. Henrich. Point-to-point trajectory planning of flexible redundant robot manipulators using genetic algorithms. *Journal of Robotica*, 2002, **20**: 269–280.