

MODELLING AND OPTIMIZATION APPROACH FOR TRAJECTORY PLANNING OF THREE FREEDOM PLANAR MANIPULATORS

**JIN-CHAO GUO, ZHENG LIU
and GUANG-ZHAO CUI**

Department of Electric and Information Engineering
Zhengzhou University of Light Industry
Zhengzhou 450002
Henan Province
P. R. China
e-mail: gjc@zzuli.edu.cn

Abstract

In this paper, we present a method for the problem of the optimal trajectory planning of redundant robot manipulators in the presence of fixed obstacles. Quadrinomial and quintic polynomials are used to describe the segment of the trajectory. Cultural based PSO (CBPSO) algorithm is proposed to design a collision-free trajectory for planar redundant manipulators. Kinematics redundancy is integrated into the presented method as planning variables. CBPSO optimizes the trajectory and ensures that obstacle avoidance can be achieved. Simulations are carried out for different obstacles to prove the validity of the proposed algorithm. Different test data generated by GA, QPSO, and CBPSO are provided with a tabular comparison. Simulation studies show CBPSO has potential online usage in engineering and distinct fast computation speed compared with the other two algorithms. Results demonstrate the effectiveness and capability of the proposed method in generating optimized collision-free trajectories.

2010 Mathematics Subject Classification: 65K99.

Keywords and phrases: trajectory planning, CBPSO algorithm, redundant robot manipulators.

Received May 8, 2011

© 2012 Scientific Advances Publishers

1. Introduction

Over the past decades, there are many researches on robot manipulators involved mainly modelling, control accuracy, and trajectory tracking. Moreover, the geometric problems with Cartesian paths related to workspace and singularities can be avoided, if the point-to-point task is planned in joint space. Marcos et al. [9] present a new technique to solve the inverse kinematics problem of redundant manipulators, which uses a fractional differential of order α to control the joint positions. Bu et al. [2] present a novel method based on trajectory planning to avoid the detachment of joint elements of a manipulator with clearances. By taking the time interval of each trajectory segment as the adjustable parameter, the dynamic response spectrums of joint forces are obtained, which can be used to facilitate the selection of the adjustable trajectory parameters so as to avoid the detachment between joint elements. The obstacle avoidance problem, where robotic manipulators are required to move from an initial position to a specified final goal without collision with other objects in the workspace becomes very important. Trajectory tracking with obstacle avoidance is rather difficult because of the high non-linearity essence of manipulator dynamics with existence of obstacles.

Evolution algorithm and other heuristic methods are now used widely on many optimization problems. Compared with inverse Jacobian matrix method, the heuristic method is better in design complexity, computation time etc. Kumar et al. [8] propose an online inverse-forward adaptive scheme. A feed-forward network is used to learn the forward kinematic map of the redundant manipulator. The proposed inverse-forward adaptive scheme effectively approximates the forward map around the joint angle vector. In order to satisfy the joint angle constraints, it provides the network inversion algorithm with an initial hint for the joint angle vector. The simulations and experiments are carried out on a 7 DOF PowerCube™ manipulator. Arakelian et al. [1] deal with the analytically tractable solution for input torques minimization of two degrees of freedom serial manipulators based on minimum energy control and optimal redistribution of movable masses. The suggested approach is

illustrated by numerical simulations carried out using ADAMS software. Gasparetto and Zonotto [3] present a novel method for smooth trajectory planning of robot manipulators. An objective function containing a term proportional to the integral of the squared jerk (defined as the derivative of the acceleration) along the trajectory is considered. The proposed method enables one to set kinematic constraints on the robot motion, expressed as upper bounds on the absolute values of velocity, acceleration, and jerk. The algorithm has been tested in simulation yielding good results. Yue et al. [12] studied the problem of point-to-point trajectory planning of flexible redundant robot manipulator in joint space. Quadrinomial and quintic polynomials were used to describe the segment, which connects the initial, intermediate, and final points in joint space. The proposed trajectory to minimize vibration of end-effector is achieved on GA. Zha [13] presents a unified approach to optimal pose trajectory planning for robot manipulators in Cartesian space through a genetic algorithm (GA) enhanced optimization of the pose ruled surface. For a given point-to-point task, the maximum dynamic payload-carrying capacity of a flexible redundant robot manipulator (FRM) is established in the paper. A finite-element model is presented for describing the dynamics of the system considering the added payload and the flexibility of the FRM. This method is useful especially for FRMs performing repeated point-to-point operations [11].

Much research has been done on robot manipulator trajectory control and many evolution and swarm intelligence algorithms have been used to optimize the trajectory, however, from previous analysis, we can see that GA cannot be used in online situations at present, just for its complex crossover and mutation factors. QPSO can only remember the last better performance of manipulator about trajectory. Cultural based particle swarm optimization (CBPSO) algorithm is a novel heuristic optimal algorithm. Compared with GA and QPSO, CBPSO is more effective and less time-consuming. In robot manipulator trajectory planning, it is hoped that the time-consuming question will be resolved with CBPSO.

Having these ideas in mind, the paper is organized as follows: Section 2 introduces the fundamentals of the kinematics and dynamics of redundant manipulators. Based on the acquired concepts, Section 3 presents the

robot arm trajectory planning method with cultural based particle swarm optimization (CBPSO) algorithms and Section 4 presents the simulation and experimental results and also gives analysis subsequently. Finally, Section 5 makes the main conclusion.

2. Robot Arm Trajectory Problem Formulation

Non-redundant robots often encounter problems in practical applications. For example, internal singularities can cause excessive joint speed requirements that can not be realized by the actuators, thus causing errors in a specified trajectory. In order to avoid singular configurations, the robot is often operated in a limited workspace. In point-to-point motion, a non-redundant robot can sometimes avoid obstacles in its workspace, but as the trajectory requirements increase, its ability to avoid obstacles becomes limited.

In order to overcome the shortcomings inherent in non-redundant robots, redundant robots have been utilized in industrial applications to increase flexibility and dexterity around a restricted task space in the presence of obstacles. They can also provide a better ability to avoid singular configurations and the excessive velocities and accelerations encountered at singularities. Most applications of redundant robots are based on a motion trajectory composed of a sequence of spatial displacements of the end-effector of the robot arm, namely, the trajectory is a path for the end-effector. Planning a trajectory requires searching for a continuous motion that takes the robot arm from a given starting configuration, to a specified end position in the task space, without colliding with any obstacles.

2.1. Problem description

A planar flexible robotic manipulator with three flexible links is studied in this work. A 3-link planar robot manipulator configuration is shown in Figure 1. In order to calculate the joints trajectory accurately, a complete description of the manipulator's workspace must be known in advance (e.g., the position of the starting point and the goal, the shape and orientation of the obstacles etc.).

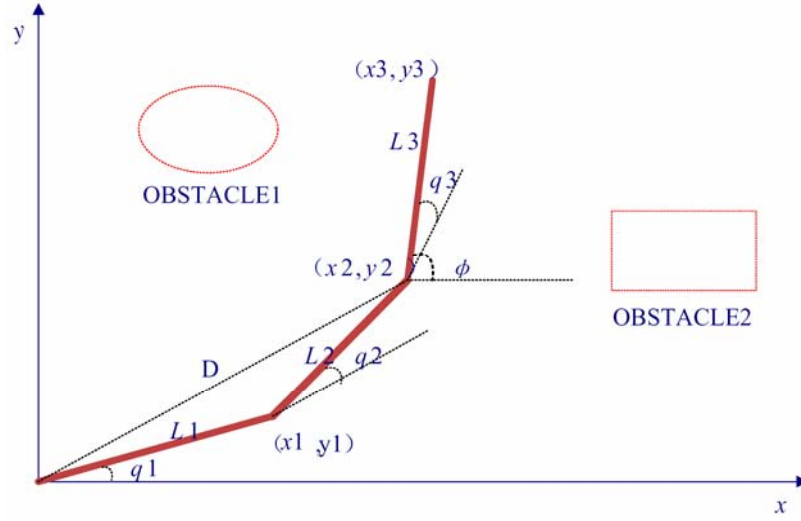


Figure 1. Three DOF planar robot manipulator configuration.

Where L_1 , L_2 , and L_3 are the length of the links; (x, y) is the Cartesian coordinate of the end link; q_1 , q_2 , and q_3 represent the joint angular motions. The shadowed circle and rectangle represent the obstacles existing in the workspace of the robotic manipulator.

Based on the geometric characters, trigonometry can be used to obtain coordinate equations between the links

$$x_1 = L_1 \cos(q_1), \quad (1)$$

$$y_1 = L_1 \sin(q_1), \quad (2)$$

$$x_2 = x_1 + L_2 \cos(q_1 + q_2), \quad (3)$$

$$y_2 = y_1 + L_2 \sin(q_1 + q_2), \quad (4)$$

$$x_3 = x_2 + L_3 \cos(q_1 + q_2 + q_3), \quad (5)$$

$$y_3 = y_2 + L_3 \sin(q_1 + q_2 + q_3). \quad (6)$$

During initialization, we take the right-arm system ($q_2 > 0$). If an initial condition is given, such as the Cartesian coordinate and the

azimuth of the starting point, we can get all the geometric information about other links using the inverse process of the above equations. The presented method will conduct the planning problem for the FRM in joint space. This means that various problems related to workspace and singularities encountered in Cartesian space can be easily avoided. There are only two important points (i.e., the initial and final point of a possible path) that should be achieved in point-to-point trajectory planning. Corresponding to the two points, there are infinite possible configurations due to kinematics redundancy. This means the end-effector can start from different poses and end with different poses for a given task. Thus, the redundancy problem with respect to initial and final points is the determination of the initial and final postures. For a robot, the number of degrees of freedom of a manipulator is n and the number of end-effectors degree of freedom is m . These two postures can be described by $2(n - m)$ parameters in joint space, where $(n - m)$ are the redundant degree of freedom of the robot manipulator.

In our research, we assume the initial posture is known. The postures of final point and medium via point i will be determined by using CBPSO. The problems of redundancy corresponding to intermediate via points will be solved efficiently by planning in joint space instead of in Cartesian task space as well, as described in the following subsection.

2.2. Point-to-point trajectory representation

According to point-to-point trajectory method, a trajectory consists of several segments with continuous acceleration at the intermediate via point, as shown in Figure 2. The position of each intermediate point is supposed to be optimized according to the environment. This is useful especially when there is an obstacle in the working area. Of course, the intermediate points can also be given as particular points that should be passed through.

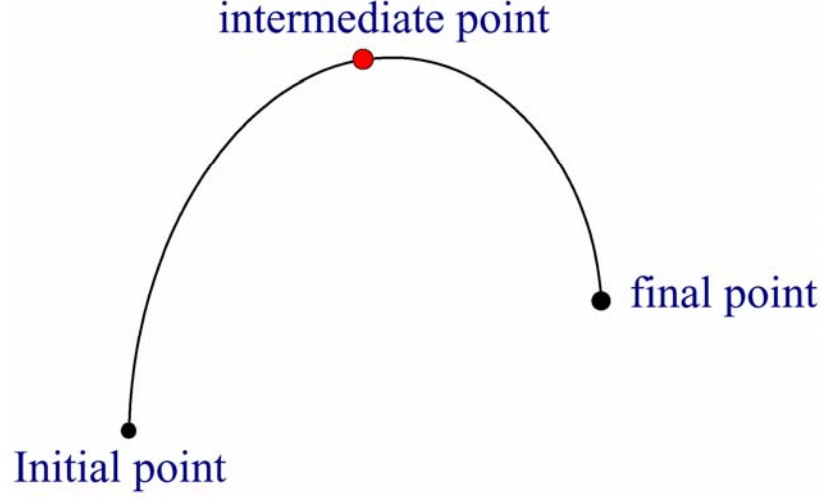


Figure 2. Intermediate points on the point-to-point path.

Here, we adopt the method proposed by Yue et al. A quadrinomial and a quintic polynomial are used to model the segments of the trajectory [12]. Let us assume there are m_p intermediate via points between the initial and the final points. Between the initial points to m_p intermediate via points, a quadrinomial is used to describe these segments as [12]:

$$p_{i,i+1} = a_{i0} + a_{i1}t_i + a_{i2}t_i^2 + a_{i3}t_i^3 + a_{i4}t_i^4, \quad (i = 0, 1, \dots, m_p - 1), \quad (7)$$

where a_{i0} to a_{i4} are constant coefficients and must be conformed to the following constraints:

$$p_i = a_{i0}, \quad (8)$$

$$p_{i+1} = a_{i0} + a_{i1}T_i + a_{i2}T_i^2 + a_{i3}T_i^3 + a_{i4}T_i^4, \quad (9)$$

$$\bullet \\ p_i = a_{i1}, \quad (10)$$

$$\bullet \\ p_{i+1} = a_{i1} + 2a_{i2}T_i + 3a_{i3}T_i^2 + 4a_{i4}T_i^3, \quad (11)$$

$$\bullet\bullet \\ p_i = 2a_{i2}, \quad (12)$$

where T_i is the execution time from point i to the neighbour intermediate point. The five unknown constant coefficients can be solved as

$$a_{i0} = p_i, \quad (13)$$

$$\overset{\bullet}{a}_{i1} = \overset{\bullet}{p}_i, \quad (14)$$

$$\overset{\bullet\bullet}{a}_{i2} = \overset{\bullet\bullet}{p}_i / 2, \quad (15)$$

$$a_{i3} = (4\overset{\bullet}{p}_{i+1} - \overset{\bullet}{p}_{i+1} T_i - 4p_i - 3\overset{\bullet\bullet}{p}_i T_i^2) / T_i^3, \quad (16)$$

$$a_{i4} = (\overset{\bullet}{p}_{i+1} T_i - 3\overset{\bullet}{p}_{i+1} + 3p_i + 2\overset{\bullet}{p}_i T_i + \overset{\bullet\bullet}{p}_i T_i^2 / 2) / T_i^4. \quad (17)$$

The intermediate point $(i + 1)$'s acceleration can be obtained as

$$\overset{\bullet\bullet}{p}_i = 2a_{i2} + 6a_{i3}T_i + 12a_{i4}T_i^2. \quad (18)$$

The segment between the intermediate and the final points can be represented as a quintic polynomial

$$p_{i,i+1} = b_{i0} + b_{i1}t_i + b_{i2}t_i^2 + b_{i3}t_i^3 + b_{i4}t_i^4, \quad (i = m_p). \quad (19)$$

Just like a_{i0} to a_{i4} ; b_{i0} to b_{i5} can be calculated by the same method

$$b_{i0} = p_i, \quad (20)$$

$$\overset{\bullet}{b}_{i1} = \overset{\bullet}{p}_i, \quad (21)$$

$$\overset{\bullet\bullet}{b}_{i2} = \overset{\bullet\bullet}{p}_i / 2, \quad (22)$$

$$b_{i3} = (20\overset{\bullet}{p}_{i+1} - (8\overset{\bullet}{p}_{i+1} + 12\overset{\bullet}{p}_i)T_i - 20p_i - (3\overset{\bullet\bullet}{p}_i - \overset{\bullet\bullet}{p}_{i+1})T_i^2) / 2T_i^3, \quad (23)$$

$$b_{i4} = (\overset{\bullet}{p}_{i+1} T_i - 3\overset{\bullet}{p}_{i+1} + 3p_i + 2\overset{\bullet}{p}_i T_i + \overset{\bullet\bullet}{p}_i T_i^2 / 2) / T_i^4, \quad (24)$$

$$b_{i5} = (12\overset{\bullet}{p}_{i+1} - (6\overset{\bullet}{p}_{i+1} + 6\overset{\bullet}{p}_i)T_i - 12p_i - (\overset{\bullet\bullet}{p}_i - \overset{\bullet\bullet}{p}_{i+1})T_i^2) / 2T_i^5. \quad (25)$$

As formulated above, the parameters to be determined are the joint angles of each intermediate via point ($n \times m_p$ parameters), the joint angular velocities of each intermediate point ($n \times m_p$ parameters), the execution time for each segment ($m_p + 1$ parameters), and the posture of the final configuration ($n - m$).

3. CBPSO Algorithm for Robot Arm Trajectory Planning

3.1. PSO algorithm

Particle swarm optimization (PSO), introduced by Kennedy and Eberhart [7] in 1995, has been inspired from animals' social behaviours, which are illustrated by their social actions resulting in population survival.

PSO is a self-adaptive population based method in which, behaviour of the swarm is iteratively generated from the combination of social and cognitive behaviours [10]. A swarm can be imagined as consisting of members called *particles*. Particles cooperate with each other to achieve desired behaviours or goals. In PSO algorithms, each particle keeps track of its own position and velocity in the problem space. The initial position and velocity of a particle are generated randomly. At each iteration, the new positions and velocities of the particles are updated by using the following two equations:

$$\vec{v}_i^{t+1} = \omega \cdot \vec{v}_i^t + c_1 r_1 [\vec{p}_i^t - \vec{x}_i^t] + c_2 r_2 [\vec{p}_g^t - \vec{x}_i^t], \quad (26)$$

and

$$\vec{x}_i^{t+1} = \vec{x}_i^t + \vec{v}_i^{t+1}, \quad (27)$$

where ω is a linearly varying inertia weight over the iterations; t is the number of the current iteration; c_1 and c_2 are acceleration coefficients; and r_1 and r_2 are random numbers between 0 and 1. \vec{x}_i^t , \vec{p}_i^t , and \vec{v}_i^t are the position, the local best, and the velocity of the i -th particle at iteration t , respectively; and \vec{p}_g^t is the global best of all particles.

3.2. Cultural algorithm

Cultural algorithm was proposed by Reynolds, which are derived from human culture evolution process. In cultural algorithms, dual evolution structure (Jin and Reynolds, 1999), consisting of population space and belief space, is adopted. In population space, PSO algorithm is adopted and operators aiming at a set of possible solutions to the problem are realized. In belief space, implicit knowledge is extracted from better individuals in population and stored in a way. Then, they are utilized to direct the evolution process in population space so as to induce population escaping from the local optimal solutions. And the algorithms also provide a model for extraction and utilization of the evolution information. Its basic framework as shown in Figure 3.

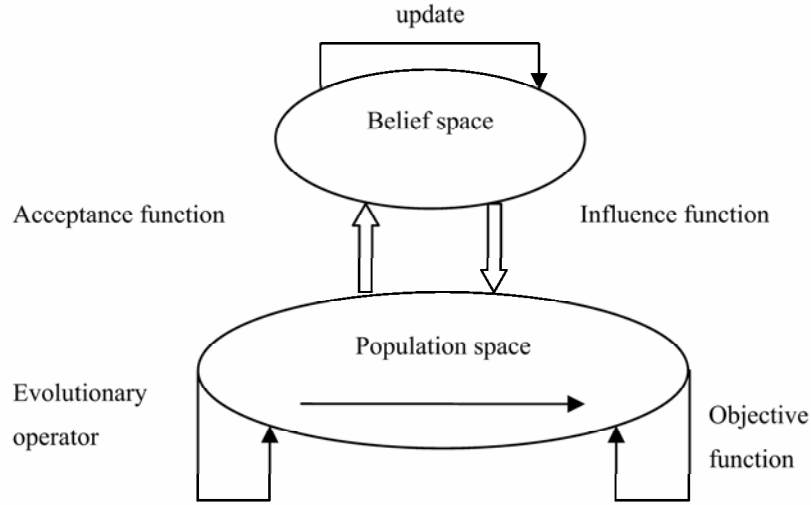


Figure 3. Cultural algorithm framework.

At each iteration, we choose particles with better performance in the population space as parents and send these parents particles to the belief space. Then, the parent particles randomly cross in belief space and the hybrid new child particles having better performance will replace the same number of parent particles in the population space. The update function is as following:

$$child(x) = p \cdot parent_1(x) + (1 - p) \cdot parent_2(x). \quad (28)$$

3.3. The cost function

Considering the calculation expense, only one intermediate via point is selected to construct the trajectory. For a planar robot manipulator with three links, there are nine parameters should be optimized, which means the solution space is nine dimensions.

The cost function is an important component of CBPSO, which is the so-called object function. In our work, the cost function consists of four indices. All indices are translated into penalty functions to be minimized. Each index is computed individually and is integrated in the cost function evaluation. The fitness function f_f adopted for evaluating the candidate trajectories in free workspace is defined as

$$f_f = \beta_1 f_{ot} + \beta_2 f_{qd} + \beta_3 f_{cd} + \beta_4 t_T, \quad (29)$$

where $\beta_i (i = 1, \dots, 4)$ are the weighting factors and they are subjected to the following relationship:

$$\sum_{i=1}^4 \beta_i = 1. \quad (30)$$

The f_{ot} index represents the amount of excessive driving, when the maximum torque $\tau_{i, \max}$ is matched, a penalty is add to the item. The f_{ot} index can be calculated according to the following equations [5]:

$$f_i^j = \begin{cases} 0, & |\tau_i^j| < \tau_{i, \max}, \\ \left| |\tau_i^j| - \tau_{i, \max} \right|, & \text{otherwise,} \end{cases} \quad (31)$$

and

$$f_{ot} = \sum_{j=1}^b \sum_{i=1}^a f_i^j, \quad (32)$$

where a is number of robot links and b is number of joint positions from the initial to final configuration.

The index f_{qd} represents the total joint travelling distance of the manipulator as criteria

$$f_{qd} = \sum_{i=1}^a \sum_{j=2}^b |q_{i,j} - q_{i,j-1}|. \quad (33)$$

The index f_{cd} represents total Cartesian trajectory length

$$f_{cd} = \sum_{j=2}^b d(p_j - p_{j-1}), \quad (34)$$

where p_j is Cartesian coordinate of the j -th specified intermediate point on the planning trajectory and $d()$ is a function that calculates the distance between points.

The index t_T represents the total time required for robot manipulator motion

$$t_T = t_1 + t_2, \quad (35)$$

where t_1 is the execution time from start to intermediate point and t_2 is the time from intermediate to final target.

Avoidance objective function f_{ob} has been combined with free space fitness function f_f to form overall fitness function f , as shown below:

$$f = f_f / f_{ob}. \quad (36)$$

Combined with f_{ob} , the robot manipulator has the ability to avoid the obstacle collision during its movement from point-to-point inside the workspace. f_{ob} can be written as Equation (37).

$$f_{ob} = \begin{cases} 1, & \sum_{i=1}^a \sum_{j=1}^b (link_{ij} \cap obstacle) = 0, \\ 0, & \text{otherwise.} \end{cases} \quad (37)$$

4. Experimental Simulation and Analysis

This section presents the results of robot case study. For a 3-link robot case, $m_p = 1$, $n = 3$, and one degree of freedom of redundancy for the final point are selected. In robot manipulator parameter configuration, all data are kept the same as [6].

In the following case studies, one and more than one obstacles inside the workspace are considered, respectively. The difference between GA [4], QPSO [5], and CBPSO is also provided with a tabular comparison.

When one and more than one obstacles exists in the workspace, the optimized results are shown in, Figures 4-7. Figures show the shorter Cartesian path of the end-effector when obstacles and other limitation conditions are taken into account.

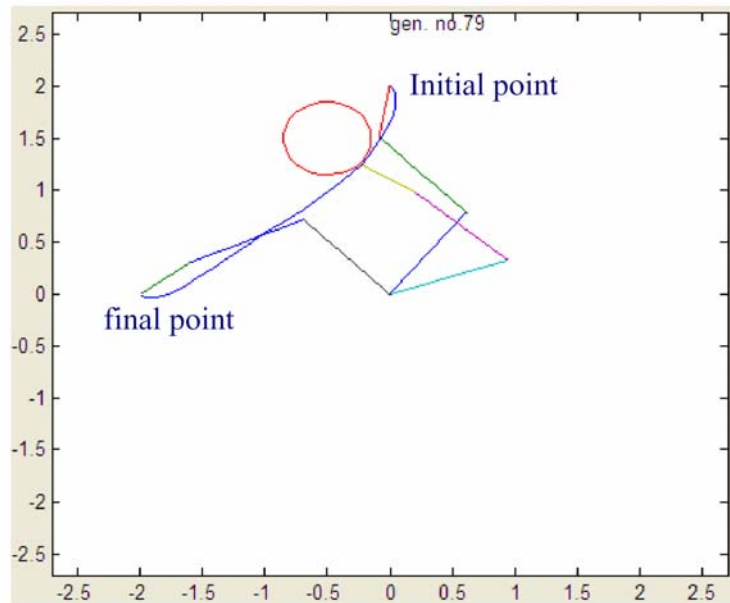


Figure 4. Cartesian path of the end-effector with one obstacle in workspace.

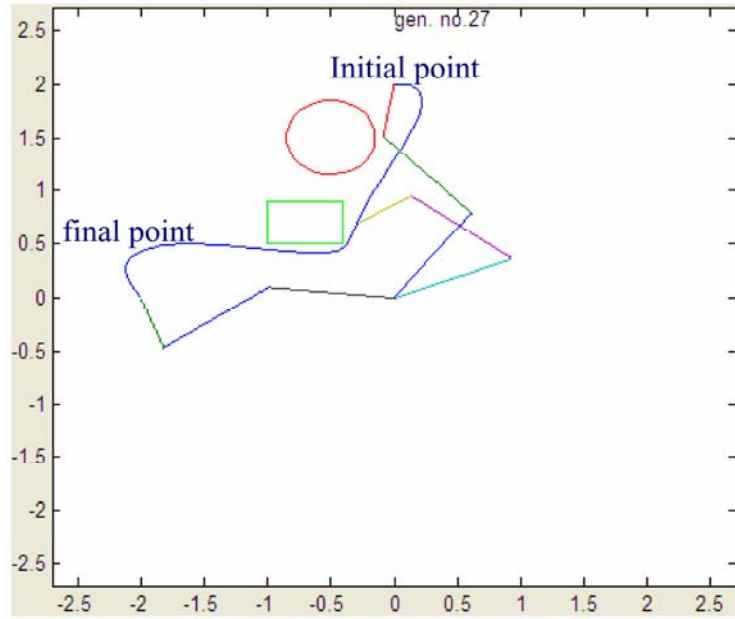


Figure 5. Cartesian path of the end-effector with two obstacles in workspace.

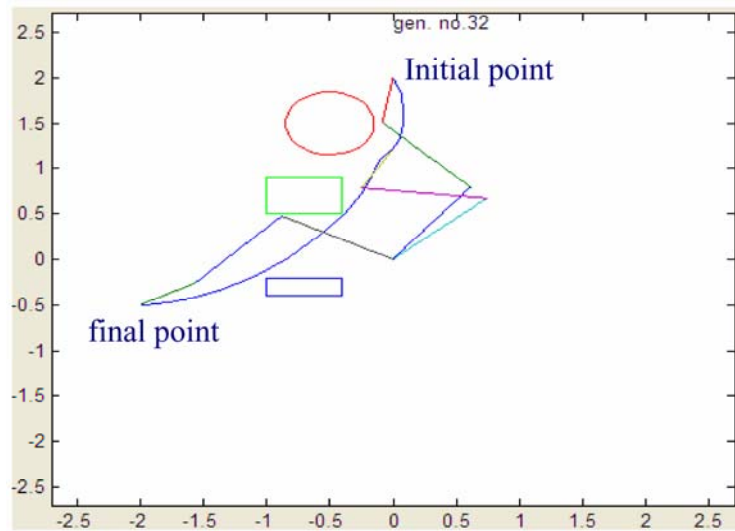


Figure 6. Cartesian path of the end-effector with three obstacles in workspace.

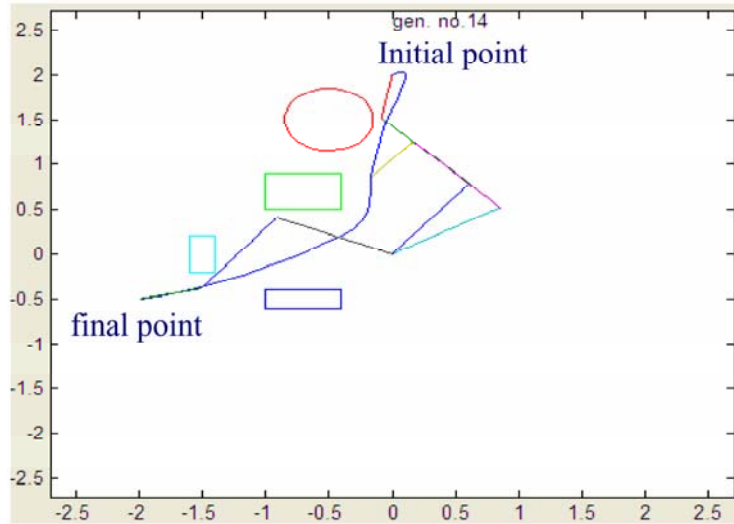


Figure 7. Cartesian path of the end-effector with four obstacles in workspace.

Table 1 shows test data of computation time (s) generated by GA, QPSO, and CBPSO, respectively, corresponding to the different number of obstacles existence workspace.

Table 1. The comparison of different algorithms in computing time

	One obstacle workspace	Two obstacles workspace	Three obstacles workspace	Four obstacles workspace
GA algorithm	70.55	80.17	none	None
QPSO algorithm	48.42	55.46	none	None
CBPSO algorithm	43.15	52.84	61.91	64.65

*Here, the computation time is the average value of 20 turns.

As noted from Table 1, compared with GA and QPSO, CBPSO has a dramatic characteristic of fast convergence ability with almost the same computation precision.

Figures 4-7 illustrate CBPSO has better performance that the manipulator can arrive the final point among different number of obstacles. The above figures illustrate that the robot manipulator can avoid most types of obstacle in setting space. Compared to other methods, the proposed planning model is more generic, and can be used to reduce the computing cost and storage in robot planning.

Figures 8-11 illustrate that CBPSO apply in the industrial robot manipulator that has four degree of freedom from Googol Technology Limited Company (as Figure 8). We use Visual C++ as the program tool and debug the industrial robot manipulator in order to evaluate the effect of the CBPSO in plane coordinates. We find that the result of the experiment is essentially fulfilled, the industrial robot manipulator avoid the obstacle. The trajectory of manipulator is quite smooth.



Figure 8. Robot manipulator with three freedoms used in experiment.



Figure 9. Initial pose of robot manipulator.



Figure 10. Intermediate pose of robot manipulator.



Figure 11. Final pose of robot manipulator.

Figures 8-11 represent three pose of robot manipulator and illustrate that the robot manipulator can avoid the cylindrical obstacle under the condition of using CBPSO.

5. Conclusion

The problem of obtain the point-to-point trajectory of robot manipulator in the case, where fixed obstacles are studied in the workspace. The model and algorithm as demonstrated in this paper are generally reliable and effective, if control parameters are appropriately selected. Optimization methods have been successfully implemented for robot manipulators. Trajectory planning method based on CBPSO to minimize vibration and time expense considering other limitations are presented. Kinematics redundancy is considered as a planning variable in the presented method. Quadrinomial and quintic polynomials are used to describe the segments that connect the initial, intermediate, and final points in joint space. Various problems related to workspace and singularities in Cartesian space are avoided by planning in joint space. Case studies show CBPSO has good computation precision and distinct fast computation speed compared with GA and QPSO.

Acknowledgement

The authors would like to thank Prof. Guangzhao Cui and Doc. Yanfeng Wang, who work in Key Lab. of Information-based Electrical Appliances of Henan Province. Thanks for their kindness and huge help in instrument provision. Many experiments are executed in the key lab. This work is supported by the Natural Science Foundation of Henan Province of China under grant (No. 092300410036).

References

- [1] V. Arakelian, J.-P. L. Baron and P. Mottu, Torque minimization of the 2-DOF serial manipulators based on minimum energy consideration and optimum mass redistribution, *Mechatronics* 21 (2011), 310-314.
- [2] W. H. Bu, Z. Y. Liu, J. R. Tan and S. M. Gao, Detachment avoidance of joint elements of a robotic manipulator with clearances based on trajectory planning, *Mechanism and Machine Theory* 45 (2010), 925-940.
- [3] A. Gasparetto and V. Zanutto, A new method for smooth trajectory planning of robot manipulators, *Mechanism and Machine Theory* 42 (2007), 455-471.
- [4] D. E. Goldberg, *Genetic Algorithm in Search, Optimization, and Machine Learning*, Addison-Wiley, 1989.
- [5] J. C. Guo, X. J. Wang and X. W. Zheng, Trajectory planning of redundant robot manipulators using QPSO algorithm, *Proceedings of the 8th World Congress on Intelligent Control and Automation* (2010), 6-9.
- [6] B. I. Kazem, A. I. Mahdi and A. T. Qudah, Motion planning for a robot arm by using genetic algorithm, *JJMIE* 2 (2008), 131-136.
- [7] J. Kennedy and R. C. Eberhart, Particle swarm optimization, *Proceedings of the 1995 IEEE International Conference on Neural Networks*, IEEE, Piscataway (1995), 1942-1948.
- [8] S. Kumar, L. Behera and T. M. McGinnity, Kinematic control of a redundant manipulator using an inverse-forward adaptive scheme with a KSOM based hint generator, *Robotics and Autonomous Systems* 58 (2010), 622-633.
- [9] M. D. Graça Marcos, J. A. Tenreiro Machado and T.-P. Azevedo-Perdicoulis, A fractional approach for the motion planning of redundant and hyper-redundant manipulators, *Signal Processing* 91 (2011), 562-570.
- [10] J. Vesterstrom and J. Riget, *Particle Swarms: Extensions for Improved Local, Multi-Modal and Dynamic Search in Numerical Optimization*, MSC Thesis, Dept. Computer Science, Univ. Aarhus, Aarhus C, Denmark, 2002.

- [11] S. G. Yue, S. K. Tso and W. L. Xu, Maximum-dynamic-payload trajectory for flexible robot manipulators with kinematic redundancy, *Mechanism and Machine Theory* 36 (2001), 785-800.
- [12] S. G. Yue, D. Henrich, X. L. Xu and S. K. Toss, Point-to-point trajectory planning of flexible redundant robot manipulators using genetic algorithms, *Journal of Robotica* 20 (2002), 269-280.
- [13] X. F. Zha, Optimal pose trajectory planning for robot manipulators, *Mechanism and Machine Theory* 37 (2002), 1063-1086.

