

---

# Path Planning Optimization for Teaching and Playback Welding Robot

Yuehai Wang\*, Ning Chi

College of Information Engineering, North China University of Technology Jinyuanzhang Rd, Shijingshan  
Dist. Beijing, 8610-88801652

Corresponding author, e-mail: wangyuehai@ncut.edu.cn\*, chining@ncut.edu.cn

## Abstract

*Path planning for the industrial robot plays an important role in the intelligent control of robot. Tradition strategies, including model-based methods and human taught based methods, find it is difficult to control manipulator intelligently and optically. Thus, it is hard to ensure the better performance and lower energy consumption even if the same welding task was executed repeatedly. A path planning optimization method was proposed to add learning ability to teaching and playback welding robot. The optimization was divided into the welding points sequence improvement and trajectory improvement, which was done both on-line and off-line. Points sequence optimization was modeled as TSP and was continuously improved by genetic algorithm based strategy, while the trajectory between two welding points was on-line improved by an try-and-error strategy where the robot try different trajectory from time to time so as to search a better plan. Simulation results verified that this control strategy reduced the time and energy cost as compared with the man-made fix-order sequence. Our method prevents the robot from the computation-intensive model-based control, and offers a convenient way for self-improvement on the basis of human teaching.*

**Keywords:** Path planning optimization, Teaching and playback robot, Genetic Algorithm (GA), TSP

**Copyright © 2013 Universitas Ahmad Dahlan. All rights reserved.**

## 1. Introduction

Path planning for mobile robot or robot manipulator will construct a safe and effective moving path in its working space to complete the given mission. Generally speaking, there will be several paths and the robot must select one according to certain criteria, such as shortest path, minimum energy cost, or minimum time cost. Finding minimum-time planning strategies for robot manipulators, given actuator constraints, has been a long-standing concern in the robotics literature. This interest is largely motivated by the obvious relationship between execution time of specific tasks and productivity. In this sense, the path planning is a constrained optimization problem which becomes very hard to solve due to the highly nonlinear multi-input dynamics and strong mechanical coupling between the robot's joints.

Since path planning for the industrial robot plays an important role at various application of mobile robot, welding robot and multiple robot system, it was intensively researched during the past several decades. In the early years, various model-based control methods were proposed to provide near-minimum time solutions [1, 2] or true minimum time solutions [3, 4]. These control techniques include linear control, optimal control, adaptive nonlinear control, sliding mode control, feedback control, inverse dynamics method, singular perturbation approach, direct strain feedback control, quasi-tracking approach, nonlinear vibration feedback control [5], and linear quadratic regulator control [6], etc. In the recent several years, the adaptive techniques, such as energy-based robust control [5], adaptive variable structure controller [7], and self-tuning adaptive control schemes [8] were proposed on the basis of a truncated model, ARMA model, or linear state space models. This intelligent control approach is to seek a more efficient, cost-less, and even the optimal trajectory planning for a welding robot. These methods, however, need a large amount of calculation, and also need various assumptions or simplifications on the manipulator dynamics to get a robot models.

On the other hand, teaching and playback robot can accomplish the given task after taught by human, so it can avoid the difficulties in obtaining an abstract model and model based control strategy. Therefore, a large amount of industrial robots, including most welding robots, adopt teaching and playback way for controlling [9]. In this way, the whole working procedure

will follow the teaching method without any change. This artificial sequence is usually, however, neither optimal nor economical, and it is also impossible to be improved by the robot itself even if the given task was executed repeatedly. Therefore, the intelligent approach that enables the robot to seek better joints sequence and improves the performance automatically and on-line through repeatedly executing a same task is urgently need.

In this paper, a path planning optimization method was proposed to augment the machine learning capability to the host computer of a teaching and playback welding robot that is used to weld various chips or integrated circuit boards by welding from a starting point to the ending point. The controller was improved in two steps: firstly the better welding points sequence was continuously constructed by genetic algorithm (GA) based on a TSP model, then the trajectory between two welding points was on-line improved by an try-and-error strategy where the robot try different trajectory from time to time so as to search a better path. For the fixed sequence and low efficiency problem of the traditional teaching playback robot, this paper gives a controller that can find a better and intelligent trajectory by self-study while free from the drawbacks resulting from model uncertainties and model truncations.

## 2. Research Method

### 2.1. Welding Procedure and Ways of Optimization

In a real welding engineering task, the robot must make a detail plan when the parts, operation, resource, the number and position of welding points, etc are taken into account. The robot manipulator will start from the base point with initial pose, move to every next welding point, weld the part fixed in the clamps, and move back to the original point after welding the entire points. Since the limitation of operation and resources, the performance improvement is mainly focus on the sequence among different welding points and the trajectory between welding point to point (PTP). PTP means only the destination pose, but not the trajectory itself, was considered while planing for the path. It is reasonable to make a obstacle-free assumption for the integrated circuit welding task.

For the circuit contains hundreds, and even several hundred, welding points, the intuitive teach sequence from human operator is impossible at all to be optimal. And the model-based control is also infeasible due to its complexity and computation load. To improve the performance of the robot controller but not to impose extra work to human operator, we augment the controller with machine-learning ability which enable the robot find better control program during the same task was repeatedly executed.

### 2.2. Idea and Procedure of Optimization

The optimization was divided into two parts: the welding points sequence improvement and trajectory improvement, which was done alternatively and both on-line and off-line. Points sequence optimization was modeled as TSP and was continuously improved by GA-based strategy, while the trajectory between two welding points was on-line improved by an try-and-error strategy where the robot try different trajectory from time to time so as to search a better one. This idea can be shown in Figure 1.

When a new mission begins, the controller construct the graph coresponding to the human taught procedure and then optimized it off-line by the GA approach which will be described detail in the next part. In the welding process, the robot will use the planned point as the next welding point by certain probability ( $1-\alpha$  in Figure 1), or randomly choose another not-welded point as the next welding point in another probability ( $\alpha$  in Figure 1). These procedure will probably find new better path. After the next target point was determined by either way, the robot will move its manipulator to weld. So it has to solve the inverse kinematics problem to obtain the parameters to drive the servo moters attached in every joints. For the initial path, these parameters was taught by human operator. But for the newly found path, the robot had to calculate by itself. And more often than not, there will be several groups of parameters that move the manipulator to the destination pose. In order to find a better trajectory, the robot will also follow the exist route to the target point by certain probability ( $1-\beta$  in Figure 1), while try another route by another probability ( $\beta$  in Figure 1). In both case, whenever the new path or better route is found, the graph information would be updated. These process will repeated until all the points were welded and task was finished. Then, the GA based TSP optimization

algorithm would run one more time to get the new best point sequence so that the robot can used the newest route whenever the same welding task repeated.

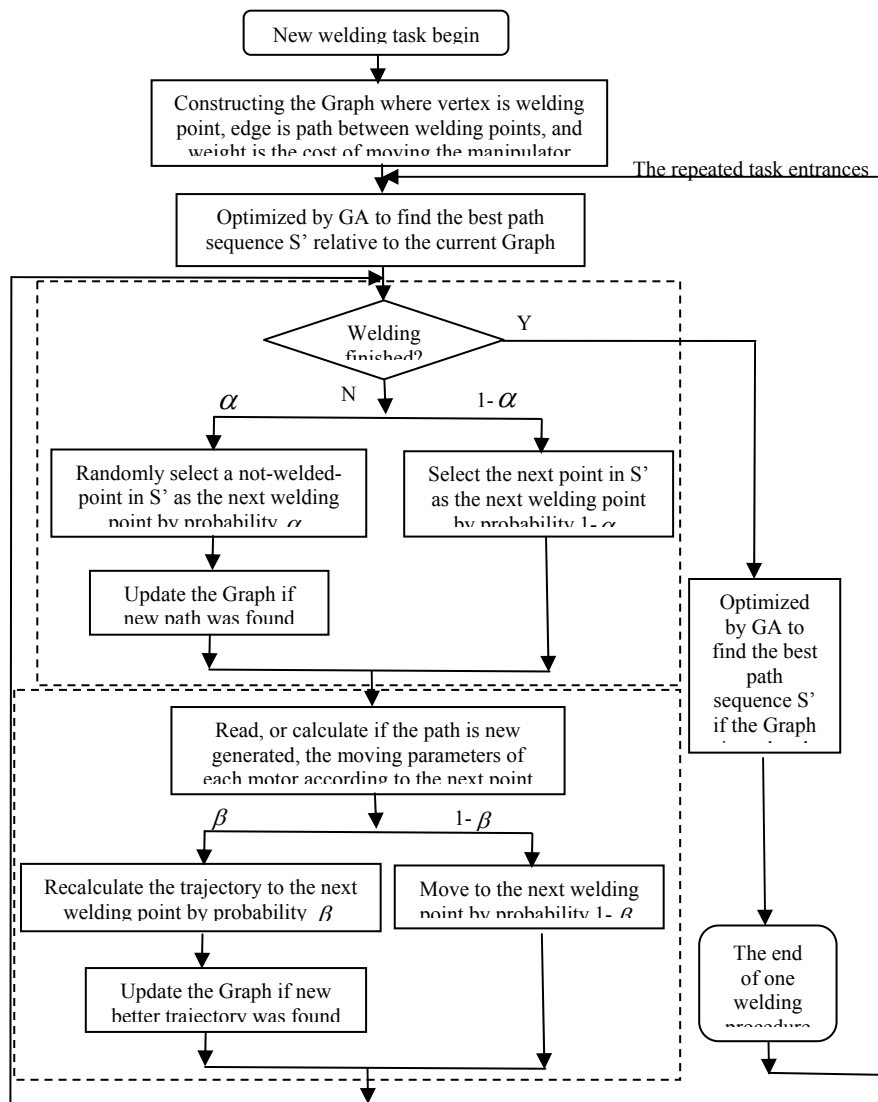


Figure 1. Ideas and Procedure About the Performance Optimization for the Teaching and Playback Welding Robot

### 2.3. Optimization to Welding Point Sequence

#### 2.3.1. Description of the Problem

In the welding task, the manipulator visits every point to weld and then back to its origin point for the next mission. It can be modeled as tourist salesman problem (TSP) where the manipulator is the salesman who was required to visit every welding point only once and back to the first point with minimum path cost or time.

Let we number the welding point as  $1, 2, \dots, n$ , and denote this welding TSP problem as following graph  $G$ :

$$G = (V, A, C) \tag{1}$$

where the vertex set  $V$  denote the points that to be welded, and  $V = \{1, 2, \dots, n\}$ . The initial value of  $V$  can be calculated from the teaching program. The arc set  $A$  denotes all the

possible paths for the manipulator. Again, the initial value of  $A$  can be extracted from the teaching program. Generally speaking, there probably exists path during every two welding points since there is no constraint for the order of welding. And then, the corresponding graph  $G$  will be complete undirected graph. But these paths were added to  $A$  step by step in the random searching process. The cost matrix  $C$  denotes the cost (such as time and energy) for the robot to pass the arc  $(i, j)$ , and  $C = \{c_{ij} \mid (i, j) \in A\}$ .

Define the decision variable as:

$$x_{ij} = \begin{cases} 1, & (i, j) \in \{welding\_path\_set\} \\ 0, & otherwise \end{cases} \quad (2)$$

Then the TSP problem is to find the minimum Hamilton route in the weighted graph  $G$ , as shown in (3):

$$\begin{aligned} MinZ &= \sum_i \sum_j c_{ij} x_{ij} \\ \sum_{i=0}^n x_{ij} &= 1 & j = 0, 1, \dots, n \\ \sum_{j=0}^n x_{ij} &= 1 & i = 0, 1, \dots, n \\ \sum_{i, j \in S} x_{ij} &< |S| - 1 & S \subseteq V, 2 \leq |S| \leq n - 2 \\ x_{ij} &\in \{0, 1\} & i, j \in V, i \neq j \end{aligned} \quad (3)$$

where  $|S|$  is the number of vertex in graph  $G$ , and the second and the third in (3) mean there is only one input arc and only one output arc for every vertex in  $S$ , while the fourth and fifth in (3) ensure there is no sub-loop route to be generated. That is to say, any subset of the welding points must not isolate from other subsets [10].

Although it is NP hard problem and quite difficult to find an optimal solution, it is still possible for the robot to ensure that better path plan on the basis of human taught and not worse than the previous one.

### 2.3.2. The GA-based Optimization Strategy

GAs are adaptive heuristic search algorithm premised on the evolutionary ideas of natural selection and genetic. The basic concept of GAs is designed to simulate processes in natural system necessary for evolution, specifically those that follow the principles first laid down by Charles Darwin of survival of the fittest. As such they represent an intelligent exploitation of a random search within a defined search space to solve a problem.

First pioneered by John Holland in the 60s, GAs has been widely studied, experimented and applied in many fields in engineering worlds. Not only does GAs provide an alternative method to solving problem, it consistently outperforms other traditional methods in most of the problems link. Many of the real world problems involved finding optimal parameters, which might prove difficult for traditional methods but ideal for GAs [11].

GA was employed to optimize the construction of point sequence according to the situation about welding mission and the development of several intelligent algorithms to solve TSP problem. It was done off-line first to construct a best sequence for the current  $G$ , and then this sequence was amended step by step on line in the repeating execution of the same task.

As described in section 2.2, GA is mainly used off line to find a best point route before executing the welding task. And in the process of welding, the robot will either follow the planned point in the best sequence, or randomly select other unfinished point as the next welding point, both of which by some probability. In case of new path was found, the graph  $G$  will be updated

incrementally. After the welding mission finish, GA will search a latest best route for the possible next same task.

**2.3.3. Genetic Algorithm Description**

Typical GA requires candidate encoding, initial population generation, fitness function designing, genetic operators designing and controlling parameters setting.

The most nature coding schema is permutation encoding where every chromosome is a string of numbers, which represents number in a sequence. E.g. the chromosome for a 9 points welding path planning looks like Figure 2 where chromosome A represents the robot will weld point 1→5→3→2→6→4→7→9→8. It should be noted that some crossover and mutation corrections must be made to leave the chromosome consistent (i.e. have real sequence in it).

Chromosome A	1	5	3	2	6	4	7	9	8
Chromosome B	8	5	6	7	2	3	1	4	9

Figure 2. Example of Chromosome with Permutation Encoding

The fitness function is defined over the genetic representation and measures the quality of the represented solution. The fitness function is always problem dependent. In welding robot here, the optimization index is energy cost or task time, as shown in (4):

$$T = d(P_0, P_{v_i}) + \sum_{k=1}^{n-1} d(P_{v_k}, P_{v_{k+1}}) + d(P_{v_i}, P_{v_0}) \tag{4}$$

where  $P_{v_i}$  is  $i^{th}$  welding point is  $v^{th}$  permutation and  $v = 0, 1, \dots, C_n^n$ ,  $d(P_i, P_j)$  is the best cost between point  $P_i$  and  $P_j$ ,  $i, j = 0, 1, \dots, n$ , including the starting point and the ending point. The fitness function is then defined in (5):

$$f = 1/T \tag{5}$$

**2.4. Robot Kinematics for Point to Point Movement**

Once the target point was determined, the robot should move its manipulator to the desired pose to prepare the welding job. A manipulator is composed of serial links which are affixed to each revolute or prismatic joint from the base frame through the end-effector as shown in Figure 3.

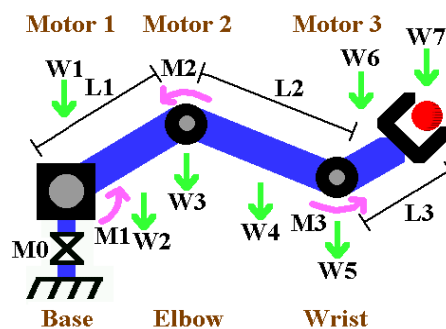


Figure 3. Working Simulation of the Welding Robot Arms

There are mainly two different spaces used in kinematics modelling of manipulators namely, Cartesian space and Quaternion space [12]. Denavit and Hartenberg showed that a general transformation between two joints requires four parameters, which was known as the Denavit-Hartenberg parameters and have become the standard for describing robot kinematics.

The robot kinematics is divided into forward kinematics and inverse kinematics. Forward kinematics will calculate the position and orientation of the end-effector in terms of the joint variables. It is straightforward and there is no complexity deriving the equations. Hence, there is always a forward kinematics solution of a manipulator. Inverse kinematics is however, going to tell how the robot move its arms and joints to a prescribed position with given pose. It is a much more difficult problem than forward kinematics. The solution of the inverse kinematics problem is computationally expansive and generally takes a very long time in the real time control of manipulators. The relationship between forward and inverse kinematics is illustrated in Figure 4.

We use the Denavit-Hartenberg method, which is the most common method, that uses four parameters to describe the robot kinematics. As shown in Figure 5, these parameters  $a_{i-1}$ ,

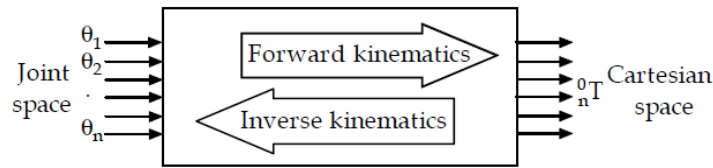


Figure 4. The Schematic Representation of Forward and Inverse Kinematics

$\alpha_{i-1}$ ,  $d_i$  and  $\theta_i$  are the link length, link twist, link offset and joint angle, respectively. A coordinate frame is attached to each joint to determine DH parameters.  $Z_i$  axis of the coordinate frame is pointing along the rotary direction of the joints.

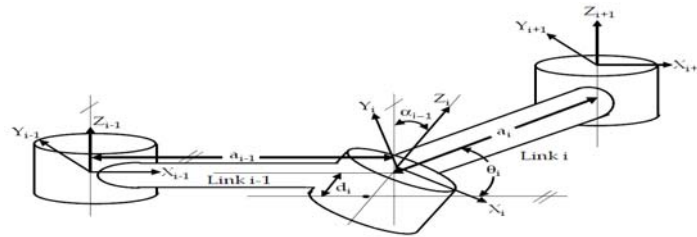


Figure 5. Coordinate Frame Assignment for a General Manipulator

The general transformation matrix  ${}^{i-1}T_i$  for a single link can be obtained as follows:

$${}^{i-1}T_i = R_x(\alpha_{i-1}) \cdot D_x(a_{i-1}) \cdot R_z(\theta_i) \cdot Q_i(d_i) \tag{6}$$

where  $R_x$  and  $R_z$  present rotation,  $D_x$  and  $Q_i$  denote translation. The forward kinematics of the end-effector with respect to the base frame is determined by multiplying all of the  ${}^{i-1}T_i$  matrices.

$${}^{base}_{end\_effector}T = {}^0T_1 \cdot {}^1T_2 \cdot \dots \cdot {}^{n-1}T_n \tag{7}$$

Tasks to be performed by a manipulator are in the Cartesian space, whereas actuators work in joint space. Cartesian space includes orientation matrix and position vector. However, joint space is represented by joint angles. The conversion of the position and orientation of a

manipulator end-effector from Cartesian space to joint space is called as inverse kinematics problem. i.e. to solve the joint  $\theta_1, \theta_2, \dots, \theta_n$  given that  ${}^0_nT$  is known.

The solution of the inverse kinematics problem strictly depend on the robot structures. Usually the end-effector is designed to be enough flexible so that it can arrive a given position with different ways. Thus, the inverse kinematics solution for a manipulator whose structure comprises of revolute joints generally produces multiple solutions.

### 2.5. Optimization of Point to Point Movement

Once the target point is given, the robot move to it according to a recorded trajectory which was taught by human at the very begining. Since there exist multiple solutions for the inverse kinematics, the robot should obtain some solution from time to time. And it's also true that it is computation intensive and long time process, the robot can not solve the inverse kinematics frequently.

In our approach, a learning probability  $\beta$  was given to controll the better solution seeking actions. When to move to a new target, robot will follow the existing path by the probability  $1 - \beta$ , and it will also solve the inverse kinematics and hope to find a better path by the probability  $\beta$ . If a better path was found, it was recorded and end-effector follows the new path. Otherwise, it will follow the existing path to maintain the performance not to decrease. Hence, it was done by self-learning on the basis human teaching program automatically.

### 2.6. Adaptive Control of Optimization Process

It is very intuitive and simple to control the learning procedure in the optimizing process by setting two learning probability  $\alpha$  and  $\beta$ , which control the possibility to find better solution in the optimization of point sequence and in the optimization of PTP trajectory, respectively.

In the beginning of optimization, there is only one welding point route and there is only one PTP moving trajectory between every two points. Thus, initial value of  $\alpha$  and  $\beta$  were set to big values to accelerate the searching process. After hundreds and thousands reparation, the better solution is incrementally found and the total performance trend to stabilize. If the system performance did not change for a long time, it maybe means our approach find the best controller program for the given task. In this case, learning probability  $\alpha$  and  $\beta$  should be set to zero to make the robot follows the existing best solution. As a conclusion, it is reasonable to set the value of  $\alpha$  and  $\beta$  proportion to the change rate of the system performance.

Let  $\eta$  denote the system performance, either system energy or the task time, and  $\Delta\eta$  denote the difference of system performance of the latest two generation. If the newest system performance becomes worse, we simply set  $\Delta\eta$  to zero. In our approach, the adaptive learning probability was adjusted as shown in (8).

$$\begin{aligned} \alpha &= \min(V_a, \Delta\eta / \eta) \\ \beta &= \min(V_b, \Delta\mu / \eta) \end{aligned} \quad (8)$$

Where  $V_a$  and  $V_b$  are the initial value of the learning probability  $\alpha$  and  $\beta$ .

## 3. Results and Analysis

We used system energy and time performance as the evaluation criteria to optimize the robot path. A robot simulation model was used to weld three kinds of chips with 36, 120, and 400 welding-points respectively, and each kind of experiment was simulated for 5000 rounds.

According to the experimental result  $M = 2^{\lfloor \log_2(n-1) + 1 \rfloor / 2}$ , the population size is set to 40, 100, 120 for the our three circuit board situations that contains 36, 120, and 400 points, respectively [10]. Every chromosome is initialized as a random number sequence and an initial population is created from a random selection of chromosomes. The next generation of population is generated by the traditional roulette wheel selection method with the best

individual always retained. The cross over and mutation operation is fixed by special process to prevent illegal chromosome being produced. The value of cross over rate is 0.8, and the value of mutation rate is 0.005. The initial value of the learning probability  $V_a = 0.5$  and  $V_b = 0.2$ .

To simplify the simulation, the manipulator is abstracted as three joints that connect the base, upper arms, lower arms, and end-effectors as shown in Figure 3. Then, the energy index is defined to be the sum of absolute value of the control pulses for the step motor M1, M2 and M3. Here the the control pulses that drives the link clockwise rotate is defined as positive, and a negative value means the pulses will drive the link to rotate counter clockwise. Table 1 gives some examples about the energy index in our experiment.

The first group of experiment is focus on the energy used for welding certain type of chip. The average number of the energy after 5000 rounds simulation was listed in table 2. It can be seen from table 2 that the energy consumption has a large increment in the first 1000 rounds, and has decrease to some extent in the second 1000 rounds. And it seems that robot can find the best solution in the third 1000 rounds since the energy still decrease and keep stable in it repeatedly running in the fourth and fifth 1000 rounds.

Table 1. Example About the Energy Index of the Welding Robot Manipulator

Step ID	M1	M2	M3	Energy
1	30	6	-7	43
2	100	-36	-10	146
3	-80	0	20	100
...	...	...	...	...

Table 2. The Results on the System Energy Index

Experiments Number	Welding points Number	the average number of the absolute value of control motor pulses		
		36	120	400
1		645	2140	9200
1000		1425.2	2962.8	12024.8
2000		842.8	2434.2	9107.3
3000		662.6	1850.7	8563.9
4000		642.1	1848.2	8554.6
5000		639.8	1845.4	8550.5

The second group of experiment is focus on the time used for welding certain type of chip. The first row of data was obtained from the teaching procedure which serves as the benchmark. And also, the average time in the first 5000 rounds for the robot to finish the welding task is given in table 3, where we can also see the same tendency as in the energy case after numbers executing and self-learning.

Table 3. The Results on the System Time Index

Experiments Number	Welding points Number	Time of welding the chip (unit: Seconds)		
		36	120	400
1		49	204	470
1000		60.5	286.5	504.7
2000		48.3	210.6	485.3
3000		43.2	184.5	402.8
4000		43.2	183.8	400.3
5000		43.1	183.5	399.7

From these experiments and simulations, we can see that it is possible for the robot to optimize its motion trajectory by self-study automatically, and thus decrease the energy and time consumption through the repetition of assigned work. Since no human tell the robot how to do, we can say that the robot has somewhat intelligence in trajectory plan.



#### 4. Conclusion

In this paper, the convenient path planning optimization for teaching and playback welding robot was proposed to enable to robot find more economical welding sequence in term of time and energy automatically by repeatedly doing the same welding task.

The optimization was divided into the welding points sequence improvement and trajectory improvement, which was done both on-line and off-line. Points sequence optimization, modeled as TSP and solved by GA method, runs once before every round of the same welding mission, which is so-call off-line optimization. For the on-line optimization, a try-and-error strategy was employed to select a different target point by certain probability ( $\alpha$  in this paper) and to solve a better trajectory between two points by another probability ( $\beta$  in this paper). These random searching actions were controlled by the learning probability  $\alpha$  and  $\beta$ , which was also adaptive according to change rate of the system energy.

Simulation results verified that, in contrast with the fix-sequence traditional controlling approach, the teaching and play back robot now can possibly decrease the time and energy consumption when same welding task was executed again and again.

#### References

- [1] Slotine J, Yang H. Improving the Efficiency of Time-Optimal Path-Following Algorithms. *IEEE Trans. Automat. Contr.* 1989; 5(1): 118-124.
- [2] Lin C, Chang P, Luh J. Formulation and optimization of cubic polynomial trajectories for industrial robots. *IEEE Trans. Automat. Contr.* 1983; 28(12): 1066-1074.
- [3] Shiller Z. Time-optimal motion of robotic manipulators. PhD Disertation. Massachusetts: Massachusetts Institute of Technology; 1987.
- [4] Kang Shin, McKay N. Minimum-time Control of Robotic Manipulators with Geometric Path Constraints. *IEEE Trans. Automat. Contr.* 1985; 30(6): 531-541.
- [5] Lee T, Ge S, Wang Z. Adaptive Robust Controller Design for Multi-link Flexible Robots. *Mechatronics.* 2001; 11: 951-967.
- [6] Khairudin M, Mohamed Z, Husain R. Dynamic Model and Robust Control of Flexible Link Robot Manipulator. *Telkomnika*, 2011; 9(2): 279-286.
- [7] Yang J, Lian F, Fu L. *Adaptive Robust Control for Flexible Manipulators*, Proceedings of IEEE int. conf. Robotics Autom. Nagoya. 1995; 1: 1223-1228.
- [8] Chen J, Meng C. Modeling and Adaptive Control of a Flexible One-link Manipulator. *Robotica.* 1990; 8: 339-345.
- [9] Wang K, Zhang J, Zhao B, Et al. *Research on control technology of trajectory tracking for robotic welding*. Proceedings of Int. Conf. Comput. Intell. Softw. Eng. Wuhan. 2010; 1: 1-4.
- [10] Luo S. The Welding Route Planning and Its Imitation of the Robot Used in Car Body Welding Process. Thesis. Shanghai: Tongji University; 2006.
- [11] Yan X, Wu Q, Liu H. An Improved Robot Path Planning Algorithm Based on Genetic Algorithm. *Telkomnika*, 2012; 10(8).
- [12] Kucuk S, Bingul Z. *Robot kinematics: Forward and Inverse Kinematics*. Industrial Robotics Theory Modelling & Control. ARS/pIV, Germany. 2006:117-148.