International Journal of Control and Automation
Vol. 8, No. 2 (2015), pp. 351-362
http://dx.doi.org/10.14257/ijca.2015.8.2.33

Trajectory Planning for a Welding Robot Based on the Bezier Curve

Caidong Wang, Xinjie Wang and Hui Zheng

College of Mechanical and Electrical Engineering,
Zhengzhou University of Light Industry,
Zhengzhou, 450002, China
vwangcaidong@163.com

Abstract

According to the requirements for tower crane welding robots, precisely planning the welding trajectory is essential. The optimal model for the least-time trajectory planning for the robot is established by Bezier curve, which has the characteristic of sub-section processing. With the consideration of constraints such as joint angular velocity and acceleration, the specific operation steps of genetic algorithm (GA) optimization are presented. A weld seam trajectory example is then proposed, and the simulation of a robot's trajectory planning is carried out. The result shows that the curve is smooth and steady, and can satisfy the requirements of tower crane welding operations. This method significantly reduces the motion time, and improves working efficiency.

Keywords: *Welding robot, trajectory planning, Bezier curve, genetic algorithm, optimization design*

1. Introduction

Tower cranes are essential equipment for the construction industry. With the rapid development of the world's infrastructure, the requirements for safety and reliability of tower cranes are increasing [1]. Most tower crane owners still adopt manual welding for their production. The development of enterprises is restricted by the problems of labor intensity, poor working conditions, low productivity, and a lack of quality control [2]. Welding robots have many advantages such as stable weld quality, increased productivity, and improved working conditions: as such they have been widely used in the: automotive, aerospace, ship-building, machine processing, electrical, and electronics industries [3]. Some large-scale construction machinery companies, which have introduced welding robots, are planning to explore the application of flexible robot welding technology on flat-head and boom tower crane structural parts. However, the welding robots are expensive, and they are difficult to use for tower crane welding applications. Therefore, this work has significance for the enhancement of the level of automation of tower crane manufacture by evolving the key technology of a tower crane welding robot.

It is difficult to establish a precise system and model because of the high welding quality required for a tower crane's standard section as they are welded: the welding robot's complex working environment, and its load change dramatically. During motion control of the robot, the angular acceleration constraint, angular velocity constraint, and angle constraint of each joint are the major constraints. In the general case of low-speed running, as long as the joint angles are not overrun, it has little influence on the trajectory planning. But, when the robot running speed is high, the joint angular acceleration and angular velocity can easily exceed their safe limits, which will force the drive current to become too large, or indeed cause an accident. At this point, the various constraints should be considered on the trajectory planning of the robot, and the most commonly

ISSN: 2005-4297 IJCA

method is to optimize the running time. Many researchers have proposed methods for calculating the time-minimal trajectories for a robot, *e.g.*, the curvature constraints should be considered when a mobile car is turning [4-6]. Chwa imposed joint velocity constraints and torque constraints on the trajectory of the manipulator as it intercepted fast-flying objects by optimizing the time [7]. Saravanan used the NURBS curve to map the robot manipulators' trajectory so that it can satisfy various constraints [8]. Algorithms for map building and path planning in an outdoor environment based on multi-sensor data fusion are available [9]. Based on the requirements of a tower crane standard section welding robot, which needs to plan its weld trajectory precisely during its operation, the time optimal model for robot trajectory planning is established by the use of Bezier curves [10]. Additionally, the joint's angular velocity and acceleration constraints are considered, and the simulation of a robot's trajectory planning undertaken.

2. Welding Robot: Overall Program

The main task of a tower crane welding robot is to complete the welding operation of the tower crane standard section's main branch. The welding robot system comprises: a welding manipulator, a welding locator, welding equipment, bearings, control systems, *etc.* The welding system for double location is demanded in the welding process. The manipulator is an essential part of the tower crane standard section welding system and directly determines the weld quality. To ensure that the welding torch can complete its welding work in an arbitrary posture, the 6-degree-of-freedom serial manipulator is adopted. According to the design principle of the manipulator that: "… arm and wrist should be separated …" [11], the 3-D model of the manipulator is designed, as shown in Figure 1. The front three joints form the arm structure and determine the position of the welding torch. The rear three joints' axes are mutually perpendicular and constitute the wrist joints which determine the posture of the welding torch. The 6-degree–of-freedom manipulator coordinate system is established by the D-H (Denavit-Hartenberg) method. The rear three joint axes of the manipulator intersect at one point, which is the reference point of the wrist. To make the kinematic equations simpler, joints 4, 5, and 6 are set as a reference point in the wrist. The coordinate system is established as shown in Figure 1 and the D-H parameters are shown in Table 1.

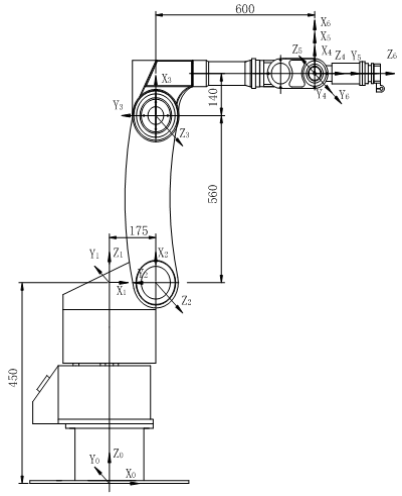

Figure 1. The D-H Coordinate System

Table 1. D-H Parameters

Rod number i	a_{i-1}	α_{i-1}	θ_i	d_i	variable range
	mm	(°)	(°)	mm	(°)
1	0	0	θ_1 (0)	d_1	0 to 340
2	a_1	-90	θ_2 (-90)	0	0 to 171
3	a_2	0	θ_3 (0)	0	0 to 190
4	a_3	-90	θ_4 (0)	d_4	0 to 360
5	0	90	θ_5 (0)	0	0 to 270
6	0	-90	θ_6 (0)		-360 to 360

3. Trajectory Planning Method: the Bezier Curve

3.1. Expression of Fourth-order Cubic Bezier Curve

The N[th] order vector equation of the Bezier curve is:

$$\begin{aligned}\theta(u) &= C_n^0(1-u)^n P_0 + C_n^1(1-u)^{n-1}uP_1 + \cdots + C_n^j(1-u)^{n-j}u^j P_j + \cdots C_n^n u^n P_n \\ &= \sum_{j=0}^{n} C_n^j(1-u)^{n-j}u^j P_j = \sum_{j=0}^{n} B_{j,n}(u)P_j\end{aligned} \tag{1}$$

Where $0 \le u \le 1$, P_0, $P_1 \cdots P_n$ are the space vector, $C_n^j = \dfrac{n!}{j!(n-j)!}$, $B_{j,n} = C_n^j(1-u)^{n-j}u^j$.

Based on the given space vector points P_0, P_1, P_2 and P_3, the fourth-order cubic Bezier plane curve can be derived as:

$$\theta(u) = \mathbf{u}^{\mathrm{T}}\mathbf{M}\mathbf{p} \quad 0 \le u \le 1 \tag{2}$$

Where

$$\mathbf{M} = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{u} = \begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix} \quad \mathbf{p} = \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}$$

3.2. Bezier Curve Constraint Conditions

To make the trajectory formed by the Bezier curve guarantee continuity of robot joint angular velocity and angular acceleration at the connections, we must derive conditions such that two Bezier curves should be satisfied at the connections. V_0, V_1, V_2, and V_3 are the control points of the first Bezier curve. a_1, a_2, and a_3 are the edge vector of the Bezier curve. W_0, W_1, W_2, and W_3 are the four control points of the second Bezier curve. b_1, b_2, and b_3 are the edge vectors (where W_0 and V_3 coincide, $W_0 = V_3$).

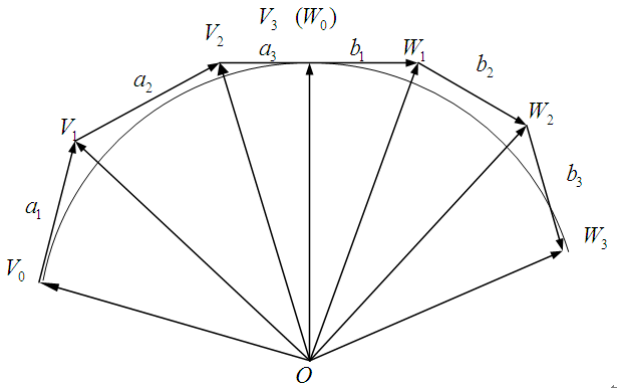

Figure 2. Stitch Figure of Two Bezier Curves

We assume that:

$$b_1 = \alpha a_3\ (\alpha > 0), b_2 = -\beta a_2 + \gamma a_3\ (\beta > 0, \gamma > 0)\ .$$

Then we obtain:

$$\begin{cases} W_1 = W_0 + b_1 = (1+\alpha)V_3 - \alpha V_2 \\ W_2 = W_1 + b_2 = \beta V_1 - (\alpha + \beta + \gamma)V_2 + (1+\alpha+\gamma)V_3 \end{cases} \tag{3}$$

The Bezier curve $\theta_1(u)$ that be constructed by four control points V_0, V_1, V_2 and V_3 can be derived as:

$$\theta_1(u) = (-u^3 + 3u^2 - 3u + 1)V_0 + (3u^3 - 6u^2 + 3u)V_1 + (-3u^3 + 3u^2)V_2 + u^3 V_3 \tag{4}$$

$$\frac{d\theta_1(u)}{du} = (-3u^2 + 6u - 3)V_0 + (9u^2 - 12u + 3)V_1 + (-9u^2 + 6u)V_2 + 3u^2 V_3$$

$$\frac{d^2\theta_1(u)}{du^2} = (-6u + 6)V_0 + (18u - 12)V_1 + (-18u + 6)V_2 + 6uV_3$$

If $u = 1$, we obtain:

$\theta_1(1) = V_3$,

$$\frac{d\theta_1(u)}{du}\Big|_{u=1} = -3V_2 + 3V_3 = 3(V_3 - V_2) = 3a_3,$$

$$\frac{d^2\theta_1(1)}{du^2}\Big|_{u=1} = 6V_1 - 12V_2 + 6V_3 = -6a_2 + 6a_3\ .$$

The Bezier curve $\theta_2(u)$ that be constructed by four control points W_0, W_1, W_2 and W_3, can be derived as:

$$\theta_2(u) = (-u^3 + 3u^2 - 3u + 1)W_0 + (3u^3 - 6u^2 + 3u)W_1 + (-3u^3 + 3u^2)W_2 + u^3 W_3 \tag{5}$$

In the same way, we obtain the formula $\frac{d\theta_2(u)}{du}$, $\frac{d^2\theta_2(u)}{du^2}$ from the derivative of equation 5.

If u=0, we obtain

$$\theta_2(0) = W_0 = V_3,$$

$$\frac{d\theta_2(u)}{du}\Big|_{u=0} = -3W_0 + 3W_1 = 3(W_1 - W_0) = 3b_1 = 3\alpha a_3,$$

$$\frac{d^2\theta_2(u)}{du^2}\Big|_{u=0} = 6W_0 - 12W_1 + 6W_2 = -6\beta a_2 + 6(\gamma - \alpha)a_3\ .$$

For the first and second order derivatives of Bezier curves are continuous at the connections, $\frac{d\theta_1(u)}{du}\Big|_{u=1} = \frac{d\theta_2(u)}{du}\Big|_{u=0}$, $\frac{d^2\theta_1(1)}{du^2}\Big|_{u=1} = \frac{d^2\theta_2(u)}{du^2}\Big|_{u=0}$ .We obtain $\alpha = 1, \beta = 1, \gamma = 2$.

So connect the two Bezier curves smoothly, the edge vector of the corresponding feature polygon should satisfy the following relationship:

$$b_1 = a_3 \quad b_2 = 2a_3 - a_2 \tag{6}$$

The two sections of the Bezier curves satisfy equation 5 during trajectory planning for the manipulator and the values of first and second derivatives at the control points are continuous, that is to say, the angular velocity and angular acceleration of the joint are continuous.

By indicating the relationship of data points between two adjacent sub-section curves, equation 3 can be rewritten as:

$$\begin{cases} W_1 = 2V_3 - V_2 \\ W_2 = 4V_3 - 4V_2 + V_1 = 2V_3 - 3V_2 + V_1 + W_1 \end{cases} \tag{7}$$

Assuming that the i^{th} section section Bezier curve connection data points are P_i , P_{i+1} and its control points are V_i、$V_i^{(1)}$、$V_i^{(2)}$、V_{i+1} , by the nature of the Bezier curve, $V_i = P_i, V_{i+1} = P_{i+1}$.

$$\begin{cases} V_i^{(1)} = 2V_i - V_{i-1}^{(2)} \\ V_i^{(2)} = 2V_i - 3V_{i-1}^{(2)} + V_{i-1}^{(1)} + V_i^{(1)} \end{cases} \tag{8}$$

So the trajectory of the i section Bezier curve of the manipulator is

$$\theta_i(u) = \mathbf{u}^{\mathrm{T}}\mathbf{M}\begin{bmatrix} V_i \\ V_i^{(1)} \\ V_i^{(2)} \\ V_{(i+1)} \end{bmatrix} \tag{9}$$

Equation 9 is a vector equation: V_i , $V_i^{(1)}$, $V_i^{(2)}$, and V_{i+1} are a set of control points. u is a parameter such that $0 \le u \le 1$ which gives the joint space trajectory of the manipulator. Vector control points have two components: the time component and joint angle component, respectively expressed by V_{ti} and V_{qi} .The corresponding trajectory point $\theta_i(u)$ also has time and joint angle components, they are respectively expressed by $t_i(u)$ and $q_i(u)$.

$$t_i(u) = \mathbf{u}^{\mathrm{T}}\mathbf{M}\begin{bmatrix} t_i \\ t_i^{(1)} \\ t_i^{(2)} \\ t_{i+1} \end{bmatrix} \tag{10}$$

$$q_i(u) = \mathbf{u}^{\mathrm{T}}\mathbf{M}\begin{bmatrix} q_i \\ q_i^{(1)} \\ q_i^{(2)} \\ q_{i+1} \end{bmatrix} \tag{11}$$

Each Bezier trajectory is only decided by the four control points according to the Bezier trajectory formation process, when a period of trajectory running conditions or power does not meet the requirements, some adjustment is needed. The welding robot's

working environment is constantly changing, so the trajectory planning method has to make a working trajectory plan quickly and accurately.

4. Optimization of the Bezier Curve Trajectory

Because the Bezier curve has the advantage of sub-section processing, and each sub-section is only determined by four adjacent control points, changing one vertex of the characteristic polygon will only affect the adjacent four curves relate to the vertex, the other curves do not change. This property of this trajectory optimization means that the entire trajectory of a manipulator can be optimized step-by-step. As a result, the whole running time optimization of the manipulator along the Bezier curve can be converted into the sub-section running time optimization for each sub-section.

(1) The Objective Function. The sub-section optimization is used for the robot running time of each sub-section of Bezier curve here.

$$T = \min \sum_{i=1}^{m-1} h_i \tag{12}$$

Where T is the total running time of the manipulator along the whole Bezier curve, h_1, $h_2, \cdots h_{m-1}$ are the running time required for the manipulator along each sub-section of the Bezier curve, and m is the number of data points.

(2) Design Variables. In equation (12), h_1, $h_2, \cdots h_{m-1}$ are independent variables, so we set $h_1, h_2, \cdots h_{m-1}$ as the design variables.

(3) Constraint Conditions.

1) Velocity constraints

Setting the joint velocity of a certain trajectory of manipulator to $\dot{\theta}(u)$, the maximum joint velocity should occur at time t_i or in the interval (t_i, t_{i+1}).

$$\dot{\theta}_{\max} = \max\left\{\left|\dot{\theta}_i\right|, \left|\dot{\theta}_{ix}\right|\right\} \tag{13}$$

Where $\dot{\theta}_{\max}$ is the absolute value of the maximum velocity according to its actual occurrence, $\left|\dot{\theta}_i\right|$ is the absolute value of the velocity at time t_i, $\left|\dot{\theta}_{ix}\right|$ is the maximum of the absolute value of the velocity in the interval $(t_i, t_{i+1})(i = 1, 2, \cdots m-1)$. Then $\left|\dot{\theta}_{ix}\right|$ can be solved by golden section method.

$$\frac{dt(u)}{du} = (-3u^2 + 6u - 3)t_i + (9u^2 - 12u + 3)t_i^{(1)} + (-9u^2 + 6u)t_i^{(2)} + 3u^2(t_i + h_i)$$

$$\frac{dq(u)}{du} = (-3u^2 + 6u - 3)q_i + (9u^2 - 12u + 3)q_i^{(1)} + (-9u^2 + 6u)q_i^{(2)} + 3u^2 q_{i+1}$$

Then:

$$\dot{\theta}(u) = \frac{d\theta(u)}{dt} = \frac{dq(u)}{du}\frac{du}{dt} = \frac{dq(u)}{du} \Big/ \left(\frac{dt}{du}\right) = \frac{A_1 u + A_2 u + A_3}{B_1 u + B_2 u + B_3} \tag{14}$$

Where $A_1 = -3q_i + 9q_i^{(1)} - 9q_i^{(2)} + 3q_{i+1}$, $A_2 = 6q_i - 12q_i^{(1)} + 6q_i^{(2)}$, $A_3 = -3q_i + 3q_i^{(1)}$, $B_1 = 9t_i^{(1)} - 9t_i^{(2)} + 3h_i$, $B_2 = 6t_i - 12t_i^{(1)} + 6t_i^{(2)}$, $B_3 = -3t_i + 3t_i^{(1)}$.

2) Acceleration constraints

$$\ddot{\theta}(u) = \frac{d\dot{\theta}(u)}{dt} = \frac{d\dot{\theta}(u)}{du} \cdot \frac{du}{dt} = \frac{q''t' - q't''}{t'^3}$$

$$t'' = \frac{dt'}{du},\ q'' = \frac{dq'}{du},\ t' = \frac{dt(u)}{du},\ q' = \frac{dq(u)}{du}$$

$$t'' = (-t_i + 3t_i^{(1)} - 3t_i^{(2)} + 6t_i + 6h_i)u + 6t_i - 12t_i^{(1)} + 6t_i^{(2)} = D_1 + D_2$$

$$D_1 = -t_i + 3t_i^{(1)} - 3t_i^{(2)} + 6t_i + 6h_i$$

$$D_2 = 6t_i - 12t_i^{(1)} + 6t_i^{(2)}$$

$$q'' = (-q_i + 3q_i^{(1)} - 3q_i^{(2)} + 6q_{i+1})u + 6q_i - 12q_i^{(1)} + 6q_i^{(2)} = C_1 u + C_2$$

$$C_1 = -q_i + 3q_i^{(1)} - 3q_i^{(2)} + 6q_{i+1},\quad C_2 = 6q_i - 12q_i^{(1)} + 6q_i^{(2)}$$

$$\ddot{\theta}_i(u) = \left[\left(B_1u^2 + B_2u + B_3\right)\left(C_1u + C_2\right) - \left(D_1u + D_2\right)\left(A_1u^2 + A_2u + A_3\right)\right] / \left(A_1u^2 + A_2u + A_3\right)$$

The maximum angular acceleration should occur at time t_i or in the interval (t_i, t_{i+1}).

$$\ddot{\theta}_{\max} = \max\left\{\left|\ddot{\theta}_i\right|, \left|\ddot{\theta}_{ix}\right|\right\} \tag{15}$$

$\left|\ddot{\theta}_i\right|$ is the joint angular acceleration of the manipulator at time t_i, $\left|\ddot{\theta}_{ix}\right|$ is the maximum absolute angular acceleration of the manipulator in the interval $(t_i,\ t_{i+1})$ $(i = 1, 2, \cdots m-1)$ and $\left|\ddot{\theta}_{ix}\right|$ can be solved by golden section method.

(4) Genetic algorithm (GA) optimization steps

GA was used in this work to solve the problem of robot trajectory planning. Assume that the robot trajectory has m nodes (including the start, and end, points), $[t_1, t_2]$, $[t_2, t_3]$, $\cdots\cdots$ $[t_{m-1}, t_m]$ are taken as time series. The length between two adjacent nodes is $h_i = t_{i+1} - t_i$ $(i = 1, 2, \cdots m-1)$. Where t_i is the time at which the welding robot arrives at a node. Optimization steps of GA were as follows [12]:

1) Coding

The time range of each Bezier curve for the welding robot is encoded as chromosomes of the GA. Float code is adopted here, by which a large area can be expressed.

2) Initial population

In the range of h_i, a number of individuals are generated randomly.

3) The distribution function of individual's fitness value in the population

For every individual in population, its fitness value is allocated by the following formula:

$$f = \begin{cases} \dfrac{1}{h_i} & \text{Satisfy the constraint conditions} \\ \dfrac{1}{h_i^{\max}} & \text{Not meeting the constraint conditions} \end{cases} \tag{16}$$

Where $h_i^{\max}$ is the maximum value of h_i. The individual which does not meet the constraint conditions has the minimum value according to equation 16.

4) Choice

Roulette selection is adopted here. The fitness value of an individual j in the population is set to f_i $(j = 1, 2 \cdots N)$. Under the roulette selection method, the selective probability of individual i is:

$$p_i = f_i / \sum_{j=1}^{N} f_j \tag{17}$$

5) Crossover

Arithmetic crossover operators. x_1 and x_2 are set as two parent individuals. In the action of arithmetic crossover operators, the progeny can be expressed as:

$$\begin{cases} x_1' = \lambda_c x_1 + (1-\lambda_c) x_2 \\ x_2' = (1-\lambda_c) x_1 + \lambda_c x_2 \end{cases}$$

Where λ_c is a random number between [0, 1].

6) Mutations

Non-uniform mutation is adopted here. The individual x_k' produced by non-uniform mutation based on x_k is calculated by:

$$x_k' = \begin{cases} x_k + \Delta(g, x_k^U - x_k) \\ x_k - \Delta(g, x_k - x_k^L) \end{cases} \tag{18}$$

Where x_k^U and x_k^L are the upper and lower boundary, g is the current number of iterations. $\Delta(g, y) = y \cdot \lambda_m \cdot (1 - g / G)^b$. Where λ_m is a random number between [0, 1], G is the maximum number of iterations, and b is a shape coefficient (set to 3 here). The specific parameters and steps are as follows.

Step 1: The population size N = 50, genetic algebra G = 80, crossover probability λ_c = 0.6, mutation rate λ_m = 0.1, given m data points P_i and (m- 1) time interval $[t_{i+1} - t_i]$.

Step 2: The control points q_i (the joint angle component of $V_i^{(1)}$) can be found by equation 8 based on the m data points P_i.

Step 3: When g = 1, the initial population h_i is generated randomly, the corresponding t_i can be achieved based on h_i.

Step 4: The Bezier curve can be obtained based on q_i and t_i, and it should be checked whether or not the curve meets the speed constraints.

Step 5: The adaptive value of individuals in the population can be obtained from equation 17.

Step 6: Take the operations of roulette wheel selection, arithmetic crossover, and non-uniform mutation on the current population.

Step 7: g = g + 1.

Step 8: If g < G, then switch to step (3). Otherwise, terminate the algorithm to yield the optimal population h_i.

5. Simulation Analysis

Taking the trajectory of welding robot welds on the standard section of a tower crane's main branch as an example, we chose eight nodes on the welding trajectory curve in Cartesian space (Table 2). Additional nodes 1 and 8 are obtained based on the initial conditions and boundary conditions, which sets the initial point velocity of the trajectory in each sub-section to zero. Putting the nodal data into the robot inverse kinematic equations, we obtain the corresponding joint angle of the eight nodes as shown in Table 2. Table 3 lists the constraints of the robot, which are determined by the structure of the robot and the motor parameters.

As shown in Table 2, due to the tasks' specific requirements, the third, fourth, and sixth joint angles changed little, therefore only those curves for the first, second, and fifth joint angles are shown. To verify the effect of GA optimization, the optimized running graphs and initial running graphs of each joint angle based on Bezier curves are plotted in Figures 3 to 5, respectively.

Table 2. The Values of the Nodes Unit: Degree

Nodes	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6
1	124.29	69.80	126.34	133.26	87.12	121.35
2	131.15	77.56	128.33	134.74	100.38	131.12
3	147.32	57.22	129.27	134.63	101.24	131.23
4	154.57	54.34	131.18	134.65	102.16	124.33
5	157.41	75.23	135.42	134.37	127.87	126.58
6	178.41	65.78	134.43	135.18	138.63	129.24
7	181.33	50.45	134.28	136.24	126.45	128.69
8	184.25	41.38	134.83	135.31	118.89	126.76

Table 3. The Constraints of Robot's Joint Angle Velocity and Acceleration

Constraints	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6
$\dot{\theta}$ (rad/s)	1.1	0.52	1.3	0.875	0.875	1.22
$\ddot{\theta}$ (rad/s^2)	3.5	2.62	3.5	3.5	3.5	3.5

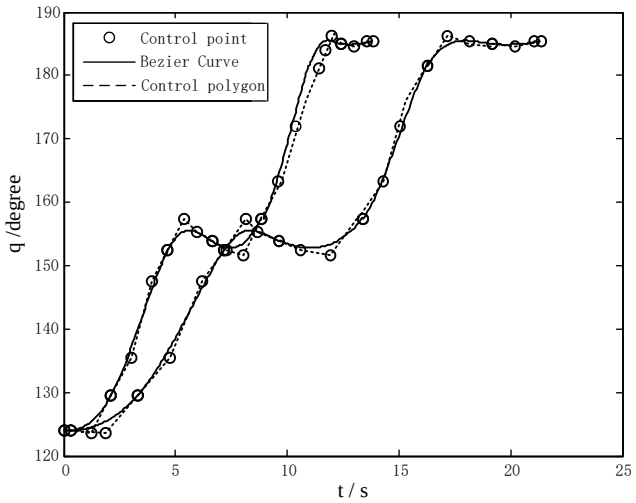

Figure 3. The Angle Optimization Curve of the First Joint

Figures 3 to 5 show that the angle variation curve of the first, second, and sixth joints were smooth, stable, and that they met the basic requirements of manipulator operation. Manipulator running time parameters are shown in Table 4 where the running time along a predetermined trajectory of the manipulator was seen to have been significantly reduced: the effect was perfect, and it could meet the velocity constraints. Each joint velocity curve can be obtained by equation (14). The robots' total running time, from the starting point to the end, has been reduced from 21.245 s to 14.122 s. The reduction of the running time improved the robots' working efficiency. Simulation results showed that the robots' running time had been reduced compared to the initial time taken. Each joint

displacement curve became smooth after GA optimization. Therefore the results realized the time optimal trajectory planning of the robot, and achieved the expected target.

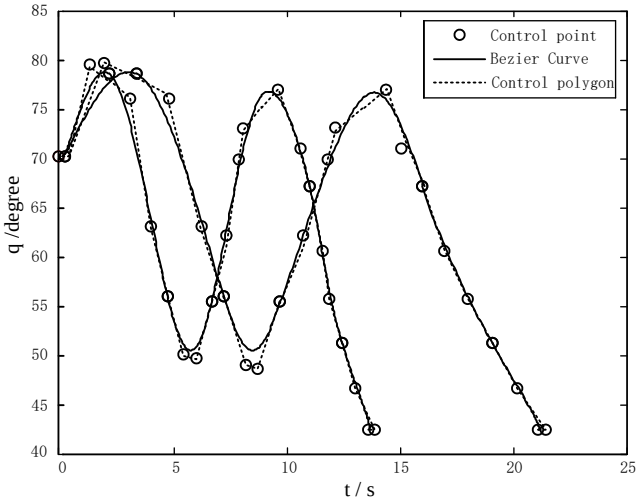

Figure 4. The Angle Optimization Curve of the Second Joint

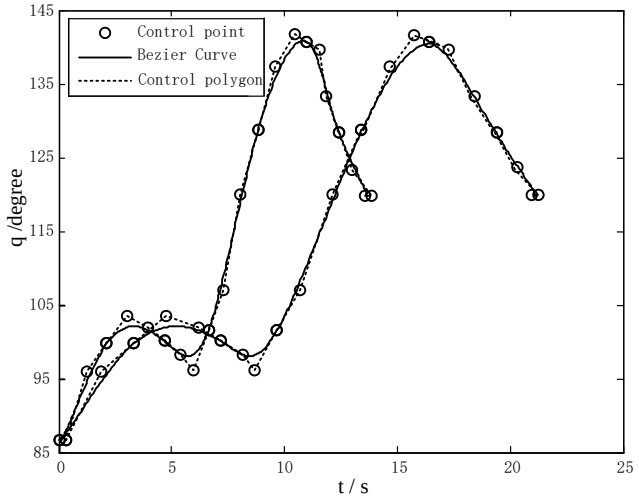

Figure 5. The Angle Optimization Curve of the Fifth Joint

Table 4. Time Optimization Results Unit: s

	h_1	h_2	h_3	h_4	h_5	h_6	h_7	Total time
Initial value	2.931	3.920	2.621	3.432	3.216	2.983	2.142	21.245
Optimal value	2.314	2.236	2.068	2.305	2.198	1.617	1.384	14.122

6. Conclusions

This work studied the application of the Bezier method in trajectory planning of a tower crane welding robot. According to the time optimal method, a robot trajectory planning model was established based on a Bezier curve method. The simulation results showed that the algorithm was simple, easy to implement, and reliable. The method could achieve the time optimal trajectory of a robot with kinematic constraints. This work only

took the velocity constraint into consideration during trajectory planning and did not consider the driving force constraints. The robot trajectory optimization method was undertaken by off-line programming, and did not consider real-time planning goals. Therefore, the planning of an optimal trajectory for such a robot under actual operational conditions, to realize on-line, real-time tracking may be a useful objective of future work.

Acknowledgements

This work is supported by Scientific and technological project in Henan Province (122102210435), Program for IRTSTHN (2012IRTSTHN013) and Doctoral Research Fund of Zhengzhou University of Light Industry.

References

[1]. L. W. Wang and L. Wang, Construction Machinery, vol. 39, no. 12, **(2008)**, pp. 21-25.
[2]. L. Hu, S. Xin and T. Wei, Journal of Harbin University of Science and Technology, vol. 17, no. 3, **(2012)**, pp. 105-109.
[3]. Y. L. Xu, T. Lin and S. B. Chen, Metal processing (thermal processing), vol. 39, no. 8, **(2010)**, pp. 32-36.
[4]. M. Haddad, W. Khalil and H. E. Lehtihet, IEEE Transactions on Robotics, vol. 26, no. 5, **(2010),** pp. 954-962.
[5]. L. Hu, S. Xin and T. Z. Wei, Journal of Harbin University of Science and Technology, vol. 17, no. 3, **(2012)**, pp. 105-109.
[6]. M. Jang, E. Lee and S. Choi, International Journal of Control and Automation, vol. 5, no. 4, **(2012)**, pp. 117-129.
[7]. D. Chwa, K. Junho and Y. C. Jin, IEEE Transactions on Systems, Man, And Cybernetics, vol. 35, no. 6, **(2005)**, pp. 831-843.
[8]. R. Saravanan, S. Ramabalab and C. Balamurugan, Robotica, no. 26, no. 6, **(2008),** pp. 753-765.
[9]. F. Yan, Y. Zhang and W. Wang, Int. J. Computer Application in Technology, vol. 44, no. 4, **(2012)**, pp. 276-283.
[10]. Y. M. Wang, Mechanical Science and Technology, vol. 20, no. 3, **(2001)**, pp. 350-352.
[11]. P. Wenger, D. Chablat and M. Baili, Journal of Mechanical Design, vol. 127, no. 1, **(2005)**, pp. 150-155.
[12]. L. F. Tian and C. Curtis, Mechatronics, vol. 14, no. 5, **(2004)**, pp. 455-470.

Authors

Caidong Wang is an Associate Professor at Zhengzhou University of Light Industry, China. He received his PhD from The School of Mechatronics Engineering, Harbin Engineering University, China, in 2011. His main research interests are mechatronics and robotic technology.

Xinjie Wang is a Professor at Zhengzhou University of Light Industry, China. She received her bachelor's, master's, and PhD degrees from Huazhong University of Science and Technology, China, in 1982, 1989, and 2005, respectively. Her main research interests include robotics and automatic control.

Hui Zheng is presently a graduate student majoring in mechanical design and theory at Zhengzhou University of Light Industry, China. He received his bachelor's degree in 2012 from Henan Polytechnic University, China. His main research interest is parallel robot control technology.