Robotica: page 1 of 19. © Cambridge University Press 2018
doi:10.1017/S0263574718001145

An optimal trajectory planning method for path tracking of industrial robots

Xianxi Luo†*, Shuhui Li‡, Shubo Liu†
and Guoquan Liu†

†*Jiangxi Engineering Research Center for New Energy Technology and Equipment, East China University of Technology, Nanchang 330013, China*
‡*Department of Electrical and Computer Engineering, University of Alabama, Tuscaloosa 35401-1956, USA*

(Accepted October 08, 2018)

SUMMARY

This paper presents an optimal trajectory planning method for industrial robots. The paper specially focuses on the applications of path tracking. The problem is to plan the trajectory with a specified geometric path, while allowing the position and orientation of the path to be arbitrarily selected within the specific ranges. The special contributions of the paper include (1) an optimal path tracking formulation focusing on the least time and energy consumption without violating the kinematic constraints, (2) a special mechanism to discretize a prescribed path integration for segment interpolation to fulfill the optimization requirements of a task with its constraints, (3) a novel genetic algorithm (GA) optimization approach that transforms a target path to be tracked as a curve with optimal translation and orientation with respect to the world Cartesian coordinate frame, (4) an integration of the interval analysis, piecewise planning and GA algorithm to overcome the challenges for solving the special trajectory planning and path tracking optimization problem. Simulation study shows that it is an insufficient condition to define a trajectory just based on the consideration that each point on the trajectory should be reachable. Simulation results also demonstrate that the optimal trajectory for a path tracking problem can be obtained effectively and efficiently using the proposed method. The proposed method has the properties of broad adaptability, high feasibility and capability to achieve global optimization.

KEYWORDS: Industrial robot, Trajectory planning, Genetic algorithm, Work piece placement

1. Introduction

For the past few decades, applications of industrial robot manipulators have increased rapidly in manufacturing lines to achieve fast, precise and secure production. An industrial robot is a highly non-linear and multivariable coupled system, and it is usually subjected to complex kinematic, dynamic constraints and other specific requirements of a task when executing the task. Trajectory planning is to schedule the optimal motion with respect to time and meet the requirements of the task without violating the constraints. It is a key research in industrial robot.[1] Our problem is targeting on tracking a predetermined path in a 3D work space. The two welding tasks[2,3] for automatic welding robot system, which are showed in Fig. 1, are typical examples of path tracking applications. Unlike pick-and-place operations that do not concern the intermediate movement path between the initial and final positions of the robot's end-effector, path tracking operation in these applications must track the predetermined weld seams with a pre-specified accuracy and velocity (task requirements).[4] At the same time, the constraints of position, velocity, acceleration and jerk of each joint should be considered so as to ensure the performance with a low overshoot, tracking error and velocity fluctuation. In addition, the smoothness, minimum execution time and minimum energy consumption are often demanded

* Corresponding author. E-mail: xianxi_luo@yahoo.com

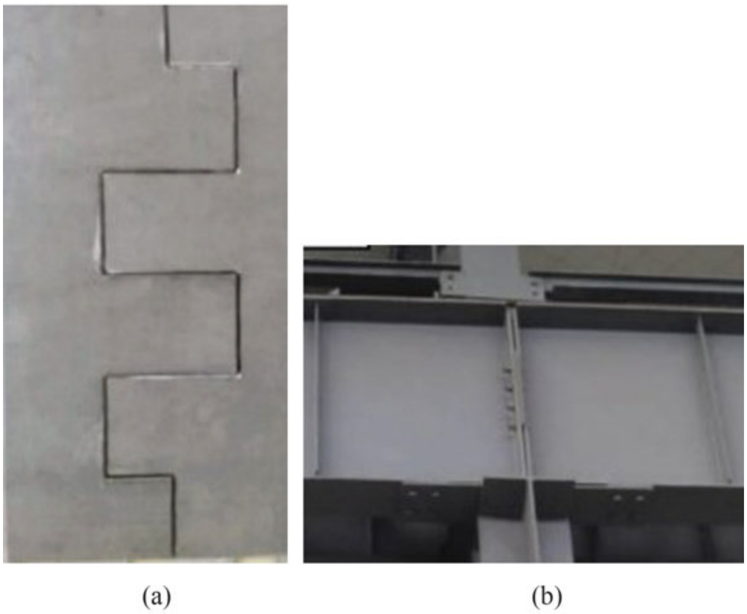


Fig. 1. Typical tasks of path tracking operation. (a) A workpiece to be welded. (b) Rectangular fillet welds on steel beam.

in practical applications. It is hard to satisfy so many requirements simultaneously with traditional programming methods such as the manual leading through technique. An offline trajectory planning and validation are essential and vital for such applications. Numerous methods for trajectory planning have been developed by different researchers. The commonly used are the polynomial interpolation algorithms in joint space and linear interpolation in Cartesian space. To overcome the discontinuities in torques and accelerations, algebraic spline interpolations were widely used in trajectory planning, e.g. cubic, quartic, quintic, trigonometric and synchronized trigonometric S-curves.[1] Recently, Zou proposed an algorithm using Non-Uniform Rational B-Spline (NURBS) curve interpolation to reduce the chord error and yield a smooth trajectory.[5] These researches successfully solved the problem of the smoothness of the trajectory. However, few of these researchers considered the kinematic constraints and the demand for minimum execution time and jerk.

The method of time-optimal trajectories through interpolating was first discussed in 1980s and developed quickly. Chen proposed the Harmony Search algorithm to solve the optimization problem for minimum execution time subject to kinematic and mechanical constrains with quintic B-splines interpolation.[6] Rubio[7] presented the methods to optimize execution time considering the dynamic characteristics in the presence of obstacles. Recently, Bu and Xu proposed to carry out trajectory planning using B-Spline curve and search the solution for minimum execution time using genetic algorithm (GA).[8] To increase the feasibility of the method, Xidias proposed an approach to clutter the 3D workspace with static objects and generate the trajectory considering the kinematic constraints and the presence of obstacles, and GA was applied to find the minimum time to perform the requested tasks.[9]

Besides the minimum execution time, the methods for a minimum jerk and minimum energy consumption trajectory have been studied. Decreasing the jerk leads to the reduction of the joint position errors and system vibration.[10] Later, Huang continued the work to develop an algorithm for an optimal time-jerk trajectory planning.[11] In order to minimize the energy consumption, Luo applied the Lagrange interpolation method to express the joint trajectory function and obtain optimal trajectory.[12] Liu applied the fourth-order Runge–Kutta, multiple shooting and traversing methods to solve the problem of energy-optimal trajectory planning for palletizing robot with high local capacity and high speed.[13]

Most of the techniques for minimum time, jerk and energy trajectory planning are optimization problems subject to constraints of kinematics, dynamics and the environmental obstacles. As the constraints increase in complexity, solutions cannot be easily obtained by analytic methods. There are many researches focusing on solving the problems using intelligent algorithms. GA was frequently applied in the cases considering several different constraints.[8,14] Lin studied the method based on particle swarm optimization and K-means clustering to derive the near optimal solution of a minimum-jerk joint trajectory.[15] Gao proposed a method of an improved teaching-learning-based optimization algorithm, which possesses better ability to escape from the local optimum.[16] Traditional trajectory generation techniques and optimization-based control strategies were merged into a unified framework for simultaneous motion planning and control as proposed by Zanchettin and Rocco.[17]

Based on these theories and techniques, the researchers were gradually extended to the problem of the practical path tracking applications. Besides all the requirements previously mentioned, a high tracking accuracy and specified space velocity along with the prescribed path is usually required. In order to reduce the deviation between the actual and desired tracks, Munasinghe segmented the desired path into straight lines and corners, and applied the interval analysis method to plan the trajectory for a three-DOF robot.[18] Zou proposed an algorithm based on NURBS curve interpolation with adaptive velocity to reduce the chord error.[5] Gao presented an adaptive algorithm through analysis of the fixed tracks to manage the joint speed and obtain a higher positioning accuracy.[19] Chen, Le and Yildirim applied image processing,[20] inverse Jacobian matrix[3] and artificial neural network[2] to recognize and track the welding seams (the prescribed tracks) in actual welding robot systems to increase positioning and trajectory accuracy.

However, it was found in our research that although the techniques using interpolation to generate smooth trajectories are suitable for pick-and-place applications, they cannot be directly applied for path tracking applications because the intermediate tracking accuracy and space velocity cannot be guaranteed. The optimal time and energy algorithms as mentioned in previous references are just focusing on searching for optimal intermediate curves passing through some taught points to minimize the objective functions, in which only the kinematic and dynamic constraints are satisfied, while the task requirements of accuracy and velocity are not included. This causes these methods unable to be applied for path tracking tasks directly. The above-mentioned references on path tracking applications mainly focused on the ways of improving the accuracy and feasibility of the trajectories. Few of them studied the methods of further optimization for time and energy efficient trajectories.

Different from the previous studies, the method of trajectory planning proposed in this paper is theoretically special and novel in the following aspects: (1) an optimal path tracking formulation strategy focusing on the least time and energy consumption without violating the kinematic constraints; (2) a special mechanism to discretize a prescribed path integration for segment interpolation to fulfill the optimization requirements of a task with its constraints; (3) a novel GA optimization approach that transforms a target path to be tracked as a curve with optimal translation and orientation with respect to the world Cartesian coordinate frame; (4) an integration of the interval analysis and GA algorithm to overcome the challenges for solving the special trajectory planning optimization problem. In the practical aspect, the contribution includes the validation of the proposed method is validated via computer simulation of a six-DOF robot, PUMA560 instead of a simple three-DOF robot through a comprehensive performance evaluation and measuring metrics.

The paper is organized as follows. The trajectory planning problem as well as the kinematic and dynamic constraints are described in Section 2. The interval analysis and GA algorithm as well as how to incorporate the two together are presented in Section 3. Simulation of the proposed algorithm and the result discussion are presented in Section 4, followed by final remarks and conclusions in Section 5.

2. Problem Formulation for Optimal Trajectory Planning

For a path tracking application, the geometric shape of a target path is usually predefined by a task, and the tracking velocity and error tolerance are predefined too. Traditionally, a series of adjacent configurations in the joint space were planned as a function over time along a curve which should be tracked with a specified velocity and error tolerance without violating the kinematic and dynamic constraints. A further problem is whether there exists an optimal transformation of the curve to make the trajectory both feasible and economical, which is analogous to changing the position and orientation of the work piece in a robot's work space.

In this paper, the curve to be tracked is defined in a Cartesian coordinate frame referred to as the definition frame. Before the optimization, the definition frame is consistent with the world frame. Since the trajectory is planned over time in this paper, the curve to be tracked is represented by a time function as shown by (1).

$$\Gamma\left(t\right) = \left(x\left(t\right), y\left(t\right), z\left(t\right)\right) \tag{1}$$

$\Gamma(t)$ represents the space curve, $x(t)$, $y(t)$ *and* $z(t)$ are the coordinates functions over time t in the definition frame. Then, this function is discretized with a sampling time T_s. Therefore, (1) is rewritten as (2a). Since T_s appears on both sides, (2a) can be simplified as (2b).

$$\Gamma\left(kT_s\right) = \left(x\left(kT_s\right), y\left(kT_s\right), z\left(kT_s\right)\right) \tag{2a}$$

$$\Gamma\left(k\right) = \left(x\left(k\right), y\left(k\right), z\left(k\right)\right) \tag{2b}$$

where T_s is the sampling time. $k = 0, 1, \ldots$ are the indices of the sampling points on the curve.

According to the robotics theory, each sampling point on the curve is regarded as the origin of the end effector's coordinate frame, and described as applying a homogenous transformation matrix $H(\cdot)$ with respect to the definition frame. When considering the end effector's orientation toward the tangential plane of the current sampling point on the curve in a 3D space, an additional factor $T_o(\cdot)$ should be multiplied to the right of (2b). Therefore, the curve can be represented by applying a series of homogenous transformation matrices over time as shown by

$$T_\Gamma\left(k\right) = H\left(x\left(k\right), y\left(k\right), z\left(k\right)\right) T_o\left(k\right) \tag{3}$$

where $H(\cdot)$ is homogenous transformation matrix. $T_o(k)$ is the additional factor matrix, it could be further explained as the homogenous transformation in accordance with the required orientation of the end effector with respect to the definition frame at time $t = kT_s$. $T_\Gamma(k)$ represents the homogenous transformation sequence defined with respect to the definition frame. The actual orientation requirements of the end effector are determined by the task, e.g. the welding torch orientation with respect to the plane of the welding seam, for better operation quality.

As previously mentioned, the trajectory could be further optimized by moving and rotating the definition frame with respect to the world frame. When the transformations are applied to the definition frame, the curve defined in the frame is translated or rotated accordingly. However, the relative positions and orientations of different points on the curve are kept unchanged.

The translational and the rotational displacements of the definition frame are selected as the decision variables in this paper. With respect to the world frame, translational displacements in x–y–z axes are noted as (x_T, y_T, z_T) and the rotational displacements are represented in the form of roll-pitch-yaw $(\theta_r, \theta_p, \theta_y)$, both of which correspond to the case of changing the position of placing the work piece. By multiplying homogenous matrix $T(x_T, y_T, z_T)$ and $T_{rpy}(\theta_r, \theta_p, \theta_y)$, which corresponds to (x_T, y_T, z_T) and $(\theta_r, \theta_p, \theta_y)$, respectively, to the left of $T_\Gamma(k)$, the trajectory generated in the definition frame is transformed into the world frame as $T_{\mathrm{OP}\Gamma}(k)$, which could be a potential solution to the problem of an optimal trajectory planning in a homogenous sequence.

$$T_{\mathrm{OP}\Gamma}\left(k\right) = T\left(x_T, y_T, z_T\right) T_{rpy}\left(\theta_r, \theta_p, \theta_y\right) T_\Gamma\left(k\right) \tag{4}$$

$T_{\mathrm{OP}\Gamma}(k)$ are the homogenous transformation matrices serial of the discretized optimal space curve.

$T(x_T, y_T, z_T)$ is the homogenous transformation matrix corresponding to the translation with respect to the origin of the world frame. $T_{rpy}(\theta_r, \theta_p, \theta_y)$ is the homogenous transformation matrix corresponding to roll-pitch-yaw rotational transformation with respect to the world frame.

We considered in this paper that a planned trajectory should satisfy the following constrains:

1. Task constraints

For a task, the moving velocities of the end effector along the curve should be kept stable within specified values, and the tracking error of the robot is also predefined, as given by

$$v = \begin{cases} \leq v_r \text{ along straight line} \\ \leq v_t^{\max} \text{ at corners} \end{cases} \tag{5a}$$

$$e \leq \rho \tag{5b}$$

where v is the moving velocity in workspace, v_r is the rated velocity, $v_t^{\max}$ is the maximum tangential velocity, e and ρ are the tracking error and error tolerance of the trajectory, respectively.

2. Kinematic constraints

For a robot, each configuration on the trajectory must meet the kinematic constraints in terms of position, velocity, acceleration and jerk for each joint to make the movement realizable, flexible and stable. These kinematic constraints are represented by

$$\begin{cases} \text{Joint postitions} & |q_i\,(k)| \leq q_i^{\max} \\ \text{Joint velocities} & |\dot{q}_i\,(k)| \leq \dot{q}_i^{\max} \\ \text{Joint accelerations} & |\ddot{q}_i\,(k)| \leq \ddot{q}_i^{\max} \\ \text{Joint jerks} & \left|\dddot{q}_i\,(k)\right| \leq \ddot{q}_i^{\max} \end{cases} \tag{6}$$

where $q, \dot{q}, \ddot{q}, \ddot{q}$ are joint position, velocity, acceleration and jerk, respectively. i is the index of the joint, and k represents the sampling index.

3. Dynamic constraints

In practice, a motor used to actuate a joint could only provide limited torque, power and energy. Therefore, each configuration on the trajectory must meet these dynamic constraints to prevent the motor from overloading as given by

$$\begin{cases} \text{Joint torque} & |\tau_i\,(k)| \leq \tau_i^{\max} \\ \text{Joint power} & |P_i\,(k)| \leq P_i^{\max} \\ \text{Joint energy} & |E_i\,(k)| \leq E_i^{\max} \end{cases} \tag{7}$$

where τ, $P\,and\;E$ are the joint torque, power and energy, respectively. The meanings of i and k are the same as those used in (6). The dynamic constraints are not considered in this paper temporarily because these constraints depend on the size of electric motors, characteristics of rotating components, positions of a robot, etc. and are more related to the transient and dynamic performance of a robot system. Therefore, it is a different topic.

Finally, the objective function used in this paper is defined as a weighed summation of the total absolute displacements for all the robot joints as given by

$$J = \sum\nolimits_{k=1}^{n} \sum\nolimits_{i=1}^{m} C_i\,|q_i\,(k) - q_i\,(k-1)| \tag{8}$$

where J is the value of objective function, C_i is the weighed coefficient and $q_i(k)$ is the position of the ith joint at the kth sampling time step. n is the total number of the discretized points on the curve. m is the total number of the joints. The goal of the optimal trajectory planning considered in this paper is to improve the robot energy efficient by minimizing the objective function (8). For the same displacements in different joints, the closer a joint is to the base, the more energy is consumed, and the larger value the corresponding coefficient C_i should be set.

For simplicity, the decision variables are vectored as $X = (x_T, y_T, z_T, \theta_r, \theta_p, \theta_y)$. In general, the strategies used in this paper include (1) planning the trajectory of a path tracking robot as a time series in a homogenous form, (2) determining the optimal vector X by minimizing the objective function (8) subject to the constraints of the inequalities (5) to (7) and finally (3) deriving the optimized discrete-time sequence of joint coordinate configurations for the robot. Details are presented in Section 3.

3. Algorithm for Trajectory Planning Optimization

To solve the optimization problem described in the previous section, a special technique is developed in this paper. This involves three steps. First, a target curve is divided into segments in the time domain and discretized into a time sequence. Each end point of a segment is represented as a homogenous

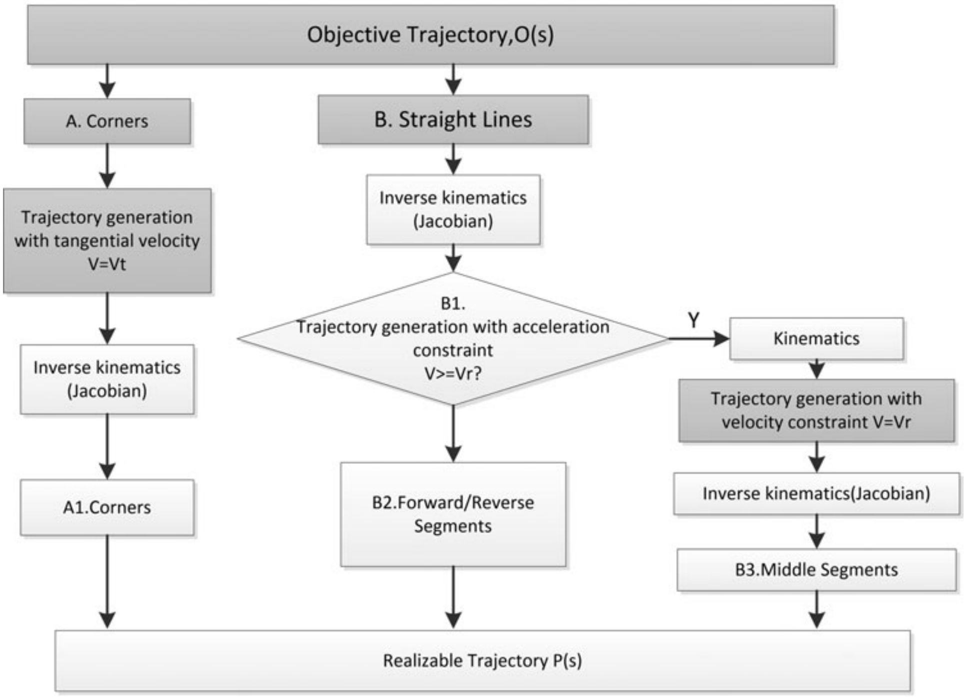

Objective Trajectory,O(s)
A. Corners
B. Straight Lines
Trajectory generation with tangential velocity V=Vt
Inverse kinematics (Jacobian)
B1. Trajectory generation with acceleration constraint V>=Vr?
Y
Kinematics
Trajectory generation with velocity constraint V=Vr
Inverse kinematics (Jacobian)
A1.Corners
B2.Forward/Reverse Segments
Inverse kinematics(Jacobian)
B3.Middle Segments
Realizable Trajectory P(s)

Fig. 2. The flow chart of the proposed trajectory planning method.

transformation matrix considering the orientation of the end effector approaching the target curve (Eq. (3)). Second, a GA is developed to search for the optimal decision variables by minimizing the objective function (8) without violating the listed constraints (5)–(7). Finally, the GA result is substituted into Eq. (4) to setup the optimal trajectory in the form of a joint coordinate configuration sequence, and the trajectory feasibility and robot performance are validated.

3.1. Segmenting a target curve into homogenous series

The proposed trajectory planning method is shown in Fig. 2. First, any geometric shape or curve of a path is described in the definition frame as $\Gamma(k)$. Then, the curve is divided into several segments consisting of corners and straight lines. For a corner segment, the trajectory is first planned in the Cartesian space with specified tangential velocities and then transformed into a joint space. Each straight-line segment is divided into three pieces: a forward piece (B1 and B2 in Fig. 2) with acceleration, a middle piece (B3) with a uniform velocity and a reverse piece with deceleration. Forward and reverse pieces are planned to ensure at least one joint move with its maximum acceleration/deceleration while ensuring that the inequality (5) is not violated. The middle piece is planned with a specified velocity in the Cartesian space. Finally, these segments were merged properly into a realizable trajectory $T_\Gamma(k)$.

3.1.1. Trajectory planning for a corner segment. Figure 3 shows the methodology developed in this paper for the trajectory planning of a corner segment $\angle ABC$. As it is difficult to balance the accuracy and efficiency for a sharp corner in the trajectory planning, trade-offs are usually made in industrial applications by substituting the sharp corner with a circular arc and by ensuring that the trajectory is constrained within an error tolerance ρ. To make the arc radius as large as possible, the arc should be tangential to the intersection of the tolerance cylinder and in the plane of ΔABC. In Fig. 3, the trajectory composed of two solid lines and an arc is used in the paper to replace a theoretical sharp corner $\angle ABC$. To construct the curve considering error tolerance, points A', B' and C' are determined from the points A, B and C as follows. All the symbols are depicted in Fig. 3.

First, the angle of the corner is obtained from Fig. 3 as

$$\beta = \cos^{-1}(\left(AB^2 + BC^2 - AC^2\right)/2 \cdot AB \cdot BC) \tag{9}$$

https://doi.org/10.1017/S0263574718001145

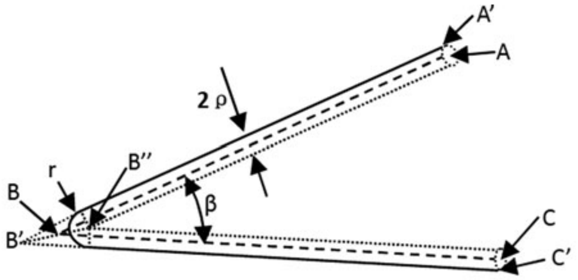


Fig. 3. A sharp corner and the realizable corner trajectory.

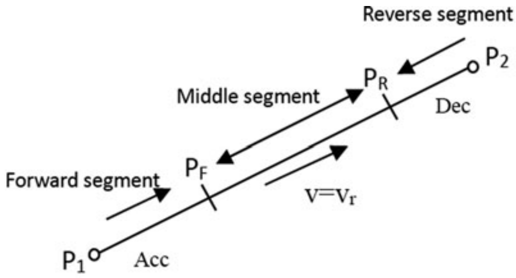


Fig. 4. Trajectory planning for a straight line.

Second, according to the geometry characteristics, the radius of the arc can be deduced as

$$r = 2\rho / \left(1 - \sin\frac{\beta}{2}\right) \tag{10}$$

Third, the minimum required time $t_{\min}$ and number of time steps N to track the arc are approximately estimated using the maximum tangential velocity $v_t^{\max}$.[18]

$$t_{\min} = (\pi - \beta)\, r/v_t^{\max} \tag{11}$$

$$N = \frac{t_{\min}}{T_s} \tag{12}$$

In (12), the number of time steps N is chosen as an upward rounded integer which requires that the tangential velocity should be adjusted accordingly after N is determined. By separating the arc into N equal pieces to obtain the Cartesian coordinate sequence of the discrete points for the corner (arc), the interval time between two adjacent points is used as the sampling time, T_s. The Cartesian coordinate sequence is noted as Γ_1.

3.1.2. Trajectory planning for a straight line. Figure 4 depicts the trajectory planning procedure for a straight line used in this paper. The trajectory planning involves three pieces: a forward piece $P_1 - P_F$, a middle piece $P_F - P_R$ and a reverse piece $P_{R\_}P_2$. However, the positions of P_F, P_R are unknown and should be determined through the planning. Basically, the proposed method is to divide a straight line (from P_1 to P_2) into equal distance pieces, index the via points (or reference points) by $j = 0, 1, 2 \ldots$. and then map these via points in a joint space as $q^j = (q_1^j, q_2^j, \ldots q_i^j, \ldots)$, where q_i^j represents the displacement of the robot's ith joint at the jth via point. For forward and reverse pieces, the two end points are either the start and end points, or the intersect point with an arc. The position and velocity of these points on the path are known. A forward piece is planned from P_1, whereas a reverse piece is planned from P_2 as shown in Fig. 4.

Then, an important issue is to determine the time interval between two via points. As the use of the maximum acceleration could reduce the execution time in forward/reverse pieces, the minimum time $t_i^{j\ \min}(k)$ required for the ith joint to move from the jth via point to $(j+1)$th via point at sampling

https://doi.org/10.1017/S0263574718001145

step k happens at the maximum acceleration and is obtained as

$$t_i^{j\ \min}(k) = \begin{cases} \frac{-\dot{q}_i^j + \sqrt{\dot{q}_i^{j^2} + 2\ddot{q}_i^{\max}\Delta q_i^j}}{\ddot{q}_i^{\max}} & \left(\text{if } \Delta q_i^j > 0\right) \\ \frac{-\dot{q}_i^j - \sqrt{\dot{q}_i^{j^2} + 2\ddot{q}_i^{\max}\Delta q_i^j}}{\ddot{q}_i^{\max}} & \left(\text{if } \Delta q_i^j < 0\right) \end{cases} \tag{13}$$

where $\Delta q_i^j = q_i^{j+1} - q_i^j$ represents the displacement of the ith joint when moving from the jth via point to $(j+1)$th via point. The minimum feasible time should ensure that each joint can complete the prescribed movement. That is,

$$t^{j\ \min}(k) = \max_i \left\{t_i^{j\ \min}(k)\right\} \tag{14}$$

Adjust the time $t^{j\ \min}(k)$ to be the multiple of the sampling time T_s with upward rounding. Assume that the minimum feasible time after the adjustment is $\tau^j(k)$. Then, the acceleration and velocity are adjusted with this new time interval $\tau^j(k)$. According to the principle of constant acceleration movement, the actual acceleration for each joint at the section of the piece between the jth and $(j+1)$th via point is expressed as

$$\ddot{q}_i^j = \frac{2\left(\Delta q_i^j - \dot{q}_i^j \tau^j\right)}{\tau^{j^2}} \tag{15}$$

Then, the trajectory in the joint coordinate between the two via points, including velocity and position, are generated according to Eqs. (16) and (17).

$$\dot{q}_i^j\left(k_j + 1\right) = \dot{q}_i^j\left(k_j\right) + \ddot{q}_i^j T_s \tag{16}$$

$$q_i^j\left(k_j + 1\right) = q_i^j\left(k_j\right) + \dot{q}_i^j\left(k_j\right) T_s + \frac{1}{2}\ddot{q}_i^j T_s^2 \tag{17}$$

In the above two equations, $k_j = 0, 1, \ldots, \frac{\tau^j}{T_s} - 1$ denoted the index of the discretized trajectory point with sampling time T_s along the line piece between the jth and $(j+1)$th via point. Later, the index is updated with the uniformly numbered index k when the segments are merged.

By this way, the forward/reverse pieces of the trajectory are produced. As the end effector velocity increases and arrives to the points with the rated velocity v_r, these points are determined as the beginning (P_F) or the end (P_R) point of the middle piece of the straight line. The corresponding indices are noted as k_F, k_R.

With Eqs. (16) and (17), the sequence of the joint configurations with the sampling interval time T_s is derived. These joint configurations are transformed into the Cartesian space and the sequence is noted as Γ_2.

For any middle piece of the straight line $(k_F \leq k < k_R)$, the velocity of the end effector in the Cartesian space is kept as the rated velocity v_r. Thus, the position of the end effector in the Cartesian space along the remaining line pieces are deduced by (18).

$$\begin{bmatrix} x(k+1) \\ y(k+1) \\ z(k+1) \end{bmatrix} = T_s \begin{bmatrix} v_r^x \\ v_r^y \\ v_r^z \end{bmatrix} + \begin{bmatrix} x(k) \\ y(k) \\ z(k) \end{bmatrix}; \quad (k_F \leq k < k_R) \tag{18}$$

where v_r^x, v_r^y, v_r^z are the velocity components along the x–y–z axes. The sequence produced by Eq. (18) in the Cartesian coordinates is denoted as Γ_3.

Thus, three sequences Γ_1, Γ_2 and Γ_3 are derived. Γ_1 is the sequence of the arcs. Γ_2 is the sequence generated from the acceleration/deceleration segments of the lines. Γ_3 is the sequence generated from the middle segments of the lines.

3.1.3. Merge the segments as a trajectory in homogenous consequence. There are three types of sequences Γ_1, $\Gamma_{2,}\Gamma_3$ in the Cartesian coordinates with the same time interval T_s. By merging them together in the order of the segments' positions, the discrete-time series of the trajectory $\Gamma(k)$ in the Cartesian space is derived. Considering the orientation of the end effector approaching the space curve, the target trajectory in homogenous form sequence $T_\Gamma(k)$ is expressed as Eq. (3).

3.1.4. Generate the trajectory in joint space. It is inconvenient to operate a robot with a trajectory in the Cartesian space. Instead, the target homogenous matrix series $T_\Gamma(k)$ is mapped into the joint space as $q(k) = (q_1(k), q_2(k), \ldots q_i(k), \ldots)^T$.

Generally, there is no effective mathematical expression for inverse kinematics.[21] The method of resolved-rate motion control with Jacobian matrix is usually used to determine the transformation.

A special velocity vector at time $t = kT_s$ comprised of the translational and rotational velocity components is defined as

$$v\,(k) = \left(v_x\,(k)\,,\, v_y\,(k)\,,\, v_z\,(k)\,,\, \omega_x\,(k)\,,\, \omega_y\,(k)\,,\, \omega_z\,(k)\,,\right)^T \tag{19}$$

where v_x, v_y and v_z represent the translational velocity along the axes *x, y* and *z* and ω_x, ω_y and ω_z represent rotational components along axis *x, y,* and *z*, respectively.

The definition of Jacobian matrix is

$$\dot{q}\,(k) = J(q\,(k))^{-1} v\,(k) \tag{20}$$

Applying Eqs. (20) to (21), we can obtain Eq. (22)

$$q\,(k+1) = q\,(k) + T_s \dot{q}\,(k) \tag{21}$$

$$q\,(k+1) = q\,(k) + J(q\,(k))^{-1} v\,(k)\,T_s \tag{22}$$

The vector Δ is defined to represent the incremental translation and rotation from time kT_s to $(k+1)T_s$.

$$\Delta = v\,(k)\,T_s = (\Delta x,\, \Delta y,\, \Delta z,\, \Delta\varphi_x,\, \Delta\varphi_y,\, \Delta\varphi_z)^T \tag{23}$$

where Δx, Δy and Δz represent the incremental translation along the axes *x, y,* and *z*, and $\Delta\varphi_x$, $\Delta\varphi_y$, *and* $\Delta\varphi_z$ represent incremental rotation angle along axes *x, y,* and *z*, respectively.

As the poses were represented in the homogenous form, the increment from $T_\Gamma(k)$ to $T_\Gamma(k+1)$ is

$$\Delta\,(T_\Gamma\,(k)\,,\, T_\Gamma\,(k+1)) = \begin{pmatrix} t\,(k+1) - t\,(k) \\ \text{vex}\left(R\,(k+1)\,R^T\,(k) - I_{3\times 3}\right) \end{pmatrix} \tag{24}$$

where $I_{3\times3}$ is an identity matrix, and $t(k)$, $R(k)$ are extracted from the following homogenous matrix:

$$T_\Gamma\,(k) = \begin{pmatrix} R\,(k) & t\,(k) \\ 0_{1\times3} & 1 \end{pmatrix} \tag{25}$$

Note: $t(k+1)$, $R(k+1)$ are similarly defined.vex$(\cdot)$ is an operator that extracts upper right triangle elements from the matrix $(R(k+1)R^T(k) - I_{3\times3})$ with specified principle shown in Eqs. (26) and (27).

$$R\,(k+1)\,R^T\,(k) - I_{3\times3} = \begin{pmatrix} 0 & -\Delta\varphi_z & \Delta\varphi_y \\ \Delta\varphi_z & 0 & -\Delta\varphi_x \\ -\Delta\varphi_y & \Delta\varphi_x & 0 \end{pmatrix} \tag{26}$$

$$\text{vex}\left(R\,(k+1)\,R^T\,(k) - I_{3\times3}\right) = (\Delta\varphi_x,\, \Delta\varphi_y,\, \Delta\varphi_z)^T \tag{27}$$

Thus, the procedure to map the homogenous pose $T_\Gamma(k)$ into the joint space is described by the following steps:

https://doi.org/10.1017/S0263574718001145

(i) Initialize with $k = 0$, and then determine the first joint coordinates $q(0)$ corresponding to the homogenous pose $T_{\Gamma}(0)$ by analyzing the characteristics of the robot. This is realized by placing the start point of the target space curve coincidental to a known initial pose (both in Cartesian space and joint space) with enough flexibility.
(ii) Apply a forward kinematic transformation to map the joint coordinate vector $q(k)$ to actual $T_{a\Gamma}(k)$.
(iii) Import the next homogenous pose $T_{\Gamma}(k+1)$ and calculate the incremental translation and rotation Δ from $T_{a\Gamma}(k)$ to $T_{\Gamma}(k+1)$ with Eqs. (25)–(27), (24). Then, calculate $\dot{q}(k)$ with (20).
(iv) Calculate the joint coordinates $q(k+1)$ using Eq. (22) and save the result.
(v) Set $k = k + 1$ and go to step (ii) until there is no further homogenous pose.

By this procedure, the trajectory in joint space $q(k), \dot{q}(k), \ddot{q}(k), \dddot{q}(k)$ $k = 0, 1, 2\ldots$ is generated. It should be noted that in the steps of (ii) and (iii), substituting $T_{a\Gamma}(k)$ for $T_{\Gamma}(k)$ is to compensate the linearizing errors induced by applying Jacobian matrix in Eq. (22).

3.2. Optimization with Genetic Algorithm (GA)

The objective of the GA-based optimization is to search for the optimal translational and rotational displacements $X_{\mathrm{op}} = (x_{\mathrm{op}T}, y_{\mathrm{op}T}, z_{\mathrm{op}T}, \theta_{\mathrm{op}r}, \theta_{\mathrm{opp}}, \theta_{\mathrm{op}y})$ for the definition coordinate frame (or initial frame), which is initially coincident with the world frame. The procedure is analogous to searching for the best position and orientation to place the work piece to make the process more feasible and efficient. The decision variables consist of the linear displacements in x, y, z axes (x_T, y_T, z_T) and the angular displacements in roll-pitch-yaw form $(\theta_r, \theta_p, \theta_y)$, with respect to the world frame. The objective function is defined by Eq. (8).

As the Cartesian space is easier to describe and understand, the trajectories are mostly planned in the Cartesian space. However, the constraints and the objective function are defined in the joint space. Furthermore, the kinematic transformations (forward or inverse) are highly non-linear. Consequently, the complexity increases as the number of DOF increases. Therefore, a multi-variant optimization problem is difficult to solve by using analytic methods.

In principle, GAs are stochastic search algorithms. These algorithms find the fittest individuals and combine their coding geneses to generate new offspring generations with specific principles. It has been proved that GAs are more robust and useful than other search techniques. This paper develops a GA-based algorithm for optimal trajectory planning as the explained below.

1. Decision variables and constraints

The decision variables (or parameter set) are $X = (x_T, y_T, z_T, \theta_r, \theta_p, \theta_y)$. These variables are physically constrained not only by the range of each variable (the interval group (28)), but also by the non-linear constraint functions describing the work space that the robot could reach (expressed as inequality (29)), or the subspace determined by the tasks.

$$\begin{cases} x_T \in \left(x^{\min}, x^{\max}\right) \\ y_T \in \left(y^{\min}, y^{\max}\right) \\ z_T \in \left(z^{\min}, z^{\max}\right) \\ \theta_r \in \left(\theta_r^{\min}, \theta_r^{\max}\right) \\ \theta_p \in \left(\theta_p^{\min}, \theta_p^{\max}\right) \\ \theta_r \in \left(\theta_r^{\min}, \theta_r^{\max}\right) \end{cases} \tag{28}$$

$$C\left(x_T, y_T, z_T, \theta_r, \theta_p, \theta_y\right) \leq 0 \tag{29}$$

2. Individual, chromosome and gene

An individual in this application is any probable value of X. It is sometimes referred to as a chromosome, while the vector entries of an individual are referred to as genes. Each gene contains a real number representing the value of the corresponding value of the component in vector X. The value of each gene is selected randomly from the constraints (the interval group (28) and inequality (29)) or the new generated individuals, namely offspring. The initial population is chosen at random within a variable range and satisfies the nonlinear constraint function.

3. Fitness function

The fitness function $F(X)$ is the function applied to assess the fitness of each individual in the population. The GA algorithm attempts to find the individual that minimize the value of the fitness function.

In this application, although the objective function (8) shown in Section 2 is defined as the weighted summation of the displacements of the joints in the joint space subject to the kinematic and dynamic constraints, where the relationship between the decision variable X and the fitness value is not defined explicitly. Thus, the objective function (8) cannot be used as the fitness function directly.

Here, the fitness function $F(X)$ is designed as the following steps:

(i) The trajectory is planned with the original space curve to derive the homogenous form sequence $T_\Gamma(k)$ using the method described in Sections 3.1.1 to 3.1.3.
(ii) The input decision variables X are substituted into Eq. (4) by multiplying each matrix in the sequence of $T_\Gamma(k)$. The trajectory in homogenous form sequence $T_{\mathrm{OP}\Gamma}(k)$ is generated corresponding to $T_\Gamma(k)$.
(iii) The procedure described in Section 3.1.4 is used to map the homogenous poses $T_{\mathrm{OP}\Gamma}(k)$ onto the joint space.
(iv) The output of the fitness function is then determined for different possible cases according to the feasibility of the trajectory as shown below:

$$F(X) = \begin{cases} J = \sum_{j=1}^{n} \sum_{i=1}^{m} C_i \left| q_i^j - q_i^{j-1} \right| & \left(\text{feasible trajectory}\right) \\ \quad M \quad \left(\text{violation of inequality } (6), \text{unfeasible}\right) \end{cases} \tag{30}$$

M is an arbitrarily selected number as long as the value greater than J for any feasible trajectory. In this way, the relationship between input X and output of the fitness function $F(X)$ is setup.

4. Producing the next generation

All the individuals of current population are assessed by the fitness function. Some of the individuals are selected as the parents to produce the next generation by a random selection process which would give higher probabilities to those with better finesses (lower fitness function value). The GA creates three types of children for the next generation:

(i) Elite children are the individuals in the current generation with the best fitness values and survived to the next generation.
(ii) Crossover children are the individuals generated by a crossover operator from the selected parent pairs. The algorithm creates a crossover child by assigning one of the randomly selected genes from two parents to the same gene of the child's chromosome. A crossover function is used to carry out this operation.
(iii) Mutation children are the individuals created by randomly changing the genes of individual parents.

Both crossover and mutation are essential to the GA. Crossover enables the algorithm to extract the best genes from different individuals and recombine them into potentially solution. Mutation adds to the diversity of a population and generate individuals with better fitness values, thereby make the algorithm escape from local minima solution to global minima solution.

After creating a new generation, the algorithm repeated the procedure to select parents and generate offspring until all stop conditions are satisfied. The individual with the best fitness found by the algorithm is the optimal solution $X_{\mathrm{op}} = (x_{\mathrm{opT}}, y_{\mathrm{opT}}, z_{\mathrm{opT}}, \theta_{\mathrm{opr}}, \theta_{\mathrm{opp}}, \theta_{\mathrm{opy}})$ to the trajectory planning problem.

https://doi.org/10.1017/S0263574718001145

Table I. DH parameters of PUMA 560.

j	θ	d	a	α
1	$q1$	0	0	1.571
2	$q2$	0	0.4318	0
3	$q3$	0.15	0.0203	-1.571
4	$q4$	0.4318	0	1.571
5	$q5$	0	0	-1.571
6	$q6$	0	0	0

Table II. The kinematic constraint of PUMA560[24].

Joint N	1	2	3	4	5	6
$q_i^{\max}$(rad)	π	$3\pi/4$	$3\pi/4$	$3\pi/4$	$3\pi/4$	$3\pi/4$
$\dot{q}_i^{\max}$(rad/s)	8	10	10	5	5	5
$\ddot{q}_i^{\max}$(rad/s^2)	10	12	12	8	8	8
$\dddot{q}_i^{\max}$(rad/s^3)	30	40	40	20	20	20

5. Stop conditions

GA stops when it reaches a stopping condition. There are many options such as specified generation, time, fitness limit or stall test. The choice for an application depends on the characteristics of the robot model.

3.3. Generating the final trajectory

After finding the optional solution, $X_{\text{op}} = (x_{\text{opT}}, y_{\text{opT}}, z_{\text{opT}}, \theta_{\text{opr}}, \theta_{\text{opp}}, \theta_{\text{opy}})$, generating the optimal trajectory for the specified space curve is relatively simple. The trajectory planned in the definition frame with homogenous form sequence $T_\Gamma(k)$ is imported and multiplied by the optimization matrices on the left, finally transformed into the optimal trajectory $T_{\text{OP}\Gamma}(k)$ with respect to the world frame.

Executing the procedure mentioned in Section 3.1.4 transforms the homogenous poses $T_{\text{OP}\Gamma}(k)$ into a joint configuration sequence $q(k)$ as a trajectory in the joint space. This also allows for $\dot{q}(k), \ddot{q}(k), \dddot{q}(k),\ k = 0, 1, 2, \ldots$ to be calculated. The feasibility of the generated optimal trajectory is validated by emulating the trajectory and checking whether the constraints are violated.

It is needed to point out that although the trajectory planning based on the GA algorithm takes some time, there is no trajectory planning involved during the execution of a robot to follow the planned trajectory so that there is no effect on the execution speed of the robot.

4. Simulation and Discussion

The proposed method was emulated with PUMA560. The robot has six-DOFs and a spherical joint wrist making it capable to attain any pose in the Cartesian work space. The DH parameters are listed in Table I. The kinematic constraints are listed in Table II. Due to the complexity, the kinematics transformation and Jacobian matrix are not presented in the paper. The readers should refer to the references.[22,23]

The simulation conditions for the movement parameters of the robot were set as follows:

1. Rated velocity: $v_r = 0.15$ m/s.
2. Tangential velocity: $v_t^{\max} = 0.15$ m/s.
3. Trajectory error tolerance: $\rho = 0.001$ m.

End-effector orientation in the roll-pitch-yaw form with respect to the target curve was set as a constant vector $(\theta_{\text{or}}, \theta_{\text{op}}, \theta_{\text{oy}}) = (-pi/2, 0, 0)$, meaning the axis z of the end-effector frame was perpendicular to the x–o–z plane in which the target curve was initially defined. In practice, the orientation could be chosen according to the productive requirements and could be varying over time.

To reduce the searching time and produce a realizable result, the searching range was set to

$$\begin{cases} x_T \in (-0.87, 0.87) \\ y_T \in (-0.87, 0.87) \\ z_T \in (-0.87, 0.87) \\ \theta_r \in (-1.57, 1.57) \\ \theta_p \in (-1.57, 1.57) \\ \theta_r \in (-1.57, 1.57) \end{cases} \tag{31}$$

The reachable work space for PUMA560 was emulated and verified, and the translation from the world place was given by the non-linear inequality:

$$0.0226 \le x_T^2 + y_T^2 + z_T^2 \le 0.7289 \tag{32}$$

Three typical target curves (line, circle and corner) were defined in the x–o–z plane (unit: m) of the definition coordinate frame (initially coincided with the world frame). The task curve was defined from the origin of the frame to make it easier to describe. Through translational and rotational transformations of the definition frame with respect to the word frame, the curve could be placed to any reachable position and orientation in the world frame.

Case 1 – Line: start point (0,0,0); end point (0.3,0,0.5).

The case for a line was first selected to show the effectiveness of the proposed method. Although a line was initially defined, later it was found that the trajectory was unrealizable if the definition frame was unmoved or unadjusted. For example, the trajectory of dashed line in Fig. 5(a) is unrealizable.

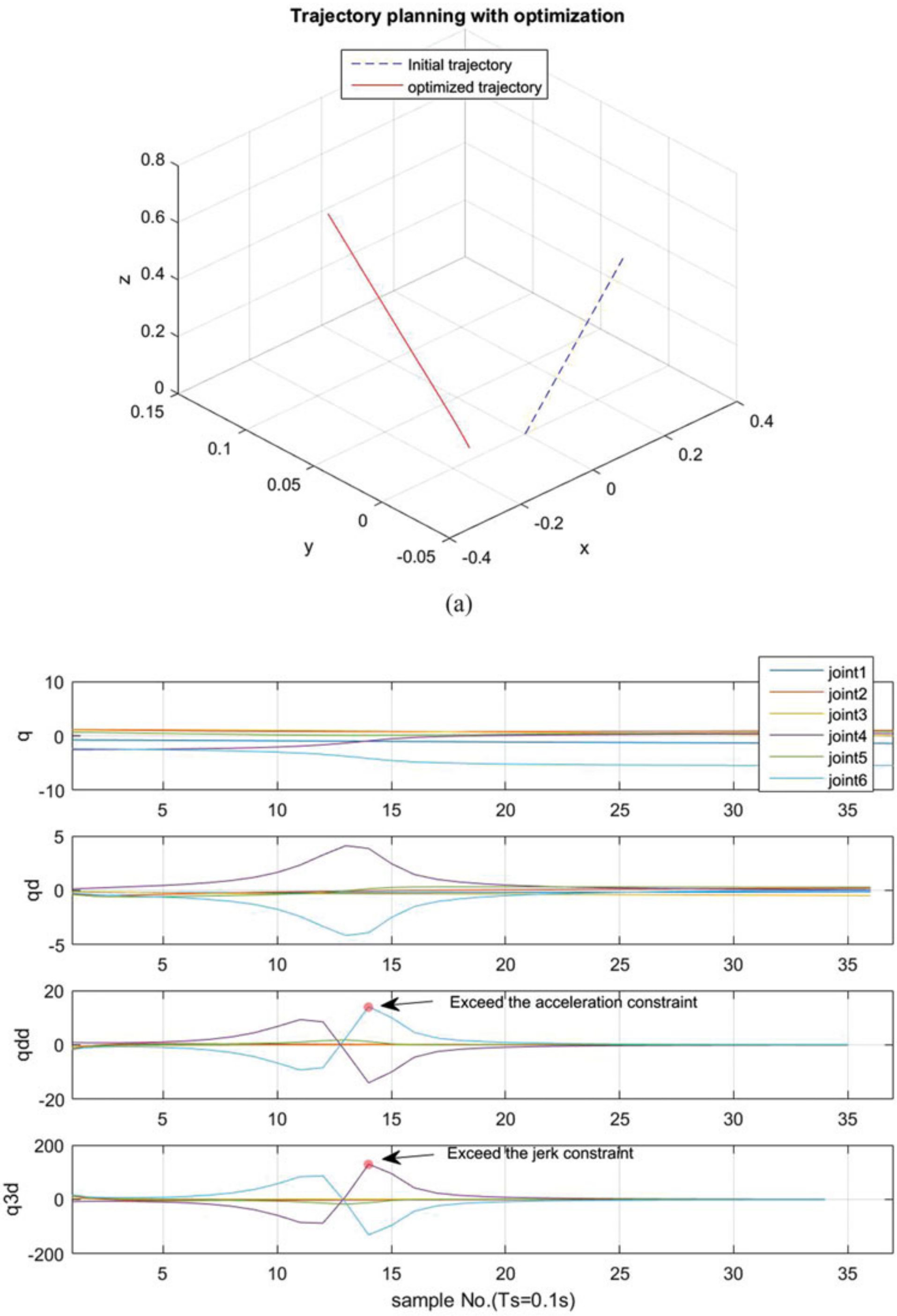
An individual created using the GA was randomly picked out (not the optimal one). The decision variable corresponding to the individual was X = (−0.1322, *0.0610, 0.1829, 0.0312,* −0.4124, *0.5020).* Figure 5(a) shows the trajectory (red solid line) transformed upon the definition frame corresponding to the translation and rotation quantities. Figure 5(b) shows the variation of the joint variables of the randomly chosen trajectory transformed from the initial one. The symbol q, qd, qdd, $q3d$ in the figure represents the corresponding variable q, $\dot{q}$, $\ddot{q}$, $\dddot{q}$.

From Fig. 5(a), it was found that the shape of the trajectory remains unchanged after the homogenous transformation upon the translation and rotation, which proves the correctness of the method of discretization and transformation of the proposed method. From Fig. 5(b), it was found that when proceeding the randomly chosen trajectory, although the joint coordinate and velocity were within the constraints, the accelerations and jerks at some points obviously exceeded the constraints ($\ddot{q}_6 \le 8$; $\dddot{q}_6 \le 20$). This indicates that even by manually leading a robot, which is commonly applied in practical applications, to define the trajectory, the planned acceleration and jerk may violate the constraints. Therefore, the procedure of optimization and validation is necessary and critical for such applications.

Case 2 – Circle: start point (0,0,0); circle radius 0.2; center (0,0, −0.2).

The case of a circle was selected to demonstrate the capability of the method to maintain the relationship between the trajectory plane and the orientation of the end effector. Figure 6(a) shows one animation capture of the robot movement. Figure 6(b) shows the relative space relation of the trajectory by transformation with the optimized variable set and the initial one. The dashed circle is the originally defined curve, while the solid line is the optimized curve.

From Fig. 6(a), it was found that the trajectory was realized in the robot's work space, and the z-axis of the end effector remained perpendicular to the plane in which the circle was located. Theoretically, the orientation could be arbitrarily defined by changing the value of roll-pitch-yaw according to the requirement of the application. This case shows that as long as the trajectory defined using Eq. (4) as homogenous transformation matrices, the orientation of the end effector would



(a)

(b)

Fig. 5. The trajectory planning and optimization for a line. (a) The optimized trajectory (randomly chosen individual) with respect to the initial one upon the transformation of translation and rotation quantities. (b) The joint variables of the optimized trajectory.

remain in the same orientation relative to the curve plane. From Fig. 6(b), it was found the shape of the optimized trajectory was quite similar to that of the initial one, which means that the target curve is accurately tracked. Furthermore, the center position of the circle moved and the plane where the curve located was changed, which means that the translation and orientation transformation upon the definition coordinate frame can migrate the task curve to the ideal places. Therefore, the foundation

https://doi.org/10.1017/S0263574718001145

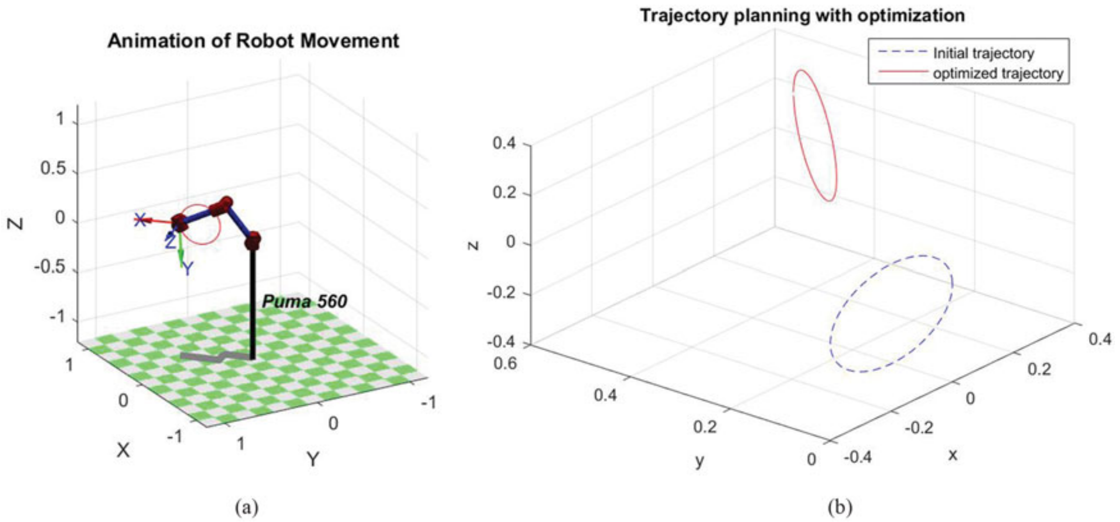


Fig. 6. Trajectory planning and optimization for a circle. (a) Animation capture of the robot movement. (b) The space relation of initial and optimized trajectory.

of the proposed method is verified. Actually, the method not only works for the trajectory of a line and circle but also works for any shapes and spatial curves in 3D space, because the definition coordinate frame is 3D and any spatial curve could be defined in the frame.

Case 3 – Corner: start point (0,0,0); corner point (0.5,0,0); end point (0.3,0,0.5).

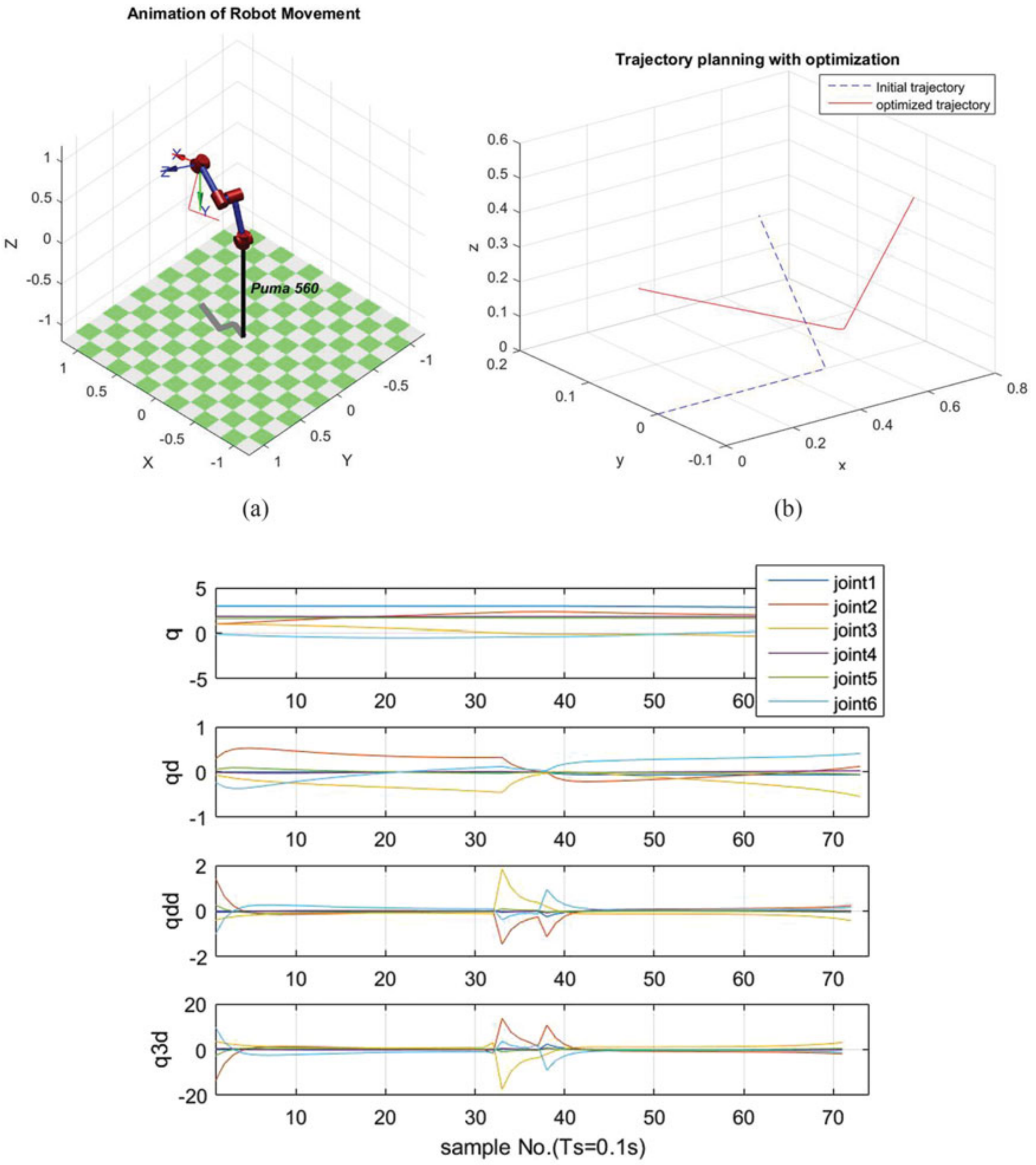
The case of a sharp corner was selected to verify the trajectory planning method for a path tracking operation around a corner. A corner trajectory was planned as two lines with forward/reverse/straight pieces and a circular arc, as discussed in Section 3. The case is basically a synthesis of the proposed method. As previously mentioned in Section 3.1, although there are many different potential curves, any curve can be divided into a combination of corners (circular arcs) and lines. This case is applied to represent the effectiveness of the proposed method because any curve segmentation strategy and the algorithm for implementing the trajectory planning are the same. Figure 7(a) shows an animation capture to emulate the proposed method. The target trajectory was successfully realized. Figure 7(b) shows the space relation of the initial and the optimized curves. The initially defined curve is the blue dashed line, and the optimized curve is the red solid line. Figure 7(c) shows the variation of the joint variables in the process of proceeding to the optimized trajectory.

From Fig. 7(a), we found that the trajectory of the sharp corner trajectory was successfully realized with the specified orientation approaching the plane which the curve was located in. From Fig. 7(b), we concluded that with the proposed method, the optimized trajectory was changed in both translational and rotational displacements with respect to the original one. This result of the transformation corresponding to the decision variables $X = (0.0902,\ 0.2090,\ 0.2020,\ 0.3263,\ 0.3016,\ -0.2522)$ was searched by the GA algorithm, and the fitness value was *0.9433*. In addition, the sharp corner was substituted with a circular arc. However, by the previously mentioned interval analysis, the tracking accuracy was assured. For the straight-line segments, the space proceeding velocity of the end effector was maintained at the specified value of the rated velocity except at the links to the arc, start or end point positions. Such performances are acceptable in most engineering scenarios.

From Fig. 7(c), it was found that each variable was constrained within the specified range, which means that the optimized trajectory satisfies the kinematic constrains and it is feasible for the robot.

In these simulation studies, the objective function was innovated to act as a fitness function according to Eq. (30), where it was expressed as weighed sum of absolute travel distance of each joint. The weighed coefficients were set as *(0.3, 0.2, 0.2, 0.1, 0.1, 0.1)*. Intuitively, as the joint approached the base, more power was needed to drive the mechanism, and larger coefficient should be set to multiply the absolute travel distances. Although there was not a strictly designed energy minimization objective function, this approximation resulted in effective and efficient behaviors in general applications. It

https://doi.org/10.1017/S0263574718001145



(a)

(b)

(c)

Fig. 7. Trajectory planning for a sharp corner. (a) Animation capture of the robot movement. (b) The space relation of initial and optimized trajectory. (c) The joint variables of the optimized trajectory.

was noted that changing the structure or parameters of the objective function would lead to a different optimal solution.

The GA algorithm in this paper is implemented by using the GA solver in Matlab Global Optimization Toolbox. In the above simulation, the equations and initial parameters required by the GA algorithm solver include

1. fitness Function: Eq. (30);
2. number of variables: 6, corresponding to dimension in the world frame such as $X = (x_T, y_T, z_T, \ \theta_r, \theta_p, \theta_y)$;

3. constraints: the initial searching range defined by Eq. (31), the reachable work space defined by Eq. (32), and the kinematic constraint defined in Table II;
4. population size for the GA algorithm: 30;
5. initial population: stochastically generated in the initial searching ranges;
6. selection option: stochastic uniform;
7. elite count: 0.05*populationsize;
8. crossover fraction: 0.8;
9. stopping conditions: stall generations — the algorithm stops when the average relative change in the fitness function value over stall generations is less than Function tolerance 1e-3.

The initial parameter selection is based on the following considerations: (a) the characteristics of the targeted problem, (b) the user guide of the global optimization toolbox in MATLAB, (3) previous experience on similar GA-based optimization problems and (d) trial and error.

Through the simulation studies, the scientific merit and properties of the proposed algorithm can be summarized as follows:

1. By interval analysis, any spatial curve (task path) can be segmented into lines and arcs. By adjusting the steps for each segment, the task requirements of space velocity and tracking accuracy are fulfilled. So the proposed method has wide adaptability in this type of applications.
2. The procedure of optimization selects the translation and orientation vectors as decision variables. During the iteration process of setting different values for these variables, mapping the initial curve to a new placement, calling the procedure of interval analysis to generate the trajectory and evaluating the results with the objective function and kinematic constraints, the optimal trajectory is found. Hence, the procedure assures the optimal choice and the complex constraint requirements and the feasibility of the generated trajectory is ensured.
3. As the inherent property of the GA algorithm, the calculation can escape from the trap to local minimum points by mutation operation. Thus, the generated result is the global optimal one.
4. It can be predicted that if the thresholds of the acceleration and jerk constraints are set at lower values, the smoothness of the trajectory could be improved. Because GA algorithm will discard the individuals with higher acceleration and jerk values, which means unstable movement.

Although the proposed method solved the problem successfully, it can be further improved. It took almost 2 h to get an optimal solution for Case 3. Obviously, it could take much longer time for a more complicated curve. As a result, the proposed method can be improved regarding the following three aspects: (1) implement the proposed method with high performance computing and parallel processing technology to reduce computing time; (2) research the trade-off relation between tracking accuracy and sampling time T_s to improve the computing speed for the trajectory planning; (3) determine more appropriate kinematic constraints according to the actual poses of the robot. Research in this area could further enhance the potential of industrial robot trajectory tracking.

5. Conclusion

A new trajectory planning method for robot path tracking is studied in this paper. The problem is to plan the trajectory with a specified geometric path, while allowing the position and orientation of the path to be arbitrarily selected within the specific ranges. The optimal objective is to realize the trajectory with both least energy and time consumption without violating the task constraints, kinematic constraints and dynamic constraints. The problem is analogous to that of optimizing the trajectory of the torch of a welding robot to track a seam on a surface of a work piece.

Three main procedures are proposed in this paper to solve the optimization problem:

1. The target curve is defined in a definition frame that initially coincides with the world frame. By interval analysis, the target curve is segmented into straight line sections and circular arc sections. The task constraints and the smoothness are considered by piecewise planning of the trajectory. The procedure of mapping homogenous sequence into the joint space is also proposed.
2. The translation and orientation vectors upon the definition frame with respect to the world frame are selected as the decision variables. GA algorithm is developed and applied to obtain the optimal transformation to the initial curves by assessing the alternatives with objective function and the

kinematic constraints simultaneously. Especially, this includes how to design the fitness function properly and effectively.

3. Combine the results to build up an optimal trajectory and then validate it via computer simulation.

The simulation demonstrated that it is not a sufficient condition to formulate a trajectory by just considering that each point in the trajectory is reachable. Consequently, the widely used manually leading-through method could be unfeasible for such an application. The simulation also showed that the proposed method has the properties of wide adaptability, high feasibility and capability of global optimization. It should be emphasized that GA algorithm is effective when searching for an optimal solution only if the optimal trajectory planning problem and the fitness function are properly designed.

Future research will target on improving and verifying the properties of the proposed method by comparing it with other algorithms (e.g. Particle Swam Optimization) and considering the existence of interferences. The performance and innovation of the proposed method will be considered and evaluated as well when being applied in under-actuated (DOF<6) robots or redundant (DOF>6) robots and applied for online trajectory planning.

Acknowledgements

The work is supported under the grant projects of Chinese National Sciences Foundation *(No. 61463003, 51409047, 61463004)*, the Chinese Scholarship Council *(No.201508360120)* and other projects *(GJJ13466, GJJ13467, JXNE2017-01)*.

References

1. Z. Li, G. Li and Y. Sun, "Development of articulated robot trajectory planning," *Int. J. Comput. Sci. Math.* **8**(1), 52–60 (2017).
2. S. Yildirim and B. Ulu, "Design of Artificial Neural Network Predictor for Trajectory Planning of an Experimental 6 DOF Robot Manipulator," Proceedings of the 4th Conference on Mechanisms, Transmissions and Applications MeTrApp2017, Springer, Trabzon, Turkey, Netherlands (Jul. 3, 2017–Jul. 5, 2017).
3. J. Le, H. Zhang and X. Chen, "Realization of rectangular fillet weld tracking based on rotating arc sensors and analysis of experimental results in gas metal arc welding," *Robot. Comput.-Integrated Manuf.* **49**, 263–276 (2018).
4. L. Jingwei, T. Yifei and W. Shaofeng, "Welding robot kinematics analysis and trajectory planning," *Telkomnika (Telecommun. Comput. Electron. Control)* **14**(2A), 92–100 (2016).
5. Q. Zou, W. Guo and F. Y. Hamimid, "A Novel Robot Trajectory Planning Algorithm Based on NURBS Velocity Adaptive Interpolation," Proceedings of the *International Conference on Mechanical Design ICMD2017*, Springer, Beijing, China, Netherlands (Oct. 13, 2017–Oct. 15, 2017).
6. Y. D. Chen *et al.*, "Optimal trajectory planning for industrial robots using harmony search algorithm," *Ind. Robot-an Int. J.* **40**(5), 502–512 (2013).
7. F. Rubio *et al.*, "Industrial robot efficient trajectory generation without collision through the evolution of the optimal trajectory," *Robot. Autonomous Syst.* **86**, 106–112 (2016).
8. J. Bu and J. Xu, "Research on time optimal trajectory planning of 7-DOF manipulator based on genetic algorithm," *Acta Technica CSAV (Ceskoslovensk Akademie Ved)* **61**(4), 189–199 (2016).
9. E. K. Xidias, "Time-optimal trajectory planning for hyper-redundant manipulators in 3D workspaces," *Robot. Comput.-Integrated Manuf.* **50**, 286–298 (2018).
10. H. S. Liu, X. B. Lai and W. X. Wu, "Time-optimal and jerk-continuous trajectory planning for robot manipulators with kinematic constraints," *Robot. Comput.-Integrated Manuf.* **29**(2), 309–317 (2013).
11. J. Huang *et al.*, "Optimal time-jerk trajectory planning for industrial robots," *Mechanism Mach. Theory* **121**, 530–544 (2018).
12. L. P. Luo *et al.*, "Trajectory planning for energy minimization of industry robotic manipulators using the Lagrange interpolation method," *Int. J. Precision Eng. Manuf.* **16**(5), 911–917 (2015).
13. Y. Liu *et al.*, "A method of energy-optimal trajectory planning for palletizing robot," *Math. Problems Eng.* **2017**, 1–10 (2017).
14. J. Tao *et al.*, "Genetic algorithm based homing trajectory planning of parafoil system with constraints," *Zhongnan Daxue Xuebao (Ziran Kexue Ban)/Journal of Central South University (Science and Technology)* **48**(2), 404–410 (2017).
15. H. I. Lin, "A fast and unified method to find a minimum-jerk robot joint trajectory using particle swarm optimization," *J. Intell. Robot. Syst.* **75**(3–4), 379–392 (2014).
16. X. S. Gao, Y. Mu and Y. Z. Gao, "Optimal trajectory planning for robotic manipulators using improved teaching-learning-based optimization algorithm," *Ind. Robot-an Int. J.* **43**(3), 308–316 (2016).
17. A. M. Zanchettin and P. Rocco, "Motion planning for robotic manipulators using robust constrained control," *Control Eng. Practice* **59**, 127–136 (2017).

18. S. R. Munasinghe, “Trajectory planning and control of industrial robot manipulators,” **In:** *Industrial Robotics: Theory, Modelling and Control* (Sam Cubero, ed.) (Intech, 2006) pp. 335–348.
19. M. Gao *et al.*, “Adaptive Velocity Planning for 6-DOF Robots with Fixed Tracks,” *Proceedings of the 25th IEEE International Symposium on Industrial Electronics ISIE2016*, Institute of Electrical and Electronics Engineers Inc, Santa Clara, CA, USA (Jun. 8, 2016–Jun. 10, 2016).
20. K. Chen *et al.*, “A Path-Planning Algorithm of the Automatic Welding Robot System for Three-Dimensional Arc Welding using Image Processing,” *Proceedings of the 13th International Conference on Ubiquitous Robots and Ambient Intelligence URAI2016*, Institute of Electrical and Electronics Engineers Inc, Xian, China (Aug. 19, 2016–Aug. 22, 2016).
21. P. Corke, *Robotics, Vision and Control* (Springer, Berlin, Heidelberg, 2011).
22. C. Lin, P. Chang and J. Luh, “Formulation and optimization of cubic polynomial joint trajectories for industrial robots,” *IEEE Trans. Autom. Control* **28**(12), 1066–1074 (1983).
23. K. Tse and C. Wang, “Evolutionary optimization of cubic polynomial joint trajectories for industrial robots,” *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, San Diego, CA (1998).
24. T. Chettibi *et al.*, “Minimum cost trajectory planning for industrial robots,” *Eur. J. Mech. a-Solids* **23**(4), 703–715 (2004).