Hybrid Time-Energy Optimal Trajectory Planning for Robot Manipulators With Path and Uniform Velocity Constraints

Wenchuang Sang[1,2], Ning Sun[1,2,*], Chenglin Zhang[1,2], Zehao Qiu[1,2], Yongchun Fang[1,2]
[1]Institute of Robotics and Automatic Information Systems,
College of Artificial Intelligence, Nankai University, Tianjin 300350, China
[2]Institute of Intelligence Technology and Robotic Systems,
Shenzhen Research Institute of Nankai University, Shenzhen 518083, China

***Abstract*—For robot manipulators executing specified tasks along predefined paths (for example, grinding, welding, and spraying), it is usually desired that the end effector possesses a uniform linear velocity distribution without violating robot joint constraints. In addition, the time-energy indicator is minimized during the motion. Based on the path parameterization method, this paper presents a pseudo acceleration allocation method which is continuous of the path jerk. Then the particle swarm optimization algorithm with inequality constraints is utilized to derive the best trajectory parameters to achieve the optimality of time and energy consumption. At last, the effectiveness of the proposed time-energy trajectory planning method is verified through numerical simulations.**

***Index Terms*—Robot Manipulators, Constrained Paths, Velocity Constraints, Optimal Trajectory Planning, Particle Swarm Optimization**

I. Introduction

With the rapid development of industry in the 21st century, robot manipulators are widely applied to carry out complex, dangerous, and repetitive work, which greatly improves productivity. It is significant to plan reasonable operation trajectories for robots performing specific tasks before being put into use. Additionally, in order to fully exploit robots' capabilities to further increase working efficiency and simultaneously reduce energy consumption, time-energy optimal trajectory planning is taken into consideration.

For some application scenarios, such as grinding, welding, and spraying, the moving path of a robot is usually strictly constrained [1], which implies that trajectory planning is completed in the task space. In order to minimize the executing time of a robot moving along a specified path, the phase plane technique [2]–[5] is adopted to search switching points of the path acceleration, and then optimal velocity profiles are generated with joint torques constrained. With the predefined path parameterized, the variables to be optimized of the optimization problem are reduced [6], [7]. Reference [6] uses the dynamic programming approach to construct optimal trajectories for arbitrary optimization objectives with multiple types of constraints. The convex optimization method [8]–[11] is applied to assign velocity along the specified path as well, which transforms the original time-optimal trajectory planning problem into a convex optimal control problem. Compared with the original numerical integration (NI) method [2], [3], reference [12] proposes a novel algorithm ensuring time optimality and completeness with both torque and velocity constraints, which is conducive to the robot safe operation. In [13], the back and forward check (BFC) algorithm is proposed to determine the optimal pseudo velocity along the path with both velocity and acceleration constraints, in which the path parameters are discretized. The works in [7], [14] achieve smooth trajectories through the input shaping method and constrained torque rates, respectively.

Aside from the minimum time problem, energy optimization is an extraordinarily noteworthy issue, due to limited energy resources carried on robots, high costs arising from tremendous energy consumption, and environmental concerns [15]–[17]. With the optimality of time and energy being taken into account simultaneously, time-energy optimal path tracking solutions are provided. In [6], the dynamic programming approach is applied to solve both time-optimal and time-energy optimal problems. The works in [18]–[20] acquire bang-bang control inputs by solving two-point boundary value problems with equality and inequality constraints. Reference [21] uses the convex optimization method to construct the optimal trajectory for a 6-degree-of-freedom (DOF) industrial robot.

Based on the previous achievements mentioned above, the main works of this paper are listed below: 1) In order to meet constraints of a specific task that requires the end effector to possess a wide range of uniform velocity, an improved interpolation method with transition functions is proposed to assign the pseudo acceleration along the predefined path. 2) The particle swarm optimization (PSO) algorithm with inequality constraints is adopted to minimize the cost function and find optimal trajectory parameters. 3) Numerical simulation results of a 6-DOF serial manipulator are given to illustrate the performance of the proposed trajectory optimization scheme.

*Corresponding author, E-mail: sunn@nankai.edu.cn

This work is supported by the National Key R&D Program of China under Grant 2018YFB1309000, the National Natural Science Foundation of China under Grant U20A20198, and the Natural Science Foundation of Tianjin under Grant 20JCYBJC01360.

978-89-93215-23-6/22/$31.00 ACA

The remainder of this paper is structured as follows. In Section II, we introduce the interpolation method and formulate the optimization problem. Then, Section III provides a 6-DOF manipulator example moving along a specified path. Next, in Section IV, simulation results are given and analyzed. At last, we provide conclusions in Section V.

II. Problem Formulation and Solution

A. Problem Formulation

Considering an n-DOF robot manipulator, the dynamics model [7], [10] can be obtained by Lagrange formulation and expressed as

$$M(\boldsymbol{q})\ddot{\boldsymbol{q}}+C(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}+G(\boldsymbol{q})+\mu\dot{\boldsymbol{q}}=\boldsymbol{\tau}, \tag{1}$$

where $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, and $\ddot{\boldsymbol{q}} \in \mathbb{R}^n$ denote the vectors of joint positions, velocities, and accelerations, respectively, $M(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ represents the mass matrix, $C(\boldsymbol{q},\dot{\boldsymbol{q}}) \in \mathbb{R}^{n\times n}$ is the coupling matrix of centrifugal and Coriolis force, $G(\boldsymbol{q}) \in \mathbb{R}^n$ indicates the vector of gravity, $\mu \in \mathbb{R}^{n\times n}$ is a diagonal matrix and the main diagonal element $\mu_i \in \mathbb{R}$ $(i=1,2,\cdots,n)$ is the viscous friction coefficient of joint i, and $\boldsymbol{\tau} \in \mathbb{R}^n$ means the vector of joint torques. Noting that if an end effector is connected to the manipulator, it can be regarded as a component of the end link, and then (1) can be updated by identification [7].

For the robot moving along a predefined path, a parameter s is introduced, which means the distance from the starting point to a certain point on the path. Then the position and orientation of the end effector in the reference Cartesian coordinate system can be parameterized as

$$\boldsymbol{P}=\boldsymbol{f}(s), \tag{2}$$

where $\boldsymbol{f}(\cdot)$ indicates that the posture of the robot's end effector is delivered as a continuous function of s.

Furthermore, according to [12], $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, and $\ddot{\boldsymbol{q}}$ can be derived as follows:

$$\boldsymbol{q}=invkine(\boldsymbol{P})=\boldsymbol{g}(s), \tag{3}$$

$$\dot{\boldsymbol{q}}=\boldsymbol{g}'(s)\dot{s}, \tag{4}$$

$$\ddot{\boldsymbol{q}}=\boldsymbol{g}''(s)\dot{s}^2+\boldsymbol{g}'(s)\ddot{s}, \tag{5}$$

where $invkine(\cdot)$ means the inverse kinematics model of the robot manipulator, $\dot{s}$ and $\ddot{s}$ denote the velocity and acceleration along the path, respectively, $\boldsymbol{g}(s)$ is the parametric representation of joint positions, $\boldsymbol{g}'(s)$ and $\boldsymbol{g}''(s)$ satisfy the equations of $\boldsymbol{g}'(s)=\partial\boldsymbol{g}/\partial s$ and $\boldsymbol{g}''(s)=\partial^2\boldsymbol{g}/\partial s^2$, respectively.

For some application scenarios of robots moving along specified paths, such as polishing the edges of metal ingots, it is required that the end effector possesses a wide range of uniform linear velocity in the task space to ensure the uniformity of products' quality. Since previous trajectory planning using the path parametric method, which can hardly meet the requirement, a path velocity assignment method which is illuminated by the trapezoidal trajectory is proposed.

The path acceleration corresponding to the trapezoidal velocity profile is shown in Fig. 1. However, the acceleration and jerk discontinuity will cause a significant impact on the equipment life. It is consequently considered to add transition functions at the discontinuous points to modify the trajectory of the path acceleration. According to [22], [23], trigonometric functions are taken into account and the analytical expression of the improved trajectory (see Fig. 2) is exhibited as follows:

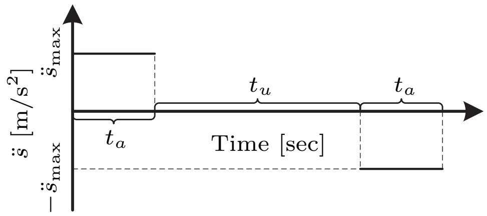

Fig. 1. Acceleration of the trapezoidal trajectory.

$$\ddot{s}=\begin{cases}\dfrac{\ddot{s}_{\max}}{2}\left[1-\cos\left(\dfrac{t}{t_b}\pi\right)\right] & 0\le t<t_1,\\ \ddot{s}_{\max} & t_1\le t<t_2,\\ \dfrac{\ddot{s}_{\max}}{2}\left[1-\cos\left(\dfrac{t-t_a}{t_b}\pi\right)\right] & t_2\le t<t_3,\\ 0 & t_3\le t<t_4,\\ -\dfrac{\ddot{s}_{\max}}{2}\left[1-\cos\left(\dfrac{t-t_a-t_u}{t_b}\pi\right)\right] & t_4\le t<t_5,\\ -\ddot{s}_{\max} & t_5\le t<t_6,\\ -\dfrac{\ddot{s}_{\max}}{2}\left[1-\cos\left(\dfrac{t-2t_a-t_u}{t_b}\pi\right)\right] & t_6\le t\le t_7,\end{cases} \tag{6}$$

where t_a, t_b, and t_u are time durations of the constant acceleration, uniform velocity, and the transition section, respectively, t_i $(i=1,2,\cdots,7)$ represents the time switching point of the trajectory, and $\ddot{s}_{\max}$ is the maximum acceleration along the path. The third-order derivative of s with respect to time, which denotes the path jerk, can be derived to be continuous according to (6).

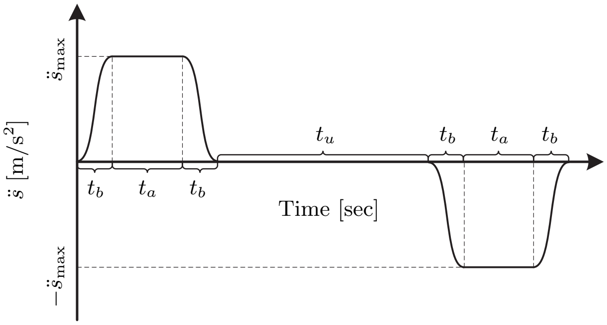

Fig. 2. Improved trajectory transited with trigonometric functions.

As both $\ddot{s}$ and $\dddot{s}$ are continuous, the improved trajectory is used for path interpolation. The total motion time of the robot is taken as t_f, which satisfies the following equation:

$$t_f=t_u+2t_a+4t_b. \tag{7}$$

Then, the overall length of the specified path, denoted as s_e, can be calculated as

$$\begin{aligned} s_e &= \int_0^{t_f} \dot{s}\ dt = \int_0^{t_f} \left(\int_0^{t_f} \ddot{s}\ dt \right) dt \\ &= \ddot{s}_{\max}(t_a + t_b)(t_a + 2t_b + t_u). \end{aligned}$$

It is obvious that the total length is known as long as the path is completely specified, and then a variety of trajectory expressions can be obtained by reasonably selecting t_a, t_b, and t_u. Thus, the joint positions, velocities, and accelerations can be derived according to (2), (3), (4), and (5), respectively. After that, joint torques can be further calculated in the light of (1). For the safe and stable operation of the manipulator, the following hardware constraints are imposed on joint actuators:

$$\begin{cases} |\tau_i| \leq \tau_{i\max}, \\ |\dot{q}_i| \leq \dot{q}_{i\max}, \\ |\ddot{q}_i| \leq \ddot{q}_{i\max}, \end{cases} \tag{8}$$

where $i = 1, 2, \cdots, n$ (n denotes the number of robot joints), $\tau_{i\max}$, $\dot{q}_{i\max}$, and $\ddot{q}_{i\max} \in \mathbb{R}^+$ are the maximum torque, velocity, and acceleration of each joint, respectively.

For quantitatively evaluating the performance of the robot, appropriate indicators should be introduced. According to [1], [5], for the time-optimal problem only and energy-optimal problem only, the cost functions could be separately chosen as follows:

$$\begin{aligned} J_t &= \int_0^{t_f} dt, \\ J_e &= \int_0^{t_f} \boldsymbol{\tau}^\top \boldsymbol{\tau} dt, \end{aligned}$$

where J_t and J_e denote the total motion time and energy consumption of the manipulator performing the assigned task, respectively. With the optimality of time and energy being taken into consideration simultaneously, we choose an indicator as weighted J_t and J_e as follows:

$$J = w_1 J_t + w_2 J_e, \tag{9}$$

where w_1 and $w_2 \in \mathbb{R}^+$ are for ensuring J_t and J_e in the same order of magnitude.

In general, the trajectory optimization problem considering time and energy optimality simultaneously can be described as follows:

$$\begin{cases} \ddot{s} \text{ chosen as (6)}, \\ \min\limits_{t_a, t_b, t_u} J, \\ \text{s.t.} \begin{cases} |\tau_i| \leq \tau_{i\max}, \\ |\dot{q}_i| \leq \dot{q}_{i\max}, \\ |\ddot{q}_i| \leq \ddot{q}_{i\max}, \end{cases} \end{cases} \tag{10}$$

where $i = 1, 2, \cdots, n$.

B. Problem Solution

As mentioned above, t_a, t_b, and t_u are the parameters to be optimized, once $\ddot{s}$ is expressed as (6). Due to the high speed of convergence, strong capability of global searching, and convenient implementation, the PSO [24] algorithm is chosen to optimize the trajectory parameters for this paper.

On the basis of the PSO algorithm, every single possible solution is regarded as a particle. For a particle swarm with the scale of m and the number of iterations of N, the velocity and position of a certain particle are updated as follows:

$$\begin{aligned} v_i^j =& w \cdot v_i^{j-1} + c_1 \cdot r_1 \cdot \left(X_{bi}^{j-1} - x_i^{j-1} \right) \\ &+ c_2 \cdot r_2 \cdot \left(X_g^{j-1} - x_i^{j-1} \right), \\ x_i^j =& x_i^{j-1} + v_i^j \cdot p_t, \end{aligned}$$

where $i = 1, 2, \cdots, m$, $j = i, 2, \cdots, N$, v_i^j and x_i^j represent the velocity and position of the ith particle in the jth iteration, separately, X_{bi}^{j-1} and X_g^{j-1} denote the historically optimal position of the ith particle and the globally optimal position of the population over the past j–1 iterations, respectively, w, c_1, and $c_2 \in \mathbb{R}$ indicate the velocity inertia coefficient, individual factor, and social factor, respectively, r_1 and $r_2 \in \mathbb{R}^+$ are random numbers greater than zero and smaller than one, and p_t indicates the pseudo time interval. Specially, $j = 0$ means the serial number of the initial generation of population.

According to (7), the parameters to be optimized of the problem (10) can be transformed into t_f, t_u, and t_a, which indicates that the position of a particle can be expressed as

$$x_i = [t_f, t_u, t_a],$$

where $i = 1, 2, \cdots, m$. The trajectory parameters are constrained as follows:

$$\begin{aligned} &0 < t_f \leq t_{\max}, \\ &\alpha_1 t_f < t_u < \alpha_2 t_f, \\ &0 < t_a < (t_f - t_u)/2, \end{aligned}$$

where $t_{\max} \in \mathbb{R}^+$ is the maximum acceptable motion time, α_1 and $\alpha_2 \in \mathbb{R}^+$ that satisfying $0 < \alpha_1 < \alpha_2 < 1$ represent the proportion of the constant velocity section.

In addition, the penalty function method [25], [26] is adopted to deal with the constraints (8). Firstly, the inequality group (8) is converted into the following form:

$$h_k(x_i) = \begin{cases} |\tau_j| - \tau_{j\max} & k = 1, 2, \cdots, n, \\ |\dot{q}_j| - \dot{q}_{j\max} & k = n+1, \cdots, 2n, \\ |\ddot{q}_j| - \ddot{q}_{j\max} & k = 2n+1, \cdots, 3n, \end{cases}$$

where $i = 1, 2, \cdots, m$, $j = 1, 2, \cdots, n$, and $h_k(\cdot)$ represents the value of the kth inequality. Secondly, the penalty item [26] is selected as

$$P(x_i) = \sigma \sum_{k=1}^{3n} W_k p_k(x_i), \tag{11}$$

where $\sigma \in \mathbb{R}^+$ denotes the penalty factor, $W_k = \sum_{i=1}^{m} p_k(x_i) / \sum_{k=1}^{3n} \sum_{i=1}^{m} p_k(x_i)$ is for normalizing different types of constraints, and $p_k(x_i) = \max\{0, h_k(x_i)\}$. The smaller W_k is, the fewer constraints are violated. In particular, W_k is equal to zero when all constraints are not violated. Finally, the penalty function can be expressed as follows:

$$F(x_i) = J(x_i) + P(x_i), \tag{12}$$

where $J(x_i)$ indicates the value of (9) corresponding to the particle x_i $(i = 1, 2, \cdots, m)$. The penalty function transforms the problem (10) into an unconstrained optimization problem and the fitness value of each particle can be judged accordingly.

Based on the PSO algorithm [24], [25], [27], the solution to the trajectory optimization problem is acquired as X_g^N through N iterations of the particle swarm.

III. A 6-DOF MANIPULATOR EXAMPLE

This section provides an implementation example of the proposed method based on a 6-DOF serial manipulator. The mass and D-H parameters of the robot links are shown in TABLE I, where α_{i-1}, a_{i-1}, q_i, d_i, and m_i indicate the twist, length, joint angle, offset and mass of link i, respectively. The hardware constraints and viscous friction coefficients of each joint are shown in TABLE II.

TABLE I
D-H AND MASS PARAMETERS OF THE MANIPULATOR

link i	α_{i-1} (rad)	a_{i-1} (m)	q_i (rad)	d_i (m)	m_i (kg)
1	0	0	q_1	0	1.5
2	$-\pi/2$	0.0510	q_2	0	3.5
3	0	0.2950	q_3	0	1.5
4	$-\pi/2$	0.0973	q_4	0.3435	3.0
5	$\pi/2$	0	q_5	0	2.5
6	$-\pi/2$	0	q_6	0	0.5

TABLE II
CONSTRAINTS AND VISCOUS FRICTION COEFFICIENTS OF EACH JOINT

joint i	$\dot{q}_{i\max}$ (rad/s)	$\ddot{q}_{i\max}$ (rad/s^2)	$\tau_{i\max}$ (N·m)	μ_i
1	1	1.5	10	10
2	1	1.5	65	8
3	1	1.5	45	6
4	1	1.5	25	5
5	1	1.5	10	2
6	1	1.5	5	1

The specified moving path (see Fig. 3) of the end effector is a rectangle with four corners transiting with inscribed circular arcs, where r is the radius of arcs, $\boldsymbol{P}_i$ $(i = 1, 2, 3, 4)$ represents the coordinates of the rectangle's corner, $\boldsymbol{P}_{i1}$ and $\boldsymbol{P}_{i2}$ $(i = 1, 2, 3, 4)$ represent the points where the lines are tangent to the arcs, and s_j $(j = 1, 2, \cdots, 8)$ denotes the length of each path segment. Some key parameters mentioned above are selected as follows:

$\boldsymbol{P}_1 = [0.2, -0.1, -0.05]^\top$ m, $\boldsymbol{P}_2 = [0.5, -0.1, -0.05]^\top$ m,
$\boldsymbol{P}_3 = [0.5, 0.1, -0.05]^\top$ m, $\boldsymbol{P}_4 = [0.2, 0.1, -0.05]^\top$ m,

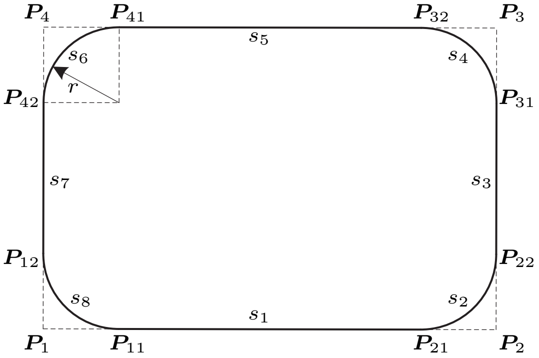

Fig. 3. Specified path of the end effector.

and then other parameters can be calculated accordingly.

In order to derive the parameterized expression of the path, the parameterization of a path segment is given. Firstly, for a straight-line path with the length of s_l, starting at $\boldsymbol{P}_s$, and ending at $\boldsymbol{P}_f$, it can be parameterized as

$$\boldsymbol{P} = \boldsymbol{P}_s + \frac{s}{s_l}(\boldsymbol{P}_f - \boldsymbol{P}_s), \tag{13}$$

where s indicates the distance from $\boldsymbol{P}_s$ to $\boldsymbol{P}$. Secondly, for a circular path with the center of $\boldsymbol{C}$, radius of r, starting at $\boldsymbol{P}_s$, and ending at $\boldsymbol{P}_f$, it can be parameterized as

$$\boldsymbol{P} = T \cdot \left[r\cos\frac{s}{r}, r\sin\frac{s}{r}, 0 \right]^\top, \tag{14}$$

where s indicates the distance from $\boldsymbol{P}_s$ to $\boldsymbol{P}$ and T is the transformation of the arc's UVW coordinate system relative to the robot's reference coordinate system. The UVW coordinate system is established as follows: the origin is set to $\boldsymbol{C}$; the U axis directs from $\boldsymbol{C}$ to $\boldsymbol{P}_s$; the W axis is perpendicular to the plane of $\boldsymbol{C}$, $\boldsymbol{P}_s$, and $\boldsymbol{P}_f$; the V axis is obtained based on the right-hand rule.

The end effector moves along the path (see Fig. 3) in the following order:

$$\boldsymbol{P}_{11} \rightarrow \boldsymbol{P}_{21} \rightarrow \boldsymbol{P}_{22} \rightarrow \boldsymbol{P}_{31} \rightarrow \boldsymbol{P}_{32} \rightarrow \boldsymbol{P}_{41} \rightarrow \boldsymbol{P}_{42} \rightarrow \boldsymbol{P}_{12} \rightarrow \boldsymbol{P}_{11}.$$

According to (13) and (14), the path can be parameterized as follows:

1) $0 \leq s < s_1$,

$$\boldsymbol{P} = \boldsymbol{P}_{11} + \frac{s}{s_1}(\boldsymbol{P}_{21} - \boldsymbol{P}_{11});$$

2) $s_1 \leq s < \sum\limits_{j=1}^{2} s_i$,

$$\boldsymbol{P} = T_1 \cdot \left[r\cos\left(\frac{s - s_1}{r}\right), r\sin\left(\frac{s - s_1}{r}\right), 0 \right]^\top;$$

3) $\sum\limits_{j=1}^{2} s_i \leq s < \sum\limits_{j=1}^{3} s_i$,

$$\boldsymbol{P} = \boldsymbol{P}_{22} + \frac{s - \sum\limits_{j=1}^{2} s_i}{s_3}(\boldsymbol{P}_{31} - \boldsymbol{P}_{22});$$

4) $\sum_{j=1}^{3} s_i \le s < \sum_{j=1}^{4} s_i,$

$$\boldsymbol{P} = T_2 \cdot \left[r\cos\left(\frac{s-\sum\limits_{j=1}^{3} s_i}{r}\right), r\sin\left(\frac{s-\sum\limits_{j=1}^{3} s_i}{r}\right), 0 \right]^\top;$$

5) $\sum_{j=1}^{4} s_i \le s < \sum_{j=1}^{5} s_i,$

$$\boldsymbol{P} = \boldsymbol{P}_{32} + \frac{s-\sum\limits_{j=1}^{4} s_i}{s_5}(\boldsymbol{P}_{41} - \boldsymbol{P}_{32});$$

6) $\sum_{j=1}^{5} s_i \le s < \sum_{j=1}^{6} s_i,$

$$\boldsymbol{P} = T_3 \cdot \left[r\cos\left(\frac{s-\sum\limits_{j=1}^{5} s_i}{r}\right), r\sin\left(\frac{s-\sum\limits_{j=1}^{5} s_i}{r}\right), 0 \right]^\top;$$

7) $\sum_{j=1}^{6} s_i \le s < \sum_{j=1}^{7} s_i,$

$$\boldsymbol{P} = \boldsymbol{P}_{42} + \frac{s-\sum\limits_{j=1}^{6} s_i}{s_7}(\boldsymbol{P}_{12} - \boldsymbol{P}_{42});$$

8) $\sum_{j=1}^{7} s_i \le s \le \sum_{j=1}^{8} s_i,$

$$\boldsymbol{P} = T_4 \cdot \left[r\cos\left(\frac{s-\sum\limits_{j=1}^{7} s_i}{r}\right), r\sin\left(\frac{s-\sum\limits_{j=1}^{7} s_i}{r}\right), 0 \right]^\top.$$

After that, the interpolation and optimization methods mentioned in Section II can be applied to generate optimal trajectory profiles.

IV. Simulation Results

In this section, we verify the performance of the proposed trajectory optimization method through simulation based on the 6-DOF manipulator discussed in Section III. TABLE III lists the parameters of the performance criteria and PSO algorithm, as described in Section II. The simulation is run three times to verify the convergence of the algorithm. Fig. 4 shows the fitness value of the globally optimal particle in each iteration of the three simulations. TABLE IV lists the optimal trajectory parameters, fitness value, and penalty value of each simulation. It is found that the optimal particles of the three simulations are almost the same and the fitness values of the globally optimal particles all converge to 16.249, which indicates that the PSO algorithm can effectively converge and obtain the best solution. Therefore, both time and energy optimality are achieved.

TABLE III
Parameters of the PSO algorithm

Parameters	Value	Unit
w_1	1	s^{-1}
w_2	2×10^{-4}	$\mathrm{N}^{-2}\cdot\mathrm{m}^{-2}\cdot\mathrm{s}^{-1}$
σ	1	–
m	50	–
N	100	–
w	0.7298	–
c_1	1.4962	–
c_2	1.4962	–
p_t	0.55	–
$t_{\max}$	20	s

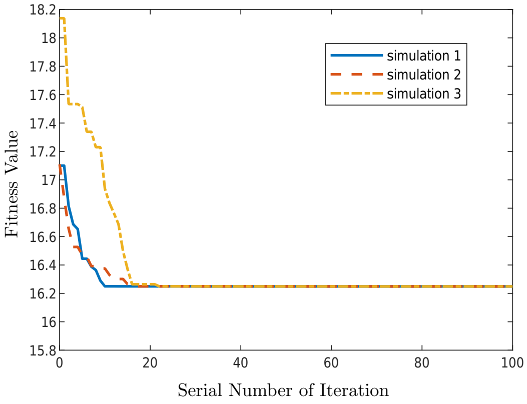

Fig. 4. Fitness of X_g^j $(j = 0, 1, \cdots, N)$.

TABLE IV
Optimal trajectory parameters, fitness value, and penalty value of each simulation

Simulation i	t_f (s)	t_u (s)	t_a (s)	fitness	penalty
1	10.498	8.394	0.585	16.249	0
2	10.498	8.394	0.523	16.249	0
3	10.498	8.394	0.597	16.249	0

Finally, the optimal trajectory parameters are acquired as the average of the three simulation results, which are calculated as follows:

$$t_f = 10.498 \text{ s},\ t_u = 8.394 \text{ s},\ t_a = 0.568 \text{ s}. \tag{15}$$

And t_b is calculated according to (7). The fitness value and penalty value of (15) can be acquired as 16.249 and 0, respectively, according to (12) and (11). The corresponding states of the manipulator are depicted in Figs. 5, 6, and 7. In Fig. 5, the red dashed lines represent the boundary conditions. It is found that all constraints are not violated and fitness converges to the minimum value while the parameters of (6) are selected as (15).

In Fig. 8, the red dashed lines, the blue solid lines, the black star, and the yellow arrows represent the predetermined path (see Fig. 3), the calculated moving path (while all joints move according to the trajectories shown in Figs. 5, 6, and 7), the

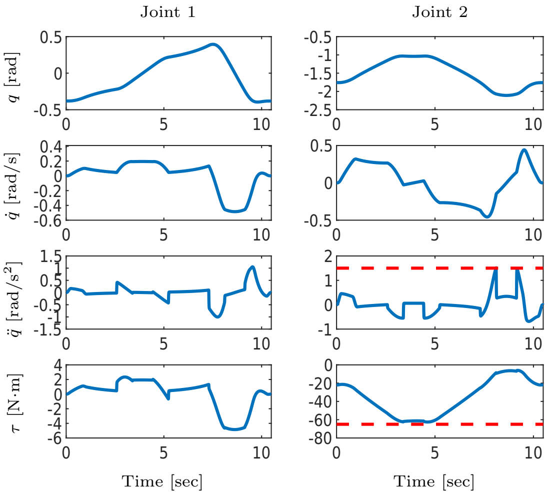

Fig. 5. States of the joint 1 and joint 2.

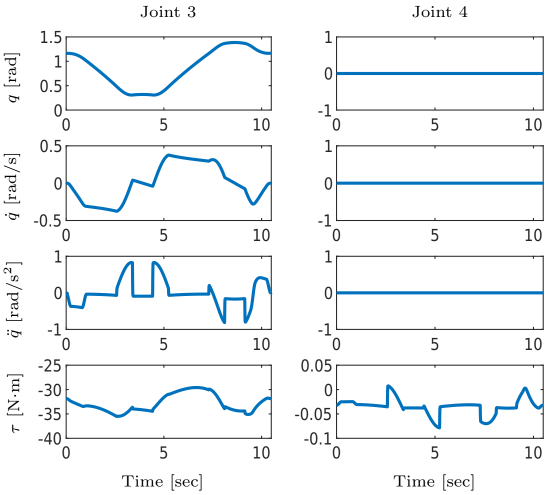

Fig. 6. States of the joint 3 and joint 4.

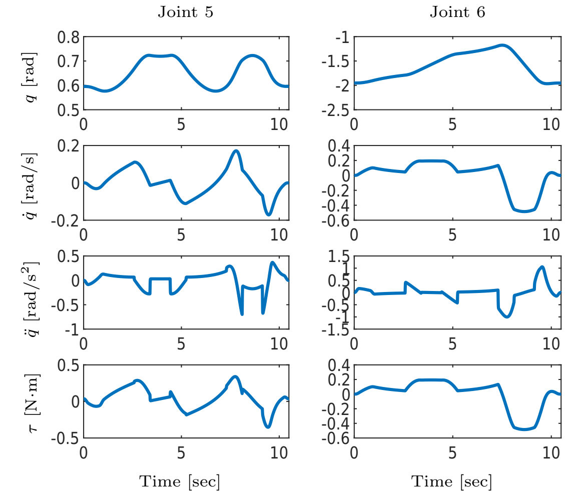

Fig. 7. States of the joint 5 and joint 6.

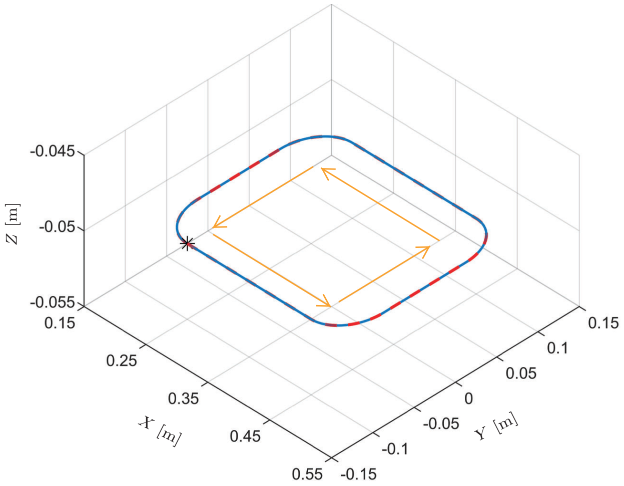

Fig. 8. Moving path of the end effector.

starting point, and the moving directions of the end effector, respectively, which proves that the planned trajectories can properly drive the robot's movement.

In general, from the simulation results, it is concluded that the proposed method can effectively generate time-energy optimal trajectories for robots moving along predefined paths to possess a wide range of uniform velocity without violating constraints.

V. CONCLUSION

In this paper, for allocating a wide range of uniform speed segment and minimizing the cost function for robots moving along specified paths, a time-energy optimal trajectory planning technique based on path parameterization and the PSO algorithm has been given. The proposed method assigns pseudo acceleration along the path in segments and finds the best trajectory parameters in limited steps. Finally, the effectiveness of the method is verified through simulation results of a 6-DOF serial manipulator.

REFERENCES

[1] J. Gregory, A. Olivares, and E. Staffetti, Energy-optimal trajectory planning for robot manipulators with holonomic constraints, *Systems & Control Letters*, vol. 61, no. 2, pp. 279-291, 2012.

[2] J. E. Bobrow, S. Dubowsky, and J. S. Gibson, Time-optimal control of robotic manipulators along specified paths, *The International Journal of Robotics Research*, vol. 4, no. 3, pp. 3-17, 1985.

[3] K. G. Shin and N. D. McKay, Minimum-time control of robotic manipulators with geometric path constraints, *IEEE Transactions on Automatic Control*, vol. 30, no. 6, pp. 531-541, 1985.

[4] J. -J. E. Slotine and H. S. Yang, Improving the efficiency of time-optimal path-following algorithms, in *Proceedings of the 1988 American Control Conference (ACC)*, Atlanta, GA, USA, pp. 2129-2134, 1988.

[5] Z. Shiller, On singular time-optimal control along specified paths, *IEEE Transactions on Robotics and Automation*, vol. 10, no. 4, pp. 561-566, 1994.

[6] K. G. Shin and N. D. McKay, A dynamic programming approach to trajectory planning of robotic manipulators, *IEEE Transactions on Automatic Control*, vol. 31, no. 6, pp. 491-500, 1986.

[7] T. Zhang, M. H. Zhang, and Y. B. Zou, Time-optimal and smooth trajectory planning for robot manipulators, *Interntional Journal of Control, Automation and Systems*, vol. 19, no. 1, pp. 521-531, 2021.

[8] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl, Time-optimal path tracking for robots: A convex optimization approach, *IEEE Transactions on Automatic Control*, vol. 54, no. 10, pp. 2318-2327, 2009.

[9] A. K. Singh and K. M. Krishna, A class of non-linear time scaling functions for smooth time optimal control along specified paths, in *Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, Germany, pp. 5809-5816, 2015.

[10] P. Reynoso-Mora, W. Chen, and M. Tomizuka, A convex relaxation for the time-optimal trajectory planning of robotic manipulators along predetermined geometric paths, *Optimal Control Applications and Methods*, vol. 37, no. 6, pp. 1263-1281, 2016.

[11] K. Hauser, Fast interpolation and time-optimization with contact, *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1231-1250, 2014.

[12] P. Y. Shen, X. B. Zhang, and Y. C. Fang, Complete and time-optimal path-constrained trajectory planning with torque and velocity constraints: Theory and alpplications, *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 2, pp. 735-746, 2018.

[13] M. X. Yuan, B. Yao, D. D. Gao, X. C. Zhu, and Q. F. Wang, A novel algorithm for time optimal trajectory planning, in *Proceedings of the ASME 2014 Dynamic Systems and Control Conference (DSCC)*, San Antonio, TX, USA, vol. 1, pp. V001T02A003, 2014.

[14] D. Constantinescu and E. A. Croft, Smooth and time-optimal trajectory planning for industrial manipulators along specified paths, *Journal of Robotic Systems*, vol. 17, no. 5, pp. 233-249, 2000.

[15] P. Tokekar, N. Karnad, and V. Isler, Energy-optimal velocity profiles for car-like robots, in *Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, pp. 1457-1462, 2011.

[16] N. K. Yang, D. Chang, M. Johnson-Roberson, and J. Sun, Robust energy-optimal path following control for autonomous underwater vehicles in ocean currents, in *Proceedings of the 2020 American Control Conference (ACC)*, Denver, CO, USA, pp. 5119-5124 , 2020.

[17] Y. H. Wang, B. Ning, F. Cao, B. De Schutter, and T. J. J. van den Boom, A survey on optimal trajectory planning for train operations, in *Proceedings of the 2011 IEEE International Conference on Service Operations, Logistics and Informatics (SOLI)*, pp. 589-594, 2011.

[18] Z. Shiller, Time-energy optimal control of articulated systems with geometric path constraints, *Journal of Dynamic Systems, Measurement, and Control*, vol. 118, no. 1, pp. 139-143, 1996.

[19] J. B. Aldrich and R. E. Skelton, Time-energy optimal control of hyper-actuated mechanical systems with geometric path constraints, in *Proceedings of the 44th IEEE Conference on Decision and Control (CDC)*, Seville, Spain, pp. 8246-8253, 2005.

[20] M. Bamdad, Time-energy optimal trajectory planning of cable-suspended manipulators, *Cable-Driven Parallel Robots*, Springer, pp. 41-51, 2013.

[21] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl, Time-energy optimal path tracking for robots: A numerically efficient optimization approach, in *Proceeding of the 2008 IEEE International Workshop on Advanced Motion Control (AMC)*, Trento, Italy, pp. 727-732, 2008.

[22] N. Sun, Y. Fang, X. Zhang, and Y. Yuan, Transportation task-oriented trajectory planning for underactuated overhead cranes using geometric analysis, *IET Control Theory & Applications*, vol. 6, no. 10, pp. 1410-1423, 2012.

[23] Y. Fang, J. Hu, W. H. Liu, Q. Q. Shao, J. Qi, and Y. H. Peng, Smooth and time-optimal S-curve trajectory planning for automated robots and machines, *Mechanism and Machine Theory*, vol. 137, pp. 127-153, 2019.

[24] J. Kennedy and R. Eberhart, Particle swarm optimization, in *Proceedings of the 1995 International Conference on Neural Networks (ICNN)*, Perth, WA, Australia, vol. 4, pp. 1942-1948, 1995.

[25] K. E. Parsopoulos and M. N. Vrahatis, Particle swarm optimization method for constrained optimization problems, *Intelligent technologies-theory and application: New trends in intelligent technologies*, vol. 76, no. 1, pp. 214-220, 2002.

[26] Ö. Yeniay, Penalty function methods for constrained optimization with genetic algorithms, *Mathematical and Computational Applications*, vol. 10, no. 1, pp. 45-56, 2005.

[27] X. H. Hu and R. Eberhart, Solving constrained nonlinear optimization problems with particle swarm optimization, in *Proceedings of the 6th World Multiconference on Systemics, Cybernetics and Informatics*, Orlando, FL, USA, vol. 5, pp. 203-206, 2002.