Asymmetric S-curve Motion Planning for Industrial Manipulators Based on Evolutionary Computation Algorithm

Yen-Ju Lin, and Ming-Yang Cheng, *Member, RST*

***Abstract*—Motion planning is an important factor affecting the efficiency of industrial robot manipulators. This paper develops an asymmetric 7-segment S-curve motion planning architecture for enabling industrial robot manipulators to complete tasks in the shortest time possible while still satisfying physical constraints. If S-curve motion planning is performed in the Cartesian space, the moving path of the industrial robot manipulator can be specified. However, because of the nonlinear mapping between the Cartesian space and the joint space, physical constraints in the joint space may be violated. To cope with the aforementioned problem, this paper proposes an approach that exploits evolutionary computation algorithms to determine the parameters of the asymmetric 7-segment S-curve so as to achieve time-optimized motion planning without violating the physical constraints in both Cartesian space and joint space. The results of simulations and experiments conducted on a 6-DOF industrial manipulator indicate that the proposed approach is feasible.**

***Index Terms*—Asymmetric S-curve, Motion Planning, Industrial Manipulator, Evolutionary Computation Algorithm**

I. Introduction

Riding the wave of Industry 4.0 while facing the challenge of ever-increasing labor costs and the impact of the COVID-19 pandemic, the trend of industrial manufacturing/production has shifted towards automation and customization. In order to meet the needs of customization as well as small-volume and large-variety production, the multi-axis industrial robot manipulator is one of the best solutions and is now widely used in various fields due to its high degree of freedom. For an industrial robot manipulator to execute a specific task such as contour following, two of the most important goals are reduction of contour error and improvement of work efficiency. With a well-designed control scheme, the contour error can be reduced such that the contouring accuracy can be substantially improved. As for the improvement of work efficiency, appropriate motion planning is crucial. Ideal motion planning on the velocity command will not only satisfy all physical constraints such as kinematics constraints (i.e. joint velocity, joint acceleration, joint jerk) and dynamics constraints (i.e. joint torque), but also achieve the goal of time optimization. In general, motion planning can be performed either in the joint space or in the Cartesian space. One of the advantages of motion planning on the joint velocity profile is that it is straightforward and easy to implement, as the servomotors are installed at the joints. Another advantage is that it is free of kinematics singularity. Nevertheless, it also has a major drawback. Since the mapping between the joint space and the Cartesian space is highly nonlinear, the original work path without motion planning in the joint space may be very different from the one having motion planning in the joint space. As a result, motion planning in the joint space is not suitable for contour following tasks. On the other hand, for motion planning in the Cartesian space, given a feedrate profile under motion planning, an interpolator is developed to generate motion commands in the Cartesian space. The motion commands in the Cartesian space are then converted into commands in the joint space to check whether or not the kinematics and dynamics constraints are satisfied. However, because of the nonlinear mapping between the Cartesian space and the joint space, kinematics or dynamics constraints in the joint space may be violated and a kinematics singularity may also occur. To cope with the aforementioned difficulty, this paper focuses on motion planning problems in the Cartesian space for industrial robot manipulators. There have been many past studies on the motion planning problem in the Cartesian space. In [1], the velocity profile is represented by 5th-order quantic splines. A heuristic approach is employed to gradually increase the velocity while satisfying the jounce constraint. Boryga and Graboś used 5th-order, 7th-order and 9th-order polynomials to construct velocity profiles such that the jerks for the start point and the end point of the moving path are zero so as to reduce vibration and improve positioning accuracy [2]. In [3], B-splines are used to construct the velocity profile; the binary search method is used to raise the control points one by one until the pre-determined maximum searching time is reached. In [4], the tool path is represented in NURBS form. The feedrate profile is represented by a cubic spline, which is a function of the parameters of the NURBS curve. Optimization algorithms are used to find the shortest moving time. In [5], the tool path is represented by a NURBS curve, while the velocity profile is constructed by a sine curve. The critical point and the breakpoint on the tool path are found by scanning the tool path in advance. As such, the velocities at the critical point and the breakpoint will be further adjusted. In [6], the tool path is represented by a NURBS curve; a bi-directional scan approach is employed to search for the physical limit of the critical point, and the S-curve motion planning for the feedrate profile between two critical points is then performed. The results in [7] reveal that the velocity profile represented by a 3rd-order S-curve performs better under vibration than the velocity profile represented by a trapezoidal curve. In [8], a square-wave jerk curve is used to construct a 3rd-order S-curve. A look-ahead

This work was supported in part by the Ministry of Science and Technology of Taiwan, under Grant MOST 108-2221-E-006-217-MY2.

Yen-Ju Lin was with the Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan; e-mail: yenju417@gmail.com).

Ming-Yang Cheng is with the Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan. (corresponding author: 886-6-2757575-62396; e-mail: mycheng@mail.ncku.edu.tw).

algorithm is employed to lessen the velocity jump problem between two adjacent linear segments so as to maintain a constant feedrate throughout the entire task. In [9] and [10], a square-wave jerk curve is used to construct an asymmetric 3rd-order S-curve, in which the time durations for the acceleration zone and the deceleration zone differ. In [11], a square-wave jerk curve is used to construct an asymmetric 3rd-order S-curve motion planning algorithm for 5-axis machine tools, in which the time durations for the positive jerk zone and the negative jerk zone differ. Compared with the asymmetric 3rd-order S-curve adopted in [9] and [10], the one adopted in [11] is more asymmetric and more flexible for motion planning. In addition, the time optimal solution is obtained using an optimization algorithm.

Similar to the idea adopted in [11], the time durations for the positive jerk zone and the negative jerk zone are different in the motion planning algorithm proposed in this paper. In addition, two zero-jerk sections have been added. As a result, the asymmetric S-curve for motion planning proposed in this paper has seven segments. With the characteristic of asymmetry, the velocity profiles can be planned flexibly. In addition, this paper proposes an approach that employs the Particle Swarm Optimization (PSO) algorithm to search for the optimal parameters of the asymmetric 7-segment S-curves that will yield a solution to satisfy both Cartesian space and joint space physical constraints while being near time-optimal. Finally, a 6-DOF industrial manipulator is used as an experimental platform, in which simulations and experiments are carried out to verify the effectiveness of the proposed approach. Simulations and experimental results have indicated that the asymmetric 7-segment S-curve motion planning approach developed in this paper performs better in machining time compared with other S-curve-based approaches.

The remainder of this paper is organized as follows. Section II gives a brief view on S-curve acceleration/deceleration and particle swarm optimization. Section III introduces the proposed asymmetric 7-segment S-curve motion planning approach. Section 4 provides the simulation and experimental results. Section 5 is the conclusion.

II. Brief Review on S-curve Acceleration/Deceleration and Particle Swarm Optimization

This section will provide a brief review on S-curve ACC/DEC (acceleration/deceleration). In this paper, the Particle Swarm Optimization (PSO) algorithm is used to search for the optimal values of the parameters of the S-curve ACC/DEC architecture. Hence, a brief review on PSO will be also given in this section.

A. S-curve Acceleration/Deceleration (ACC/DEC)

S-curve ACC/DEC is one of the most common ACC/DEC planning methods. Its main advantage is that it has a continuous acceleration and deceleration curve which can prevent the servomechanism from encountering sudden acceleration and deceleration changes, while its velocity profile is smoother, as well. S-curve ACC/DEC motion planning can be divided into three parts: the acceleration zone, constant speed zone and deceleration zone. In order to simplify the calculations, the S-curve is usually symmetrical; that is, the time durations of the acceleration zone and the deceleration zone are the same. However, limitations in symmetrical properties may lead to lengthy machining time. In order to achieve the goal of time-optimal operation, this paper derives an asymmetric 7-segment S-curve, as shown in Fig. 1. Compared with the symmetrical S-curve, the asymmetric 7-segment S-curve has several advantages. One of the advantages is that the time durations for the acceleration zone and the deceleration zone can be planned flexibly. Another advantage is that the asymmetric 7-segment S-curve can achieve a higher velocity in a shorter time duration.

According to the acceleration profile shown in Fig. 1, the acceleration command can be expressed as:

$$a(t)=\begin{cases} J_1 t & , \quad 0\le t\le K_1 \\ J_1 K_1 & , \quad K_1\le t\le K_2 \\ J_1K_1 - J_3(t-K_2) & , \quad K_2\le t\le K_3 \\ 0 & , \quad K_3\le t\le K_4 \\ -J_5(t-K_4) & , \quad K_4\le t\le K_5 \\ -J_5(K_5-K_4) & , \quad K_5\le t\le K_6 \\ -J_5(K_5-K_4)+J_7(t-K_6) & , \quad K_6\le t\le K_7 \end{cases} \tag{1}$$

where $K_i=\sum_{n=1}^{i} T_n$.

By integrating $a(t)$ with respect to time, the velocity command can be expressed as:

$$v(t)=\begin{cases} \frac{1}{2}J_1t^2 & ,0\le t\le K_1 \\ J_1K_1t-\frac{1}{2}J_1K_1^2 & ,K_1\le t\le K_2 \\ -\frac{1}{2}J_1K_1^2+J_1K_1t-\frac{1}{2}J_3K_2^2+J_3K_2t-\frac{1}{2}J_3t^2 & ,K_2\le t\le K_3 \\ -\frac{1}{2}J_1K_1^2+J_1K_1K_3-\frac{1}{2}J_3K_2^2+J_3K_2K_3-\frac{1}{2}J_3K_3^2 & ,K_3\le t\le K_4 \\ -\frac{1}{2}J_1K_1^2+J_1K_1K_3-\frac{1}{2}J_3K_2^2+J_3K_2K_3-\frac{1}{2}J_3K_3^2 & \\ \quad -\frac{1}{2}J_5K_4^2+J_5K_4t-\frac{1}{2}J_5t^2 & ,K_4\le t\le K_5 \\ -\frac{1}{2}J_1K_1^2+J_1K_1K_3-\frac{1}{2}J_3K_2^2+J_3K_2K_3-\frac{1}{2}J_3K_3^2 & \\ \quad -\frac{1}{2}J_5K_4^2+J_5K_4t+\frac{1}{2}J_5K_5^2-J_5K_5t & ,K_5\le t\le K_6 \\ -\frac{1}{2}J_1K_1^2+J_1K_1K_3-\frac{1}{2}J_3K_2^2+J_3K_2K_3 & \\ \quad -\frac{1}{2}J_3K_3^2-\frac{1}{2}J_5K_4^2+J_5K_4t+\frac{1}{2}J_5K_5^2 & ,K_6\le t\le K_7 \\ \quad -J_5K_5t+\frac{1}{2}J_7K_6^2-J_7K_6t+\frac{1}{2}J_7t^2 & \end{cases} \tag{2}$$

By integrating $v(t)$ with respect to time, the position command can be expressed as:

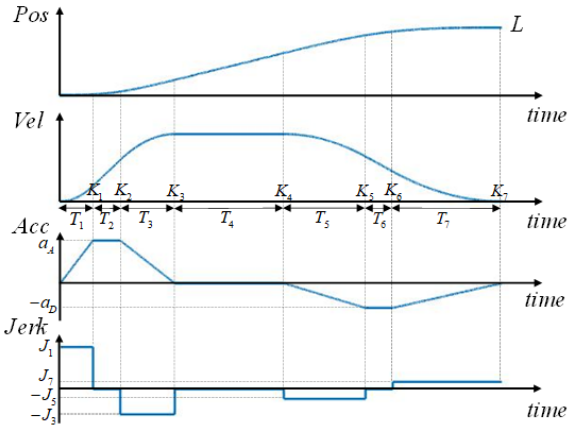

Fig. 1. Asymmetric 7-segment S-curve

$$p(t)=\begin{cases}\frac{1}{6}J_1t^3 & ,0\le t\le K_1\\ \frac{1}{6}J_1K_1^3-\frac{1}{2}J_1K_1^2t+\frac{1}{2}J_1K_1t^2 & ,K_1\le t\le K_2\\ \frac{1}{6}J_1K_1^3-\frac{1}{2}J_1K_1^2t+\frac{1}{2}J_1K_1t^2+\frac{1}{6}J_3K_2^3-\frac{1}{2}J_3K_2^2t & \\ \quad+\frac{1}{2}J_3K_2t^2-\frac{1}{6}J_3t^3 & ,K_2\le t\le K_3\\ \frac{1}{6}J_1K_1^3-\frac{1}{2}J_1K_1^2t-\frac{1}{2}J_1K_1K_3^2+J_1K_1K_3t+\frac{1}{6}J_3K_2^3 & \\ -\frac{1}{2}J_3K_2^2t-\frac{1}{2}J_3K_2K_3^2+J_3K_2K_3t+\frac{1}{3}J_3K_3^3-\frac{1}{2}J_3K_3^2t & ,K_3\le t\le K_4\\ \frac{1}{6}J_1K_1^3-\frac{1}{2}J_1K_1^2t-\frac{1}{2}J_1K_1K_3^2+J_1K_1K_3t+\frac{1}{6}J_3K_2^3 & \\ -\frac{1}{2}J_3K_2^2t-\frac{1}{2}J_3K_2K_3^2+J_3K_2K_3t+\frac{1}{3}J_3K_3^3-\frac{1}{2}J_3K_3^2t & ,K_4\le t\le K_5\\ \quad+\frac{1}{6}J_5K_4^3-\frac{1}{2}J_5K_4^2t+\frac{1}{2}J_5K_4t^2-\frac{1}{6}J_5t^3 & \\ \frac{1}{6}J_1K_1^3-\frac{1}{2}J_1K_1^2t-\frac{1}{2}J_1K_1K_3^2+J_1K_1K_3t+\frac{1}{6}J_3K_2^3 & \\ -\frac{1}{2}J_3K_2^2t-\frac{1}{2}J_3K_2K_3^2+J_3K_2K_3t+\frac{1}{3}J_3K_3^3-\frac{1}{2}J_3K_3^2t & ,K_5\le t\le K_6\\ +\frac{1}{6}J_5K_4^3-\frac{1}{2}J_5K_4^2t+\frac{1}{2}J_5K_4t^2-\frac{1}{6}J_5K_5^3+\frac{1}{2}J_5K_5^2t-\frac{1}{2}J_5K_5t^2 & \\ \frac{1}{6}J_1K_1^3-\frac{1}{2}J_1K_1^2t-\frac{1}{2}J_1K_1K_3^2+J_1K_1K_3t+\frac{1}{6}J_3K_2^3 & \\ -\frac{1}{2}J_3K_2^2t-\frac{1}{2}J_3K_2K_3^2+J_3K_2K_3t+\frac{1}{3}J_3K_3^3-\frac{1}{2}J_3K_3^2t & \\ +\frac{1}{6}J_5K_4^3-\frac{1}{2}J_5K_4^2t+\frac{1}{2}J_5K_4t^2-\frac{1}{6}J_5K_5^3+\frac{1}{2}J_5K_5^2t & ,K_6\le t\le K_7\\ -\frac{1}{2}J_5K_5t^2-\frac{1}{6}J_7K_6^3+\frac{1}{2}J_7K_6^2t-\frac{1}{2}J_7K_6t^2+\frac{1}{6}J_7t^3 & \end{cases} \tag{3}$$

B. Particle Swarm Optimization (PSO)

Particle swarm optimization is a method for the optimization of continuous nonlinear functions [12], [13]. At the outset, all particles will randomly search for a solution in the multi-dimensional solution space and obtain the fitness value of the solution. As the number of iterations increases, each particle will modify the direction for finding solutions based on its own experience and the best experience of every particle. By means of cooperation and competition among particles, the best solution with the best fitness value can be found.

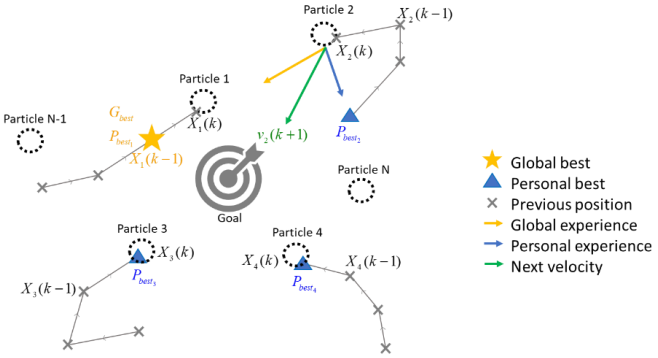

Fig. 2. Particle Swarm Optimization

Fig. 2 illustrates the idea of particle swarm optimization. Assume that there are N particles to search for the best solution in a D-dimensional solution space. The position of particle n is $X_n=(x_{n1}, x_{n2},\ldots, x_{nD})$ and the best position of particle n (Personal best) is $P_{bestn}=(pbest_{n1}, pbest_{n2},\ldots, pbest_{nD})$. The best position of all particles (Global best) is $G_{best}=(gbest_1, gbest_2,\ldots,gbest_D)$. The next velocity is $V_n=(v_{n1}, v_{n2},\ldots, v_{nD})$, while the next position of the particle can be calculated using (4) and (5):

$$v_{nd}(k+1)=w(k)\times v_{nd}(k)+c_1\times rand_1\times(pbest_{nd}-x_{nd}(k)) +c_2\times rand_2\times(gbest_d-x_{nd}(k)) \tag{4}$$

$$x_{nd}(k+1)=x_{nd}(k)+v_{nd}(k+1) \tag{5}$$

where $w(k)$ is the inertia weight, c_1 and c_2 are the acceleration constants, and *rand* is a random number in the range of [0,1]. Fig. 3 shows the flowchart for the particle swarm optimization algorithm.

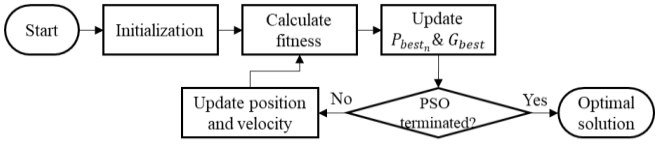

Fig. 3. Flowchart of the PSO algorithm

III. THE PROPOSED ASYMMETRIC 7-SEGMENT S-CURVE MOTION PLANNING APPROACH

The procedures of the proposed asymmetric 7-segment S-curve motion planning approach will be elaborated in the following.

Suppose that the path length is L and the velocity at the endpoint of the path is zero. Substituting $t=\sum_{i=1}^{7}T_i=K_7$ into (2) and (3) will yield

$$\begin{aligned} &v(K_7)=0\\ &p(K_7)=L \end{aligned} \tag{6}$$

Expanding (6) will result in

$$a_A(T_1+2T_2+T_3)-a_D(T_5+2T_6+T_7)=0$$

$$a_A\left[T_1\begin{pmatrix}T_1+3T_2+3T_3+3T_4\\+3T_5+3T_6+3T_7\end{pmatrix}+T_2\begin{pmatrix}3T_2+6T_3+6T_4\\+6T_5+6T_6+6T_7\end{pmatrix}+T_3\begin{pmatrix}2T_3+3T_4+3T_5\\+3T_6+3T_7\end{pmatrix}\right]$$
$$-a_D\left[T_5\left(T_5+3T_6+3T_7\right)+T_6\left(3T_6+6T_7\right)+2T_7^2\right)\Big]=6L \tag{7}$$

By solving (7) for a_A and a_D, one will have

$$a_A = \frac{6L(T_5+2T_6+T_7)}{(T_1+2T_2+T_3)\left[T_5(2T_5+6T_6+3T_7)+T_6(3T_6+3T_7)+T_7^2\right]} + (T_5+2T_6+T_7)\begin{bmatrix} T_1(T_1+3T_2+3T_3+3T_4)+T_2(3T_2+6T_3+6T_4) \\ +T_3(2T_3+3T_4) \end{bmatrix}$$

$$a_D = \frac{6L(T_1+2T_2+T_3)}{(T_1+2T_2+T_3)\left[T_5(2T_5+6T_6+3T_7)+T_6(3T_6+3T_7)+T_7^2\right]} + (T_5+2T_6+T_7)\begin{bmatrix} T_1(T_1+3T_2+3T_3+3T_4)+T_2(3T_2+6T_3+6T_4) \\ +T_3(2T_3+3T_4) \end{bmatrix} \quad (8)$$

From Fig. 1, it is clear that both a_A and a_D are positive. In particular, both a_A and a_D described by (8) must satisfy the following inequality:

$$\begin{aligned} a_A &> 0 \\ a_D &> 0 \end{aligned} \quad (9)$$

If inequality (9) is satisfied, then one can obtain the following constraints for the jerk profile in the Cartesian space:

$$\begin{aligned} J_1 &= a_A/T_1 < J_{\max} \\ J_3 &= a_A/T_3 < J_{\max} \\ J_5 &= a_D/T_5 < J_{\max} \\ J_7 &= a_D/T_7 < J_{\max} \end{aligned} \quad (10)$$

where J_{max} is the maximum jerk in the Cartesian space.

From Fig. 1, it is also clear that the maximum velocity will be reached at time instant K_3. Substituting $t = \sum_{i=1}^{3} T_i = K_3$ into (2) will yield the maximum velocity. As a result, one will have the following constraints:

$$\begin{aligned} a_A &< a_{\max} \\ a_D &< a_{\max} \\ v(K_3) &< v_{\max} \end{aligned} \quad (11)$$

where a_{max} is the upper bound of the maximum acceleration in the Cartesian space and v_{max} is the upper bound of the maximum velocity in the Cartesian space.

Substituting T_1 ~T_7 and J_1, J_3, J_5, J_7 into (3) will yield the position command. It is evident that the position command starts from zero in Fig. 1. In other words, to obtain actual position command $p(t)$, one needs to take the initial coordinate of the starting point of the work path into account. By using actual position command $p(t)$ in the Cartesian space and robot inverse kinematics, one can obtain joint angle $q_i(t)$, joint velocity $\dot{q}_i(t)$, joint acceleration $\ddot{q}_i(t)$, and joint jerk $\dddot{q}_i(t)$. For the i^{th} joint, the joint angle constraint, joint velocity constraint, joint acceleration constraint, and joint jerk constraint are listed in the following:

$$\begin{aligned} q_{i_{\min}} < q_i(t) &< q_{i_{\max}} \\ \max\left(|\dot{q}_i(t)|\right) &< \dot{q}_{i_{\max}} \\ \max\left(|\ddot{q}_i(t)|\right) &< \ddot{q}_{i_{\max}} \\ \max\left(|\dddot{q}_i(t)|\right) &< \dddot{q}_{i_{\max}} \end{aligned} \quad (12)$$

where $q_{i_{\min}}$ and $q_{i_{\max}}$ are the lower bound and the upper bound of the i^{th} joint angle, respectively; $\dot{q}_{i_{\max}}$ is the upper bound of the i^{th} joint velocity; $\ddot{q}_{i_{\max}}$ is the upper bound of the i^{th} joint acceleration; and $\dddot{q}_{i_{\max}}$ is the upper bound of the i^{th} joint jerk.

To check whether the joint torque is satisfied or not, one needs to compute the joint torque through the dynamic model of the robot manipulator [14] described by (13)

$$\tau = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) \quad (13)$$

where $M(q)$ is the $n\times n$ inertia matrix; $C(q,\dot{q})$ is the $n\times n$ Coriolis and centripetal matrix; and $G(q)$ is the $n\times 1$ gravity vector.

Before using (13) to check whether the joint torque constraint is satisfied, the system model parameters in (13) need to be identified in advance. In general, to perform system identification, (13) is rewritten in the form of a barycentric parameter as described by (14). The approach proposed in [15] is employed in this paper to identify the system parameters:

$$\tau = \Phi(q,\dot{q},\ddot{q})\beta \quad (14)$$

where regression matrix $\Phi(q,\dot{q},\ddot{q})$ consists of joint position, joint velocity, and joint acceleration; and β is the vector of barycentric parameters to be identified.

After the system model parameters have been identified, one can compute the joint torques at each time instant. Inequality (15) is the joint torque constraint:

$$\max\left(|\tau_i(t)|\right) < \tau_{i_{\max}} \quad (15)$$

where $\tau_{i_{\max}}$ is the upper bound of the i^{th} joint torque.

Since the goal is to find the best solution for T_1 ~T_7 of the asymmetric 7-segment S-curve, the particle position in the PSO algorithm is a set of 7-dimensional values, and each dimension corresponds to T_1 ~T_7. When the particle has a new position solution, it is necessary to check whether this solution satisfies the constraints described by (12) and (15), and then calculate the fitness described by (16).

$$fitness = -\frac{1}{T_1+T_2+T_3+T_4+T_5+T_6+T_7} \quad (16)$$

The search process is illustrated as follows：

Step 1: Initialize particle position X_n

The positions of all particles are randomly distributed in the search space.

$X_n=(T_{n1}, T_{n2}, T_{n3}, T_{n4}, T_{n5}, T_{n6}, T_{n7})$, $n=1,2,\ldots, N$

N is set as 500. $T_{ni} \in \left[0,\ T_{i_{\max}}\right]$ where $T_{i_{\max}}$ is an estimate of the maximum value of each time instant.

Step 2: Check physical constraints

Φ Cartesian space physical constraints

jerk : $J_1 < J_{\max}$, $J_3 < J_{\max}$, $J_5 < J_{\max}$, $J_7 < J_{\max}$

acceleration : $0 < a_A < a_{\max}$, $0 < a_D < a_{\max}$

velocity : $v(K_3) < v_{\max}$

where $a_A = \frac{6L(T_5+2T_6+T_7)}{\Delta}$, $a_D = \frac{6L(T_1+2T_2+T_3)}{\Delta}$

$$\Delta=(T_1+2T_2+T_3)\begin{bmatrix} T_5(2T_5+6T_6+3T_7) \\ +T_6(3T_6+3T_7)+T_7^2 \end{bmatrix}+(T_5+2T_6+T_7)\begin{bmatrix} T_1(T_1+3T_2+3T_3+3T_4) \\ +T_2(3T_2+6T_3+6T_4) \\ +T_3(2T_3+3T_4) \end{bmatrix}$$

2 Joint space physical constraints

$$\begin{aligned} angle &: \quad q_{i_{\min}} < q_i(t) < q_{i_{\max}} \\ velocity &: \quad \max\left(\left|\dot{q}_i(t)\right|\right) < \dot{q}_{i_{\max}} \\ acceleration &: \quad \max\left(\left|\ddot{q}_i(t)\right|\right) < \ddot{q}_{i_{\max}} \\ jerk &: \quad \max\left(\left|\dddot{q}_i(t)\right|\right) < \dddot{q}_{i_{\max}} \\ torque &: \quad \max\left(\left|\tau_i(t)\right|\right) < \tau_{i_{\max}} \end{aligned}$$

Step 3: Calculate fitness

If particle X_n has satisfied all the physical constraints in *Step 2*, then its fitness can be calculated. If not, then the fitness is set as 0. The fitness function is set as follows:

$$fitness = -\frac{1}{T_1+T_2+T_3+T_4+T_5+T_6+T_7}$$

Step 4: Update P_{best_n} & G_{best}

In the first iteration, each particle's $P_{best_n} = X_n \cdot G_{best}$ is X_n with the minimum fitness. In subsequent iterations, if the current *fitness*(X_n) of the particle is less than *fitness*(P_{best_n}), then P_{best_n} needs to be updated; otherwise, the previous one is kept. G_{best} also needs to be updated when *fitness*(X_n) of a certain particle is less than *fitness*(G_{best}).

Step 5: PSO termination

In this paper, the termination condition is reached when the number of times *G*_best remains unchanged reaches 100. If the termination condition is satisfied, then stop the algorithm. Otherwise, go to the next step.

Step 6: Update particle velocity and position

Update the particle velocity and position according to (2) and (3), then go to *Step* 2.

IV. Simulation and Experimental Results

In the simulation, three different types of S-curve are compared—an asymmetric 7-segment S-curve, a symmetric 7-segment S-curve and an asymmetric 5-segment S-curve [11]. Each type of S-curve is conducted on the work path separately. The work path is designed to avoid the robot singularity in advance and its information is shown in Table .

TABLE I WORK PATH INFORMATION

Trajectory information	X (cm)	Y (cm)	Z (cm)	α (deg)	β (deg)	γ (deg)	Trajectory length (cm)
Start point	57	-25	25.3	0	0	180	72.28
End point	17	20	65.3				

The simulation results of motion planning of different types of S-curve are listed in Table II.

TABLE II MOTION PLANNING RESULTS FOR DIFFERENT TYPES OF S-CURVE

time	T_1(s)	T_2(s)	T_3(s)	T_4(s)	T_5(s)	T_6(s)	T_7(s)	total(s)
asymmetric 7-segment	0.462	0.418	0.479	0.048	0.751	0.029	0.308	2.495
symmetric 7-segment	0.464	0.419	0.464	0.010	0.464	0.419	0.464	2.704
asymmetric 5-segment	0.856	0.790	0	0.719	0.320	-	-	2.685

It can be seen from Table II that the asymmetric 7-segment S-curve developed in this paper has the shortest machining time. Compared with the symmetric 7-segment S-curve and the asymmetric 5-segment S-curve, the machining time for the proposed asymmetric 7-segment S-curve can be reduced by 7.7% and 7.1%, respectively.

The motion planning results obtained by the asymmetric 7-segment S-curve in the simulation are used as the joint angle commands for the 6-DOF industrial manipulator in the experiment. The experimental results are shown in Figs. 4~9.

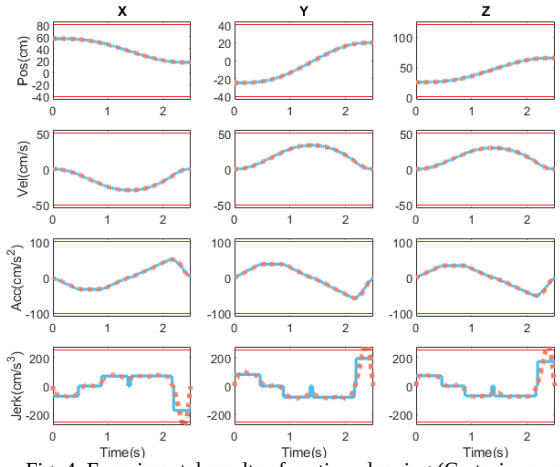

Fig. 4. Experimental results of motion planning (Cartesian space)

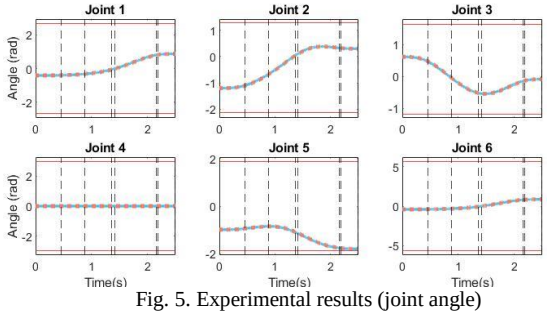

Fig. 5. Experimental results (joint angle)

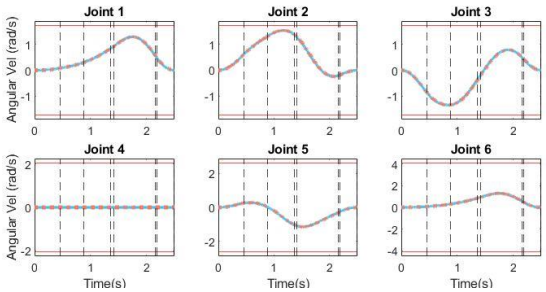

Fig. 6. Experimental results (joint velocity)

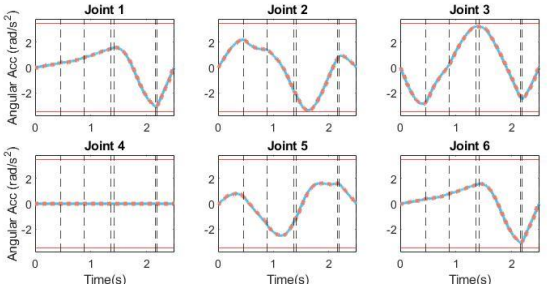

Fig. 7. Experimental results (joint acceleration)

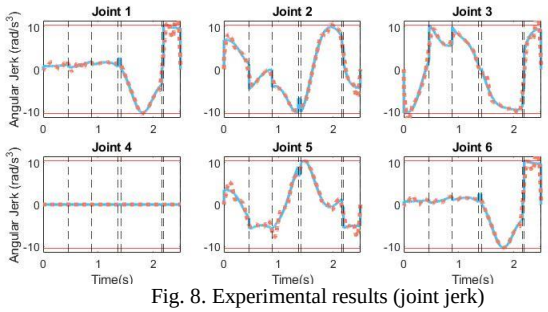

Fig. 8. Experimental results (joint jerk)

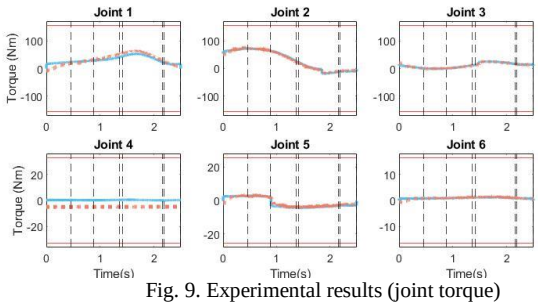

Fig. 9. Experimental results (joint torque)

It can be seen from

Fig. 4~9 that the motion planning results satisfy physical constraints in both Cartesian space and joint space. Especially in Fig. 8, a certain joint almost reaches the upper/lower bound of the constraint in each time segment, meaning motion planning can give full play to the capabilities of the machine and approach time optimization.

V. Conclusion

Motion planning is an important factor that affects the efficiency of industrial manipulators. Using the proposed PSO-based asymmetric 7-segment S-curve motion planning approach, the motion planning results satisfy both the Cartesian space physical constraints and joint space physical constraints. In addition, the obtained path command is near time optimal. Moreover, the results of both simulations and experiments conducted on a 6-DOF industrial manipulator indicate that the proposed approach is feasible.

References

[1] A. Bharathi and J. Dong, "Feedrate optimization for smooth minimum-time trajectory generation with higher order constraints," *The International Journal of Advanced Manufacturing Technology,* vol. 82, no. 5-8, pp. 1029-1040, 2016.

[2] M. Boryga and A. Graboś, "Planning of manipulator motion trajectory with higher-degree polynomials use," *Mechanism and Machine Theory,* vol. 44, no. 7, pp. 1400-1419, 2009.

[3] F. Liang, J. Zhao, and S. Ji, "An iterative feed rate scheduling method with confined high-order constraints in parametric interpolation," *The International Journal of Advanced Manufacturing Technology,* vol. 92, no. 5, pp. 2001-2015, 2017.

[4] T.-C. Lu and S.-L. Chen, "Time-optimal feedrate algorithm for non-uniform rational B-spline tool paths with process and machine tool constraints," in *Proc. of the 2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 2019, pp. 752-757.

[5] A.-C. Lee, M.-T. Lin, Y.-R. Pan, and W.-Y. Lin, "The feedrate scheduling of NURBS interpolator for CNC machine tools," *Computer-Aided Design,* vol. 43, no. 6, pp. 612-628, 2011.

[6] X. Du, J. Huan, and L.-M. Zhu, "A complete S-shape feed rate scheduling approach for NURBS interpolator," *Journal of Computational Design and Engineering,* vol. 2, no. 4, pp. 206-217, 2015.

[7] M. A. a. H. Şen, "S-curve Motion Profile Design for Vibration Control of Single Link Flexible Manipulator," *Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Dergisi,* vol. 23, no. 68, pp. 661-676, 2021.

[8] Y. Chen, X. Ji, Y. Tao and H. Wei, "Look-ahead algorithm with whole S-curve acceleration and deceleration," *Advances in Mechanical Engineering,* vol. 5, 2013.

[9] Y. Bai, X. Chen, and Z. Yang, "A generic method to generate as-curve profile in commercial motion controller," in *Proc. of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2017.

[10] K.-H. Rew and K.-S. Kim, "Using asymmetric S-curve profile for fast and vibrationless motion," in *Proc. of the 2007 International Conference on Control, Automation and Systems*, 2007, pp. 500-504.

[11] T.-C. Lu and S.-L. Chen, "Genetic algorithm-based S-curve acceleration and deceleration for five-axis machine tools," *The International Journal of Advanced Manufacturing Technology,* vol. 87, no. 1, pp. 219-232, 2016.

[12] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. of the ICNN'95-International Conference on Neural Networks*, 1995, vol. 4, pp. 1942-1948.

[13] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proc. of the 1998 IEEE International Conference on Evolutionary Computation*, 1998, pp. 69-73.

[14] K. S. Fu, R. Gonzalez, and C. G. Lee, *Robotics: Control, Sensing, Vision, and Intelligence*. Tata McGraw-Hill Education, 1987.

[15] J. Swevers, W. Verdonck, and J. De Schutter, "Dynamic Model Identification for Industrial Robots," *IEEE Control Systems Magazine*, vol. 27, no. 5, pp. 58-71, Oct. 2007.

Yen-Ju Lin received her B.S. degree in mechanical engineering from National Cheng Kung University, Tainan, Taiwan, in 2019, and her M.S. degree in electrical engineering from National Cheng Kung University, Tainan, Taiwan, in 2022. Her research interests include robotics, motion planning and optimization algorithms.

Ming-Yang Cheng received his B.S. degree in control engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1986, and his M.S. and Ph.D. degrees in electrical engineering from the University of Missouri, Columbia, in 1991 and 1996, respectively. From 1997 to 2002, he held several teaching positions at Kao Yuan Institute of Technology, Kaohsiung, Taiwan; Dayeh University, Changhua, Taiwan; and National Kaohsiung First University of Science and Technology, Kaohsiung, Taiwan. Since 2002, he has been with the Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan, where he is currently a Professor. His research interests include motion control, motor drives, visual servoing, and robot control.