

Received April 10, 2020, accepted April 29, 2020, date of publication May 4, 2020, date of current version May 19, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2992343

A Planning Method for Multi-Axis Point-to-Point Synchronization Based on Time Constraints

SHAN-DA WANG¹, XIAO LUO², SHI-JIE XU¹, QING-SHENG LUO¹, BAO-LING HAN³,
GUAN-HAO LIANG¹, AND YAN JIA³

¹School of Mechatronic Engineering, Beijing Institute of Technology, Beijing 100081, China

²School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China

³School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China

Corresponding author: Xiao Luo (luox@bit.edu.cn)

ABSTRACT In this paper, a third-order S-curve is used to model the motion issue. A closed method of solving is derived for the optimal point-to-point(P2P) time issue based on the model. Based on the P2P time-optimal solution, a closed multi-axis synchronization algorithm (MASA) method is proposed. This method considers the kinematic equation of the S-curve based on the inverse function of the multivariate function. Therefore, the inverse function of the execution time is derived via the kinematic equation of the P2P time optimization issue and solving a set of smaller bandwidth parameters is an important attempt in this paper. Otherwise, based on the MASA proposed in this paper, a synchronous computing framework suitable for the planning of robot space poses was designed. This framework can effectively solve problems with robot pose planning. The innovations of this article can be summarized as:(1)Based on the third-order S-curve, this paper deduced a complete method for solving the P2P time-optimal curve.(2) Based on the method for solving the time-optimal curve, this paper further derived three synchronization methods based on time constraints.(3) Based on the MASA proposed in this paper, a synchronous computing framework suitable for the planning of robot space poses was designed. The continuity and smoothness of the robot joint spatial motion can be guaranteed in the Descartes space. Few scholars have studied the closed solution method of multi-axis synchronization. The research in this paper promotes the theoretical research of motion control.

INDEX TERMS Motion control, multi-axis synchronization, S-curve, time optimal.

I. INTRODUCTION

In recent years, the robotics and digital processing industries based on motion control have developed rapidly. For a multi-motor electromechanical system, multi-axis motion planning has always been one of the difficulties and a topic of interest. Multi-axis motion planning requires multiple effectors to move in coordination so that the system can provide the specified task trajectory. The requirements of multi-axis motion planning for an electromechanical system are as follows.

Firstly, considering the performance of the motor and the actual needs of the application scenario, the motion of the motor must not exceed the specified bandwidth. Secondly, considering the stability and trajectory accuracy of the system, the motion trajectory of the motor should be continuous

The associate editor coordinating the review of this manuscript and approving it for publication was Ning Sun¹.

and smooth. Thirdly, in industrial applications, time optimality has always been an important performance indicator. Fourthly, multi-axis synchronization is also a big challenge. Finally, on the basis of the above, it is necessary to satisfy the principle of lowest energy as much as possible.

There are two application scenarios for multi-axis synchronization. One is point-to-point(P2P) motion, that is to say there exists only start and end points in the whole motion process. The other is multi-point motion, that is to say there exist many points in the whole motion process. This thesis focuses on studying the problem of multi-axis synchronization with P2P. The reasons are as follows.

Firstly, the P2P multi-axis synchronization problem is an important and basic issue in robotics and motion control. However, this part of research lacks model-based theoretical research. If we can make a theoretical breakthrough on this issue, it can not only be directly applied to real scenes, but also help other scholars to do more in-depth research.

Secondly, the current multi-axis P2P synchronization algorithms have some defects. Specific analysis will be made below.

Finally, the author's research fields also cover the multi-point multi-axis synchronization issue. We hope to apply the theory of multi-axis P2P synchronization to multi-axis multi-point synchronization. In fact, based on the research in this article, some results have been obtained. The detail will be introduced below.

A. CURRENT RESEARCH AND SHORTCOMING OF P2P MOTION

The LSPB(Linear segment with parabolic blend) planning method has been proposed previously. It ensures that the velocity continuously changes within the limits of velocity and acceleration. Moreover, the calculation is simple, which can solve some practical problems. However, it causes the acceleration to hop and significantly impacts the motor in practical applications. This impact also greatly influences the accuracy of the tip trajectory. Therefore, continuous S-curve planning has been proposed for the position curve C^2 [1]. This planning method uses a bounded input quantity integral of Jerk to obtain the acceleration, with the acceleration curve changing smoothly. Curves of this type have been widely used because they are time-optimal, smooth and easy to calculate. At the same time, a great deal of research has been conducted to eliminate the disadvantages of this method [2]–[7], and good results have been achieved. To further improve the control effect and trajectory accuracy, some studies have combined the S-curve and the kinematic model of the system. For example, in reference [8], a kinematic model was used to solve the vibration problem during the acceleration and deceleration of the robot in the production and processing lines.

It should be noted that although the S-curve has a continuous acceleration curve, it is not differentiable. A great deal of research has been conducted to address this issue. A traditional S-curve can be interpreted as a continuous and differentiable position curve constructed by piecewise cubic polynomial functions. Therefore, a higher-order polynomial function can be used to describe the motion curve, which is further evolved into a higher-order S-curve to overcome this problem [9], [10]. In addition, some scholars have used continuous and infinitely differentiable harmonic functions [11], [12] or the Sigmoid function [13] to construct a Jerk curve, which can make the Jerk and acceleration of the S-curve smoother and enables the motors to run more smoothly.

However, when the model order increases, the solution to the trajectory becomes more complicated, especially when solving the multi-axis synchronization issue. To solve this problem, some dynamic optimization algorithms have been introduced in the field. For example, in Reference [14], an optimization method based on MPC was used to achieve synchronized planning of multi-axis and multi-position points. In Reference [13], the scaling factor and binary search

methods were used to solve the multi-axis synchronization issue. In addition, more intelligent algorithms have attracted attention from researchers in the field. In Reference [15], a sextic polynomial was used to describe the joint motion curve and to search for feasible paths in complex geometric environments in combination with the GA and PSO algorithms. In Reference [16], the PSO algorithm was used to optimize the curve parameters. In References [17], [18], a more novel intelligent algorithm was used to optimize the target trajectory directly in the joint space.

However, there are following shortcomings for the existing P2P synchronization solutions from the current research situation.

The first problem is how to solve the optimal curve quickly and efficiently when the target position is too close. Current methods use iteration or search to find suitable parameters.

Secondly, current synchronous solutions are focused on control methods, optimization methods, and intelligent algorithms. Although these methods can solve the synchronization problem, some algorithms cannot guarantee the global optimality, and some research results cannot be applied in real-time systems because they require a large amount of calculations.

In summary, few scholars have studied the model-based closed solution method for P2P time optimization and multi-axis P2P synchronization problems. This greatly limits the application range of the algorithm and reduces the solution efficiency. In practical applications, it relies heavily on high-cost hardware, which seriously restricts the application effect. Therefore, model-based analysis and derivation are extremely meaningful in this situation.

B. RESEARCH OF THIS ARTICLE

Based on the study of the third-order S-curve equation of motion, a closed solution method for uniaxial P2P time optimal curve is derived. This algorithm can directly calculate the time-optimal trajectory of P2P without iteration or optimization. This algorithm can solve the motion parameters in a closed and efficient way.

Then, based on the closed solution of uniaxial axis P2P time optimal curve, the inverse mapping with adjustable bandwidth parameters is solved by using the inverse function of multiple functions. This theory reveals the changing process of S-curve and the relations between parameters. Before that, no more scholars have developed relevant research.

After solving the inverse mapping, an effective multi-axis synchronization algorithm (MASA) is proposed. Different from the previous research, the algorithm is composed of closed solution formulas with high efficiency and low calculation cost.

In the end, a synchronization calculating framework is designed for robot planning based on MASA. It is effective and provides the guarantee for continuity and smoothness of robot joint space motion.

In addition, for the problem of P2P multi-axis synchronization, this article conducts a series of relatively basic

theoretical derivations and gives a set of new ideas for multi-axis synchronization. More importantly, this theory and framework can be applied to high-order S-curves or other composite S-curves. In theory, this article also has greater innovation and contribution.

The rest of the paper is structured as follows. In Chapter II, the time optimality of the S-curve is demonstrated, the iterative update equation of the S-curve is derived, and the parameters of the third-order S-curve of the research object are defined and described. In Chapter III, a closed solving method for time-optimal planning is derived for the 7-segment S-curve. In Chapter IV, the multi-axis synchronization issue is transformed into a parameter optimization issue based on time constraints. Then, according to the system bandwidth parameters, three different optimization methods are derived. In Chapter V, experiments and data analyses are conducted for the proposed method. Chapter VI summarizes the work and describes future research.

II. S-CURVE PLANNING MODEL

The concept of the S-curve was first proposed in [1]. Many articles pointed out that the S-curve was demonstrated by the Pontryagin's minimum-value principle to have the time optimality under a given bandwidth. However, the S-curve is a third-order integral system with amplitude limitation. Similarly, n th order integral systems with amplitude limitation can achieve the goal of time optimality by stepping the control quantity to its maximum value; however, a more complicated judgment of amplitude limitation is required. The dual-integral system is taken as an example to demonstrate the existence conditions of time optimality.

A. PROOF OF TIME OPTIMALITY

A simple dual integral system and its state space equation are as follows:

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \quad (1)$$

The boundary conditions of the system are:

$$\begin{aligned} |u| &\leq M \\ x_1(t_0) &= 0, \quad x_2(t_0) = 0 \\ x_1(t_f) &= S, \quad x_2(t_f) = 0 \end{aligned} \quad (2)$$

The optimal control quantity u^* is obtained to minimise the performance indicator:

$$J = \int_{t_0}^{t_f} dt = t_f - t_0 \quad (3)$$

According to the minimum-value principle, the Hamilton function of the system is:

$$H[x(t), u(t), t] = 1 + \lambda_1 x_2 + \lambda_2 u \quad (4)$$

The co-state equations and governing equations are:

$$\begin{cases} \dot{\lambda}_2 = -\frac{\partial H}{\partial x_2} = -\lambda_1 \\ \dot{\lambda}_1 = -\frac{\partial H}{\partial x_1} = 0 \end{cases} \quad (5)$$

The optimal control quantity is:

$$u^*(t) - \text{sign}[\lambda_2(t)]M = \begin{cases} M & \lambda_2(t) < 0 \\ -M & \lambda_2(t) > 0 \end{cases} \quad (6)$$

It is thus clear that the time-optimal control is a switch-type control, which requires the control variable to be maximum. The above second-order integral system is a typical velocity-displacement system, where x_1 is the displacement, x_2 is the velocity and \dot{x}_2 is the control quantity. The system is required to start from a zero initial state and maintain a zero velocity after reaching the specified position S . When the control quantity reaches its maximum value, the shortest-time performance can be guaranteed. However, in actual scenarios, the velocity value cannot be infinitely increased, so it is necessary to stop the input of the control quantity when the velocity is increased to its maximum value. The planning equation based on this idea is the LSBP mentioned earlier. However, LSBP suffers from sudden changes in acceleration, which can be directly solved by increasing the system order. The motor bandwidth is generally limited, i.e. the maximum velocity V_m , the maximum acceleration A_m and the maximum value of *jerk* J_m . Thus, this is a three-integral system with an upper limit, with $x = [S \ V \ A]^T = [x_1 \ x_2 \ x_3]^T$ as the state variable, where S is the system displacement, V is the system velocity and A is the system acceleration. The control quantity u is *jerk*, and the $|u| \leq J_m$.

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u \quad (7)$$

For the above displacement system, the same method can demonstrate that the goal of the shortest time can be achieved by ensuring that the control quantity \dot{x}_3 always takes its maximum value J_m when it is acting. The switching control based on this type of input quantity evolves into the third-order S-curve after adding relevant amplitude limiting logics. As the entire motion process can be presented by seven continuous piecewise functions, it is also called a 7-segment S-curve. As the Jerk input of the 7-segment S-curve is a step signal, discontinuous points remain in the acceleration curve, which can be solved by increasing the system order. Higher-order integral systems are also obtained using the maximum-value principle. Another method is to use a smooth and differentiable function to construct a Jerk curve so that it has an approximate optimality under the current order, e.g. [11], [12]. As all of the time-optimal curves completely follow the bang-bang control rate or approximate square wave control rate, they also have similar geometric or mathematical properties. This paper analyses and derives the 7-segment S-curve. This is because in a third-order system, a

polynomial equation of up to the third degree may be present. According to Galois theory, polynomial equations below the fourth degree have closed solutions, which means that all problems in the 7-segment S-curve can have closed solutions. At the same time, the 7-segment S-curve has been widely used because of its C^2 continuity. At the same time, the use of the 7-segment S-curve allows for easier recognition of the changing law of the system equations and provides a more holistic idea for related research by other scholars. Therefore, using the 7-segment S-curve is not only of practical significance but also of academic significance.

B. S-CURVE PARAMETERIZATION AND ITERATION EQUATION

The S-curve can be interpreted as a limited multi-integral system, and multiple integrations of the control quantity is the final output of the system, that is, the displacement quantity. The displacement expression of the S-curve can be expressed as:

$$S = \underbrace{\int_{t_f}^{t_s} \dots \int_{t_f}^{t_s} u(t) dt}_n \quad (8)$$

where $u(t)$ is a time-varying system input and n is the system order, that is, there is an n multiple integral sign. Considering the amplitude limitation, $u(t)$ will continue to output within a continuous period of time, or stop outputting within a certain period of time. In other words, $u(t)$ is segmented. Thus, the input series $U = [u_1 \dots u_i]$ can be used to denote the output of each segment, and the time series $T = [T_1 \dots T_i]$ can be used to denote the duration of each segment in U . U and T can be used to uniquely denote a given S-curve model. Note that t_p is the p th sampling cycle.

$$\begin{cases} S_p = \sum_{k=0}^{n-1} \frac{(t_p)^k}{k!} S_{p-1}^{(k)} + \frac{(t_p)^n}{n!} u_p \\ S_p^{(1)} = \sum_{k=1}^{n-1} \frac{(t_p)^k}{k!} S_{p-1}^{(k)} + \frac{(t_p)^{n-1}}{(n-1)!} u_p \\ S_p^{(2)} = \sum_{k=2}^{n-1} \frac{(t_p)^k}{(k-1)!} S_{p-1}^{(k)} + \frac{(t_p)^{n-2}}{(n-2)!} u_p \\ \vdots \\ S_p^{(n)} = u_p \end{cases} \quad (9)$$

Take $n=3$ as an example, the above equation is written in the form of iteration such that:

$$\begin{aligned} S_p &= S_{p-1} + V_{p-1}t_p + \frac{1}{2}A_{p-1}t_p^2 + \frac{1}{6}J_p t_p^3 \\ V_p &= V_{p-1} + A_{p-1}t_p + \frac{1}{2}J_p t_p^2 \\ A_p &= A_{p-1} + J_p t_p \\ J_p &= u_p \end{aligned} \quad (10)$$

where S_p is the system output during the p th sampling period, V_n is the system velocity during the p th sampling period, A_n is the system acceleration during the p th sampling period and J_n is the system input during the p th sampling period. When $p = 1$, S_0 , V_0 and A_0 are the initial position, initial velocity and initial acceleration of the system, respectively. Note that all of the other S-curve models can be written in similar forms.

C. INTRODUCTION TO THE 7-SEGMENT S-CURVE

Based on the summary in II.A, S-curves of a given order can be uniquely determined by U and T . A standard 7-segment S-curve with a zero initial quantity is shown in Figure 1.

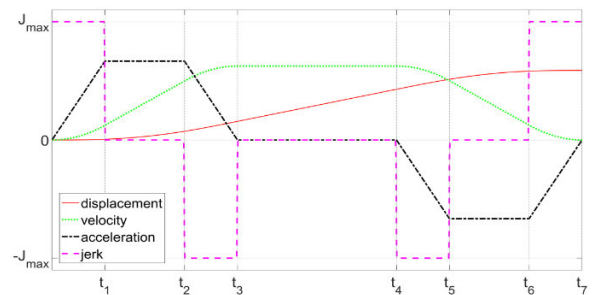


FIGURE 1. Schematic diagram of a 7-segment S-curve.

In Figure 1, the purple curve shows the output of the control quantity *jerk*. It can be seen that *jerk* maintains the maximum output J_m during the time periods of $t_1 - t_0$ and $t_7 - t_6$. On the other hand, *jerk* maintains the reverse maximum output $-J_m$ during the time periods $t_3 - t_2$ and $t_5 - t_4$. The output is stopped during the time periods of $t_2 - t_1$, $t_4 - t_3$ and $t_6 - t_5$. Note that $t_0 = 0$.

The elements in the time series T and the input series U have the following relationship:

$$\begin{cases} t_7 = \sum_{k=1}^7 T_k \\ T_p = t_p - t_{p-1}, (p = 1, \dots, 7) \\ T = [T_1 \quad T_2 \quad T_3 \quad T_4 \quad T_5 \quad T_6 \quad T_7] \\ U = [J_m \quad 0 \quad -J_m \quad 0 \quad -J_m \quad 0 \quad J_m] \end{cases} \quad (11)$$

For the input series U of a standard S-curve, when the input quantity is applied, it always takes the maximum value and takes zero in the inactive state. Then, the input series U and time series T are used to determine the S-curve. Together with (10) and (11), and the specified sampling period Smp , the S-curve can be differentiated into a series of points: $P = [P_1 \dots P_n]$.

$$n = \text{roundup}\left(\frac{t_7}{Smp}\right) \quad (12)$$

where $\text{roundup}(\ast)$ is the rounding-up function. In most articles, the S-curve model is expressed by a piecewise function. However, in a motion control program, any theoretically

continuous trajectory needs to be discretized. Therefore, the recursive method used here has certain advantages.

In summary, this section described the S-curve with the largest input quantity using two sets of parameters. In addition, the entire S-curve can be differentiated according to the recursive equation described above. Note that there is no limit on the maximum velocity and acceleration. Moreover, we did not take into account the impact of the target position on the entire velocity curve. All of these restrictions are directly reflected in each element of the time series T . This will be analyzed in detail in the next chapter in combination with the time optimality theory.

There are also some discussions based on system models and stability.

Firstly, for an n-order time-displacement system, it can be understood as an n-fold integration system. The state space model in modern control principles can describe this system very conveniently and accurately. Now we expect the system to move to the designated position. Without the intervention of any control means, the system will be unstable due to divergence. Indeed, this problem can be solved by designing a controller. However, if the output of the system is expected to reach the target point at a speed, the traditional control method will cause incompetence.

In this paper, the state space equation is used to describe the time-displacement system, and the minimum-value principle is used to derive the time-optimal control rate of the second-order system. In this case, the system is not only stable, but also can guarantee time optimality.

However, when the order of the system further increases, and the system has bandwidth constraints, the above solution process becomes more complicated. In this paper, under the guidance of the principle of minimum value, the optimal control rate of time is solved. This is the application of the optimal control method. Therefore, the derivation of this paper does not need to consider the problem of system stability.

III. TIME-OPTIMAL 7-SEGMENT S-CURVE OF P2P

In many papers, the solutions to the standard 7-segment S-curve are usually to maintain a specific contour. However, the curves are not solved based on time optimality. In this chapter, we derive the curve based on time optimality. As the goal of time optimality can directly improve the execution efficiency, the results could be of great significance for practical applications.

A. SIMPLIFICATION OF THE MODEL

In the velocity-time curve, the area enclosed by the curve and the time axis is the displacement through which the effector moves. Two simple principles are considered to help us identify the conditions for the time-optimal S-curve.

Firstly, on the premise of a given displacement, if the effector has a higher velocity, the execution time will be shorter. Thus, the velocity should be increased as much as possible to shorten the execution time. Secondly, as it is a point-to-point solution problem, zero velocity and zero

acceleration are maintained at the start and end points. For a fixed bandwidth, the velocity-time curve must be symmetric with respect to the center time point. If it is asymmetric, there must be a lower utilization rate of velocity on one side, but this makes the curve not the optimal one. Therefore, the time-optimal velocity curve must be as high as possible within the bandwidth limits and also symmetric with respect to the center time point.

To facilitate the analysis of the problem, the vector displacement S_r is simplified to a scalar distance D_r such that:

$$\begin{aligned} S_r &= S_{ref} - S_{cur} \\ D_r &= |S_r| \\ Dir &= sign(S_r) \end{aligned} \quad (13)$$

In the actual problem, the displacement has a direction Dir , which complicates the derivation process. Accordingly, all the displacements are reduced to a scalar distance D_r . The scalar quantity D_r is used to obtain a non-negative time series T and an input series moving by default in the positive direction. The opposite motion can be obtained by inverting the input series U . As a result, in all of the following derivations, a scalar distance D greater than zero is used. In the final algorithm implementation, equation (14) can be used to modify the actual direction of motion.

$$U = U \cdot Dir \quad (14)$$

In the next section, the above two criteria and assumption will be used to derive the solving method to obtain the optimal curve based on the maximum bandwidth of the system. The system bandwidth corresponds to the three determined variables of V_m , A_m and J_m . In other words, the maximum values of the first-, second- and third-order derivatives of the position curve are not allowed to exceed this limit.

B. SOLVING THE SHORTEST-TIME PROBLEM AT THE MAXIMUM BANDWIDTH

Now the entire acceleration phase of the system is directly considered. In this phase, J_m is first used to increase the system acceleration to A_m . The calculation is as follows:

$$\begin{aligned} T_1 &= T_j = \frac{A_m}{J_m} \\ U_1 &= J_m \end{aligned} \quad (15)$$

Then, the system begins to perform a uniform acceleration motion to increase the velocity. Finally, $-J_m$ is used to reduce the acceleration from A_m to 0, and the velocity at this time is V_m . Hence,

$$\begin{aligned} T_3 &= \frac{A_m}{J_m} \\ U_3 &= -J_m \end{aligned} \quad (16)$$

Whenever the system input is activated and when the acceleration is changed by A_m , the amount of velocity change incurred is:

$$V_j = \frac{A_m^2}{2J_m} \quad (17)$$

Hence, it can be deduced that:

$$\begin{aligned} T_2 &= \frac{V_m - 2V_j}{A_m} \\ U_2 &= 0 \end{aligned} \quad (18)$$

Furthermore, the distance travelled by the system during the entire ascent phase can be calculated as:

$$D_{up} = \frac{V_m^2}{2A_m} + \frac{A_m V_m}{2J_m} \quad (19)$$

As the curve is symmetric, if the system needs to increase the velocity to its maximum value in the most efficient manner and stop in the same way, a distance of $2D_{up}$ is required to be travelled through.

If the target position is $D_r \geq 2D_{up}$, the system will travel through a distance of $D_r - 2D_{up}$ at the maximum velocity of V_m . Hence,

$$\begin{aligned} T_4 &= \frac{D_r - 2D_{up}}{V_m} \\ U_4 &= 0 \end{aligned} \quad (20)$$

In summary, the system can increase the velocity to its maximum value according to the time optimality principle when $D_r \leq 2D_{up}$. The time series and input series of the S-curve at this time are:

$$\begin{aligned} T_1 &= T_3 = T_5 = T_7 = T_j \\ T_2 &= T_6 = \frac{V_m - 2V_j}{A_m} \\ T_4 &= \frac{D_r - 2D_{up}}{V_m} \\ U &= [J_m \quad 0 \quad -J_m \quad 0 \quad -J_m \quad 0 \quad J_m] \end{aligned} \quad (21)$$

In addition, it should be noted that the above equation needs to satisfy:

$$V_m - 2V_j \geq 0 \quad (22)$$

However, the system bandwidth parameters are arbitrarily specified by the user. In the case of $V_m - V_j < 0$, A_m is very high. According to (22), a new maximum acceleration \tilde{A}_m can be calculated as:

$$\tilde{A}_m = \sqrt{V_m J_m} \quad (23)$$

It can be easily demonstrated that:

$$\tilde{A}_m < A_m \quad (24)$$

\tilde{A}_m is used to replace the previous A_m to ensure that the S-curve exists. More importantly, $\tilde{A}_m < A_m$. In other words, the new maximum acceleration value \tilde{A}_m given by (23) does not exceed the maximum value A_m set by the previous user; i.e. \tilde{A}_m is within a safe and reasonable range, but is more conservative. This condition is extremely important for the completeness of the algorithm. With this method, a reasonable \tilde{A}_m for any given system bandwidth can always be found so that the S-curve exists. Thus, in the following sections, it is always assumed that the condition of $V_m - 2V_j \geq 0$ is true.

C. SOLVING THE SHORTEST-TIME PROBLEM AT A NON-MAXIMUM BANDWIDTH

In III.A, the distance conditions that can reach the maximum bandwidth were derived. The S-curve in the standard state makes full use of the bandwidth of each order. But once the target position reaches $D_r < 2D_{up}$, the effector cannot maximize the velocity. Thus, a set of reasonable bandwidth parameters needs to be re-selected to match the current distance constraint. Iterative methods have been used in previous studies to find a suitable set of parameters. However, they require additional time for the calculation, and it is always difficult to ensure their completeness. There are also scaling methods that can solve a smaller set of parameters while preserving the contour of the curve; however, they lose time optimality.

To address this problem, this paper makes the following considerations. If the target position is closer to the current position, there is not enough distance to increase the velocity to its maximum value. However, in theory, there is a slightly smaller velocity value that can serve as the target velocity and at the same time ensure the exact target distance. Hence, in this section, based on the system model, a mapping relationship is established between the target distance and the maximum velocity, which is used to solve the expected velocity value that can satisfy the distance constraint.

1) KINEMATIC SOLUTION AT A NON-MAXIMUM VELOCITY

In Section III.A, it was assumed that the target distance is only great enough, and there is enough time and distance to increase the acceleration and velocity to their maximum values. This section focuses on the issue where the target distance is enough to increase the acceleration to its maximum value, but not enough to increase the velocity to its maximum value. Thus, we expect to establish a relationship between the target position and the maximum velocity to solve this target velocity.

Firstly, boundary conditions are derived based on the above description. The upper limit of the distance is not allowed to exceed $2D_{up}$, and the lower limit is the condition that the acceleration reaches its maximum value. Hence, the distance can be expressed as:

$$D_a = 4V_j T_j = \frac{2A_m^3}{J_m^2} \quad (25)$$

Therefore, when $D_a \leq D_r < 2D_{up}$, it is the distance constraint for motion at a non-maximum velocity. Therefore, within the interval $[2V_j, V_m)$, there is a reasonable velocity value V_r such that the effector can travel a distance of D_r . Based on the above derivation, equation (19) can be used to solve V_r :

$$J_m V_r^2 + A_m^2 V_r - A_m J_m D_r = 0 \quad (26)$$

Furthermore, according to the expression of T_2 in (18), (24) can be expressed as:

$$t_a^2 + \frac{3A_m}{J_m} t_a + \frac{2A_m^3 - D_r J_m^2}{A_m J_m^2} = 0 \quad (27)$$

where t_a is the acceleration time to be solved. Equation (27) is a one-variable quadratic equation for t_a . By combining the value range of D_r and the root relationship, the unique solution to (27) can be determined as:

$$t_a = \frac{\sqrt{A_m^4 + 4D_r A_m J_m^2} - 3A_m^2}{2J_m A_m} \quad (28)$$

In summary, when $D_a \leq D_r < 2D_{up}$, the 7-segment standard S-curve degenerates into a six-segment S-curve, and the time series and input series are:

$$\begin{aligned} T_1 = T_3 = T_5 = T_7 = T_j \\ T_2 = T_6 = \frac{\sqrt{A_m^4 + 4D_r A_m J_m^2} - 3A_m^2}{2J_m A_m} \\ T_4 = 0 \\ U = [J_m \quad 0 \quad -J_m \quad 0 \quad -J_m \quad 0 \quad J_m] \end{aligned} \quad (29)$$

2) MOTION SOLUTION AT A NON-MAXIMUM ACCELERATION

If the target distance is further reduced, then $D_r < D_a$, which means that D_r is small enough, and there is no longer a need to increase the acceleration to its maximum value. According to (25), with the acceleration as an unknown quantity, a smaller acceleration A_r can be solved to satisfy the distance constraint such that:

$$A_r = \sqrt[3]{\frac{D_r J_m^2}{2}} \quad (30)$$

Equation (30) is transformed into a form of *jerk* time t_j as shown in (31).

$$t_j = \sqrt[3]{\frac{D_r}{2J_m}} \quad (31)$$

Thus, in the case of $D_r < D_a$, the 7-segment standard S-curve degenerates into a 4-segment S-curve, and the time series and input series are:

$$\begin{aligned} T_1 = T_3 = T_5 = T_7 = \sqrt[3]{\frac{D_r}{2J_m}} \\ T_2 = T_4 = T_6 = 0 \\ U = [J_m \quad 0 \quad -J_m \quad 0 \quad -J_m \quad 0 \quad J_m] \end{aligned} \quad (32)$$

3) DESIGN OF THE POINT-TO-POINT TIME-OPTIMAL ALGORITHM

According to the analyses and derivations above, the algorithm logic block diagram shown in Figure 2 can be obtained.

In essence, the algorithm establishes a mapping relationship between the target displacement and runtime, which is a continuous piecewise function. It can be seen from (33) that the definition domain of the mapping is the entire real number axis.

However, when the order of the system further increases, and the system has bandwidth constraints, the above solution process becomes more complicated. In this paper, under the guidance of the principle of minimum value, the optimal

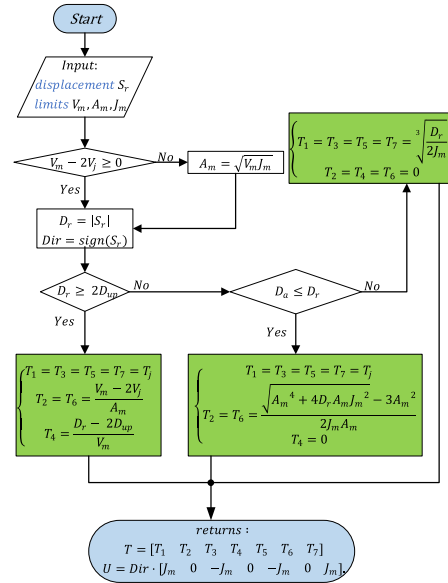


FIGURE 2. Logic diagram of the time-optimal algorithm for the P2P problem.

control rate of time should be solved. This is the application of the optimal control method. Therefore, the derivation of this paper does not need to consider the problem of system stability.

Using the above derivation and (21), (29) and (32), we can see that the mapping means that under the premise of some given system parameters, an arbitrary real number is mapped into the parameter series T , U , which denotes a smooth curve in displacement-time space in combination with (10).

$$\begin{aligned} F(S_r | V_m, A_m, J_m) \mapsto T, U \\ T = \text{sum}(T) \end{aligned} \quad (33)$$

As this mapping satisfies both surjection and injection, the algorithm is complete. That is, for any displacement independent variable, a smooth curve can be uniquely determined. This uniquely determined curve is the least time-consuming one of all feasible solutions based on the minimum-value principle in Chapter III. Therefore, $F(S_r | V_m, A_m, J_m)$ also uniquely determines the shortest running time t . $\text{sum}(T)$ in (33) is the sum of the time series T . Therefore, the function in (34) can be used to represent the mapping between the target position and the shortest running time under the given bandwidth parameters.

$$f(S_r | V_m, A_m, J_m) = t \quad (34)$$

Based on the above derivation, an inverse mapping f^{-1} related to the execution time t can be determined. With the inverse mapping f^{-1} , a desired running time t_r and target displacement S_r are given to solve a suitable set of bandwidth parameters $\tilde{V}, \tilde{A}, \tilde{J}$ to help address the multi-axis synchronization issue.

$$\begin{aligned} f(S_r | V_m, A_m, J_m) \mapsto t_{min} \\ f^{-1}(t_r | S_r, V_m, A_m, J_m) \mapsto \tilde{V}, \tilde{A}, \tilde{J} \end{aligned} \quad (35)$$

This problem will be further studied and analyzed in Chapter IV.

IV. MULTI-AXIS SYNCHRONIZATION ISSUE BASED ON TIME CONSTRAINTS

Consider $f(S, V, A, J) = t$ to be a multivariate function, which can be expressed as follows:

$$f(S, V, A, J) = \begin{cases} \frac{VA^2 + SAJ + JV^2}{\sqrt{A^4 + 4SAJ^2 + A^2}} & |S| \geq \frac{V^2}{A} + \frac{AV}{J} \\ \frac{VAJ}{2AJ} & \frac{2A^3}{J^2} \leq |S| < \frac{V^2}{2A} + \frac{AV}{2J} \\ 2\sqrt[3]{\frac{4S}{J}} & |S| < \frac{2A^3}{J^2} \end{cases}$$

$$\text{dom} = \left\{ (S, V, A, J) \mid S \in \mathbb{R}, A > 0, J > 0, V \geq \frac{A^2}{J} \right\} \quad (36)$$

It can be shown that the left and right limits of the piecewise function $f(S, V, A, J)$ at the end points are equal, and the algebraic operation of the real variable is continuous [19]; therefore, $f(S, V, A, J)$ is also continuous. An image of $f(S, V | A, J) = t$ under a given A, J can be plotted in three-dimensional coordinates, as shown in Figure 3.

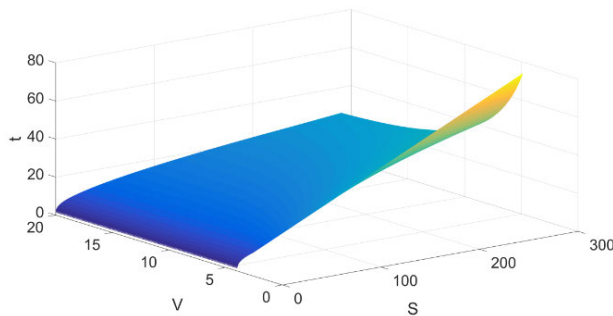


FIGURE 3. Projection of $f(S, V, A, J)$ in three-dimensional space under a given A, J .

$\delta = \{(S, V, A, J) \mid S \geq 0, A = 2, J = 1, V \geq 4\}$ is selected as the definition domain of the function in Figure 3. Firstly, it can be seen that the surface changes continuously. Secondly, from the perspective of the overall trend, under the same displacement S , the greater the velocity V is, the shorter the time t is consumed. With the increase of system bandwidth, the system can complete tasks in a shorter time and vice versa, which can be well understood in a visualized manner. It is also easy to demonstrate mathematically that S, V, A, J maintain a monotonic relationship with the running time t . Therefore, if three of the independent variables S, V, A, J are constants, f becomes a monotonic continuous unary function. Hence, f must have a monotonic inverse function f^{-1} . The inverse mapping described in (35) can be expressed as:

$$\begin{cases} f_J^{-1}(t_r | S_r, V_m, A_m, J_m) = \tilde{J} \\ f_A^{-1}(t_r | S_r, V_m, A_m, J_m) = \tilde{A} \\ f_V^{-1}(t_r | S_r, V_m, A_m, J_m) = \tilde{V} \end{cases} \quad (37)$$

As far as the multi-axis synchronization planning issue is concerned, an effector consuming less time needs to extend its motion time to synchronize with a slower effector. Therefore, the three types of inverse mapping relationships described in (37) need to solve the mapping of a smaller bandwidth parameter under a more relaxed time constraint. As the mapping of f is an injection, there is also a positive relationship shown in (38) between the parameters obtained by (37).

$$\begin{cases} f(S_r | V_m, A_m, \tilde{J}) = t_r \\ f(S_r | V_m, \tilde{A}, J_m) = t_r \\ f(S_r | \tilde{V}, A_m, J_m) = t_r \end{cases} \quad (38)$$

This smaller bandwidth parameter allows the effector to use the time t_r to complete the displacement S_r . In the following section, the details of solving the equations for the three mappings described in (37) will be deduced, and related algorithms will be designed. However, before that, some preparations need to be made.

As far as the above solution to the time optimality issue is concerned, we know that there is a maximum distance D_m that can be reached at a given time t and under a given bandwidth condition. In other words, there must exist a $D_m \geq D_r$ to ensure that the inverse solution f^{-1} is available. It is necessary to first determine the mapping of $t_r \mapsto D_m$.

Firstly, a new system constant T_{up} is derived to represent the shortest time required for the system to reach the maximum velocity V_m .

$$T_{up} = \frac{V_m - 2V_j}{A_m} + 2T_j = T_j + \frac{V_m}{A_m} \quad (39)$$

The mapping relationship of $t_r \mapsto D_m$ will be derived below in combination with T_{up} .

If the target running time t is greater than $2T_{up}$, this time is sufficient to accelerate the velocity to its maximum value. Hence, the maximum running distance is:

$$d = 2D_{up} + (t - 2T_{up})V_m \quad (40)$$

When the target running time is $t \in [4T_j, 2T_{up})$, there is not enough time to increase the velocity to V_m during this interval. The vertex velocity to which the velocity can be increased according to the optimal control rate is:

$$v_p = \left(\frac{t}{2} - 2T_j \right) A_m + 2V_j \quad (41)$$

The maximum distance can be equivalent to the product of v_p and $t/2$.

$$d = \frac{v_p^2}{A_m} + \frac{A_m v_p}{J_m} = \frac{t}{2} v_p = \frac{A_m J_m t^2 - 2A_m^2 t}{4J_m} \quad (42)$$

When the target running time is $t < 4T_j$, the acceleration cannot be maximized even if the maximum system input is used. Therefore, there is no uniform acceleration phase nor

a uniform motion phase. Hence, the maximum distance is equivalent to the product of v_p and $t/2$.

$$\begin{aligned} v_p &= \frac{J}{16}t^2 \\ d &= \frac{t}{2}v_p = \frac{J}{32}t^3 \end{aligned} \quad (43)$$

In summary, the relationship between the target time t_r and the maximum target displacement d can be obtained as shown by the function in (44).

$$\begin{aligned} f_i(t | V_m, A_m, J_m) &= \begin{cases} 2D_{up} + V_m(t - 2T_{up}) & t \geq 2T_{up} \\ \frac{A_m J_m t^2 - 2A_m^2 t}{4J_m} & 2T_{up} > t \geq 4T_j \\ \frac{J}{32}t^3 & 4T_j > t \end{cases} \end{aligned} \quad (44)$$

Secondly, in the following derivation, there will be solutions to the unary cubic equation. Complete solutions to the unary cubic equation have been derived previously. However, the resulting set of rooting equations does not distinguish between real roots and imaginary roots. The introduction of the complex field adds to the complexity of the implementation of the entire motion control logic. In Reference [20], they proposed a S.J. equation to give a more detailed discriminant and solving method, which could effectively eliminate the calculation of virtual roots.

Finally, from the analysis and derivation above, the time-based synchronization algorithm is the inverse operation for solving the time-optimal algorithm. Therefore, the time-based synchronization algorithm must also be a continuous piecewise function. It can be seen from the above derivation of the time-optimal algorithm that when the independent variable is within a certain range, it can be described by a certain kinematic equation. Then, the difficulty of the inverse mapping solution lies in determining the interval and deriving the kinematic equation.

A. SOLVING THE MINIMUM VELOCITY PROBLEM UNDER TIME CONSTRAINTS

The minimum velocity under time constraints can be expressed as:

$$f_V^{-1}(t_r | S_r, V_m, A_m, J_m) = \tilde{V} \quad (45)$$

In this case, the target displacement is S_r , the maximum system velocity is V_m , the maximum acceleration is A_m and the maximum *Jerk* value is J_m . The independent variable is the expected running time t_r . The mapping f_V^{-1} indicates that the minimum velocity \tilde{V} at which S_r can be completed using t_r is solved. To ensure the minimum velocity, the effector should increase the velocity to \tilde{V} in the most efficient way, and then make a uniform motion. As a result, the characteristic of this kind of problem is that the kinematic equation maintains the uniform motion phase as far as possible. Hence, the velocity curve in this case is close to a trapezoid.

As *Jerk* remains the same under this problem, in the case of $\tilde{V} \geq 2V_j$, the acceleration is increased to A_m , and vice versa. Hence, when $\tilde{V} \geq 2V_j$, a change in the vertex velocity of the effector does not affect the acceleration. In the case of $\tilde{V} < 2V_j$, a change in the vertex velocity of the effector affects the acceleration. The two cases correspond to different kinematic equations. Here, the distance in the critical state of $\tilde{V} = 2V_j$ is solved and denoted as D_V .

$$D_V = 4V_j T_j + 2V_j(t_r - 4T_j) \quad (46)$$

According to (46), the above case only occurs when $t_r \geq 4T_j$. Hence,

$$f_V^{-1} : \{t_r, S_r\} \mapsto [2V_j, V_m] \quad (47)$$

where $|S_r| \in [D_V, f_i(t_r)]$ and $t_r \in [4T_j, +\infty)$.

In this interval, as the acceleration increases to its maximum value and if the velocity is expected to increase further, there is a uniformly accelerated motion. We can use \tilde{V} to denote the acceleration time t_a :

$$t_a = \frac{\tilde{V} - 2V_j}{A_m} \quad (48)$$

The displacement can be expressed as:

$$D_r = (t_r - 2t_a - 4T_j)\tilde{V} + \frac{\tilde{V}^2}{A_m} + \frac{A_m \tilde{V}}{J_m} \quad (49)$$

We can transform (49) into an equation for the vertex velocity such that:

$$\tilde{V}^2 + \frac{(A_m^2 - J_m A_m t_r)}{J_m} \tilde{V} + A_m D_r = 0 \quad (50)$$

It can be judged from the root relationship that there are two positive real roots in the equation. Based on the assumptions in this case, we can know that the required \tilde{V} is within the interval $[2V_j, V_m]$. After derivation, we can find that the symmetry axis of the quadratic function in (50) is always on the right side of the interval $[2V_j, V_m]$, which is the smaller one of the two roots. Therefore, the analytic expression of \tilde{V} and the expression of time series T can be further deduced as:

$$\begin{aligned} T_1 &= T_3 = T_5 = T_7 = T_j \\ T_2 &= T_6 = \frac{t_{up}}{2} - 2T_j \\ T_4 &= t_r - t_{up} \end{aligned} \quad (51)$$

where the

$$\begin{aligned} \tilde{V} &= \frac{A_m J_m t_r - A_m^2 - \sqrt{A_m^4 - 2A_m^3 J_m t_r + A_m^2 J_m^2 t_r^2 - 4D_r A_m J_m^2}}{2J_m} \\ t_{up} &= 2(T_j + \frac{\tilde{V}}{A_m}). \end{aligned}$$

When D_r takes a smaller value, there is not enough distance to saturate the acceleration. Assume that the required vertex

acceleration is a_p . As there must be no uniformly accelerated segment, the kinematic equation can be expressed as:

$$\begin{cases} v_p = \frac{a_p^2}{2J_m} \\ D_r = \left(t_r - 4\frac{a_p}{J_m}\right)v_p + \frac{v_p^2}{a_p} + \frac{a_p v_p}{J_m} \end{cases} \quad (52)$$

Furthermore, equation (52) can be transformed into:

$$2a_p^3 - J_m t_r a_p^2 + J_m D_r = 0 \quad (53)$$

By observing (52) from the root relationship, the equation has one negative root and two positive roots. The two positive roots can be further screened according to $a_p \in [0, A_m)$. It is easy to obtain the only feasible positive real number solution using the S.J. equation proposed in [20]. Subsequently, a_p and T are transformed into:

$$\begin{cases} T_1 = T_3 = T_5 = T_7 = t_j \\ T_2 = T_6 = 0 \\ T_4 = t_r - 2t_{up} \end{cases} \quad (54)$$

where the:

$$\begin{cases} t_j = \frac{a_p}{J_m} \\ t_{up} = 2t_j; \end{cases}$$

and:

$$\begin{cases} w = \arccos\left(\frac{54S_r}{J_m t_r^3} - 1\right) \\ a_p = \frac{J_m t_r}{6} \left(\cos\frac{w}{3} - \sqrt{3}\sin\frac{w}{3} + 1\right) \end{cases}$$

Hence, a mapping relationship with the minimum velocity \tilde{V} in the case of $t_r \in [4T_j, +\infty)$ is established.

$$f_V^{-1}(t_r | S_r, V_m, A_m, J_m) = \begin{cases} \text{nosolution} & f_i(t_r) < |S_r| \\ \text{formula(51)} & D_v \leq |S_r| \leq f_i(t_r) \\ \text{formula (53), (54)} & |S_r| < D_v \end{cases} \quad (55)$$

When $t_r \in [0, 4T_j)$, the time is not enough to increase the acceleration to its maximum value. Therefore, motion can still be described using (53). Hence, f_V^{-1} can be expressed as:

$$f_V^{-1}(t_r | S_r, V_m, A_m, J_m) = \begin{cases} 4T_j \leq t_r & \begin{cases} \text{nosolution} & f_i(t_r) < |S_r| \\ \text{formula (51)} & D_v \leq |S_r| \leq f_i(t_r) \\ \text{formula (54)} & |S_r| < D_v \end{cases} \\ 0 < t_r < 4T_j & \text{formula (54)} |S_r| \leq f_i(t_r) \end{cases} \quad (56)$$

The expressions are all analytic and it is easy to verify that the function f_V^{-1} is continuous. In addition, it can be found that two lines in (56) use the same kinematic equation. When $t_r = 4T_j$, D_v takes its minimum value $4V_j T_j$ and $f_i(t_r)$ is equal to $4V_j T_j$. When $t_r < 4T_j$, $f_i(t_r) < 4V_j T_j$ is true. According to the existence conditions of D_v and the above

analysis, a logical expression of D_V is designed, which could make the whole algorithm logic more concise. The flow chart of the entire algorithm is shown in Figure 4.

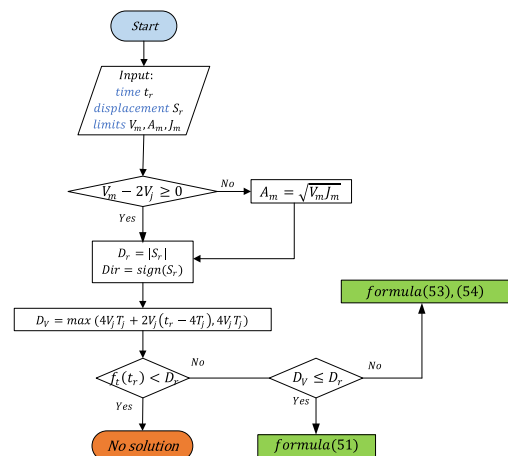


FIGURE 4. Logic diagram of the minimum velocity synchronization algorithm under time constraints.

Note that $\max(*, *)$ indicates that the greater of the two variables is taken.

B. SOLVING THE MINIMUM ACCELERATION PROBLEM UNDER TIME CONSTRAINTS

The minimum acceleration under time constraints can be expressed as:

$$f_A^{-1}(t_r | S_r, V_m, A_m, J_m) = \tilde{A} \quad (57)$$

Under time constraints, the values of the target displacement S_r , system maximum velocity, acceleration and Jerk are given. The independent variable is the expected running time t_r . The mapping f_A^{-1} indicates that the minimum acceleration \tilde{A} at which S_r can be exactly completed using t_r is solved. To ensure the minimum acceleration, the effector has a longer period of uniform acceleration, which indirectly shortens the uniform motion time. Note that the velocity curve in this case is close to a triangle.

Assume that $t_r > 2T_{up}$, then the following details are considered. When $S_r = f_i(t_r)$, both the acceleration and velocity reach their maximum values, and there is a uniform motion phase. After that, S_r is reduced continuously. In this case, we can keep the maximum velocity V_m unchanged and reduce the acceleration to slow down the entire motion. In this process, due to the constant decrease in acceleration, the uniform acceleration time gradually increases, while the uniform motion time decreases continuously. Therefore, it can be inferred that the existence of S_r and \tilde{A} enables the velocity to increase to V_m without a uniform motion phase. In this case, the motion distance is $D_A = t_r V_m / 2$.

Therefore, when $t_r \geq 2T_{up}$ and $t_r V_m / 2 \leq S_r \leq f_i(t_r)$, the motion can increase the velocity to its maximum value. Note that there exist uniform motion and uniform acceleration

phases. We then deduce the kinematic equation. Assuming that the acceleration is \tilde{A} . The uniform acceleration time is t_a :

$$t_a = \frac{V_m - \frac{\tilde{A}^2}{J_m}}{\tilde{A}} = \frac{V_m}{\tilde{A}} - \frac{\tilde{A}}{J_m} \quad (58)$$

The motion in this case can be expressed as:

$$\left(t_r - 2t_a - 4\frac{\tilde{A}}{J_m}\right)V_m + \frac{V_m^2}{A_m} + \frac{\tilde{A}V_m}{J_m} = S_r \quad (59)$$

We then transform (59) into an equation for \tilde{A} such that:

$$V_m\tilde{A}^2 + (J_m S_r - V_m t_r J_m)\tilde{A} + J_m V_m^2 = 0 \quad (60)$$

It can be judged from the root system relationship that (60) has two positive roots. Due to $t_a \in [0, t_r)$, it can be further deduced that \tilde{A} always takes a smaller root. Hence, it can be transformed into:

$$\begin{aligned} T_1 &= T_3 = T_5 = T_7 = t_j \\ T_2 &= T_6 = t_{up} - 2t_j \\ T_4 &= t_r - 2t_{up} \end{aligned} \quad (61)$$

where the

$$\tilde{A} = \frac{J_m t_r V_m - \sqrt{(S_r J_m)^2 - 2J_m^2 S_r t_r V_m + J_m^2 t_r^2 V_m^2 - 4J_m V_m^3 - J_m S_r}}{2V_m};$$

and

$$\begin{cases} t_j = \frac{\tilde{A}}{J_m} \\ t_{up} = t_j + \frac{V_m}{\tilde{A}}. \end{cases}$$

After that, when S_r continues to decrease until it is less than $t_r V_m/2$, the velocity cannot increase to its maximum value, and there is no uniform velocity phase. Therefore, the vertex velocity is $v_p = 2S_r/t_r$. The kinematic equation can be transformed into:

$$S_r = \frac{v_p^2}{\tilde{A}} + \frac{v_p \tilde{A}}{J_m} \quad (62)$$

It is further transformed into:

$$v_p \tilde{A}^2 - J_m S_r \tilde{A} + J_m v_p^2 = 0 \quad (63)$$

It can be inferred from the root system relationship that the equation contains two positive roots. Hence, there is an acceleration phase. Thus, $t_a \in [0, t_r/2]$. \tilde{A} and T are transformed into:

$$\begin{cases} T_1 = T_3 = T_5 = T_7 = t_j \\ T_2 = T_6 = \frac{1}{2}t_r - 2t_j \\ T_4 = 0. \end{cases} \quad (64)$$

where the

$$\tilde{A} = \begin{cases} \frac{J_m S_r - \sqrt{(S_r J_m)^2 - 4J_m V_m^3 - J_m S_r}}{2v_p} & v_p > 0 \\ 0 & v_p = 0; \end{cases}$$

and the

$$t_j = \frac{\tilde{A}}{J_m}.$$

When $t_r < 2T_{up}$, the maximum velocity cannot reach V_m . Accordingly, only $S_r \leq f_i(t_r)$ is needed to use (63). Therefore, the minimum acceleration mapping under time constraints can be expressed as the following piecewise function:

$$f_A^{-1}(t_r | S_r, V_m, A_m, J_m) = \begin{cases} 2T_{up} \leq t_r & \begin{cases} no\ solution & f_i(t_r) < |S_r| \\ formula\ (60),\ (61) & D_A \leq |S_r| \leq f_i(t_r) \\ formula\ (63),\ (64) & |S_r| < D_A \end{cases} \\ 0 < t_r < 2T_{up} & formula\ (63),\ (64) |S_r| \leq f_i(t_r) \end{cases} \quad (65)$$

When $t_r \geq 2T_{up}$, $D_A \leq f_i(t_r)$. When $t_r < 2T_{up}$, $f_i(t_r) < D_A$ is always true. Therefore, equation (65) can be used to determine the relationship of the size between $f_i(t_r)$ and $|S_r|$, and then D_A can be used to choose a kinematic equation.

C. SOLVING THE MINIMUM JERK PROBLEM UNDER TIME CONSTRAINTS

The minimum acceleration under time constraints can be expressed as:

$$f_J^{-1}(t_r | S_r, V_m, A_m, J_m) = \tilde{J} \quad (66)$$

In this case, the values of the target displacement S_r , system maximum velocity, acceleration and *Jerk* are given. The independent variable is the expected running time t_r . The mapping f_J^{-1} indicates that the minimum acceleration \tilde{J} at which S_r can be exactly completed using t_r is solved. The continuous decrease in *Jerk* causes simultaneous changes in acceleration and velocity.

If the maximum velocity V_m and the maximum acceleration A_m are kept unchanged, while the *Jerk* decreases, the time for the acceleration to increase to its maximum value gradually increases, and the uniform acceleration time gradually decreases. The minimum *Jerk* that can exactly increase the acceleration and velocity to their maximum values can be recalculated. It is worth noting that there is no uniform acceleration phase nor a uniform motion phase in this case.

$$\begin{cases} j = \frac{A_m^2}{V_m} \\ t_j = \frac{A_m}{j} \\ t_{up} = 2t_j \end{cases} \quad (67)$$

In addition, it can be proven that $t_{up} \geq T_{up}$. Hence, the case of $t_r \geq 2t_{up}$ is first considered. In this case, $\tilde{J} \in [j, J_m]$ and the velocity and acceleration can be kept at their maximum values. The lower bound of the distance can be calculated when $\tilde{J} = j$. In addition, in combination with the assumption

of $t_r \geq 2t_{up}$ and the processing method in Section IV.B, a more accurate $D_{J,A,V}$ expression can be obtained.

$$\begin{cases} d_{J,A,V} = (t_r - 4t_j) V_m + 2t_j V_m = t_r V_m - 2 \frac{V_m^2}{A_m} \\ d_{J,A,V,2t_{up}} = 2 \frac{V_m^2}{A_m} \\ D_{J,A,V} = \max(t_r V_m - 2 \frac{V_m^2}{A_m}, 2 \frac{V_m^2}{A_m}) \end{cases} \quad (68)$$

Hence, when $t_r \geq 2t_{up}$ and $D_r \in [D_{J,A,V}, f_i(t_r)]$, (69) can be used to describe its motion:

$$D_r = \left(t_r - 2 \frac{V_m - \frac{A_m^2}{\tilde{J}}}{A_m} - 4 \frac{A_m}{\tilde{J}} \right) V_m + \frac{V_m^2}{A_m} + \frac{A_m V_m}{\tilde{J}} \quad (69)$$

It can be further transformed into:

$$\tilde{J} = \frac{V_m A_m^2}{t_r V_m A_m - V_m^2 - A_m D_r} \quad (70)$$

Then, T and U can be transformed into:

$$\begin{aligned} T_1 &= T_3 = T_5 = T_7 = t_j \\ T_2 &= T_6 = t_a \\ T_4 &= t_r - 2t_a - 4t_j \\ U &= Dir \cdot [\tilde{J} \quad 0 \quad -\tilde{J} \quad 0 \quad -\tilde{J} \quad 0 \quad \tilde{J}] \end{aligned} \quad (71)$$

where the

$$t_j = \frac{A_m}{\tilde{J}},$$

and the

$$t_a = \frac{V_m - \frac{A_m^2}{\tilde{J}}}{A_m}.$$

According to the analysis above, when $D_r = D_{J,A,V}$, the acceleration and the velocity become saturated at the same time, and there is no uniform acceleration phase. However, an interesting phenomenon arises. As *Jerk* decreases further, the acceleration becomes unsaturated first. This conclusion can be demonstrated by the following derivation. Assume that D_r is small and no uniform acceleration phase nor a uniform motion phase is required. t_r and \tilde{J} can be used to denote the vertex velocity and vertex acceleration such that:

$$\begin{cases} a_p = \tilde{J} \frac{t_r}{4} \\ v_p = \tilde{J} \left(\frac{t_r}{4} \right)^2 \end{cases} \quad (72)$$

$a_p = A_m$ can be used inversely to solve the expression of \tilde{J} , and the $t_r \geq 2t_{up}$ assumption is substituted into v_p to get $v_p \geq V_m$. Instead, V_m is used to replace v_p and obtain $a_p \leq A_m$. At the same time, we can prove that there are cases where $v_p = V_m$ and the acceleration is not saturated.

The minimum value $D_{J,V}$ of the distance in cases where the acceleration is not saturated and the velocity is saturated is calculated below. As there is no uniform velocity phase, $D_{J,V} = V_m t_r / 2$. Therefore, when $t_r \geq 2t_{up}$ and

$D_r \in [D_{J,V}, D_{J,A,V}]$, we can continue to reduce *Jerk* while maintaining the maximum velocity V_m . Assume that the duration of *Jerk* is t_j . Hence, the kinematic equation can be transformed into:

$$D_r = (t_r - 4t_j) V_m + 2t_j V_m \quad (73)$$

It can be further transformed into:

$$t_j = \frac{t_r V_m - D_r}{2V_m} \quad (74)$$

Then, T and U can be obtained as:

$$\begin{aligned} T_1 &= T_3 = T_5 = T_7 = t_j \\ T_2 &= T_6 = 0 \\ T_4 &= t_r - 4t_j \\ U &= Dir \cdot [\tilde{J} \quad 0 \quad -\tilde{J} \quad 0 \quad -\tilde{J} \quad 0 \quad \tilde{J}] \end{aligned} \quad (75)$$

And the $\tilde{J} = V_m / t_j^2$.

The kinematic equation when $t_r \geq 2t_{up}$ and $D_r \in [0, D_{J,V}]$ is considered below. As both the acceleration and the velocity cannot be saturated, there is only variable accelerated motion. The kinematic equation can be expressed as:

$$D_r = \frac{\tilde{J}}{32} t_r^3 \quad (76)$$

Then, T and U are transformed into:

$$\begin{aligned} T_1 &= T_3 = T_5 = T_7 = \frac{t_r}{4} \\ T_2 &= T_6 = 0 \\ T_4 &= 0 \\ U &= Dir \cdot [\tilde{J} \quad 0 \quad -\tilde{J} \quad 0 \quad -\tilde{J} \quad 0 \quad \tilde{J}] \end{aligned} \quad (77)$$

where the

$$\tilde{J} = \frac{32D_r}{t_r^3}.$$

The cases of $t_r \in [2T_{up}, 2t_{up})$ are then analysed. During this period, it is possible for the effector to maximise the velocity and the acceleration. When *Jerk* decreases with D_r , the system consumes the uniform acceleration phase to slow down the acceleration process, and the uniform acceleration phase gradually decreases. Therefore, within a certain range, the effector can maintain the maximum velocity and the maximum acceleration. The lower bound distance is $D_{J,A,V} = t_r V_m / 2$. Therefore, when $t_r \in [2T_{up}, 2t_{up})$ and $D_r \in [D_{J,A,V}, f_i(t_r))$, the effector can maintain the maximum velocity and the maximum acceleration. Equations (70) and (71) can be used to obtain a solution. As D_r decreases further, a smaller *Jerk* cannot saturate the acceleration and the velocity at $t_r/2$. Hence, $v_p = a_p t_r - a_p^2 / \tilde{J}$. When D_r decreases even further and if $a_p = A_m$ is maintained, v_p gradually becomes less than V_m . If $v_p = V_m$ is maintained independent of changes to a_p and \tilde{J} , the equation cannot hold. So when D_r becomes smaller, the effector starts to reduce the vertex velocity and keeps the acceleration saturated. The motion in this case includes variable acceleration and uniform

acceleration phases. In this case, the lower bound of the distance spends all of the time on the variable acceleration motion and the vertex acceleration is A_m .

$$j = \frac{4A_m}{t_r}$$

$$D_{J,A} = j \frac{t_r^3}{32} = \frac{A_m}{8} t_r^2 \quad (78)$$

Hence, when $t_r \in [2T_{up}, 2t_{up})$ and $D_r \in [D_{J,A}, D_{J,V})$, the kinematic equation can be described as:

$$\begin{aligned} T_1 &= T_3 = T_5 = T_7 = t_j \\ T_2 &= T_6 = (t_r - 4t_j)/2 \\ T_4 &= 0 \\ U &= Dir \cdot [\tilde{J} \quad 0 \quad -\tilde{J} \quad 0 \quad -\tilde{J} \quad 0 \quad \tilde{J}] \quad (79) \end{aligned}$$

where the:

$$\begin{cases} v_p = \frac{2D_r}{t_r} \\ \tilde{J} = \frac{2A_m^2}{(A_m t_r - 2v_p)} \\ t_j = \frac{A_m}{\tilde{J}} \end{cases}$$

When D_r further reduces within the interval $[0, D_{J,A})$, the acceleration cannot maintain its maximum value. Therefore, the kinematic equation can be described and solved by (77).

The case of $t_r \in [4T_j, 2T_{up})$ is considered below. In this case, due to the short time, the velocity cannot increase to V_m but the acceleration can still increase to A_m . Hence, for $D_r \in [D_{J,A}, f_t(t_r))$, (78) and (79) can be used to solve the problem. When D_r continues to decrease, (77) can be used to solve the problem.

Finally, the case of $t_r \in (0, 4T_j)$ is considered. As the time in this case is too short, the acceleration cannot be maximized. Then, for $D_r \in [0, D_{J,A})$, (77) can be used to solve the problem.

Hence, f_J^{-1} can be further transformed into (80), as shown at the bottom of this page.

D. TIME-CONSTRAINED MULTI-AXIS SYNCHRONIZATION ALGORITHM

In the above section, we deduced the inverse mapping f_*^{-1} for different bandwidth parameters. f_*^{-1} can solve a set of smaller bandwidth parameters to allow the effector to use any given time t_r to complete the S_r motion. In this section, a set of multi-axis synchronization algorithm (MASA) is designed

based on the inverse mapping deduced above. The algorithm is shown below.

Algorithm 1 MAS Algorithm

Input: S, V, A, J, n

- 1: *Init* $t = 0$;
- 2: **for** $i = 1$ **to** n **do**
- 3: $t = \max(f(S_i, V_i, A_i, J_i), t)$;
- 4: **end for**
- 5: **for** $i = 1$ **to** n **do**
- 6: $U_i, T_i = f_*^{-1}(t, S_i, V_i, A_i, J_i)$;
- 7: **end for**
- 8: **return** U, T ;

Here, n is the number of effectors. S, V, A, J is an n -dimension array. S_i, V_i, A_i, J_i represent the target displacement, maximum velocity, maximum acceleration and maximum *Jerk* of the i th effector, respectively. U contains n 7-dimensional vectors, which store the output series of each effector. T contains n 7-dimensional vectors, which store the time series of each effector. f_*^{-1} denotes different inverse solutions.

In Reference [21], it was pointed out that an increase in *Jerk* increases the motor error. In reference [22], it was proposed that minimizing energy *Jerk* should be the goal of trajectory planning. In reference [22] introduces two optimization targets.

$$\min \max |U| \quad (81)$$

The optimization goal (81) can minimize the system input and further reduce the error. Another optimization target is:

$$\min \int_0^T U^2 dt \quad (82)$$

The optimization target (82) can minimize the energy consumption of the effector.

But the problem based on energy optimization is extremely complicated. Many scholars here use intelligent algorithms [17], [18], iterative algorithms [9], [23], or numerical methods [24] to solve. In the [25], the author gave a more rigorous mathematical derivation and theoretical proof, which cleverly transformed the energy optimal problem into a convex problem, and Solved the optimal trajectory.

To address the multi-axis synchronization issue, multiple targets, such as synchronous motion, time optimal and minimum *Jerk*, are linearly combined into an objective function.

$$f_J^{-1} = \begin{cases} \text{no solution} & f_t(t_r) < |S_r| \\ \text{formula (70), (71)} & D_{J,A,V} \leq |S_r| \leq f_t(t_r) \\ \begin{cases} \text{formula (74), (75)} \\ \text{formula(79)} \end{cases} & \begin{aligned} & 2t_{up} \leq t_r \text{ and } D_{J,V} \leq |S_r| < D_{J,A,V} \\ & 2t_{up} > t_r \text{ and } D_{J,A} \leq |S_r| < D_{J,A,V} \end{aligned} \\ \text{formula(77)} & \text{others} \end{cases} \quad (80)$$

Numerical or intelligent algorithms [9], [17], [18] have been used to optimize the objective function. These algorithms have clear logic and mature computing modules. However, it is difficult to guarantee the operating rate and convergence of these methods. Moreover, it is almost impossible to implement and use these methods in real-time systems. When facing the P2P multi-axis synchronization issue, the algorithm proposed in this paper is strict, concise, easy to implement, complete and stable. As the algorithm directly gives a closed solving equation, the computational speed is much faster than the iterative method.

In addition, if f_J^{-1} is used in MASA, multi-axis synchronization and optimization target (81) can both be achieved. The acceleration is minimized when f_A^{-1} is used. But the following derivations can be obtained:

$$\min A = \tilde{A} = \frac{1}{4} \int_0^T |U| dt \quad (83)$$

In other words, f_A^{-1} minimizes A. According to the definitions of Euclidean and McHarden distances, it can be proven that:

$$\min \sqrt{\int_0^T U^2 dt} < \min \int_0^T |U| dt \quad (84)$$

The relationship of the size between $\min \int_0^T |U| dt$ and $\min \int_0^T U^2 dt$ is hard to determine. However, both optimization targets minimize energy consumption and only use unused metrics. Under the measure of L1 norm, f_A^{-1} achieves the optimization.

V. SIMULATION AND EXPERIMENT

This chapter consists of three parts. In part 1, each inverse mapping described in Chapter III is separately tested to verify the correctness and completeness of the algorithm. In part 2, a six-degree-of-freedom serial robots is built in a simulation environment to investigate its effect on the actual application platform. In the third experiment, this paper arranges the algorithm to run in a real-time system. To verify that the algorithm has good portability, efficiency and accuracy in real-time system.

This article does not arrange comparison experiments with other algorithms for the following reasons. Firstly, the principle of minimum value and mathematical derivation prove that under the premise of the same order, the algorithm in this paper is time-optimized. Secondly, there exists no iteration or optimization process in this algorithm, which can generate the high computational efficiency. Thirdly, in the field of multi-axis synchronization algorithms, there are few closed-form algorithms. At last, it is difficult to implement and use the current complex synchronization algorithms in real-time systems.

In addition, as the method proposed in this paper is analytical, it is easy to verify that the mapping is continuous, whether it is a numerical verification method or a derivative method.

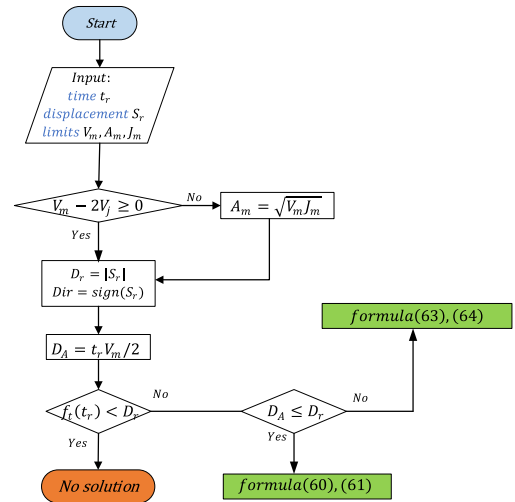


FIGURE 5. Logic diagram of the minimum acceleration synchronization algorithm under time constraints.

Therefore, no continuous verification experiment is carried out.

A. ALGORITHM SIMULATION EXPERIMENT

During the simulation experiment, the relevant algorithms are implemented in MATLAB, and a series of verification experiments is conducted. The simulation environment is a 64-bit Windows 10 operating system with MATLAB 2016b.

1) VERIFICATION OF BASIC MAPPINGS

In this experiment, the three inverse operations f_V^{-1} , f_A^{-1} and f_J^{-1} proposed in Sections IV.A to IV.C and the F proposed in Section III are implemented. Multiple datasets are used to verify the relationship between the four mappings. Then, data analyses of the three inverse mappings are carried out based on the relevant theories described above.

The first experiment is performed to verify the correctness of each mapping. That is, the experiment is conducted to verify whether the correct optimal curve can be solved with a set of given parameters.

The parameter settings are the following: $V_m = 20$, $A_m = 20$, $J_m = 30$ and target displacement $S_r = 100$. This set of data is directly introduced into the mapping F , and then $t_r = \text{sum}(T) = 6.6667$ can be calculated. Then, V_m, A_m, J_m, t_r are introduced to the three sets of inverse mappings. The four methods could be used to obtain the same time series and input series:

$$\begin{aligned}
 T &= [0.6667 \ 0.3333 \ 0.6667 \ 3.3333 \ 0.6667 \ 0.3333 \ 0.6667], \\
 U &= [30 \ 0 \ -30 \ 0 \ -30 \ 0 \ 30].
 \end{aligned}$$

According to the parameters, (10) is used to plot the time variation of position, velocity, acceleration and *Jerk*, as shown in Figure 6.

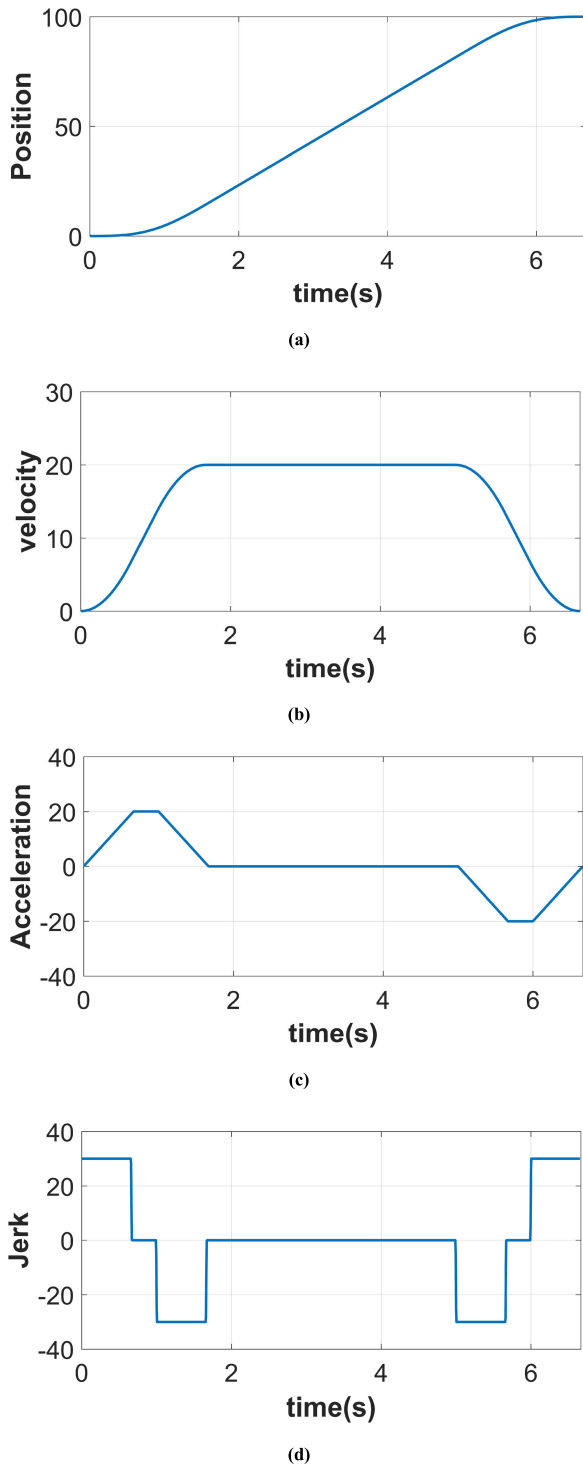


FIGURE 6. Position and derivative graphs with given parameter settings. (a) Displacement of the seven-segment S curve over time. (b) Velocity of the seven-segment S curve over time. (c) Acceleration of the seven-segment S curve over time. (d) Jerk of seven-segment S curve over time.

It can be seen that as the target displacement is large enough, the velocity and acceleration reach their maximum values. Figure 6(d) shows that the input also stays at its maximum value when activated, which means that the S-curve is

time optimal. In addition, all of the three inverse mappings are also calculated correctly with a given time t_r . This also ensures that the second cycle of MASA can be solved correctly. It is worth noting that S_r is the maximum displacement within t_r and the three inverse operations use different kinematic equations to obtain the same curve. This indicates that the inference in the inverse operation is correct.

When $S_r = -100$, it can be obtained that:

$$T = [0.6667 \ 0.3333 \ 0.6667 \ 3.3333 \ 0.6667 \ 0.3333 \ 0.6667],$$

$$U = [-30 \ 0 \ 30 \ 0 \ 30 \ 0 \ -30].$$

It can be seen that T remains non-negative and the sign of U changes. This indicates that the simplified framework of the vector displacement S_r to the scalar distance D_r proposed in this paper is correct.

After that, the parameters are set to $V_m = 20A_m = 20$, $J_m = 30$ and target displacement to $S_r = 25$. As the displacement is small, according to the algorithm designed in Figure 2, (29) can be used for the calculation, as shown in Figure 7.

It can be seen from the above set of data that the time for uniform acceleration is reduced. In this case, the vertex velocity is lower than V_m and the other bandwidth parameters reach their maximum values. In the literature, to save the contour of the curve, proportional scaling or iterative search is used to find a set of feasible parameter settings. These methods not only consume extra time and space, but also cannot ensure time optimality. The method proposed in this paper can efficiently solve the optimal parameter settings simply by solving quadratic equations.

Similarly, with $S_r = 10$, a curve can be obtained, as shown in Figure 8.

From this set of data, it can be seen that as the target displacement is further reduced, the acceleration no longer needs to reach the maximum bandwidth but the mapping F can still solve feasible and optimal parameter settings.

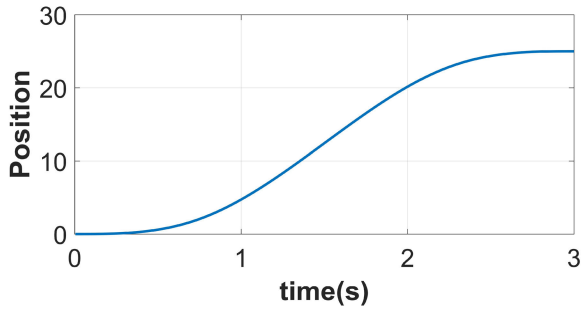
Finally, with $S_r = 0$, it can be obtained that:

$$T = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0],$$

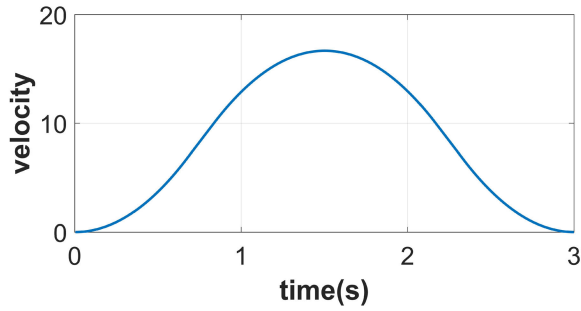
$$U = [30 \ 0 \ -30 \ 0 \ -30 \ 0 \ 30].$$

$S_r = 0$ means that the effector is not moving. This is common in multi-axis synchronization scenarios. As is shown above, although the input series contains non-zero terms, the entire non-negative time series is summed to zero. This also shows that the mapping F can correctly solve $S_r = 0$.

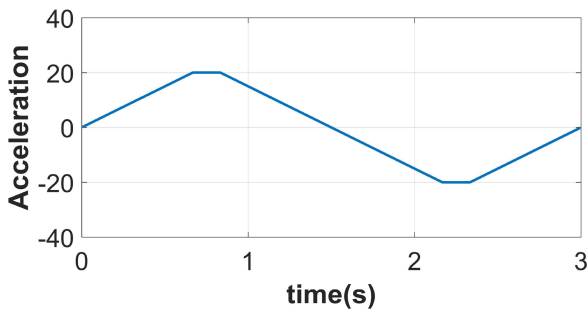
This experiment confirms the correctness, continuity and optimality of the mapping F and the correctness of the distance assumption. This method features a higher speed and lower computational complexity as compared to legacy algorithms.



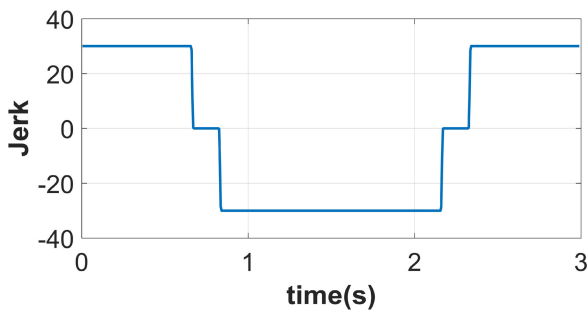
(a)



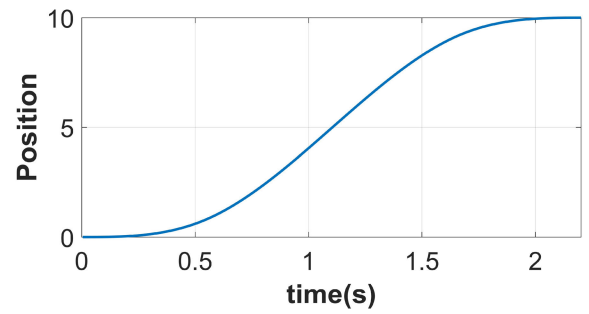
(b)



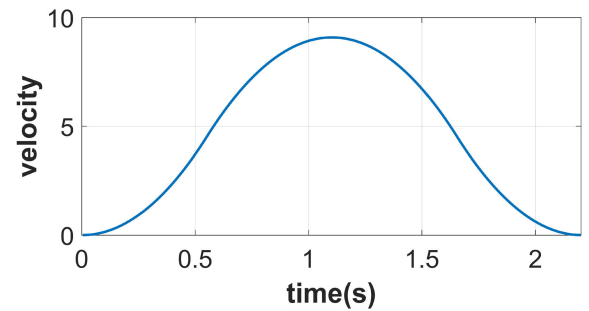
(c)



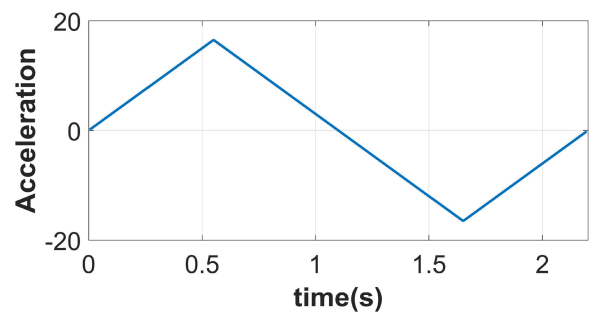
(d)



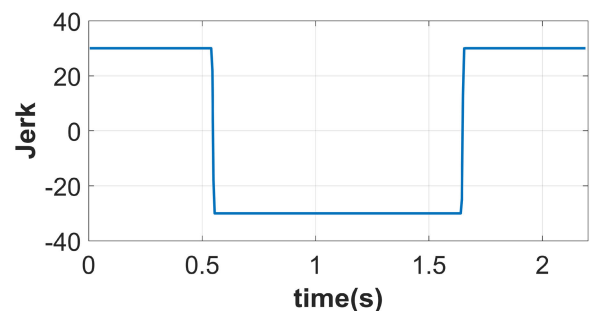
(a)



(b)



(c)



(d)

FIGURE 7. Positions and derivative graphs with $S_r = 25$. (a) Displacement of the seven-segment S curve over time. (b) Velocity of the seven-segment S curve over time. (c) Acceleration of the seven-segment S curve over time. (d) Jerk of seven-segment S curve over time.

FIGURE 8. Positions and derivative graphs with $S_r = 10$. (a) Displacement of the seven-segment S curve over time. (b) Velocity of the seven-segment S curve over time. (c) Acceleration of the seven-segment S curve over time. (d) Jerk of seven-segment S curve over time.

2) FURTHER VERIFICATION OF INVERSE MAPPINGS

In this experiment, a series of simulations are performed to verify the correctness and completeness of the three inverse mappings and to compare the differences between them.

According to the analysis above, when the bandwidth is determined, given a large S_r , the unique shortest time t_m can be determined. In the following experiments, the bandwidth parameter settings are fixed, and S_r and t_r are continuously

changed to observe the changes in the three inverse mappings. The fixed bandwidth parameter settings are the following: $V_m = 20$, $A_m = 20$ and $J_m = 30$.

The inverse mapping f_V^{-1} is first tested. According to the data in experiment V.A.1, with $S_r = 100$, the shortest time can be calculated as $t_m = 6.6667$. The parameter values of $S_r = 100$ and $t_r = 7$ are substituted into f_V^{-1} to solve the following kinematic series:

$$T = [0.6667 \ 0.2577 \ 0.6667 \ 3.8179 \ 0.6667 \ 0.2577 \ 0.6667],$$

$$U = [30 \ 0 \ -30 \ 0 \ -30 \ 0 \ 30].$$

The above series is plotted in Figure 9.

It can be seen that both the acceleration and *Jerk* reach the maximum bandwidth. But as there is more time, the vertex velocity drops in motion. In addition, it can be seen from the figure that the position curve is second-order differentiable and third-order continuous.

t_r is further increased to 20 below. We can solve the kinematic series in this case:

$$T = [0.4170 \ 0 \ 0.4170 \ 18.3319 \ 0.4170 \ 0 \ 0.4170],$$

$$U = [30 \ 0 \ -30 \ 0 \ -30 \ 0 \ 30].$$

From the displacement graph in Figure 10(a), it can be clearly seen that there is a longer uniform motion period. As can be seen from the velocity curve in Figure 10(b), the system only increases the velocity to 5.2176. From Figure 9(c), the acceleration of the curve does not increase to its maximum value. *Jerk* still maintains the maximum bandwidth. In Figure 10(b), it can be seen that the contour of the velocity curve is close to a trapezoid.

The inverse mapping f_A^{-1} is verified below. Assume that $S_r = 100$ and $t_r = 7$. The following series can be solved by f_A^{-1} :

$$T = [0.4226 \ 1.1547 \ 0.4226 \ 3.0000 \ 0.4226 \ 1.1547 \ 0.4226],$$

$$U = [30 \ 0 \ -30 \ 0 \ -30 \ 0 \ 30].$$

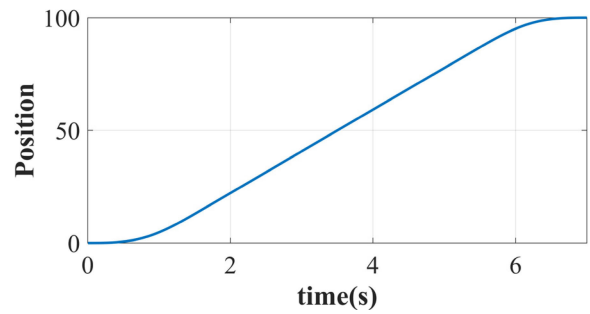
It can be seen that both the velocity and *Jerk* reach the maximum bandwidth. But as there is more time, this makes the vertex acceleration in motion decrease. In addition, it can be seen from the figure that the position curve is second-order differentiable and third-order continuous.

After the execution time t_r is further extended to 20, the following kinematic series and graphs can be solved:

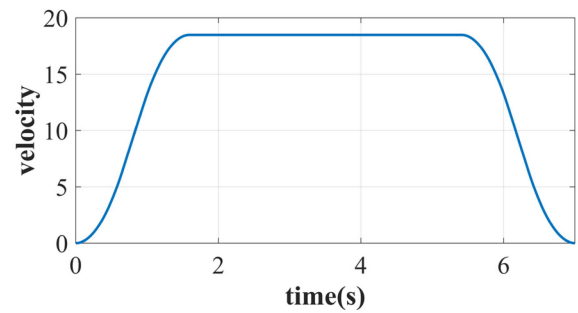
$$T = [0.0334 \ 9.9331 \ 0.0334 \ 0 \ 0.0334 \ 9.9331 \ 0.0334],$$

$$U = [30 \ 0 \ -30 \ 0 \ -30 \ 0 \ 30].$$

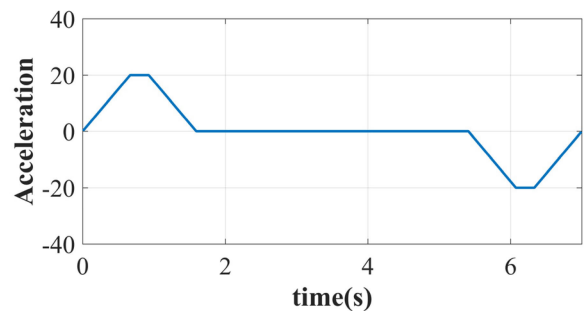
Figure 12(a) shows that the displacement curve is very curvy, which indicates that the effector is in a variable motion for a long time. Figure 12(b) shows that the effector is in a uniform acceleration phase for a long time.



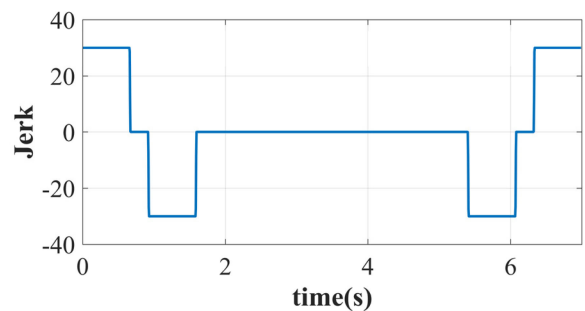
(a)



(b)



(c)



(d)

FIGURE 9. Kinematic graphs solved by f_V^{-1} with $S_r = 100$ and $t_r = 7$. (a) Displacement of the seven-segment S curve over time. (b) Velocity of the seven-segment S curve over time. (c) Acceleration of the seven-segment S curve over time. (d) Jerk of seven-segment S curve over time.

Figures 12(c) and 12(d) show that the effector starts a long uniformly accelerated motion after obtaining a low acceleration using the maximum *Jerk*. As analyzed earlier, the velocity contour in Figure 12(b) is close to a triangle.

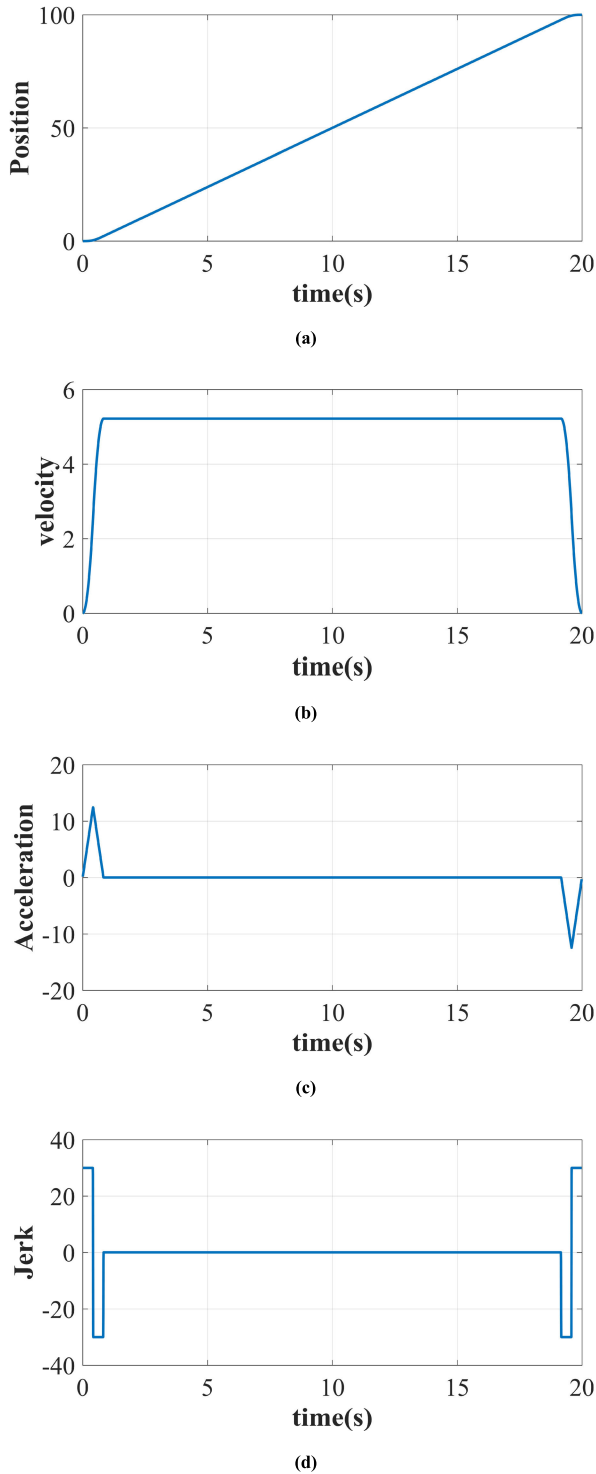


FIGURE 10. Kinematic graphs solved by f_V^{-1} with $S_r = 100$ and $t_r = 20$. (a) Displacement of the seven-segment S curve over time. (b) Velocity of the seven-segment S curve over time. (c) Acceleration of the seven-segment S curve over time. (d) Jerk of seven-segment S curve over time.

Assume that $S_r = 100$ and $t_r = 7$. The kinematic series solved by f_J^{-1} is:

$$T = \begin{bmatrix} 1 & 0 & 1 & 3 & 1 & 0 & 1 \end{bmatrix},$$

$$U = \begin{bmatrix} 20 & 0 & -20 & 0 & -20 & 0 & 20 \end{bmatrix}.$$

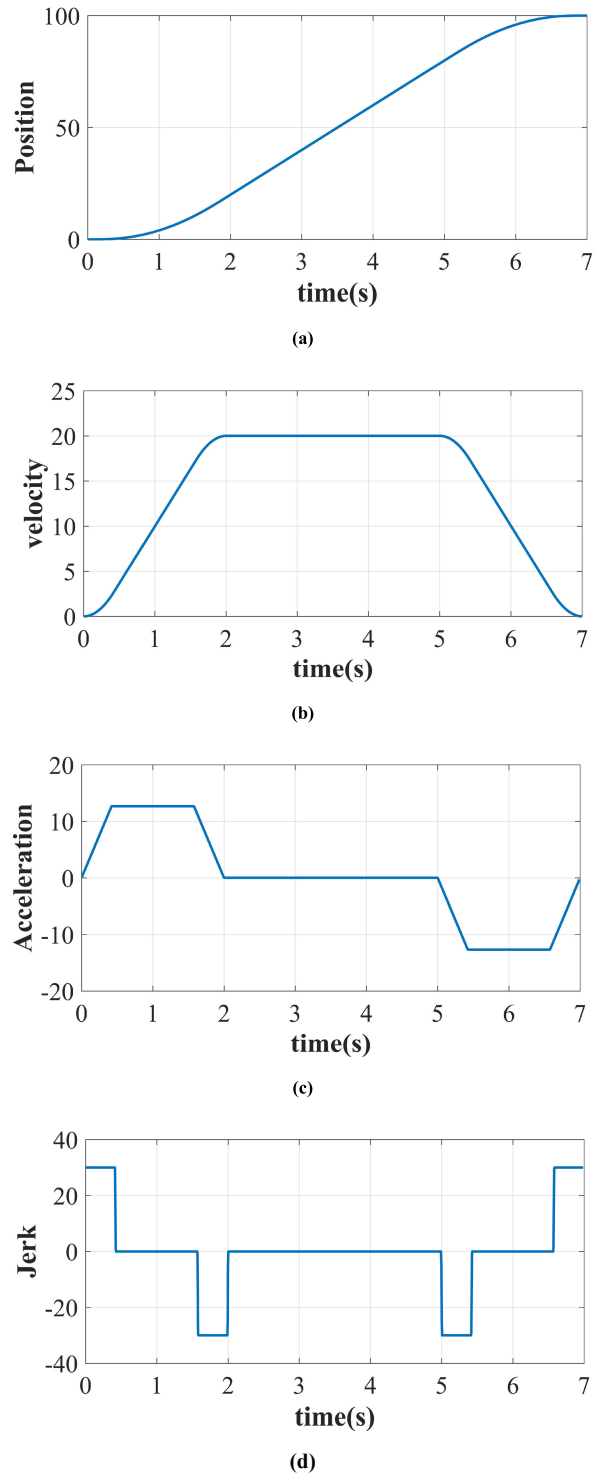
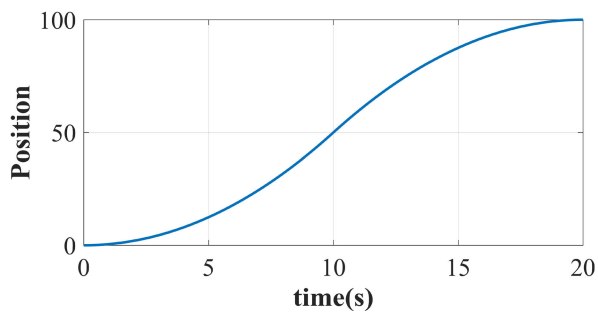


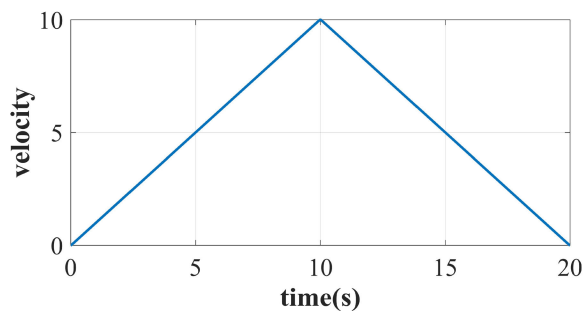
FIGURE 11. Kinematic graphs solved by f_A^{-1} with $S_r = 100$ and $t_r = 7$. (a) Displacement of the seven-segment S curve over time. (b) Velocity of the seven-segment S curve over time. (c) Acceleration of the seven-segment S curve over time. (d) Jerk of seven-segment S curve over time.

The above series is plotted in Figure 13.

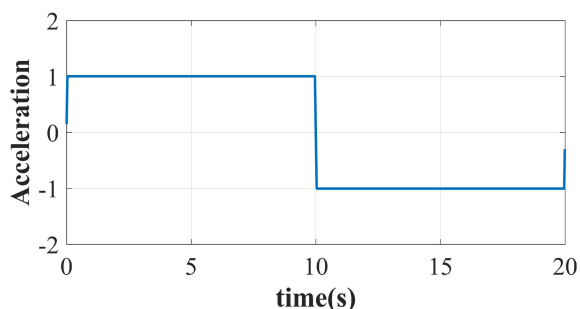
Figure 13(a) shows that the start-stop process of the effector is relatively smooth. Figures 13(b) and 13(c) show that both the velocity and acceleration reach their maximum values. Figure 13(d) shows that the Jerk of the executor is only 20 due to the increase in execution time.



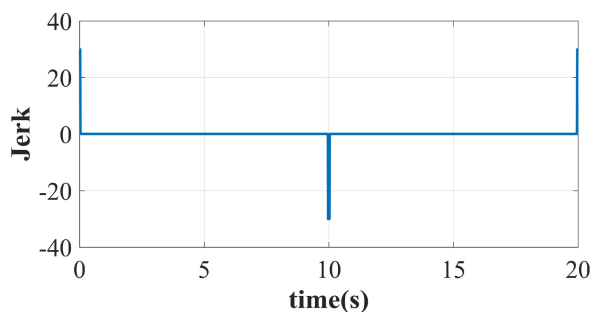
(a)



(b)



(c)



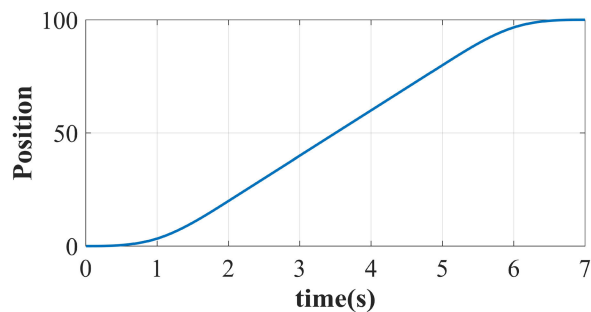
(d)

FIGURE 12. Kinematic graphs solved by f_A^{-1} with $S_r = 100$ and $t_r = 20$. (a) Displacement of the seven-segment S curve over time. (b) Velocity of the seven-segment S curve over time. (c) Acceleration of the seven-segment S curve over time. (d) Jerk of seven-segment S curve over time.

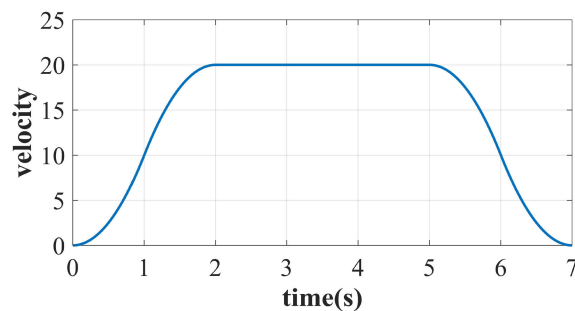
After the execution time is further extended to 20, the following kinematic series can be solved:

$$T = [5 \quad 0 \quad 5 \quad 0 \quad 5 \quad 0 \quad 5],$$

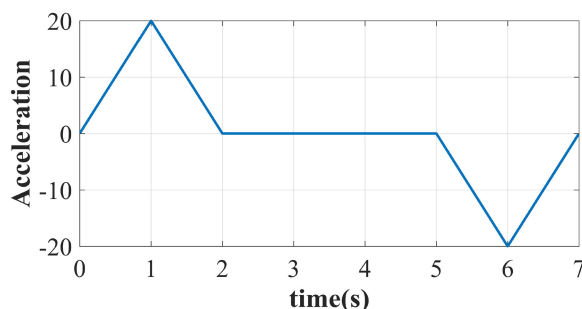
$$U = [0.4 \quad 0 \quad -0.4 \quad 0 \quad -0.4 \quad 0 \quad 0.4].$$



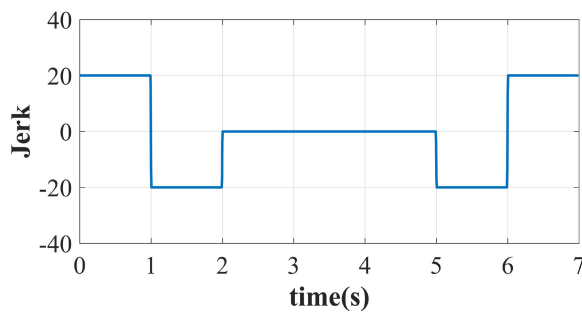
(a)



(b)



(c)



(d)

FIGURE 13. Kinematic graphs solved by f_J^{-1} with $S_r = 100$ and $t_r = 7$. (a) Displacement of the seven-segment S curve over time. (b) Velocity of the seven-segment S curve over time. (c) Acceleration of the seven-segment S curve over time. (d) Jerk of seven-segment S curve over time.

Their curve contours are shown in Figure 14.

It is clear from Figure 14(d) that the Jerk drops to 0.4. In addition, the entire motion process only contains a variable motion. Figure 14(a) shows that the entire curve is extremely

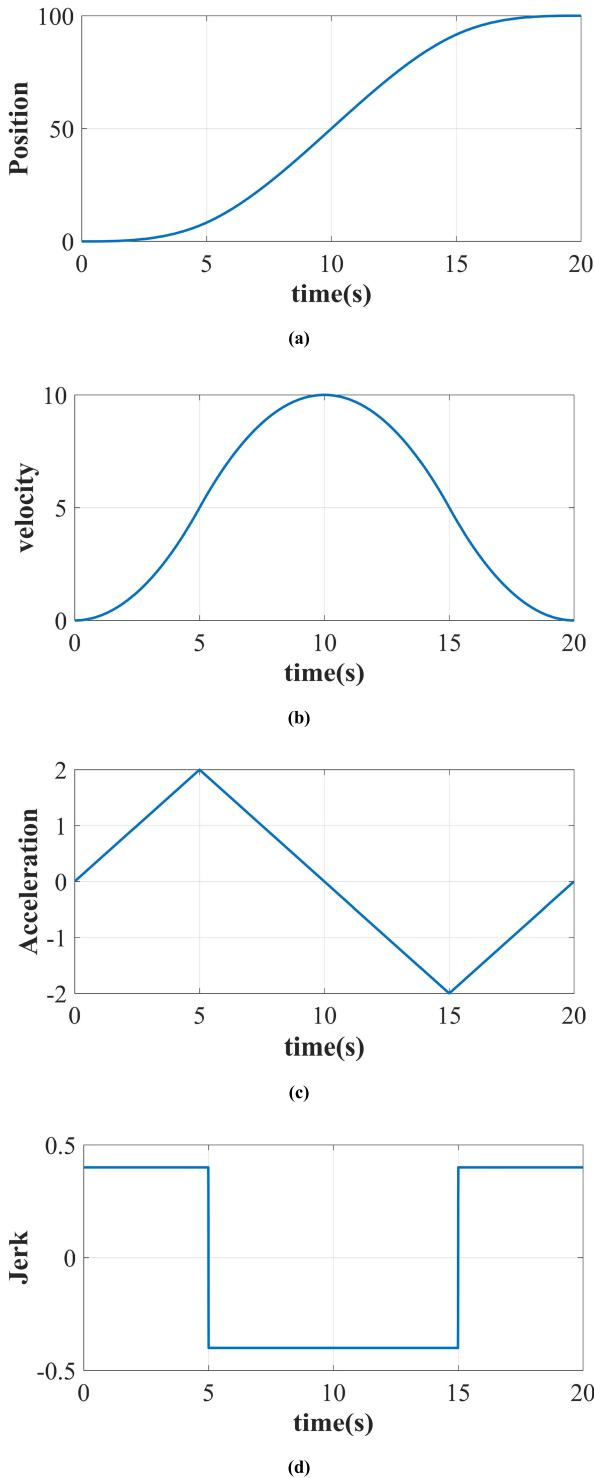


FIGURE 14. Kinematic graphs solved by f_r^{-1} with $S_r = 100$ and $t_r = 20$. (a) Displacement of the seven-segment S curve over time. (b) Velocity of the seven-segment S curve over time. (c) Acceleration of the seven-segment S curve over time. (d) Jerk of seven-segment S curve over time.

smooth. Figure 14(c) shows that the maximum acceleration of the effector is also only 2.

Finally, assume that $t_r = 1$ and $S_r = 0$. This scenario occurs when the current effector of MASA does not receive

a motion command, but other effectors need to move for 1 s. Then, all of the three inverse mappings should not lead to any motion. The calculation results are shown below.

$$\begin{cases} T_V = [0 & 0 & 0 & 1 & 0 & 0 & 0] \\ U_V = [0 & 0 & 0 & 0 & 0 & 0 & 0] \\ T_A = [0 & 0.5 & 0 & 0 & 0 & 0.5 & 0] \\ U_A = [0 & 0 & 0 & 0 & 0 & 0 & 0] \\ T_J = [0.25 & 0 & 0.25 & 0 & 0.25 & 0 & 0.25] \\ U_J = [0 & 0 & 0 & 0 & 0 & 0 & 0] \end{cases} \quad (85)$$

According to the above results, we can see that when $S_r = 0$, the three inverse mappings use different kinematic equations, and neither of them makes the effector move. This shows that the inverse mapping can correctly select the corresponding kinematic equation and calculate this special set of results.

In experiment V.A.2, a series of experiments were conducted with f_V^{-1} , f_A^{-1} and f_J^{-1} . Their implementations proved that the three inverse mappings can indeed solve the kinematic series correctly. At the same time, the related parameters can be optimized according to the above derivation.

B. ROBOT SIMULATION EXPERIMENTS

The standard DH parameter table of robot KingKong is given in Table 1.

TABLE 1. Table of standard DH parameters of KingKong.

No	d/m	a/m	α/rad
1	0.1135	0	$\pi/2$
2	0	-0.2436	0
3	0	-0.2632	0
4	0.0928	0	$\pi/2$
5	0.1103	0	$-\pi/2$
6	0.0568	0	0

The schematic diagram of KingKong’s coordinate system is shown in Figure 15.

1) VERIFICATION EXPERIMENT FOR THE JOINT SPACE

The bandwidth settings of the joint are the following: $V_{J,m} = 30deg/s$, $A_{J,m} = 40deg/s^2$, and $J_{J,m} = 80deg/s^3$. The robot is ordered to move from P_1 to P_2 .

$$\begin{aligned} P_1 &= [0^\circ \quad -150^\circ \quad 60^\circ \quad -45^\circ \quad -45^\circ \quad 45^\circ], \\ P_2 &= [100^\circ \quad -90^\circ \quad -60^\circ \quad -120^\circ \quad 45^\circ \quad 45^\circ]. \end{aligned}$$

Figure 16(a) shows that the curves are smooth and continuous, and they also reach the designated position accurately. Figure 16(b) shows that all six joints move synchronously, and the velocity curves are smooth and continuous.

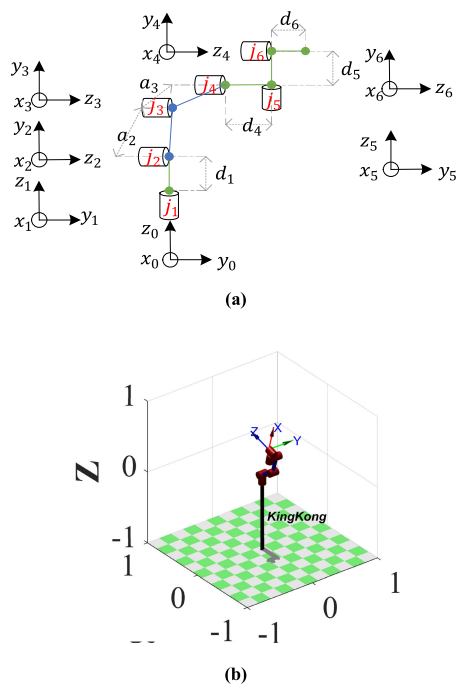


FIGURE 15. (a) Schematic for the KingKong parameters: d_k is the linkage length of the k th link and α_k is the linkage torsion of the k th link. J_k is the k th joint. (b) Visualization of the KingKong model.

According to the joint angle, the end position was recomputed with the motion trajectory shown in Figure 17.

2) VERIFICATION EXPERIMENT FOR THE DESCARTES SPACE

It is known that the interpolation of the robot's end-effector pose is difficult. Here, we combine the proposed MASA, Euler angles and parametric equations to solve this problem. According to basic robotic theories, the end transformation matrix T of a robot can be obtained through the joint angle and DH parameters. T belongs to a special Euclidean group. Through the transformation relationship between the rotation matrix and Euler angles, T can be mapped into a six-dimensional scalar to denote the pose of the robot end-effector. x, y, z denote the position P of the robot end-effector in the base coordinate system, which can be retrieved directly from T . α, β, γ are Euler angles solved by the rotation matrix of T . α, β, γ can be used to denote the attitude angle of the robot's end-effector in the base coordinate system.

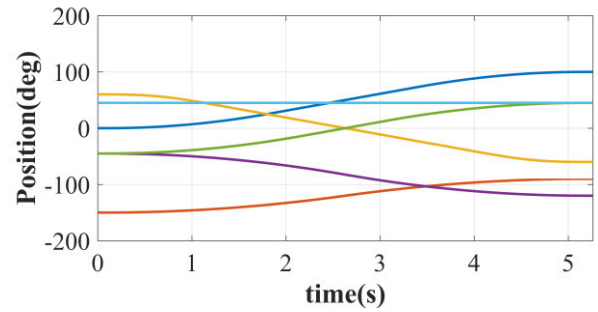
$$fkine(\vartheta_1, \vartheta_2, \vartheta_3, \vartheta_4, \vartheta_5, \vartheta_6) \mapsto T \in SE3$$

$$T \mapsto \xi = [x \quad y \quad z \quad \alpha \quad \beta \quad \gamma] \quad (86)$$

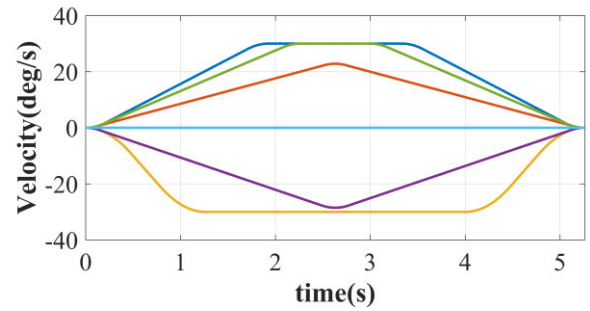
The above method can be used to determine the current pose ξ_c of the robot.

We shall first determine the target position P_r , the target attitude angle Q_r and the parametric equation Υ of the curve to be run.

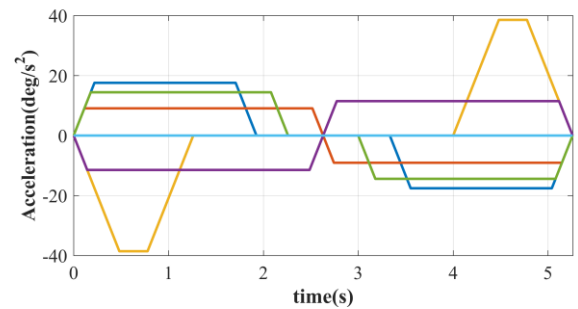
The length of the entire curve can be calculated using P_c, P_r, Υ, L . The space angle Q_Δ to be rotated can be calculated using Q_c and Q_r . Based on the bandwidth parameter



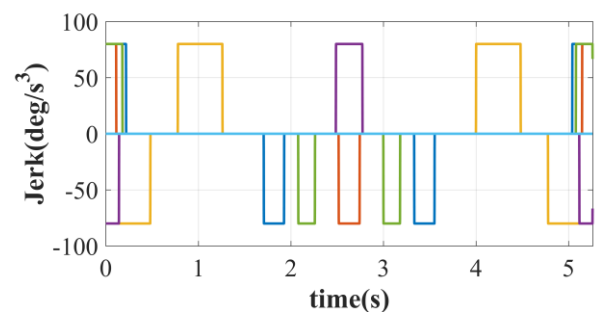
(a)



(b)



(c)



(d)

FIGURE 16. The motion trajectory of the KingKong. (a) Displacement of the seven-segment S curve over time. (b) Velocity of the seven-segment S curve over time. (c) Acceleration of the seven-segment S curve over time. (d) Jerk of seven-segment S curve over time. Axis 1 is the blue line. Axis 2 is the red line. Axis 3 is the yellow line. Axis 4 is the purple line. Axis 5 is the green line. Axis 6 is the light blue line.

settings given by the user, MASA is used with L and Q_Δ for smooth interpolation of the robot spatial pose. Four kinematic series could then be generated. The ξ_i of each cycle could be

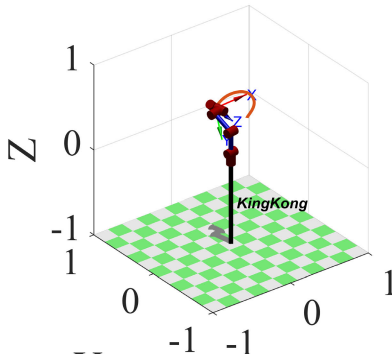


FIGURE 17. Start and end positions of KingKong’s motion.

calculated based on the kinematic series. After T_i is inversely calculated using the parametric equation Υ , it is then mapped back into the joint space by an inverse kinematics algorithm. Here, the inverse kinematic calculations and multiple solutions of the UR robot are chosen using the method described in [26].

The bandwidth parameters of the system in spatial motion are first set, where the end-effector velocity bandwidth is $V_{P,m} = 0.15 \text{ m/s}$, $A_{P,m} = 0.2 \text{ m/s}^2$ and $J_{P,m} = 0.3 \text{ m/s}^3$. The bandwidth of the spatial angle change is $V_{Q,m} = 30 \text{ deg/s}$, $A_{Q,m} = 40 \text{ deg/s}^2$ and $J_{Q,m} = 80 \text{ deg/s}^3$.

The initial position of the robot is at $[-90^\circ -150^\circ 45^\circ -90^\circ -90^\circ -90^\circ]$. The end-effector’s change matrix solved by forward kinematics is:

$$\begin{bmatrix} 0.0000 & 1.0000 & 0.0000 & -0.0928 \\ -0.2588 & 0.0000 & 0.9659 & -0.2528 \\ 0.9659 & 0.0000 & 0.2588 & 0.5710 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The pose is solved as:

$$[-0.0928 \ -0.2528 \ 0.5710 \ 90.0000 \ 75.0000 \ -180.0000].$$

The pose of the target is:

$$[-0.0928 \ -0.2528 \ 0.38 \ 120.0000 \ 120.0000 \ -160.0000].$$

The curve is a straight line, which can be calculated as follows: $L = 0.1910$, $Q_\Delta = [30.0000 \ -45.0000 \ 20.0000]$.

By introducing the data in to MASA, the following kinematic series can be solved in T_L , U_L , T_Q , and U_Q as shown at the bottom of this page.

We first plot the kinematic graphs of the robot’s end-effector pose; the results are shown in Figure 18.

Figure 18(a) shows that the robot’s end pose reaches the specified angle in a correct, smooth and synchronized way. The other panels of Figure 18 show that this planning process complies with the given system bandwidth.

It can be seen that the robot’s end-effector completes the specified position motion, and it complies with the specified bandwidth parameter settings. It can be seen from Figures 18 and 19 that the position motion and the pose motion are synchronized.

According to the pose obtained by synchronous planning, the joint angle series is inversely solved. The resulting plots are shown below.

It can be seen that the robot’s end-effector completes the specified position motion, and it complies with the specified bandwidth parameter settings. It can be seen from Figures 18 and 19 that the position motion and the pose motion are synchronized.

According to the pose obtained by synchronous planning, the joint angle series is inversely solved. The resulting plots are shown below.

Figure 20 shows that both the velocity and acceleration curves are continuous. An important advantage of this planning framework is that the entire planning process is analytical. As a result, any trajectories of the variable segments and the uniform segments in the Descartes space can be calculated. In other words, it can effectively guarantee the controllability of the trajectory space velocity when performing some specific processes.

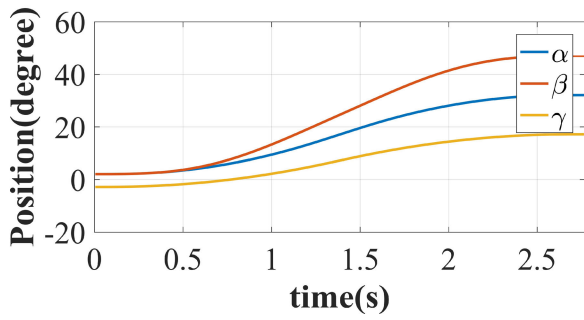
C. EXPERIMENT IN REAL-TIME SYSTEM

In the third experiment, this paper arranges the algorithm to run in a real-time system. To verify that the algorithm has good portability, efficiency and accuracy in real-time system.

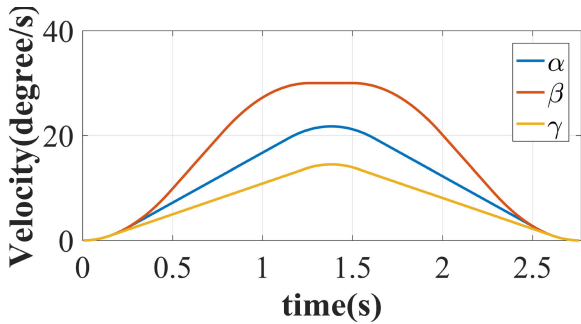
Our laboratory was commissioned to build an explosive ordnance disposal (EOD) robot with dual cooperative series robots. The controller we use is Beckhoff C-6920. This controller runs under the 32-bit Windows 7 operating system and provides the TwinCat, a real-time motion control platform. The controller works at 24V DC. As shown in Figure 21.

The motor of the series robot uses Kollmorgen RGM. RGM works at 48V DC. As shown in Figure 22.

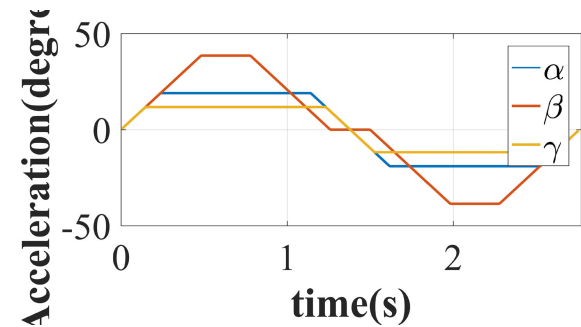
$$\begin{aligned} T_L &= [0.5682 \ 0.2435 \ 0.5682 \ 0 \ 0.5682 \ 0.2435 \ 0.5682] \\ U_L &= [0.3000 \ 0 \ -0.3000 \ 0 \ -0.3000 \ 0 \ 0.3000] \\ T_Q &= \begin{bmatrix} 0.2379 & 0.9041 & 0.2379 & 0 & 0.2379 & 0.9041 & 0.2379 \\ 0.4820 & 0.2960 & 0.4820 & 0.2400 & 0.4820 & 0.2960 & 0.4820 \\ 0.1469 & 1.0862 & 0.1469 & 0 & 0.1469 & 1.0862 & 0.1469 \end{bmatrix}, \\ U_Q &= \begin{bmatrix} 80.0000 & 0 & -80.0000 & 0 & -80.0000 & 0 & 80.0000 \\ 80.0000 & 0 & -80.0000 & 0 & -80.0000 & 0 & 80.0000 \\ 80.0000 & 0 & -80.0000 & 0 & -80.0000 & 0 & 80.0000 \end{bmatrix}. \end{aligned}$$



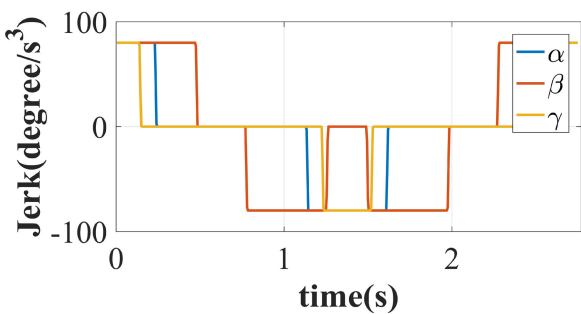
(a)



(b)



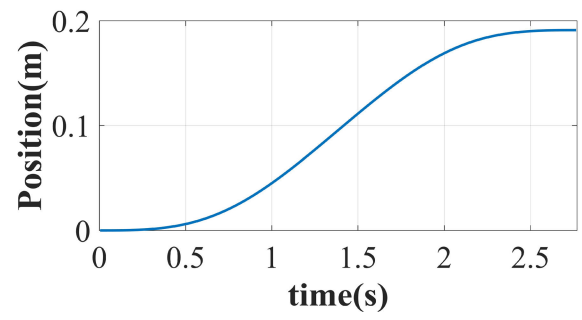
(c)



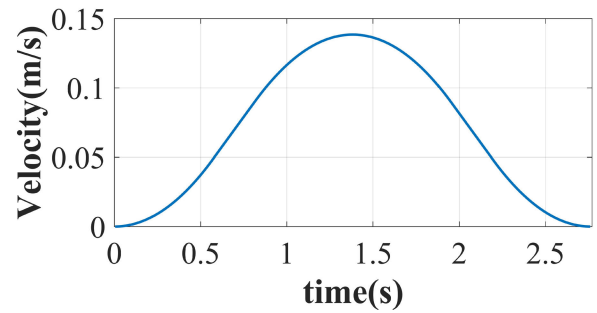
(d)

FIGURE 18. Kinematic graphs of KingKong's end-effector pose. (a) Displacement of the seven-segment S curve over time. (b) Velocity of the seven-segment S curve over time. (c) Acceleration of the seven-segment S curve over time. (d) Jerk of seven-segment S curve over time.

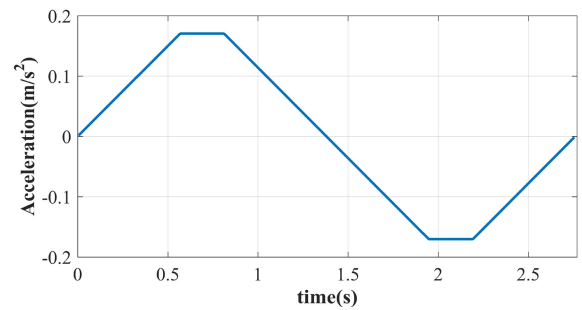
RGM communicates with C-6920 through CanOpen protocol. CANopen is a high-level communication protocol based on the control area network (Controller Area Network, CAN). It is often used in embedded systems and is also a field bus commonly used in industrial control.



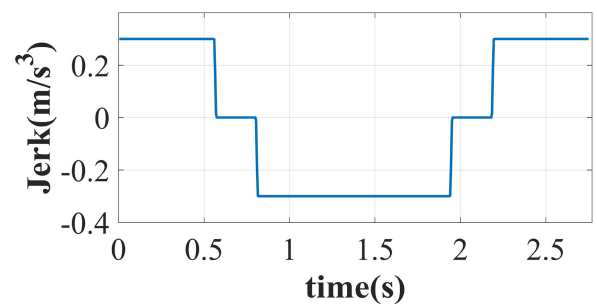
(a)



(b)



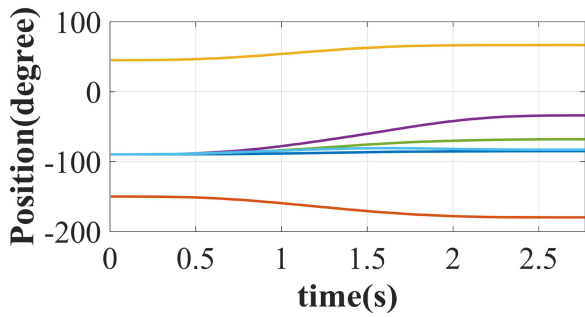
(c)



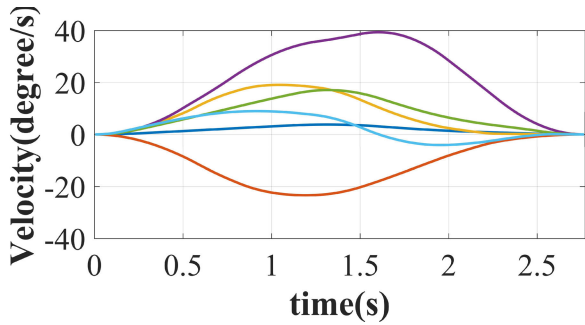
(d)

FIGURE 19. Kinematic graphs of the arc length at KingKong's joint spatial position. (a) Displacement of the seven-segment S curve over time. (b) Velocity of the seven-segment S curve over time. (c) Acceleration of the seven-segment S curve over time. (d) Jerk of seven-segment S curve over time.

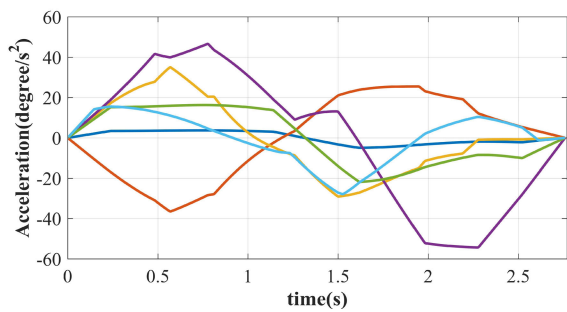
As shown in Figure 23(a) and Figure 23(b), EOD robot is composed of two serial manipulators, mobile platform and vision system. Additionally, EOD robot is characterized by a high-performance explosion-proof tank, which is convenient to handle the explosives in the process of detonation.



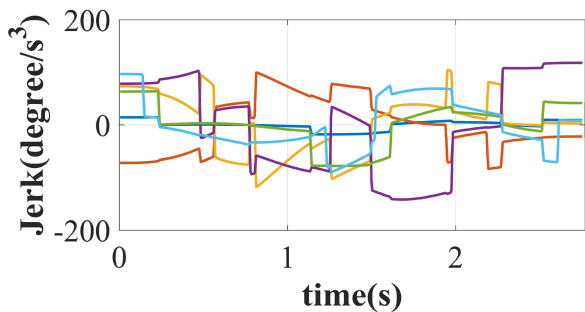
(a)



(b)



(c)



(d)

FIGURE 20. Kinematic graphs of the arc length at KingKong’s joint spatial position. (a) Displacement of the seven-segment S curve over time. (b) Velocity of the seven-segment S curve over time. (c) Acceleration of the seven-segment S curve over time. (d) Jerk of seven-segment S curve over time. Axis 1 is the blue line. Axis 2 is the red line. Axis 3 is the yellow line. Axis 4 is the purple line. Axis 5 is the green line. Axis 6 is the light blue line.

The software architecture of manipulators is shown in Figure 24.

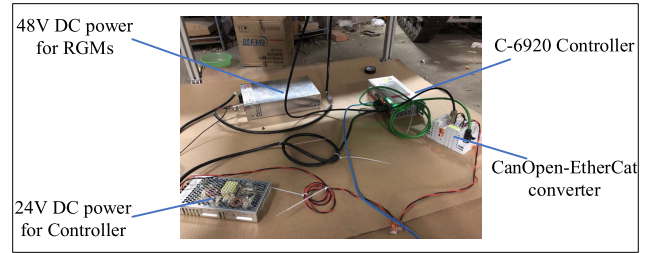
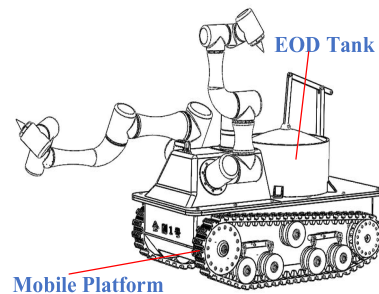


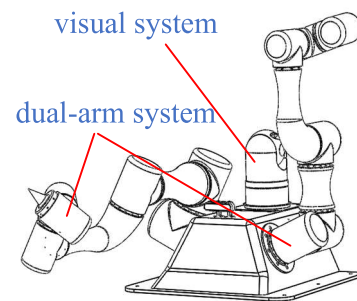
FIGURE 21. C6920 controller.



FIGURE 22. RGM of Kollmorgen.



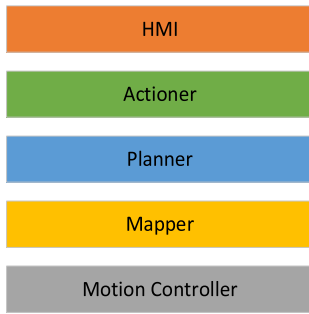
(a)



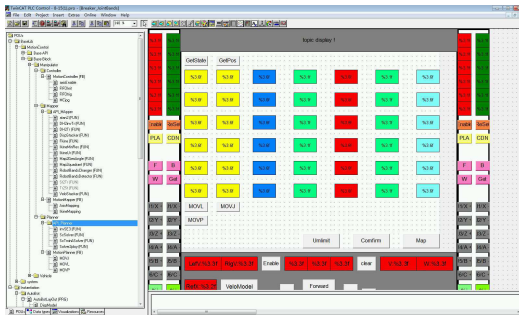
(b)

FIGURE 23. Model diagram of EOD robot.

A series of control commands and functions are provided in the Human Machine Interface (HMI) of the control terminal. These upper-level motion functions will be parsed into robot motion commands in **Actioner**. **Planner** then solves the corresponding motion sequence according to the upper-level motion instructions. The **Mapper** is responsible for converting the motion sequence into motor motion commands.



(a)



(b)

FIGURE 24. The software architecture of double manipulators.

Finally, the **MotionController** is responsible for communication and control with the motor. The MASA proposed in this article will be implemented in **Planner**. The dissipation of energy needs to be considered at any time because the battery is used for EOD robot in the complex environment. The multi-axis synchronization of EOD robot is using the f_A^{-1} mapping with energy optimal.

Based on the derivation above, Codesys language of Twin-Cat is used to achieve the MASA and the motion control module of robot, shown in Figure 25.

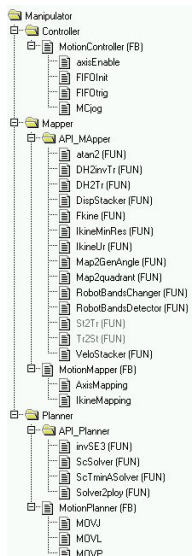


FIGURE 25. Algorithm interface of double manipulators system.

The P2P time optimal algorithm **ScSolver** and the anti-mapping **ScTminASolver** are completed in TwinCat. Related functions can be found in the folder of planner. Since the algorithm proposed in this paper only requires solving quadratic polynomial equations, the synchronization logic is clear and simple. Even if transplanted to the TwinCat platform, it does not require complicated programming and data structures. This is the larger advantage of the algorithm proposed in this paper.

EOD robot after assembly is shown in Figure 26:



(a)



(b)



(c)

FIGURE 26. EOD robot.

TABLE 2. Table of standard DH parameters of WuKong.

No	d/m	a/m	α/rad
1	0.1135	0	$\pi/2$
2	0	-0.425	0
3	0	-0.3749	0
4	0.0928	0	$\pi/2$
5	0.1103	0	$-\pi/2$
6	0.0568	0	0

EOD robot has two series robots. The robot on the left side is small and the DH performance has been shown in the Chapter V. B. The robot on the right side is named WuKong, and its DH performance is shown in the following table.

Additionally, there are offsets between the two robots and the origin of the coordinate system. The two offset matrices are:

$$Base_K = \begin{bmatrix} 1 & 0 & 0 & 0.07 \\ 0 & 0.4226 & 0.9063 & 0.1373 \\ 0 & -0.9063 & 0.4226 & 0.0894 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Base_W = \begin{bmatrix} 1 & 0 & 0 & -0.03 \\ 0 & 0.4226 & -0.9063 & -0.1346 \\ 0 & 0.9063 & 0.4226 & 0.0881 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

According to the original plan, the experiment of the EOD robot was indeed arranged here. However, due to the epidemic, we were unable to return to campus. In this regard, we take the following remedial measures to prove that the algorithm proposed in this paper has a good effect even if it is applied in a real-time platform.

Firstly, the TwinCat platform can turn any 32-bit Win7 system into a real-time controller. Our EOD robot has completed the communication and debugging of TwinCat and RGMs and been running normally for more than one year. As long as the TwinCat can output the correct position, the EOD robot can correctly execute the control instructions. Therefore, a 32-bit Win7 Industrial Personal Computer (IPC) with a lower configuration is used as the controller in this chapter. In the following experiment, this paper will analyze the performance of the algorithm in real-time system according to the closed-loop period of the controller and the control signal.

Secondly, since this set of algorithms and robot motion framework has been implemented in the TwinCat platform in 2018. During this year there are also some experiment and testing. For example, the synchronization framework of this paper has been used in [26] to demonstrate the running effect of KingKong. In addition, we also submitted some previous videos. Hope to show you the effect of the algorithm applied to the series robot.

The configuration of IPC is Win 7 system of 32-bit, celeron CPU. The closed-loop cycle of TwinCat in IPC is 2ms, that is to say TwinCat will communicate with the driver once every 2 ms. During this period, the expected positions of the actuator at the next time would be calculated.

1) EXPERIMENT OF MULTI-POINT IN JOINT SPACE

First set the three via points of WuKong:

$$P_1 = [180^\circ \quad -30^\circ \quad 45^\circ \quad -90^\circ \quad -90^\circ \quad -45^\circ];$$

$$P_2 = [90^\circ \quad -150^\circ \quad 45^\circ \quad -90^\circ \quad -90^\circ \quad -45^\circ];$$

$$P_3 = [270^\circ \quad -150^\circ \quad 45^\circ \quad -90^\circ \quad -90^\circ \quad -45^\circ];$$

The motion bandwidth of WuKong is the same as that of KingKong.

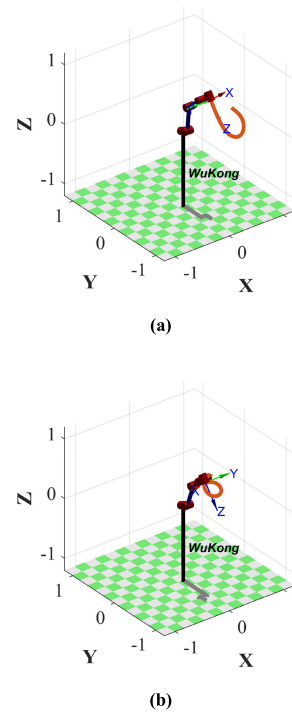


FIGURE 27. Start and end positions of WuKong's motion. (a) shows the trajectory between P1 and P2. (b) shows the trajectory between P2 and P3.

The position points planned in the controller can draw the curve of the robot end trajectory and each joint.

First of all, as is shown in Figure28 (a), (b), (c), (d), each joint has completed synchronous movement, and in the process, each order curve does not exceed the bandwidth parameter. It can be seen from the position curve that each joint has reached the specified position correctly. The speed curve is smooth, and the acceleration curve is continuous. And Jerk has always maintained the maximum output. Otherwise the Jerk curve is symmetrical. In addition, the TwinCat also runs normally without any abnormalities. This shows that the algorithm can run smoothly in a closed loop of 2ms.

2) EXPERIMENT OF MULTI POINT MOTION IN CARTESIAN SPACE

After that, KingKong will be ordered to complete a set of spatial movements to implement KingKong's tool change process.

As shown in Figure 29. KingKong has replaceable knives at the end, which can help replace different knives through the equipment box next to the tank. Among them, O is the origin of the car body coordinate system $(0 \ 0 \ 0)$. The entire tool change process is as follows.

First of all, the end of KingKong moves from any location to point $A(-0.2719 \ 0.157 \ 0.3)$ above the first tool, and then moves to point $B(-0.2719 \ 0.157 \ 0.1)$ in a straight line. After placing tool 1, KingKong will move to point A in a straight line again. At this time, there is no tool at

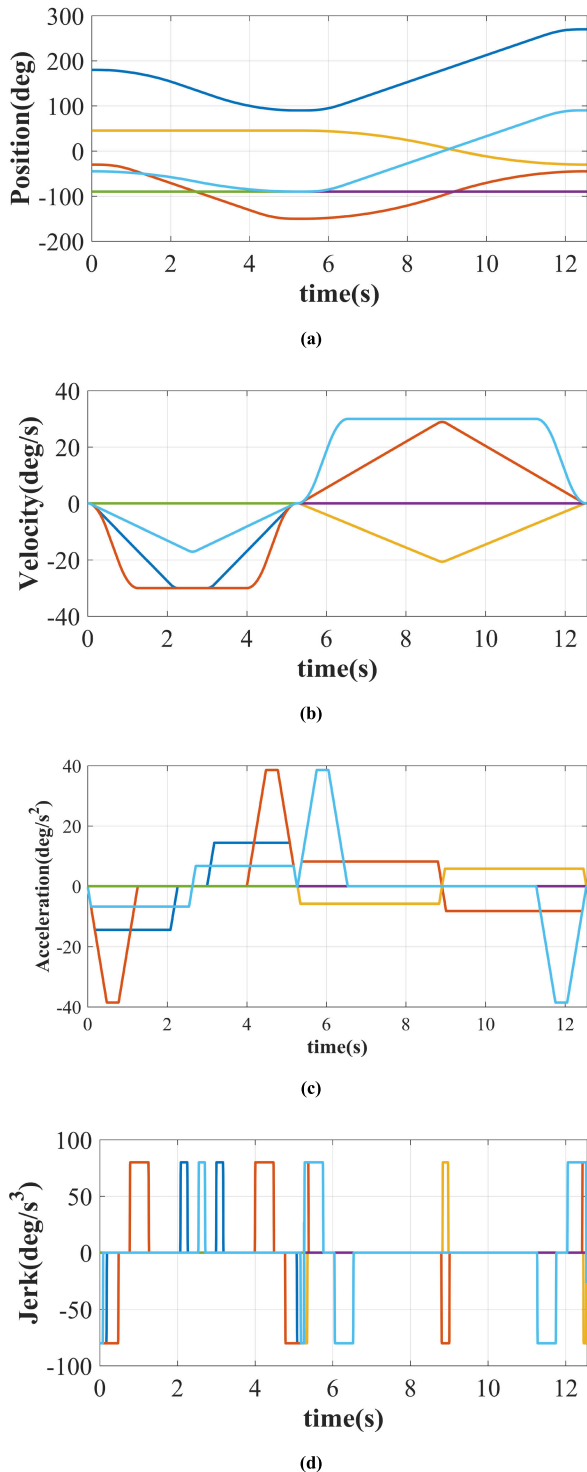


FIGURE 28. The motion trajectory of the WuKong. (a) Displacement of the seven-segment S curve over time. (b) Velocity of the seven-segment S curve over time. (c) Acceleration of the seven-segment S curve over time. (d) Jerk of seven-segment S curve over time. Axis 1 is the blue line. Axis 2 is the red line. Axis 3 is the yellow line. Axis 4 is the purple line. Axis 5 is the green line. Axis 6 is the light blue line.

the end of the KingKong. Subsequently, the robot moves to point C (−0.3484 0.23 0.3) in joint space where is above tool 2. Along with the straight line, it descends to point

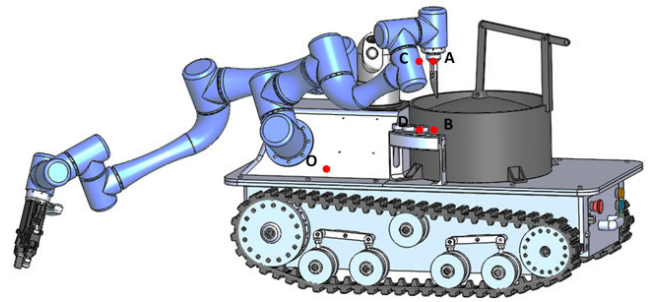


FIGURE 29. Key points of the tool change process.

D (−0.3484 0.23 0.1). After replacing the tool 2, it moves to point C in a straight line. The whole tool change process is over.

Four space locations are obtained with the key space points from the description above.

$$\begin{aligned} \xi_A &= [-0.2719 \quad 0.157 \quad 0.3 \quad 0 \quad \pi/2 \quad 0]; \\ \xi_B &= [-0.2719 \quad 0.157 \quad 0.1 \quad 0 \quad \pi/2 \quad 0]; \\ \xi_C &= [-0.3484 \quad 0.23 \quad 0.3 \quad 0 \quad \pi/2 \quad 0]; \\ \xi_D &= [-0.3484 \quad 0.23 \quad 0.1 \quad 0 \quad \pi/2 \quad 0]; \end{aligned}$$

The initial location of KingKong is:

$$P_1 = [90^\circ \quad -150^\circ \quad 45^\circ \quad -90^\circ \quad -90^\circ \quad -90^\circ];$$

Use the bandwidth parameter from the previous experiment. The planned position of each joint can be obtained in the controller. Using forward kinematics, the trajectory of the robot end in Cartesian space can be drawn.

The end of robot moves along with the correct special trajectory and complete the tool change, shown in Figure 30.

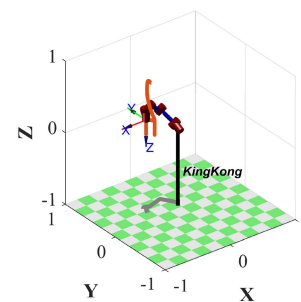


FIGURE 30. Spatial trajectory of tool change process.

From Figure 31(a), (b), (c), (d), we can see the continuous and smooth location curve, speed curve and acceleration curve. In addition, the curve in joint space is stable when the proposed algorithm and framework are operated in general cartesian space. The risk of motion shaking can be avoided by directly reducing the parameter of bandwidth.

However, there are still some major issues that need to be discussed and further studied.

Firstly, Figure 20(d) and 31(d) show that although the Descartes space curve is regular and stable, the Jerk curve

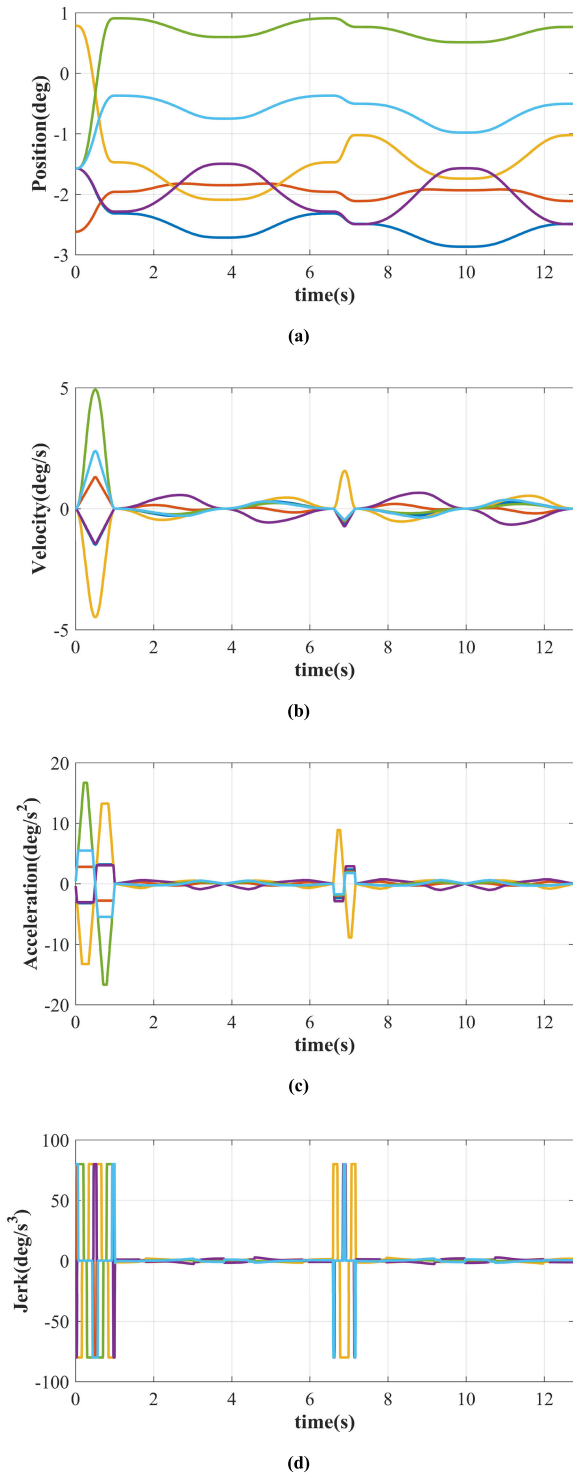


FIGURE 31. Kinematic graphs of the arc length at WuKong's joint spatial position. (a) Displacement of the seven-segment S curve over time. (b) Velocity of the seven-segment S curve over time. (c) Acceleration of the seven-segment S curve over time. (d) Jerk of seven-segment S curve over time. Axis 1 is the blue line. Axis 2 is the red line. Axis 3 is the yellow line. Axis 4 is the purple line. Axis 5 is the green line. Axis 6 is the light blue line.

mapped back to the joint space has a huge hop. This problem is caused by the third-order S-curve Jerk hopping. In the experiment described in [26], the third-order S-curve is

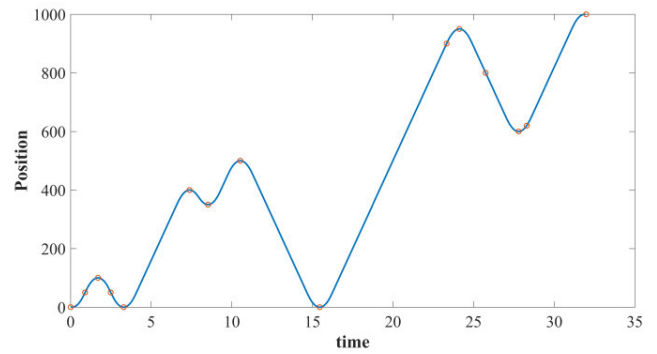


FIGURE 32. Displacement curve for the single-axis multi-point time optimality.

compared with the fifteen-segment S-curve of the sine structure, and this problem is significantly improved. Therefore, using a smoother planning equation in the Descartes space can further improve the smoothness of the joint space curve.

Secondly, it can be seen that although the overall plan achieves the bandwidth and time optimality of the Descartes space, the motion in joint space is not constrained. This is a huge problem of this framework. After all, in actual applications, developers directly obtain the parameter values of the motor. There are two more visualized solutions to this problem. Firstly, the motion in the Descartes space is directly planned through the operating bandwidth parameters of the motor. This causes the motion in the Cartesian space to be unstable. Related research can be found in [17], [18]. Secondly, due to the typical variable inertia system of the robot, the motor output torque is affected by the joint angle, and the acceleration that can be provided at different times is also time-varying. In other words, the kinematic equations have to be considered in the entire planning process if it is expected to plan the optimal trajectory in the Descartes space directly from the joint space and meet the kinematic constraints. Related research can be found in [8], [12].

VI. SUMMARY AND OUTLOOK

A. SUMMARY

Multi-axis synchronization has always been one of the highlights and difficulties in the field of motion control. Most multi-axis synchronization issues are solved by the framework of control principles or iterative algorithms. These methods have strict requirements for both hardware and software. This paper conducted analyses based on the kinematic equations of the time-optimal curves and solved the time-constrained synchronization algorithm from the perspective of multivariate function inverse mapping. The theories and methods proposed in the paper have the following contributions.

Firstly, according to the derivation and analyses of the paper, the smoothness of the S-curve can be improved by increasing the order of the system or constructing a more continuous Jerk curve. This is of great significance for the theoretical basis of the time-optimal curve.

Secondly, based on the third-order S-curve, this paper deduced a complete method for solving the P2P time-optimal curve. Compared with traditional methods, the proposed is not only fast but can also ensure time optimality.

Thirdly, based on the method for solving the time-optimal curve, from the perspective of inverse mapping of multivariate functions, this paper further derived three synchronization methods based on time constraints. The experiments showed that the algorithm is fast and effective. Users may adopt appropriate methods based on their different needs. More importantly, this algorithm is simple to implement and requires minimal computation.

Finally, based on the MASA proposed in this paper, a synchronous computing framework suitable for the planning of robot space poses was designed. This framework can effectively solve problems with robot pose planning. The continuity and smoothness of the robot joint spatial motion can be guaranteed in the Descartes space.

However, there are still problems and deficiencies in the research of the paper.

Firstly, although the paper proposed feasible theories and ideas to solve the multi-axis synchronization based on P2P time optimality, it is difficult to solve higher-order S-curves. The authors once established a fifteen-segment S-curve using a sine function. A large number of judgements and complicated equations occurred in the process of solving the time optimality and synchronization issues of such complex S-curves, which is not for solving multi-point passing issues. Therefore, combining it with an intelligent algorithm is a feasible solution.

Secondly, in the proposed MASA-based robot pose synchronization framework, when the spatial motion is mapped back into the joint space, the motion curve is unconstrained. This is an extremely important issue that needs to be considered. The introduction of kinematic equations will make the bandwidth parameters of the entire robot time-varying, which was not tackled in this paper.

B. OUTLOOK

According to the ideas and derivations in Chapter II, the P2P time-optimal curve of the system can be easily obtained. Based on the P2P optimal curve, a P2P multi-axis synchronization algorithm was derived.

However, in practical applications, the problem that the effector passes through multiple points in sequence also exists. Based on the results of this paper, the single-axis multi-point time optimization issue was solved. Based on the Rolle's median theorem in higher mathematics, a series of inferences were obtained to guide us through this problem. As the third-order S-curve was used as the motion model, solving the first-order quartic equation was involved in the multi-point motion problem. That is to say, the third-order S-curve reached the limit for solving the multi-point time-optimal curve problem in a closed manner. An example of solving this problem is shown in the figure below. It can be seen very intuitively that the curve is symmetric at

each vertex, which is in agreement with the optimal theory mentioned in this paper.

Based on the third-order S-curve single-axis multi-point time-optimal algorithm, we are thinking about whether the problem of multi-axis multi-point synchronization can be solved. However, this is really difficult, especially in relation to theoretical work. Following the idea described here, the multi-point kinematic series that consumes the longest time can be used as a benchmark to allow other effectors to be synchronized with it; however, we found that this is not feasible. The next attempt is to investigate whether it is necessary to extend the execution time of the slowest executor. If this problem can be solved, a more systematic and comprehensive multi-axis synchronization theory can be presented.

ACKNOWLEDGMENT

The authors thank their parents, their professors, all their colleagues in the past year. They also thank Dr. Chai for providing all types of help during the preparation of this manuscript.

REFERENCES

- [1] R. H. Castain and R. P. Paul, "An on-line dynamic trajectory generator," *Int. J. Robot. Res.*, vol. 3, no. 1, pp. 68–72, Mar. 1984.
- [2] K.-H. Rew and K.-S. Kim, "A closed-form solution to asymmetric motion profile allowing acceleration manipulation," *IEEE Trans. Ind. Electron.*, vol. 57, no. 7, pp. 2499–2506, Jul. 2010.
- [3] X. Broquere, D. Sidobre, and I. Herrera-Aguilar, "Soft motion trajectory planner for service manipulator robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2008, pp. 2808–2813.
- [4] R. Haschke, E. Weitnauer, and H. Ritter, "On-line planning of time-optimal, jerk-limited trajectories," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2008, pp. 3248–3253.
- [5] S. Liu, "An on-line reference-trajectory generator for smooth motion of impulse-controlled industrial manipulators," in *Proc. 7th Int. Workshop Adv. Motion Control.*, Jul. 2002, pp. 365–370.
- [6] B. Tondu and S. A. Bazaz, "The three-cubic method: An optimal online robot joint trajectory generator under velocity, acceleration, and wandering constraints," *Int. J. Robot. Res.*, vol. 18, no. 9, pp. 893–901, Sep. 1999.
- [7] H. Mu, Y. Zhou, S. Yan, and A. Han, "Third-order trajectory planning for high accuracy point-to-point motion," *Frontiers Electr. Electron. Eng. China*, vol. 4, no. 1, pp. 83–87, Mar. 2009.
- [8] C. Liu and Y. Chen, "Combined S-curve feedrate profiling and input shaping for glass substrate transfer robot vibration suppression," *Ind. Robot. Int. J. Robot. Res. Appl.*, vol. 45, no. 4, pp. 549–560, Jun. 2018.
- [9] H. Wu and D. Sun, "High precision control in PTP trajectory planning for nonlinear systems using on high-degree polynomial and cuckoo search," *Optim. Control Appl. Methods*, vol. 40, no. 1, pp. 43–54, Jan. 2019.
- [10] R. A. Osornio-Rios, R. de Jesús Romero-Troncoso, G. Herrera-Ruiz, and R. Castañeda-Miranda, "FPGA implementation of higher degree polynomial acceleration profiles for peak jerk reduction in servomotors," *Robot. Comput. Integr. Manuf.*, vol. 25, no. 2, pp. 379–392, Apr. 2009.
- [11] S. Perumaal and N. Jawahar, "Synchronized trigonometric S-curve trajectory for jerk-bounded time-optimal pick and place operation," *Int. J. Robot. Autom.*, vol. 27, no. 4, p. 385, 2012.
- [12] M. H. Ghasemi, N. Kashiri, and M. Dardel, "Time-optimal trajectory planning of robot manipulators in point-to-point motion using an indirect method," *Proc. Inst. Mech. Eng., C, J. Mech. Eng. Sci.*, vol. 226, no. 2, pp. 473–484, Feb. 2012.
- [13] Y. Fang, J. Hu, W. Liu, Q. Shao, J. Qi, and Y. Peng, "Smooth and time-optimal S-curve trajectory planning for automated robots and machines," *Mechanism Mach. Theory*, vol. 137, pp. 127–153, Jul. 2019.
- [14] J. Lin, N. Somani, B. Hu, M. Rickert, and A. Knoll, "An efficient and time-optimal trajectory generation approach for waypoints under kinematic constraints and error bounds," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 5869–5876.

- [15] A. Machmudah, S. Parman, A. Zainuddin, and S. Chacko, "Polynomial joint angle arm robot motion planning in complex geometrical obstacles," *Appl. Soft Comput.*, vol. 13, no. 2, pp. 1099–1109, Feb. 2013.
- [16] T.-C. Lu, S.-L. Chen, and E. C.-Y. Yang, "Near time-optimal S-Curve velocity planning for multiple line segments under axis constraints," *IEEE Trans. Ind. Electron.*, vol. 65, no. 12, pp. 9582–9592, Dec. 2018.
- [17] X. Gao, Y. Mu, and Y. Gao, "Optimal trajectory planning for robotic manipulators using improved teaching-learning-based optimization algorithm," *Ind. Robot, Int. J.*, vol. 43, no. 3, pp. 308–316, May 2016.
- [18] Y. Chen, L. Yan, H. Wei, and T. Wang, "Optimal trajectory planning for industrial robots using harmony search algorithm," *Ind. Robot, Int. J.*, vol. 40, no. 5, pp. 502–512, Aug. 2013.
- [19] T. Tao, *Analysis I*. Springer, 2016.
- [20] S. Fan, "A new extracting formula and a new distinguishing means on the one variable cubic equation," (in China) *Natural Sci. J. Hainan Teachers College*, vol. 2, no. 2, pp. 91–98, Dec. 1989.
- [21] M. Leahy and G. Saridis, "Compensation of unmodeled puma manipulator dynamics," in *Proc. IEEE Int. Conf. Robot. Autom.*, Mar. 1987, pp. 151–156.
- [22] K. J. Kyriakopoulos and G. N. Saridis, "Minimum jerk path generation," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 1988, pp. 364–369.
- [23] G. Field and Y. Stepanenko, "Iterative dynamic programming: An approach to minimum energy trajectory planning for robotic manipulators," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 1996, pp. 2755–2760.
- [24] M. Stilman, "Global manipulation planning in robot joint space with task constraints," *IEEE Trans. Robot.*, vol. 26, no. 3, pp. 576–584, Jun. 2010.
- [25] N. Sun, Y. Wu, H. Chen, and Y. Fang, "An energy-optimal solution for transportation control of cranes with double pendulum dynamics: Design and experiments," *Mech. Syst. Signal Process.*, vol. 102, pp. 87–101, Mar. 2018.
- [26] W. Shanda, L. Xiao, L. Qingsheng, and H. Baoling, "Existence conditions and general solutions of closed-form inverse kinematics for revolute serial robots," *Appl. Sci.*, vol. 9, no. 20, p. 4365, 2019.



SHAN-DA WANG received the B.S. degree in automation from the Kunming University of Science and Technology, Yunnan, China, in 2012, the M.S. degree in computer science from North China Electric Power University, Beijing, China, in 2016. He is currently pursuing the Ph.D. degree in mechanical engineering with the Beijing Institute of Technology. For the closed inverse solution of serial robots, a paper has been published entitled *Existence Conditions and General Solutions of Closed-form Inverse Kinematics for Revolute Serial Robots*. His major research interests include robotics, motion control, and robot dynamics.



XIAO LUO received the bachelor's degree in information and electronics and the master's and Ph.D. degrees in mechatronical engineering from the Beijing Institute of Technology, Beijing, China, in 2006, 2008, and 2012, respectively. She is currently a Lecturer with the Beijing Institute of Technology. Her current research interests include image processing, computer vision, simultaneous localization and mapping (SLAM), and automate robot.



SHI-JIE XU received the B.S. degree in mechanical engineering from the University of Science and Technology Beijing, Beijing, China, in 2019. He is currently pursuing the M.S. degree in mechanical engineering with the Beijing Institute of Technology, Beijing. His research interests include robotics, motion control, and robot dynamics.



QING-SHENG LUO received the B.S. degree in precision instrument design, the M.S. degree in mechanical optimization design, and the Ph.D. degree in mechanical engineering from the Beijing Institute of Technology, Beijing, China, in 1982, 1987, and 2001, respectively. He has directed and completed 18 scientific research projects at the level of government minister or above, which included a major project at the ministerial level and two major pre-research fund projects at the ministerial level and 15 educational reform research projects. He has been published on journals, domestic, and international conferences. He has published 200 articles on science and education, including 50 EI articles, and 11 academic monographs. He holds 16 China National Invention Patents. His research interests include specialized robot technology, mechatronical product innovative design, mechatronical system control technology, mechatronical device integration technology, and college students' scientific and technologically innovative activities. He received ten First and Second Prizes of Teaching Achievements at the ministerial and college level.



BAO-LING HAN received the B.S. degree in precision instrument design from the Beijing Institute of Technology, Beijing, China, in 1982, the M.S. degree in precision instrument from Tsinghua University, Beijing, in 1995, and the Ph.D. degree in mechanical engineering from the Beijing Institute of Technology, in 2002. She holds the Doctor of Engineering position with the Beijing University of Technology. She is currently a Professor and a Ph.D. Supervisor with the Beijing University of Technology. She is also the Vice President of the China Graphics Society and the International Society for Geometry and Graphics. She has published more than 80 academic articles and eight academic monographs (teaching materials), among which Intelligent Combat Robot received the Fifth China Excellent Publishing Award. Her main research interests include intelligent robot, optical electromechanical integration technology, and so on.



GUAN-HAO LIANG received the B.S. degree in mechanical engineering and the M.S. degree in mechanical engineering from the Beijing Institute of Technology, Beijing, China, in 2011. He joined the master's-Ph.D. Joint Program in mechanical engineering at the Beijing Institute of Technology, in 2012. His research interests include deep learning techniques adopted in wearable sensor-based human activity recognition, motion planning in industrial, and mobile robots.



YAN JIA received the B.S. degree in mechanical engineering from Southwest University, Chongqing, in 2013, and the master's degree from the Beijing Institute of Technology, Beijing. She joined the M.Sc.-Ph.D. based program in mechanical engineering at the Beijing Institute of Technology, in 2014. Her main research interests include motion planning for quadruped robots and reinforcement learning.

...