Mechanism and Machine Theory 137 (2019) 127–153

Contents lists available at ScienceDirect

Mechanism and Machine Theory

journal homepage: www.elsevier.com/locate/mechmachtheory

Research Paper

Smooth and time-optimal S-curve trajectory planning for automated robots and machines

Yi Fang, Jie Hu*, Wenhai Liu, Quanquan Shao, Jin Qi, Yinghong Peng

School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

ARTICLE INFO

Article history:
Received 30 October 2018
Revised 20 February 2019
Accepted 11 March 2019

Keywords:
Trajectory planning
S-curve motion profile
Robotic manipulator
Sigmoid function
Jerk

ABSTRACT

In this paper, a smooth and time-optimal S-curve trajectory planning method is proposed to meet the requirements of high-speed and ultra-precision operation for robotic manipulators in modern industrial applications. This method utilizes a piecewise sigmoid function to establish a jerk profile with suitably chosen phase durations such that the generated trajectories are infinitely continuously differentiable under the given constraints on velocity, acceleration and jerk. All the trajectory parameters are derived with an analytical algorithm to ensure an acceptable computational cost. This S-curve model achieves a greater efficiency than the trigonometric models, while avoiding the high complexity presented by conventional high order polynomial models. The trade-off between efficiency and smoothness can be modulated by the limit value of the snap (the derivative of jerk), this feature is advantageous for adapting to different task requirements. Furthermore, a synchronization strategy is suggested to coordinate multi-axis motions, enabling the full capabilities of the actuators to be exploited. The feasibility and practicality of the proposed approach is evaluated by the simulation and experimental studies in comparison with other benchmark techniques in the literature.

1. Introduction

In recent years, with the development of automation technology and computer science, industrial robots and automatic machines have been increasingly employed in modern manufacturing systems. As an essential issue in the robotics and machine tools field, the problem of trajectory planning has received enormous attention from a series of researchers. The goal of trajectory planning is to produce the optimum reference inputs to a control system according to the geometric path as well as the mechanical limits for carrying out the desired motion of the given task. The performance of planned trajectories is crucial for motion control since it exerts a significant and direct influence on the stability, reliability and productivity of the machinery.

Numerous trajectory generation methods have been explored during the last decades. Since there is a need of improving the productivity to increase economic efficiency in the industrial field, the time optimality is a key factor to be considered. Meanwhile, the capability to be implemented in real time requires low computational effort for a method to minimize the latency in executing the motion with as little downtime as possible [1]. From the aspect of application, the trapezoidal velocity profile, also referred to as the linear segment with parabolic blends (LSPB) trajectory, is usually regarded as the

* Corresponding author at: Institute of Knowledge Based Engineering, School of Mechanical Engineering, Shanghai Jiao Tong University, 800 Dongchuan RD, Minhang District, Shanghai 200240, China.
E-mail address: hujie@sjtu.edu.cn (J. Hu).

https://doi.org/10.1016/j.mechmachtheory.2019.03.019
0094-114X/

optimal timing law given maximum velocity and acceleration restriction [2,3]. Unfortunately, this technique suffers from the occurrence of infinite jerk spikes as a consequence of the inherent acceleration discontinuities, which can stimulate the mechanical resonance frequencies of the system and cause severe shock [4]. As a matter of fact, the yielded motion profiles are unrealistic due to delays arising from physical limits of the actuators and necessitate specialized controllers for practical implementation.

To gain superior dynamic performance, some researchers have taken the robot dynamics into consideration for the control system. Wu et al. [5] developed the dynamic model of a redundantly actuated parallel manipulator based on the virtual work principle and designed a position and force switching control scheme for the two extendible chains. The tests revealed that the contour error is similar to that of the corresponding non-redundant manipulator. More recently, in [6], the mechatronics modeling of a 2-DOF parallel manipulator was investigated by means of bond graph techniques and the forced vibration was analyzed according to the interactions between the mechanical and control subsystems. The motion control for a mobile manipulator can become more intractable than in the case of the manipulator alone since coordination between mobile base and manipulator is needed [7]. The indirect solution of open-loop optimal control problem was utilized with the full exploitation of redundancy in [8] to find the optimal trajectory for obtaining the maximum dynamic load-carrying capacity (DLCC) of redundant manipulators. Korayem et al. [9] studied the path planning using feedback linearization method for a wheeled mobile robot with two manipulators to attain singularity avoidance and a smooth motion, which remarkably improves the tracking accuracy and DLCC.

Considering the complexity of the identification of exact robot dynamic model and the implementation of algorithms for computing the dynamics, another possible solution is to introduce the jerk constraints in the trajectory planning, providing an indirect means to confine the variation rate of the actuator torques [10]. It is well accepted that jerk limitation is of great significance in diminishing the vibrations of the structure, wear on the machinery and positioning error during movement [11–14]. The trajectories with bounded jerk allow the realization of more expeditious and precise tracking by the controller. Generally, smooth motions are necessarily enforced in delicate production operations such as handling of fragile items, precise assembly or spot welding, which implies that the generated trajectories should have good continuity properties without exceeding the actuator limitations [3]. Moreover, with the growing demand for lightweight systems to achieve greater productivity, the trajectory optimization and vibration suppression of manipulators with flexible joints and links has become a crucial problem [15,16].

In view of this, more sophisticated polynomial trajectories have evolved to enable smoother movement showing a bell-shaped velocity profile with less jerkiness [17–19]. Boryga and Grabos [20] utilized the properties of the roots multiplicity to build higher-degree polynomial acceleration profiles up to 9th order for serial-link manipulators, one limitation of their method is that only the acceleration limit can be predefined. Machmudah et al. [21] employed a genetic algorithm (GA) and a particle swarm optimization (PSO) to search the feasible sixth degree polynomial joint trajectory in complex geometrical environments. Despite that higher order polynomials support better continuity, they have the tendency to exhibit a swiftly growing oscillatory characteristic and degradation of numerical accuracy with a higher degree, which may deteriorate the motion. Moreover, a substantial amount of high degree terms bring difficulty in the evaluation of polynomial coefficients and represent a considerable computation load [22].

These undesirable effects can be overcome by adding some dummy constraints to modulate the shape of the function [23], or more commonly, through exploiting a sequence of lower order piecewise polynomials, i.e., spline functions for the interpolation. The cubic spline function is favored since it is the most elegant polynomial that assures continuous acceleration, and consequently limited jerk while preventing pronounced oscillations [24,25]. Tondu and Bazaz [1] developed the analytic three-cubic method to interpolate joint positions with associated velocities. Lin presented a unified methodology to find the near optimal solution of a minimum-jerk trajectory based on PSO [26]. The problem of minimum cost motion planning was addressed in [27] as a nonlinear constrained optimization one of parameterized joint evolution vector using the sequential quadratic programming (SQP) method. The minimum time-jerk trajectory was optimized by the augmented Lagrange constrained particle swarm optimization (ALCPSO) algorithm in [28].

With the cubic spline, the virtual points are generally required to ensure the continuity of the trajectory at the endpoints. Thus, higher order splines expressed in the B-form or a blending of different order polynomials have also been invoked in some work to include more boundary conditions [11,29,30]. The applications of cubic splines as well as fifth-order B-splines with an objective function composed of two terms that are proportional to the integral of the squared jerk and the total execution time, have been investigated by Gasparetto and Zanotto [10,31]. Huang et al. [32] also described a methodology adopting the quintic B-spline interpolation in joint space but applied a non-dominated sorting genetic algorithm (NSGA-II) as a multi-objective optimization algorithm to obtain a time-jerk optimal trajectory. A composite polynomial merging cubic polynomials with Bézier curves based on quadratic Bernstein polynomials is utilized to generate a smooth trajectory for a 2-degrees-of-freedom (DOF) parallel mechanism in [33]. Kucuk [34] combined cubic spline with the 7th order polynomial to construct minimum-time smooth motion trajectories beginning and ending with zero jerk using a PSO algorithm for serial and parallel manipulators. However, the search for an optimal solution in most spline-based approaches involves various numerical optimization techniques demanding both processing time and hardware resources, and this restrict their applications in online operations to a large extent [1,35].

As an alternative for generating jerk-limited motions, S-curve trajectory methods have already been widely used in modern manufacturing facilities [36]. They can be perceived as an approximation to the trapezoidal velocity profile, and yet do not inherently cause flexible impact by carefully selecting switching times compatible with the imposed additional

constraints. A fourth-order S-shape acceleration/deceleration (Acc/Dec) control algorithm with time optimization was proposed in [37] for a computer numerical control (CNC) simulation system. Li et al. [38] designed a modified acceleration profile by using a sinusoidal waveform to suppress vibrations for high speed positioning. Nguyen et al. [39] developed a systematic approach to design time optimal polynomial S-curve motion profiles. Asymmetric motion profiles have also been investigated with growing interest in recent works to further lessen the residual response and shorten the positioning time [40]. The complete closed-form solution for asymmetric profiles of different distances was given in [41] by introducing the jerk ratio which scales down the jerk in the deceleration stage. However, the introduction of this ratio restricts the possibility to set arbitrary values for the constraints. Bai et al. [42] proposed a freeform S-curve motion profile to optimize the magnitude and phase terms in residual vibration response using an energy-based method. The asymmetrical S-curve trajectories derived from a trigonometric jerk function were suggested in [43] and [44]. Because of more trajectory parameters involved, how to establish a systematic method to derive asymmetric profiles is still a challenging topic.

Most of these abovementioned S-curve methods coped with only the calculation of monoaxial motion trajectory, whereas a quite common case encountered in reality is multi-axis motions, which can be either the spatially multi-dimensional case or the motion of mutually independent axes. To this end, one essential characteristic for a trajectory planning method is the synchronization of all active DOFs. That is, all actuators execute the trajectories with the same start and end time instants to achieve motion with maximum smoothness. Broquère et al. [45] presented a soft motion trajectory planner using cubic curves to limit the jerk, acceleration and velocity in Cartesian space for both monodimensional and multidimensional cases. Subsequently, Haschke et al. [46] introduced an algorithm to calculate synchronized time-optimal third-order trajectories with arbitrary initial conditions for service robots. The optimization problem of a trajectory passing through specified waypoints under kinematic constraints was studied in [47]. Perumaal and Jawahar [48] focused on the generation of a jerk-bounded trigonometric S-curve trajectory for pick and place tasks of a 6-DOF manipulators. The simulation results prove that their algorithm is able to generate smoother and faster motions relative to spline-based trajectories. However, the maximum allowable jerk cannot be attained during the Acc/Dec phases due to a conservative synchronization strategy, which leads to unnecessary performance surplus and an additional increase in the total execution time.

The central objective of this study is to propose a novel S-curve trajectory planning method based on piecewise sigmoid jerk function with an improved motion synchronization strategy for point to point movements. The time optimality and jerk limitation are simultaneously taken into account to achieve a rapid and precise synchronized motion compliant with prescribed constraints. In principle, this approach is general purpose and can be applied to typical serial or parallel industrial robotic manipulators with arbitrary DOFs as well as CNC equipment. The obstacle avoidance problem, which is out of the scope of this study, could be addressed by using a path planning method preceding this approach, for instance, iteratively introducing a set of intermediate via-points into the trajectory through a sampling-based algorithm.

The remainder of this article is organized as follows: Section 2 gives a brief review and discussion of common S-curve models. Next, the methodology for generating sigmoid jerk S-curve trajectories is described in Section 3. The strategy for synchronizing all movements of multi-axis mechanisms is presented in Section 4. Afterwards, in Section 5, numerical simulations on manipulators with three and six DOFs are addressed. Section 6 reports the experimental results on a real robot platform. Finally, the conclusions are drawn in Section 7.

2. State of the art in S-curve models

In this section, the S-curve profiles proposed in the relevant literature are outlined. These methods usually fall into two main categories: polynomial models and trigonometric models. Generally, a motion profile can be divided into three stages probably composed of several segments: start up stage, cruise stage with maximum velocity and brake stage, of which the cruise stage may disappear because of distance limitations.

2.1. Polynomial S-curve model

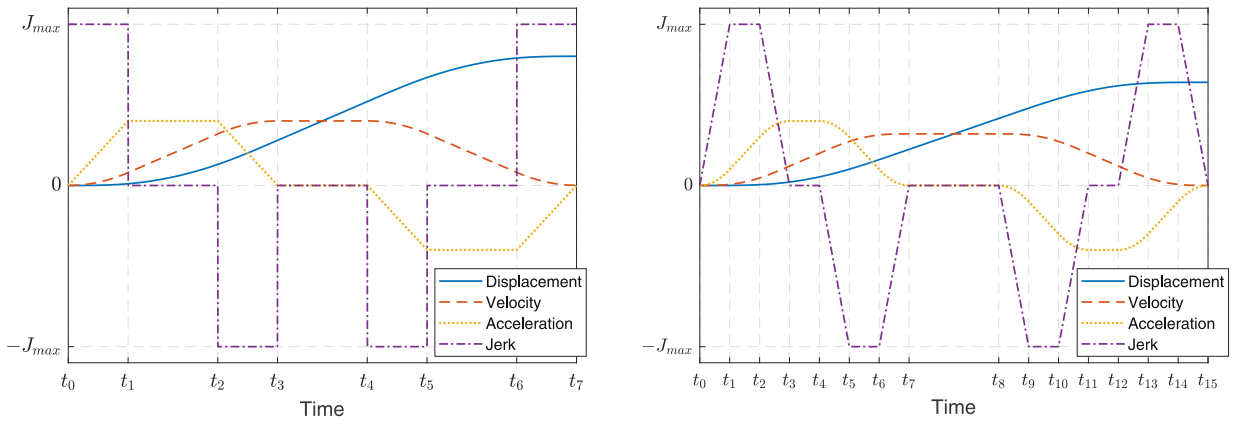
2.1.1. Cubic polynomial model

Originally put forward by Castain and Paul [49], the third order S-curve trajectory has become one of the most popularly used model in practice [50–53], because it enables obtaining the minimum time motion with constrained jerk and moderate complexity. As shown in Fig. 1(a), the canonical form of the motion profile consists of seven segments, among which the first three and the last three constitute the acceleration and deceleration stages, respectively, resulting from modifying the trapezoidal velocity profile by limiting the rise time of the acceleration waveform. By means of the celebrated Pontryagin's maximum principle [54], the time optimal jerk profile is of bang-off-bang type:

$$j(t)=\begin{cases} J_{\max} & t_0 \le t < t_1, t_6 \le t \le t_7 \\ 0 & t_1 \le t < t_2, t_3 \le t < t_4, t_5 \le t < t_6 \\ -J_{\max} & t_2 \le t < t_3, t_4 \le t < t_5 \end{cases} \tag{1}$$

where $J_{\max}$ is the maximum specified jerk value. The corresponding time evolution of displacement described by a set of polynomials can be deduced from the integration relationship among them.

In comparison to the trapezoidal trajectory featuring an impulsive jerk profile, the rectangular-shaped jerk adopted by this motion profile weakens excessive stress on the actuators and mechanical structure engendered by step changes in



(a) Third order polynomial S-curve model. (b) Fourth order polynomial S-curve model.

Fig. 1. Polynomial S-curve models.

acceleration, thus prolonging the service life of the machine. However, the trajectory has a shortcoming of jumps in jerk, and nonzero instantaneous jerks at the target points will lead to unexpected residual vibrations with accompanying extra settling time, which deteriorates the positioning accuracy.

2.1.2. Fourth order polynomial model

For industrial applications requiring high precision, it proves crucial to remove the detrimental jerk effect at the initial and final instants of motion [20]; this signifies an enforcement of a continuous jerk profile, for which an available means is the 4th order S-curve model [37,55,56], whose number of overall trajectory segments increases to fifteen as each acceleration ramp now splits into three separate phases with respect to the third order profile. In this way, the bound value on snap (time derivative of jerk) can be defined to achieve a jerk curve growing and descending linearly with prespecified slopes in lieu of step jumps. A schematic view of this model is given in Fig. 1(b). Constant snaps reflect nonsmooth changes in jerk marked by sharp turning points, making the jerk not to be differentiable.

2.1.3. Higher order polynomial model

To further enhance the smoothness of the trajectory, the continuity of snap, or even higher order derivatives, should also be taken into account. It has been indicated in [57] that a smooth snap is required as well for shaping an input motion profile with a view to compensate for unfavorable dynamic effects of the mechanical system. In general, a higher order polynomial model with a desired level of continuity can be designed by assigning a recursive template for the evolution of the highest order derivative with a finite peak value. Nevertheless, according to the study by Nguyen et al. [39], what comes with a better performance is that the number of trajectory segments will increase exponentially along with the model order, and hence can lead to a complicated selection algorithm for the switching times with no available analytical solutions, which shows a remarkable negative effect on the computational efficiency and poses a challenge for the controller. Consequently, they are not normally used unless specifically requested.

Another simplified form of polynomial-based S-curve profile involves the use of a single polynomial rather than piecewise polynomials to construct the speed curve in the Acc/Dec stages [58,59]. In spite of lower computational burden required by this type of models due to the reduced number of trajectory segments, the efficiency would inevitably decline.

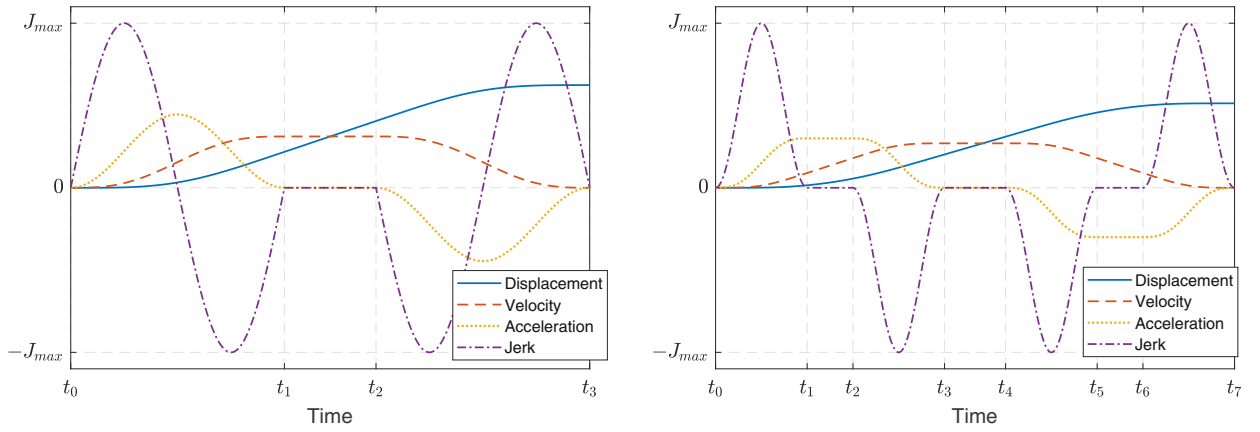
2.2. Trigonometric S-curve model

Since the formulation of the trajectory profile and the scheduling procedure will become rather cumbersome as the polynomial S-curve model becomes smoother, in the exploration of new S-curve approaches, some investigations have attempted to devise motion profiles applying a trigonometric model to ensure the continuity of the jerk.

2.2.1. Sine jerk model

Unlike the piecewise function for a polynomial model, the Acc/Dec profiles of this model are derived from a single sinusoidal jerk function expressed as follows [60]:

$$j(t) = \begin{cases} J_{\max} \sin\left(\dfrac{2\pi}{t_J}(t - t_0)\right) & t_0 \le t < t_1 \\ 0 & t_1 \le t < t_2 \\ -J_{\max} \sin\left(\dfrac{2\pi}{t_J}(t - t_2)\right) & t_2 \le t \le t_3 \end{cases} \tag{2}$$



(a) Sine jerk S-curve model. (b) Harmonic jerk S-curve model.

Fig. 2. Trigonometric S-curve models.

where t_J is the time interval during which the jerk varies.

Fig. 2(a) depicts the graphical representation of this model. Owing to its succinct expression with only three sections to be identified, it can be implemented more effortlessly. However, this convenience comes with the drawback of degraded efficiency on the grounds that the jerk and acceleration only reach their maximum values at several isolated points as a result of the omission of related phases, which obviously causes insufficient exploitation of the dynamic capacity of the driving system. Moreover, the limiting conditions of velocity and acceleration cannot take effect at once in virtually all cases if the jerk constraint is designated, in other words, only one of them can actually be reached even with a long travel distance. A feasible compensation measure for this is to extend this model to seven phases by interposing constant acceleration phases between the acceleration ramp-up and ramp-down sections [38]. Also note that the attribute of a sine jerk waveform may be simulated by a cubic polynomial with adequately selected parameters except that the sine jerk profile presents non-null continuous derivatives for any order in the interior of a phase [61].

2.2.2. Harmonic jerk model

Another trigonometric model may be constructed via replacing rectangular jerk pulses employed in the cubic polynomial S-curve trajectory by a level-shifted cosinusoidal function [39,43]:

$$j(t) = \begin{cases} \dfrac{J_{\max}}{2}\left(1 - \cos\left(\dfrac{2\pi}{t_J}(t - t_i)\right)\right) & t_0 \le t < t_1,\ t_6 \le t \le t_7 \\ 0 & t_1 \le t < t_2,\ t_3 \le t < t_4,\ t_5 \le t < t_6 \\ -\dfrac{J_{\max}}{2}\left(1 - \cos\left(\dfrac{2\pi}{t_J}(t - t_i)\right)\right) & t_2 \le t < t_3,\ t_4 \le t < t_5 \end{cases} \tag{3}$$

where t_i is the initial time instant of the corresponding segment. Although this model is defined with only seven segments (see Fig. 2(b)) as simple as the 3rd order polynomial model, it is capable of generating smooth trajectories with performance as good as that of the 5th order polynomial model. This can be explained reasonably by the fact that both models ensure continuity up to the snap level.

Through the above examination of existing S-curve profiles, it can be summarized that the trigonometric model is less efficient than the polynomial one concerning the execution time considering that the maximum acceleration or jerk cannot be continuously sustained, but it allows a more concise description of the profile compared to the latter under the same level of smoothness, thereby facilitating the searching process of optimal motion parameters. Given the strengths and weaknesses of the two types of motion profiles, it makes sense to seek a smooth S-curve model without loss of simplicity and efficiency.

3. Smooth S-curve trajectory planning

As previously described, the feasibility of a trajectory planning method relies on its ability to reach a reasonable compromise among operating efficiency, motion smoothness and computational complexity, which can be mutually contradictory when being accounted for concurrently to optimize the motion profile. Overall, the smoothness level of the trajectories increases at the expense of a prolonged execution time and more intractable computations for a given set of limiting conditions.

It needs to be remarked that in case the jerk, acceleration and velocity are successively saturated in the shortest possible time within given constraints and are persisted as long as possible, the total traveling time can be minimized [62]. Therefore,

a time-efficient trajectory should embrace the constant velocity, acceleration and jerk periods during which the velocity, acceleration and jerk can maintain their maximum allowable values. In addition, to guarantee a continuous jerk profile, the instrumental phases for smooth changes in the jerk are indispensable. Hence, it can be deduced that fifteen is the least possible number of segments necessary to compose a trajectory that can possess the time-optimality of the third order S-curve profile with the requirement of jerk continuity.

3.1. Sigmoid jerk model

With a comprehensive consideration of the above factors, a novel S-curve trajectory model is developed, where the piecewise sigmoid function is adopted to constitute a continuous jerk profile that can be integrated three times to obtain the displacement trajectory. The salient attribute of the presented method lies in enabling the planning of smooth and adjustable trajectories characterized by an extremely high degree of continuity, while maintaining time optimality in accordance with given constraints involving the boundary conditions as well as the kinematic limitations on velocity, acceleration and jerk. Additionally, the computing complexity is reasonable for the further purpose of online implementation of the trajectories as the analytic optimal solutions are derived without relying on any iterative optimization process.

Sigmoid functions are nonlinear real-valued functions with positive first derivatives everywhere that are extensively used as the activation function in neural networks, and they generally vary in a range enclosed by a pair of horizontal asymptotes. More specifically, the sigmoid function refers to the standard logistic function expressed as:

$$f(x) = \frac{1}{1+e^{-x}} \tag{4}$$

This function has several distinctive properties making it suitable for achieving a smooth transition of the jerk with no sharp points. First, it has a definition domain of $[-\infty, +\infty]$, with the return value monotonously growing from 0 to 1 and centering at the value of 0.5, so no additional gain coefficient needs to be added for the jerk function while the symmetry facilitates the derivation of optimal parameters and leads to a lower peak snap value. Second, it is analytically tractable for the differentiation operation due to the nature of the exponential in its formula. The slope of the function curve is steeper, approximately linear over the middle range of the interval around $x = 0$ and quickly flattens out to a zero value at both extremities as x approaches negative infinity or infinity. This feature can be usefully drawn on to guarantee the continuity of the global profile.

Suppose that the physical limits for maximum velocity, acceleration and jerk are given as $V_{\max}$, $A_{\max}$ and $J_{\max}$ respectively, and the target distance to be traveled is D, representing the relative displacement from the initial position that can be either positive or negative. Moreover, all derivatives of the trajectory are constrained to be zero at the initial and final positions. The kinematic profiles for the normal case of the proposed model are depicted in Fig. 3, in which the jerk curve is defined as follows:

$$j(t) = sign(D) \cdot \begin{cases} J_{\max}\dfrac{1}{1+e^{-a(1/(1-\tau_i)-1/\tau_i)}} & t_0 \le t < t_1, t_{12} \le t < t_{13} \\ J_{\max} & t_1 \le t < t_2, t_{13} \le t < t_{14} \\ J_{\max}\dfrac{1}{1+e^{a(1/(1-\tau_i)-1/\tau_i)}} & t_2 \le t < t_3, t_{14} \le t \le t_{15} \\ 0 & t_3 \le t < t_4, t_7 \le t < t_8, t_{11} \le t < t_{12} \\ -J_{\max}\dfrac{1}{1+e^{-a(1/(1-\tau_i)-1/\tau_i)}} & t_4 \le t < t_5, t_8 \le t < t_9 \\ -J_{\max} & t_5 \le t < t_6, t_9 \le t < t_{10} \\ -J_{\max}\dfrac{1}{1+e^{a(1/(1-\tau_i)-1/\tau_i)}} & t_6 \le t < t_7, t_{10} \le t < t_{11} \end{cases} \tag{5}$$

where t_i $(i = 0, 1, \ldots, 15)$ stands for the time boundary of each trajectory segment, $\tau_i = \frac{t-t_{i-1}}{t_i-t_{i-1}}$ represents the normalized time variable in the interval $[t_{i-1}, t_i]$ and a is a positive constant parameter controlling the variation rate. Here the sigmoid function is modified so that its original interval of definition is mapped to the interval $[t_i, t_{i+1}]$ $(i = 0, 2, 4, 6, 8, 10, 12, 14)$, which enables the jerk to smoothly alternate between its maximum limit value $J_{\max}$ and zero, and thus the trajectories generated are infinitely differentiable throughout.

All subfigures in Fig. 3 may be flipped about the horizontal time axis in the case of negative displacement corresponding to a movement along the opposite direction. The whole trajectory comprises fifteen segments, and the operation time of each phase is denoted as $T_i = t_i - t_{i-1} (i = 1, 2, \ldots, 15)$. Since the derivatives are assumed to be restricted as absolute values, $T_1 = T_3 = T_5 = T_7$ and $T_2 = T_6$ should hold for the acceleration stage $[t_0, t_7]$ to have a null acceleration at t_7. Correspondingly, the deceleration stage $[t_8, t_{15}]$ has similar relations $T_9 = T_{11} = T_{13} = T_{15}$ and $T_{10} = T_{14}$. Meanwhile, the velocity profile is of symmetric type for point to point motions, thus the time intervals allocated for the acceleration and deceleration stages are equal. Therefore, the shape of the trajectory depends completely on the actual maximum jerk and four time intervals: the varying jerk interval T_s, the constant jerk interval T_j, the constant acceleration interval T_a and the constant velocity interval T_v, during which the maximal snap, jerk, acceleration and velocity are achieved respectively. The total execution time T_{exe} in which the trajectory is to be traversed, is the sum of the time intervals of all fifteen phases: $T_{exe} = \sum_{i=1}^{15} T_i = 8T_s + 4T_j + 2T_a + T_v$.

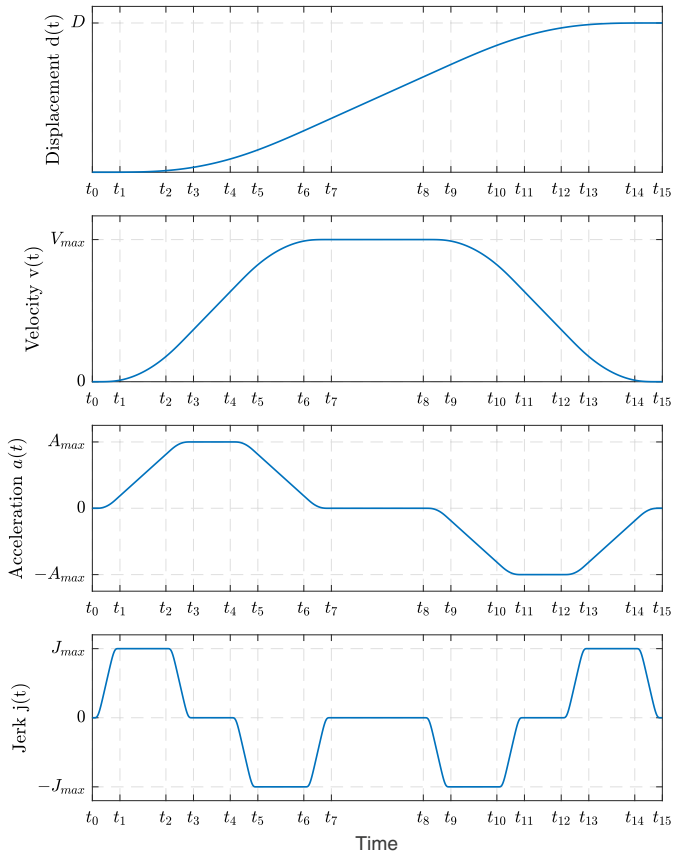


Fig. 3. Kinematic profiles of the proposed sigmoid S-curve model.

The problem of optimal S-curve trajectory planning is to calculate the times T_s, T_j, T_a and T_v for obtaining the desired displacement between an initial position and a final position within the shortest time under consideration of the given motion constraints, which can be succinctly formulated as:

$$\min T_{exe} = \int_0^{T_{exe}} 1dt$$
$$subject\ to \begin{cases} T_i \geq 0, i = 1, \ \ldots, \ 15 \\ |v(t)| \leq V_{\max} \\ |a(t)| \leq A_{\max} \\ |j(t)| \leq J_{\max} \end{cases} \tag{6}$$

The explicit formulations for the acceleration, velocity and displacement, $a(t)$, $v(t)$ and $d(t)$, respectively, are not available since the jerk function is not analytically integrable. However, due to the symmetrical characteristic of the jerk profile, the values of these kinematic variables at specific connection time instants can be achieved through the area under the curve of their derivatives. For the standard case when all fifteen segments are present as shown in Fig. 3, the following relationships can be found:

$$\begin{cases} A_{\max} = a(t_3) = a(t_4) = J_{\max}\left(T_s + T_j\right) \\ V_{\max} = v(t_7) = v(t_8) = A_{\max}\left(2T_s + T_j + T_a\right) \\ |D| = |d(t_{15})| = V_{\max}\left(4T_s + 2T_j + T_a + T_v\right) \end{cases} \tag{7}$$

In practical applications, it sometimes appears that not all the fifteen segments are included in the motion profile; one or more segments with constant or zero jerk may not exist as the desired displacement is too short or the kinematic constraints create internal restrictions in themselves. In this case, the S-curve trajectory will not be performed with the maximum permissible values of the kinematic parameters and the maximum jerk in Eq. (5) may need to be adjusted. On these grounds, it is essential to discuss various circumstances for specifying the motion parameters according to the actuator specifications and the travel distance. To distinguish the actual maximum values of velocity, acceleration and jerk during the motion from their limit values, the actual maximum values are designated as $v_{\max} = \max(v(t))$, $a_{\max} = \max(a(t))$, and

$j_{\max} = \max(j(t))$ with lowercase letters, respectively. In the following paragraphs, a suitable value is first chosen for the variation rate parameter, and then an algorithm is derived for calculating the time intervals.

3.2. Choice of the variation rate parameter

Without loss of generality, consider the time interval $[t_0, t_1]$ with a positive displacement, the snap is calculated by performing the differentiation of the jerk function:

$$s(t) = \frac{d}{dt} j(t) = J_{\max} \frac{a e^{-a(1/(1-\tau_1)-1/\tau_1)}}{\left[1 + e^{-a(1/(1-\tau_1)-1/\tau_1)}\right]^2} \left[\frac{1}{(1-\tau_1)^2} + \frac{1}{{\tau_1}^2} \right] \frac{1}{T_1}, \quad t_0 \le t \le t_1 \tag{8}$$

where $\tau_1 = \frac{t-t_0}{T_1}$. For the sake of clarity, it is denoted that $f = 1/(1-\tau_1) - 1/\tau_1$, and then the expression for the snap can be written as:

$$s(t) = J_{\max} \frac{a e^{-af}}{\left[1 + e^{-af}\right]^2} \dot{f} = \frac{a}{J_{\max}} j(t)[J_{\max} - j(t)]\dot{f}, \quad t_0 \le t \le t_1 \tag{9}$$

Apparently, this function is continuous and derivable in its definitional domain. Furthermore, due to the fact that $j(t)$ approaches zero at the boundary points in an exponential manner while the function f and its derivatives tend to infinity with power function, the snap function $s(t)$ also goes to zero exponentially at the boundary points.

Correspondingly, from formula (9) the crackle (derivative of the snap) is expressed as:

$$\dot{s}(t) = \frac{a}{J_{\max}} s(t)[J_{\max} - 2j(t)]\dot{f} + \frac{a}{J_{\max}} j(t)[J_{\max} - j(t)]\ddot{f}, \quad t_0 \le t \le t_1 \tag{10}$$

where $\ddot{f} = 2[\frac{1}{(1-\tau_1)^3} - \frac{1}{{\tau_1}^3}]\frac{1}{{T_1}^2}$. If we continue to differentiate this function, it is worthy to observe that all the terms of the snap function as well as its derivatives always include the factor $j(t)$ or its higher derivatives. As a result, it can be concluded that all the derivatives of the snap function over this interval are continuously differentiable and vanish at the endpoints. This feature provides the possibility to generate a complete trajectory with a theoretically infinite order of continuity.

Different values of the control parameter a will lead to different jerk variation rates, namely, the change of the snap profile (Fig. 4), it becomes a necessity to study the influence of the parameter a on the snap and choose an appropriate value to yield the desired motion profile. Still consider the time segment $[t_0, t_1]$, the time point when the snap achieves its extreme value in the defined interval can be obtained by solving the following equation:

$$\dot{s}(t) = 0 \Rightarrow s(t)[J_{\max} - 2j(t)]\dot{f} + j(t)[J_{\max} - j(t)]\ddot{f} = 0 \tag{11}$$

By substituting Eq. (9), the above equation can be simplified as:

$$\frac{a}{J_{\max}}[J_{\max} - 2j(t)]\dot{f}^2 + \ddot{f} = 0 \tag{12}$$

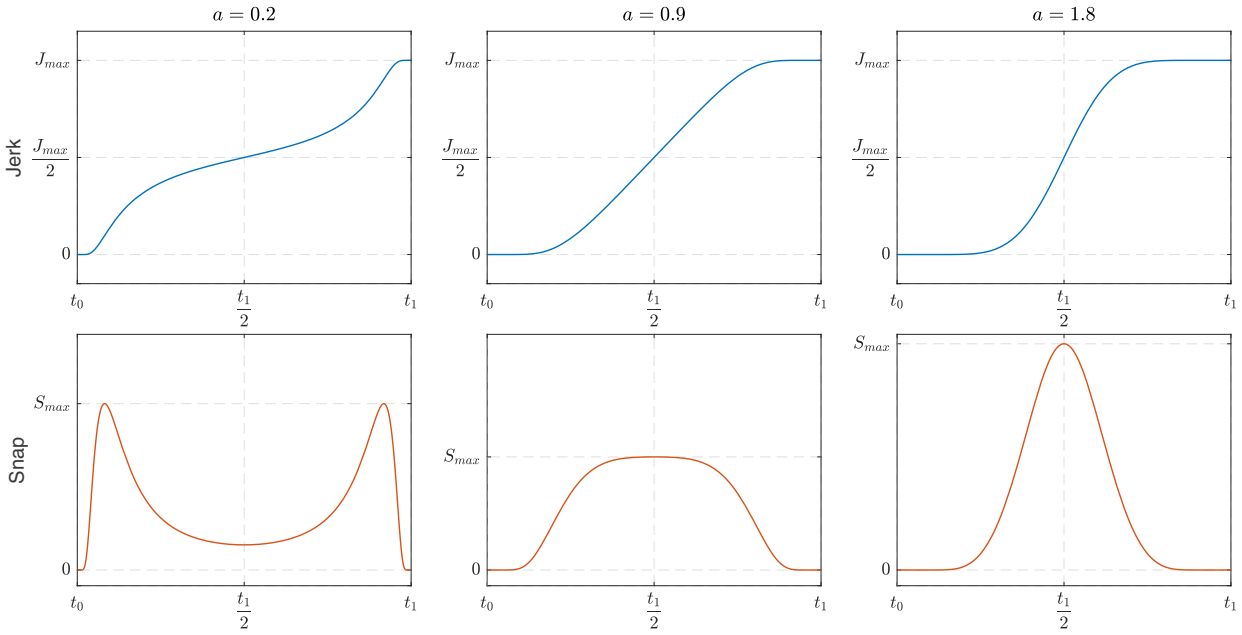


Fig. 4. Jerk and snap profile shapes under different values of a.

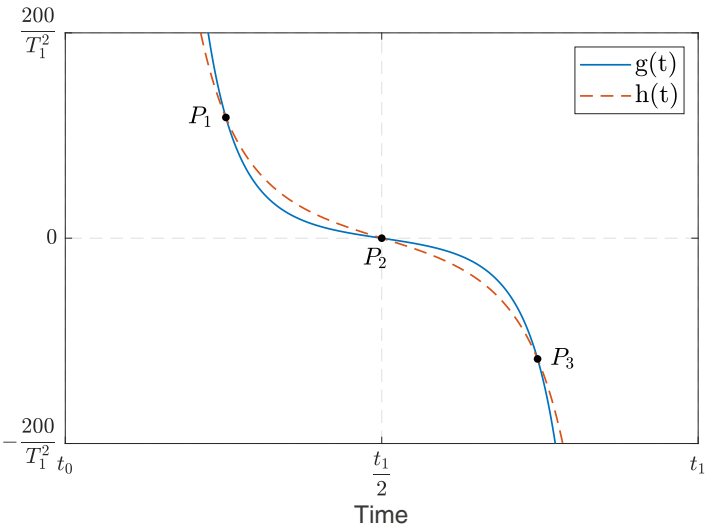


Fig. 5. Intersection points of two functions g and h ($a = 0.6$).

It is evident that one root of this equation is $\tau_1 = \frac{1}{2}$, i.e., $t = \frac{t_0+t_1}{2}$. In fact, as shown in Fig. 4, the number of roots varies depending upon the value of a, which results in different distributions of the extreme points. When the parameter a is below a certain value, the snap curve achieves its local minima at the midpoint of the time interval, at the two sides of which the two maximum extreme points are symmetrically located. Otherwise, the snap curve has a unique maximum extreme point at the midpoint of this segment. To find this critical value influencing the root number, two functions g and h are defined as follows:

$$g = \frac{a}{J_{\max}}[J_{\max} - 2j(t)]\dot{f}^2 \tag{13}$$

$$h = -\ddot{f} \tag{14}$$

As shown in Fig. 5, the number of intersection points between the two functions represents the number of roots of Eq. (12). With a small value of a, both function curves as well as two of their intersection points are rotationally symmetric about the fixed intersection point $(\frac{t_0+t_1}{2}, 0)$. The two intersection points on both sides shift gradually toward the middle and eventually coincide with the central intersection point with increasing a. A further growth in a has no more effect on the number of intersection points. It can be known that all three intersection points will happen to coincide at one point on condition that the gradients of the functions g and h are equal at the point $t = \frac{t_0+t_1}{2}$. The gradients of the two functions are respectively obtained as follows:

$$\begin{aligned}\dot{g}(t) &= \frac{a}{J_{\max}}[-2s(t)]\dot{f}^2 + \frac{2a}{J_{\max}}[J_{\max} - 2j(t)]\dot{f}\ddot{f} \\ &= -2a^2\frac{j(t)}{J_{\max}}\left[1 - \frac{j(t)}{J_{\max}}\right]\left[\frac{1}{(1-\tau_1)^2} + \frac{1}{\tau_1{}^2}\right]^3\frac{1}{T_1{}^3} + 4a\left[1 - 2\frac{j(t)}{J_{\max}}\right]\left[\frac{1}{(1-\tau_1)^2} + \frac{1}{\tau_1{}^2}\right]\left[\frac{1}{(1-\tau_1)^3} - \frac{1}{\tau_1{}^3}\right]\frac{1}{T_1{}^3}\end{aligned} \tag{15}$$

$$\dot{h}(t) = -\dddot{f} = -6\left[\frac{1}{(1-\tau_1)^4} + \frac{1}{\tau_1{}^4}\right]\frac{1}{T_1{}^3} \tag{16}$$

Substituting $t = \frac{t_0+t_1}{2}$ and setting $\dot{g}(\frac{t_0+t_1}{2}) = \dot{h}(\frac{t_0+t_1}{2})$ lead to the critical value: $a = \frac{\sqrt{3}}{2}$. When a is greater than this critical value, the snap will reach its maximal value at the midpoint of the time interval:

$$s_{\max} = s\left(\frac{t_0+t_1}{2}\right) = \frac{2aJ_{\max}}{T_1} \tag{17}$$

Since the derivative of the snap at the midpoint with respect to a is always greater than zero, namely:

$$\frac{ds\left(\frac{t_0+t_1}{2}\right)}{da} = \frac{2J_{\max}}{T_1} > 0 \tag{18}$$

it is straightforward to verify that the snap value at the midpoint has a monotonically increasing relation with the parameter a when other parameters remain constant.

From a practical perspective, the snap profile is expected to involve no more than one ramp-up and ramp-down phase, hence the case with multiple fluctuations is precluded from consideration. Meanwhile, the maximal snap should commonly be as small as possible with the same condition of the execution time and jerk limit. For these reasons, the parameter a is endowed with the value $\frac{\sqrt{3}}{2}$ in this paper. Then, with the selected value of a, in case that the snap is bounded by a given value $S_{\max}$, the relation of the actual maximal jerk value to the snap limit is given by:

$$j_{\max} = \frac{S_{\max} T_1}{\sqrt{3}} = \frac{S_{\max} T_s}{\sqrt{3}} \tag{19}$$

On this basis, the other actual maximal kinematic values can be written as:

$$\begin{cases} a_{\max} = \dfrac{S_{\max} T_s}{\sqrt{3}} \left(T_s + T_j\right) \\ v_{\max} = \dfrac{S_{\max} T_s}{\sqrt{3}} \left(T_s + T_j\right)\left(2T_s + T_j + T_a\right) \\ d_{\max} = \dfrac{S_{\max} T_s}{\sqrt{3}} \left(T_s + T_j\right)\left(2T_s + T_j + T_a\right)\left(4T_s + 2T_j + T_a + T_v\right) \end{cases} \tag{20}$$

As the snap has a direct correlation with the duration of varying jerk phases, the trade-off between the time optimality and smoothness is able to be regulated through its limit value $S_{\max}$, which is usually not imposed in the kinematic constraints for the generation of jerk-limited trajectories. Accordingly, this value can be customized by the user from the point of practical applications to meet different task requirements, bringing flexibility to the trajectory planning. If the snap limit is specified to a relatively large value, the operating efficiency will be given more emphasis, and thus will lead to a faster motion with less execution time. In contrast, a smaller value of the allowable snap signifies a smoother trajectory performed for a longer time. It should be indicated that no matter what value is chosen, the continuity of the yielded trajectories will remain unaffected.

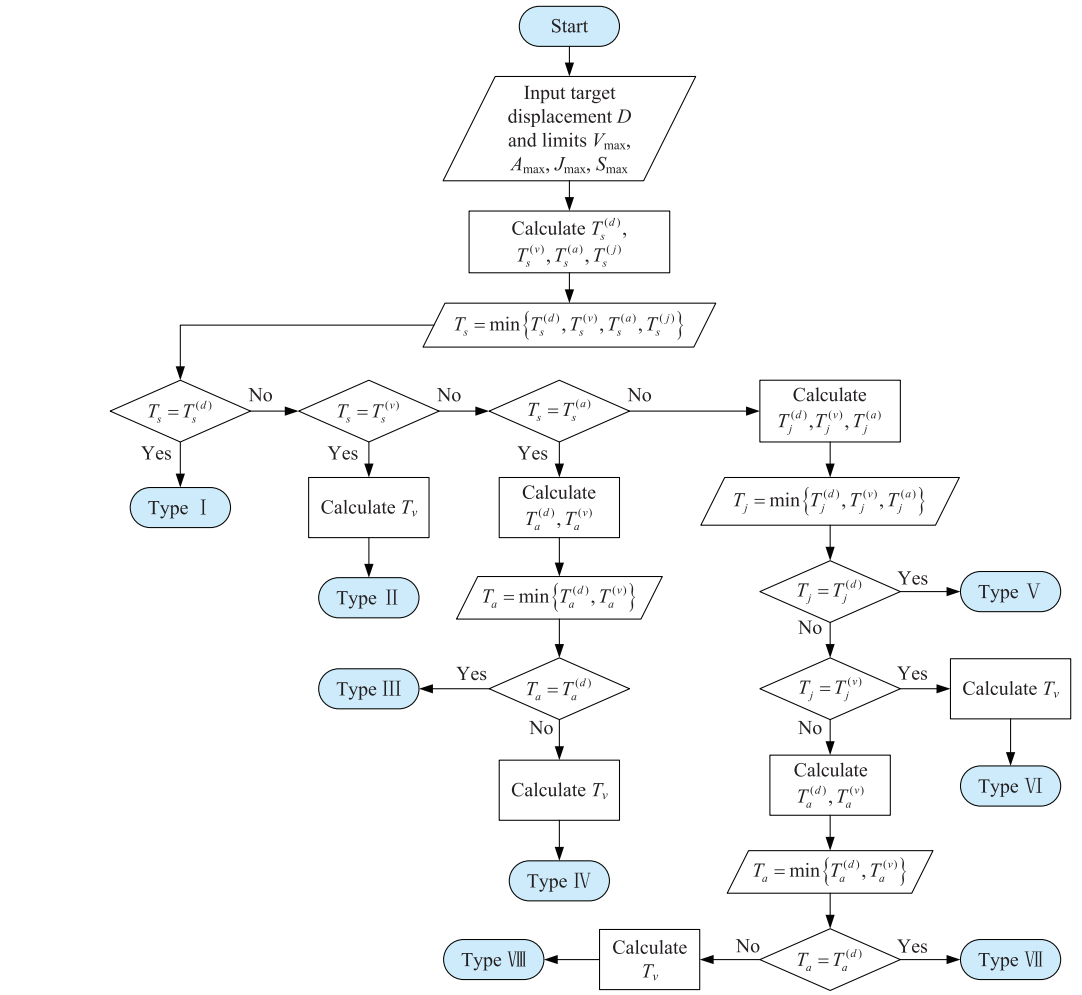
3.3. Calculation of time parameters

There is no fixed mechanism to find a valid time optimal solution for S-curve trajectory planning within prescribed constraints. In the event that the motion profile is of the general form of fifteen phases, the required time intervals can be solved by Eqs. (7) and (19). To cope with special cases corresponding to different motion parameters and profile shapes, some methods constructed constraint criteria to judge the profile type according to the ranges of given constraints, and these approaches usually results in an incomplete classification [37,63] or an intricate branching framework requiring prior analysis [43], especially in the presence of a large number of trajectory segments. A generic algorithm has been given in [55] to calculate higher-order trajectories, the basic thought of which is determining in turn the periods of a certain maximal derivative from higher to lower order; for each period, the preliminary time duration to obtain the desired displacement is progressively updated to the optimal value by checking for conflicts between the resulting trajectories and the bounds on the remaining lower order derivatives. Although this algorithm works well with high reliability, it is less reasonable from a computational point of view. All the time intervals must be calculated and revised by repeated comparisons irrespective of whether they exist or not, which gives rise to meaningless tests of the time values.

Indeed, with a view to minimize the computation complexity, it is worth recognizing that the restraining effect of lower-order constraints on a higher-order derivative has precedence over the inherent bound on the higher-order derivative. That is, a certain constant derivative period is not required when the limiting value of a lower-order derivative has been attained during the calculation of phase duration for higher-order derivatives. In view of this point, a more efficient modified algorithm is suggested to avoid redundant data handling, wherein the profile type is anticipated according to the calculation results at each step such that only the necessary time periods are dealt with. The jerk profile can be classified into eight types based on the phases involved. In each step, the upper bounds of the corresponding time interval for each constraint of lower derivatives are calculated under the limit condition of this phase, among which the minimal one is chosen. With this time, one of the bounds on lower derivatives will be obtained, then the time interval for this derivative is calculated based on the time values resulted from previous steps until the displacement requirement is satisfied. Consequently, this algorithm enables the kinematic limits to be attained in the shortest time within the bounds and thus guarantees the time optimality. Moreover, the geometric symmetry of the trajectory profiles is taken advantage of to facilitate the deduction and expression of relevant equations. It is possible to extend the basic flow of this algorithm to other S-curve models in terms of the time intervals to be resolved. The flowchart of the algorithm is illustrated in Fig. 6 and the detailed steps are described as follows:

Step 1: *Determination of the varying jerk phase duration* T_s

(1) First, the relation between the final displacement $d(t_{15})$ and the time interval T_s with the specified snap limit $S_{\max}$ is established from Eq. (20) without considering the bounds on jerk, acceleration and velocity, which implies that T_j, T_a and T_v



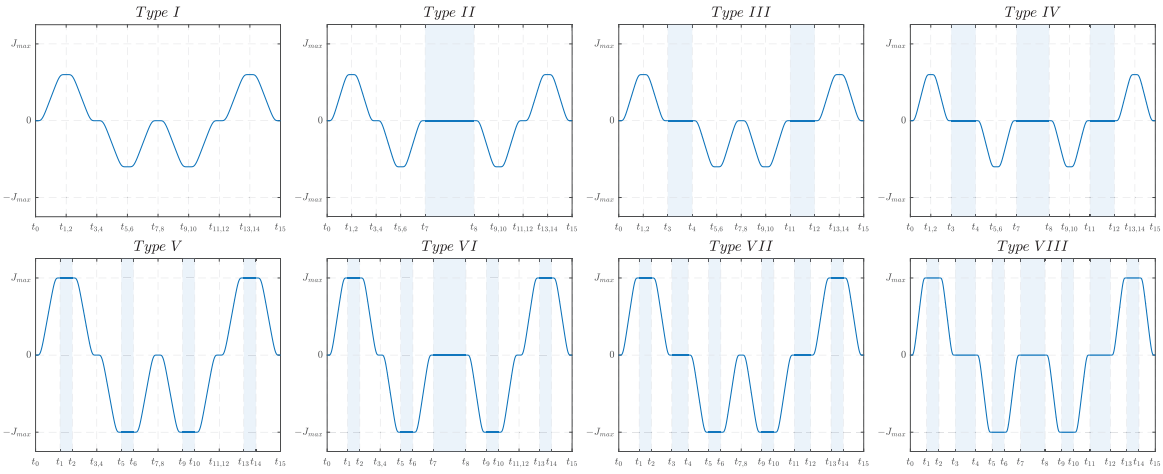


Fig. 6. Flowchart of the time parameters determination process.

are nonexistent:

$$d_{\max} = \left| d\left(t_{15} | T_j = T_a = T_v = 0\right) \right| = \frac{8 S_{\max} {T_s}^4}{\sqrt{3}} \tag{21}$$

In order to achieve the target displacement, the left side is set equal to D, and an additional superscript is added to distinguish between the values of T_s computed with different restrictive conditions below, then T_s determined by the displacement is:

$$T_s^{(d)} = \sqrt[4]{\frac{\sqrt{3}|D|}{8 S_{\max}}} \tag{22}$$

This value may violate the given kinematic constraints, therefore, it is mandatory to verify the upper bounds of T_s associated with them.

(2) Then, the limitation of maximum velocity is introduced. It is clear that the velocity reaches its peak value at the time instant t_7:

$$v_{\max} = v\left(t_7 | T_j = T_a = 0\right) = \frac{2 S_{\max} {T_s}^3}{\sqrt{3}} \tag{23}$$

In the light of the fact that $v_{\max}$ should not be larger than the velocity bound of the actuator $V_{\max}$, the corresponding critical value of T_s is obtained as follows:

$$T_s^{(v)} = \sqrt[3]{\frac{\sqrt{3} V_{\max}}{2 S_{\max}}} \tag{24}$$

(3) Next, the limitation of the maximum acceleration is taken into consideration. The maximal acceleration is reached at time instant t_3:

$$a_{\max} = a\left(t_3 | T_j = 0\right) = \frac{S_{\max} {T_s}^2}{\sqrt{3}} \tag{25}$$

Then the boundary value of T_s in accordance with the acceleration constraint is calculated by setting $a_{\max}$ equal to $A_{\max}$:

$$T_s^{(a)} = \sqrt{\frac{\sqrt{3} A_{\max}}{S_{\max}}} \tag{26}$$

(4) Finally, T_s confined by the jerk limit can be directly derived from Eq. (19):

$$T_s^{(j)} = \frac{\sqrt{3} J_{\max}}{S_{\max}} \tag{27}$$

The bound providing the strictest permissible value is the determinant one for the corresponding time interval. This way, the value of T_s that fulfils all constraints is resulted from Eq. (28) and T_s will be taken as a known parameter in the remaining deduction process of subsequent steps.

$$T_s = \min\left\{T_s^{(d)},\ T_s^{(v)},\ T_s^{(a)},\ T_s^{(j)}\right\} \tag{28}$$

Notice that among the four time intervals, T_s is the unique one that must be present in the motion profile. Depending on the magnitude relationship among the calculated time values, the existence of other time intervals is inferable so that certain steps can be omitted to promote the computation speed. Here, there exist four possible cases to be discussed:

Case 1: If $T_s^{(d)} = \min\{T_s^{(d)},\ T_s^{(v)},\ T_s^{(a)},\ T_s^{(j)}\}$, it means that the desired displacement is the sole limiting factor on the execution time. There is no need to proceed to the calculations of other time intervals, only the varying jerk phases exist, i.e., $T_j = T_a = T_v = 0$. None of the kinematic values (jerk, acceleration and velocity) can reach their limits. In this case, a practical maximum jerk $j_{\max} = S_{\max} {T_s^{(d)}}^2/\sqrt{3}$ is used to replace the limiting jerk value $J_{\max}$ in Eq. (5).

Case 2: If $T_s^{(v)} = \min\{T_s^{(d)},\ T_s^{(v)},\ T_s^{(a)},\ T_s^{(j)}\}$, the maximum available velocity can be reached without the constant jerk and acceleration segments ($T_j = T_a = 0$). The limit values of acceleration and jerk are unachievable. The motion profile is calculated using the actual attainable maximal jerk value $j_{\max} = S_{\max} {T_s^{(v)}}^2/\sqrt{3}$ instead of $J_{\max}$. Go straight to step 4.

Case 3: If $T_s^{(a)} = \min\{T_s^{(d)},\ T_s^{(v)},\ T_s^{(a)},\ T_s^{(j)}\}$, T_j is equal to zero. The constant jerk segments are absent due to the acceleration constraint. The practically maximum value attained by the jerk is $S_{\max} {T_s^{(a)}}^2/\sqrt{3}$. Go straight to step 3.

Case 4: If $T_s^{(j)} = \min\{T_s^{(d)},\ T_s^{(v)},\ T_s^{(a)},\ T_s^{(j)}\}$, go to step 2.

Step 2: *Determination of the constant jerk phase duration* T_j

(1) Further calculation starts with the construction of the relational formula between the end displacement and T_j under the hypothesis that the acceleration and velocity constraints are not contravened. The value of $T_s = T_s^{(j)}$ obtained in step 1 is substituted in the above formula to resolve T_j required by the total displacement:

$$|d(t_{15}|T_a = T_v = 0)| = |D| = J_{\max}\left(8{T_s}^3 + 16{T_s}^2T_j + 10T_s{T_j}^2 + 2{T_j}^3\right)$$

$$\Rightarrow T_j^{(d)} = \sqrt[3]{\frac{T_s^3}{27} + \frac{|D|}{4J_{\max}} + \sqrt{\frac{|D|T_s^3}{54J_{\max}} + \frac{D^2}{16J_{\max}^2}}} + \sqrt[3]{\frac{T_s^3}{27} + \frac{|D|}{4J_{\max}} - \sqrt{\frac{|D|T_s^3}{54J_{\max}} + \frac{D^2}{16J_{\max}^2}}} - \frac{5T_s}{3} \tag{29}$$

Note that here the jerk limit $J_{\max}$ is exploited to express the displacement rather than the snap limit $S_{\max}$ to simplify the computation since the operations of step 2 are performed on the basis of the prerequisite that the maximum permissible jerk value can be achieved, namely, $j_{\max} = J_{\max}$. Similar treatments are also to be applied in the later steps.

(2) The upper bound value of T_j imposed by the velocity limit is given by:

$$v(t_7|T_a = 0) = V_{\max} = J_{\max}\left(2{T_s}^2 + 3T_sT_j + {T_j}^2\right) \Rightarrow T_j^{(v)} = -\frac{3T_s}{2} + \sqrt{\frac{{T_s}^2}{4} + \frac{V_{\max}}{J_{\max}}} \tag{30}$$

(3) Then, the effect of the acceleration bound is handled following from the calculation of the maximal acceleration attained at t_3:

$$a(t_3) = A_{\max} = J_{\max}\left(T_j + T_s\right) \Rightarrow T_j^{(a)} = \frac{A_{\max}}{J_{\max}} - T_s \tag{31}$$

At this point, the duration T_j conforming to all restrictions can be selected by Eq. (32) and it will then be treated as a given quantity.

$$T_j = \min\left\{T_j^{(d)},\ T_j^{(v)},\ T_j^{(a)}\right\} \tag{32}$$

Similar to step 1, three possibilities for the minimal time value need to be distinguished:

Case 1: If $T_j^{(d)} = \min\{T_j^{(d)},\ T_j^{(v)},\ T_j^{(a)}\}$, then $T_a = T_v = 0$ and the calculation is accomplished. The motion profile is composed of the varying and constant jerk segments. Only the jerk can reach the maximal admissible value $J_{\max}$.
Case 2: If $T_j^{(v)} = \min\{T_j^{(d)},\ T_j^{(v)},\ T_j^{(a)}\}$, the constant acceleration phases are not imperative for achieving the velocity limit, namely, $T_a = 0$. Go straight to step 4.
Case 3: If $T_j^{(a)} = \min\{T_j^{(d)},\ T_j^{(v)},\ T_j^{(a)}\}$, go to step 3.

Step 3: *Determination of the constant acceleration phase duration* T_a

(1) Again, the time period T_a is calculated with the given T_s and T_j resulting from the previous steps in order that the target displacement D is achieved, disregarding the bound on the velocity:

$$|d(t_{15}|T_v = 0)| = |D| = A_{\max}\left({T_a}^2 + 3T_jT_a + 6T_sT_a + 8{T_s}^2 + 2{T_j}^2 + 8T_sT_j\right)$$

$$\Rightarrow T_a^{(d)} = \frac{-\left(6T_s + 3T_j\right) + \sqrt{\left(2T_s + T_j\right)^2 + 4|D|/A_{\max}}}{2} \tag{33}$$

(2) The next required procedure is to calculate an appropriate value of T_a under the velocity constraint:

$$v(t_7) = V_{\max} = A_{\max}\left(T_a + 2T_s + T_j\right) \Rightarrow T_a^{(v)} = \frac{V_{\max}}{A_{\max}} - 2T_s - T_j \tag{34}$$

Eq. (35) gives the expression for T_a that complies with all limiting conditions:

$$T_a = \min\left\{T_a^{(d)},\ T_a^{(v)}\right\} \tag{35}$$

Likewise, there are two conceivable circumstances:

Case 1: If $T_a^{(d)} = \min\{T_a^{(d)},\ T_a^{(v)}\}$, then $T_v = 0$ and the trajectory type is determined. The velocity cannot attain its extreme allowable value in virtue of the displacement restriction, and thus the cruise stage with maximum velocity disappears.
Case 2: If $T_a^{(v)} = \min\{T_a^{(d)},\ T_a^{(v)}\}$, the cruise stage is required for generating feasible trajectories able to accomplish the target displacement. Go to step 4.

Step 4: *Determination of the constant velocity phase duration* T_v

In the case that the target displacement is sufficiently large for the velocity to reach its limiting value, the uniform motion time is determined based on the target displacement and previously computed time intervals as follows:

$$T_v = \frac{|D - d(t_{15}|T_v = 0)|}{V_{\max}} = \frac{|D|}{V_{\max}} - \left(4T_s + 2T_j + T_a\right) \tag{36}$$

So far, all the four time periods T_s, T_j, T_a and T_v determining univocally the S-curve trajectory have been completely established with closed-form solutions by the above four steps. Normally, in real applications, the kinematic constraints are fixed for a particular machine such that the limiting conditions of time periods induced by them can be precalculated and stored in the controller, and thus only the displacement limitation needs to be accounted for during real-time implementation of the demanded trajectory. Since this algorithm guarantees the satisfaction of the system constraints while covering all available profile types, it has the capability to provide reliable and effective trajectories under various conditions specified by the user.

4. Synchronization of multi-axis motions

As described in the aforegoing, in case the method is utilized for a multi-axis mechanism containing several actuators, for instance, a multi-DOF manipulator, though each actuator can be independently controlled, the synchronization of their trajectories should be taken into account in the design of the motion profiles. As the motion constraints for each actuator usually differ from each other, the individual execution times computed separately by single-axis motion laws are also different. At this time, the trajectories of the faster actuators are preferred to be executed in a coordinated way rather than in a time-optimal manner so as to alleviate the stress on the actuators and achieve a more natural motion.

For the industrial robots, the sigmoid S-curve model proposed is applicable for trajectory planning in both joint space and Cartesian space. One important thing to note here is that, it is not sufficient to merely unify the execution time of the trajectories along each axis according to the slowest one for planning a straight line trajectory in Cartesian space, since an accurate position relationship must now hold among different motion axes over the whole path. This suggests that the velocity synchronization is needed beyond the time synchronization to endow all DOFs with identically shaped velocity profiles [46].

4.1. Time synchronized trajectory generation

The minimum possible execution times calculated for each individual actuator with the algorithm in Section 3 are assumed to be $T_{exe,1}$, $T_{exe,2}$, ..., $T_{exe,N}$, among which the largest is selected as the synchronization time:

$$T_{exe}^{sync} = \max\left\{T_{exe,1},\ T_{exe,2},\ \ldots,\ T_{exe,N}\right\} \tag{37}$$

where N is the number of actuators and $T_{exe,k} = 8T_{s,k} + 4T_{j,k} + 2T_{a,k} + T_{v,k}$ $(k = 1,\ \ldots,\ N)$. Once the synchronized execution time is determined, the motion profiles of the actuators except for the slowest one should be shaped to adapt to this time. Since the requirement of time optimality is removed, there is indeed far more than one technique that can make the profile shapes meet the synchronization constraints on the basis of the parameters selected to be regulated. Conventionally, a time-consuming recursive procedure is needed to incrementally decrease the maximum values of kinematic parameters and recompute the time intervals until the condition of the final execution time is satisfied [38,45]. To avoid this problem, in this article, the issue of trajectory parameter adjustments is tackled in two different manners: time-scaled synchronization and minimum jerk (Min-jerk) synchronization; the former seeks to make a geometrical modification to the trajectories by scaling operations while maintaining the proportional relationship among the original time intervals. The time lengths of all segments that compose the trajectory for the k^{th} actuator are stretched linearly with a synchronization factor λ_k:

$$\left(T_{s,k}^{sync}, T_{j,k}^{sync}, T_{a,k}^{sync}, T_{v,k}^{sync}\right) = \lambda_k\left(T_{s,k}, T_{j,k}, T_{a,k}, T_{v,k}\right) \tag{38}$$

where $\lambda_k = \frac{T_{exe}^{sync}}{T_{exe,k}}$, $k = 1,\ 2, \ldots,\ N$.

Then the maximum jerk of the modified trajectory for each actuator is properly recalculated with the purpose of remaining compliant with the displacement requirement after the time scaling:

$$j_{\max,k}^{sync} = \frac{j_{\max,k}}{{\lambda_k}^3} \tag{39}$$

The shapes of the jerk profiles will remain unchanged after the synchronization procedure. Accordingly, the other kinematic parameters for each actuator will be updated with the modulation of the jerk:

$$a_{\max,k}^{sync} = \frac{a_{\max,k}}{{\lambda_k}^2},\ v_{\max,k}^{sync} = \frac{v_{\max,k}}{\lambda_k} \tag{40}$$

Whilst this time-scaled synchronization manner lowers the maximal values of all kinematic variables, the second one consists in minimizing the peak value of jerk $j_{\max,\,k}$ for each individual axis, this is actualized through searching for the

jerk limit value that prolongs the duration of the trajectory to the synchronization time in the range of $[0, j_{\max,k}]$, while remaining the maximum snap and other kinematic constraints unchanged. In this case, the duration of varying jerk segments will be compressed as far as possible, leading to a shape change of the jerk profile. Seeing that the evolution trend in the motion time versus the jerk limit value approximately follows an exponential-like decay displaying an upward concave shape, the search is performed with the aid of an accelerated dichotomy formulated as follows:

Algorithm Min-jerk synchronization algorithm.

Input:
(1) Individual execution times $T_{exe,k}$ $(k = 1, 2, \ldots, N)$, the synchronization time T_{exe}^{sync}, time error bound δ.
(2) The required displacements D_k, kinematic constraints $V_{\max,k}$ and $A_{\max,k}$, actual maximal jerk $j_{\max,k}$, the snap limit $S_{\max,k}$ $(k = 1, 2, \ldots, N)$.
Output: Synchronized time parameters $(T_{s,k}^{sync}, T_{j,k}^{sync}, T_{a,k}^{sync}, T_{v,k}^{sync})$ and minimized maximum jerk $j_{\max,k}^{sync}$ $(k = 1, \ldots, N)$.

for $k = 1$ **to** N **do**
  if $T_{exe,k} = T_{exe}^{sync}$ **then**
    Preserve the original time parameters;
  else
    $J_{LB} = 0$, $J_{UB} = j_{\max,k}$, $T_{UB} = T_{exe,k}$;
    while $|T_{exe,k} - T_{exe}^{sync}| > \delta$ **do**
      $\hat{J}_{\max} = (J_{LB} + J_{UB})/2$;
      Compute the time parameters $(\hat{T}_{s,k}, \hat{T}_{j,k}, \hat{T}_{a,k}, \hat{T}_{v,k})$ and execution time $\hat{T}_{exe}$ with $(D_k, V_{\max,k}, A_{\max,k}, \hat{J}_{\max}, S_{\max,k})$;
      if $J_{LB} \neq 0$ **then**
        if $\hat{T}_{exe} < T_{exe}^{sync}$ **then**
          $\hat{J}_{\max} = [T_{LB}\hat{J}_{\max} - \hat{T}_{exe}J_{LB} + (J_{LB} - \hat{J}_{\max})T_{exe}^{sync}]/(T_{LB} - \hat{T}_{exe})$;
        else
          $\hat{J}_{\max} = [\hat{T}_{exe}J_{UB} - T_{UB}\hat{J}_{\max} + (\hat{J}_{\max} - J_{UB})T_{exe}^{sync}]/(\hat{T}_{exe} - T_{UB})$;
        end if
        Compute the time parameters $(\hat{T}_{s,k}, \hat{T}_{j,k}, \hat{T}_{a,k}, \hat{T}_{v,k})$ and execution time $\hat{T}_{exe}$ with $(D_k, V_{\max,k}, A_{\max,k}, \hat{J}_{\max}, S_{\max,k})$;
        end if
        if $\hat{T}_{exe} > T_{exe}^{sync}$, **then**
          $J_{LB} = \hat{J}_{\max}$, $T_{LB} = \hat{T}_{exe}$;
        else
          $J_{UB} = \hat{J}_{\max}$, $T_{UB} = \hat{T}_{exe}$;
        end if
      end while
    $j_{\max,k}^{sync} = \hat{J}_{\max}$, $(T_{s,k}^{sync}, T_{j,k}^{sync}, T_{a,k}^{sync}, T_{v,k}^{sync}) = (\hat{T}_{s,k}, \hat{T}_{j,k}, \hat{T}_{a,k}, \hat{T}_{v,k})$;
  end if
end for
return $\{j_{\max,k}^{sync}, (\hat{T}_{s,k}, \hat{T}_{j,k}, \hat{T}_{a,k}, \hat{T}_{v,k})\}$ $(k = 1, 2, \ldots, N)$

The algorithm presented here fastens the convergence rate relative to the traditional dichotomy, normally the minimized peak jerk for one axis can be found within 5 cycles with millisecond accuracy, making it still feasible to use this synchronization manner for online trajectory generation.

4.2. Velocity synchronized trajectory generation

To generate a straight line motion that is often employed in operational space tasks, the displacements of each axis at any time during the motion should have the same ratio as their required increments to satisfy the path constraint. In this scenario, the minimum trajectory execution times for each axis do not need separate calculations, instead, their kinematic constraints are adjusted according to the desired displacements to obtain coordinated velocity profiles:

$$\mathbf{C}_{\max,k}{}' = |D_k| \min_{k=1,\cdots,N} \left\{ \frac{\mathbf{C}_{\max,k}}{|D_k|} \right\} \tag{41}$$

where $\mathbf{C}_{\max,k} = (V_{\max,k}, A_{\max,k}, J_{\max,k}, S_{\max,k})$ represents the constraint vector considered for axis k in our trajectory model.

After the regulation of constraints, the resulting trajectory profiles of all axes will share the same number of segments with identical time intervals, so it is only necessary to calculate the time parameters for one of the axes and then apply them directly to the other axes. For each trajectory segment, at least one axis attains its original maximum allowable value in either snap, jerk, acceleration or velocity while the others are in their feasible regions. Unlike the time synchronization which does not affect the overall movement time conditioned by the largest of the calculated minimum times for each axis, the velocity synchronization could lead to a longer execution time considering that some of the original kinematic limits may be degraded, and consequently no actuators can execute the motion as fast as that before synchronization. However, the resulting execution time is still the time optimal one within which the desired straight trajectory can be accomplished without infringing any kinematic limits because the motion is certain to evolve under the limit conditions all the time.

Fig. 7 gives an example of point to point motion from P_0 (0.2, 0.2) to P_f (0.8, 0.5) in 2D space to elucidate the impact of different synchronization procedures on the trajectories. The kinematic constraints and snap limit along the two axes x and y are chosen as: $\mathbf{C}_{\max,x} = \mathbf{C}_{\max,y} = (0.5m/s, 2m/s^2, 5m/s^3, 30m/s^4)$. Here the time synchronization is performed by the scaling operation, the result of the Min-jerk synchronization algorithm is similar and is not plotted in this figure for

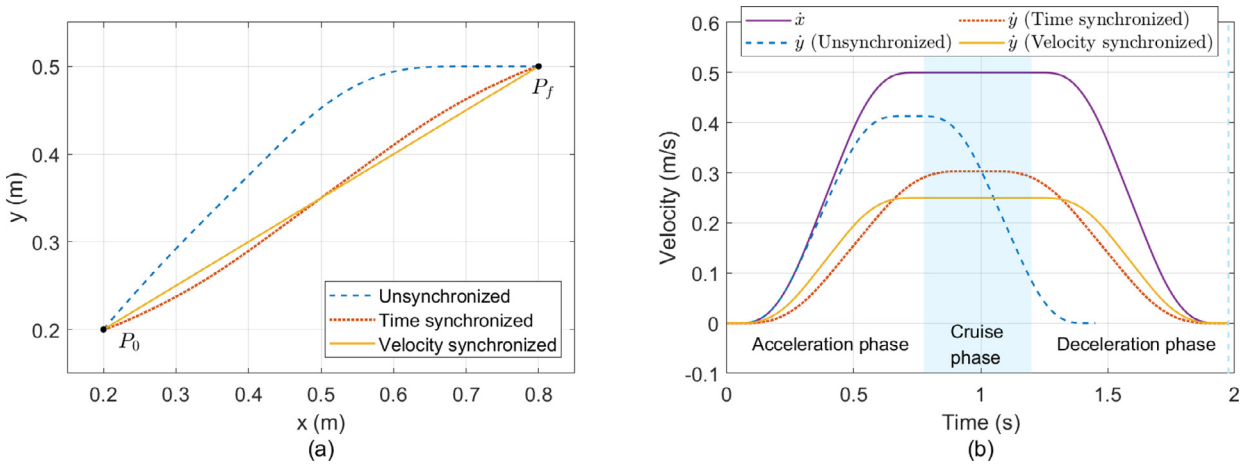


Fig. 7. Example for comparison of different synchronization procedures: (a) generated path; (b) velocity profiles.

clarity. From Fig. 7(a) it can be clearly seen that only velocity synchronized trajectories can generate a strictly rectilinear path, reflecting the linear dependence between the displacements of each DOF in the phase plane, and the velocity profiles in Fig. 7(b) reveal that both axes accelerate and decelerate simultaneously besides having the same motion time.

The two types of synchronization techniques suggested in this section can also be used in conjunction with the polynomial and trigonometric S-curve methods in Section 2 to produce desired multi-axis motions. Another possible application may be yielding a fixed-time trajectory with given constraints.

5. Numerical examples

To assess the effectiveness of the proposed approach and make a comparison with classical trajectory planning methods, two numerical examples on the generation of trajectories for robot manipulators, one in Cartesian space and another in joint space, are described in this section.

5.1. Example 1: 3-DOF manipulator

The first example represents the generation of a rectilinear motion in task space for the end-effector of a 3-DOF serial manipulator as illustrated in Fig. 8, the coordinates of the start and end points for the straight path are given in Table 1. The link lengths of the robot are assumed to be $l_1 = 0.85\ m$, $l_2 = 0.95\,m$, and $l_3 = 0.65\,m$. This task has been studied in [20] and executed with the use of higher-degree polynomials (of order 9, 7 and 5) guaranteeing the continuity of the trajectory up to the third derivative (jerk) and the same motion duration time. However, the polynomial profiles described therein only support setting the acceleration limit values while other bounds cannot be specified independently since the extreme kinematic values are mutually interrelated.

In this example, as shown in Table 2, the resulting peak values of end-effector velocities, accelerations and jerks along each axis for the three polynomials reported in [20] are taken as three groups of kinematic constraints to reveal the effect of different constraint sets on the performance of S-curve trajectory and provide a comparison between the results by the two methods. Remark that these limit values have been in accordance with the relation schema given by Eq. (41), this highlights the validity of the suggested synchronization technique. As stated earlier, the execution time of the trajectory depends on the specific user-defined value for the snap limit. First, the snap upper bounds are chosen to be no larger than the extreme values contained in the snap curves obtained from the polynomials, which are discontinuous at the endpoints. For the three constraint sets, they are set as $275.22\,\mathrm{m/s^4}$, $125.10\,\mathrm{m/s^4}$ and $41.70\,\mathrm{m/s^4}$, respectively.

According to the given conditions of the displacement and kinematic limits, the desired trajectories of the end-effector in the operational space were yielded by the presented method. Then the joint angular positions were established through inverse kinematics:

$$\theta_1 = \arctan\left(\frac{p_y}{p_x}\right)$$

$$\theta_2 = \arctan\left(\frac{p_z - l_1}{\sqrt{p_x^2 + p_y^2}}\right) - \arcsin\left(\sqrt{\frac{4l_2^2l_3^2 - \left[p_x^2 + p_y^2 + (p_z - l_1)^2 - l_2^2 - l_3^2\right]^2}{4l_2^2\left[p_x^2 + p_y^2 + (p_z - l_1)^2\right]}}\right)$$

$$\theta_3 = \arccos\left(\frac{p_x^2 + p_y^2 + (p_z - l_1)^2 - l_2^2 - l_3^2}{2l_2l_3}\right)$$

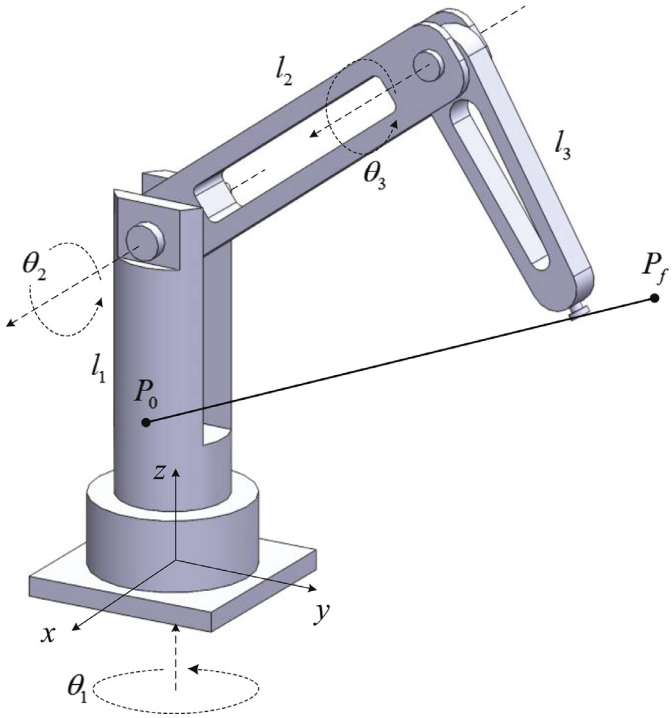


Fig. 8. Rectilinear motion of a 3-DOF robot manipulator.

Table 1
Coordinates of the desired path points and corresponding displacements for task 1.

Axis	Coordinate (m)		Displacement (m)
	Initial point P_0	Final Point P_f	
x	0.15	0	−0.15
y	0	1.3	1.3
z	0.4	0.9	0.5

Table 2
Kinematic constraints of the robot along each axis for task 1.

Kinematic constraints	Velocity (m/s)			Acceleration (m/s^2)			Jerk (m/s^3)		
	Axis x	Axis y	Axis z	Axis x	Axis y	Axis z	Axis x	Axis y	Axis z
Set 1	0.10	0.83	0.32	0.23	2.00	0.77	1.29	11.26	4.33
Set 2	0.11	0.94	0.36	0.21	1.83	0.71	0.86	7.49	2.88
Set 3	0.14	1.26	0.48	0.22	1.91	0.73	0.68	5.90	2.27

where (p_x, p_y, p_z) is the position of the end-effector in the Cartesian coordinate system. The corresponding angular velocities, accelerations and jerks were achieved by taking the numerical derivatives of angular positions with respect to time.

Figs. 9-13 represent the end-effector trajectories including displacements, velocities, accelerations, jerks and snaps in the direction of each coordinate axis under different constraint sets, and the graphic results of the joint temporal evolutions are illustrated in Figs. 14-17.

As depicted in Figs. 11-13, apart from the acceleration and jerk, the snap and even higher derivatives profiles of the end-effector also initiate and terminate at zero, making it possible to achieve improved smoothness and accuracy at the extremities relative to the trajectories depicted by the polynomials. Observing the trajectories generated with different constraint conditions, the longest constant velocity phase is reported for the first constraint set while the cruise phase disappears in the case of the third constraint set. This finding is readily understood because a lower velocity limit can be attained more quickly, particularly when looser higher-order constraints are available. These characteristics are consistent with those exhibited by the corresponding polynomial profiles, of which the time duration of a quasi-stationary movement arising from the root multiplicity decreases sequentially with the polynomial degrees of 9, 7 and 5. Also, it can be noted from the vari-

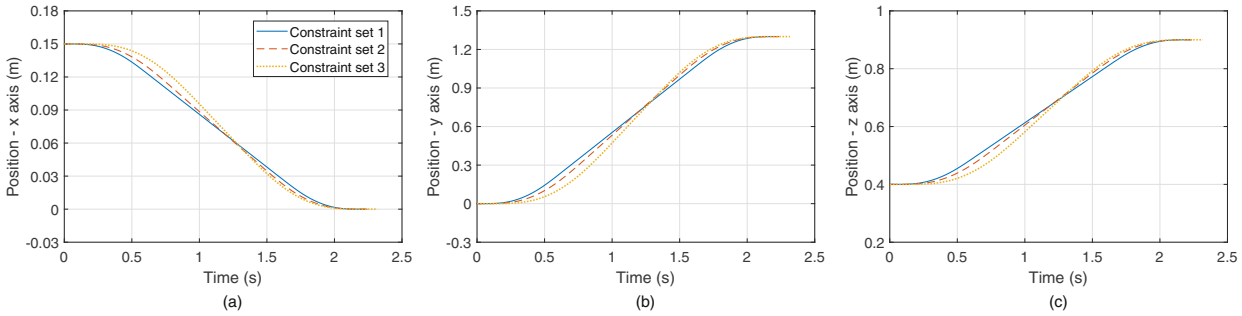


Fig. 9. End-effector position trajectories in the direction of (a) x-axis; (b) y-axis; (c) z-axis for task 1.

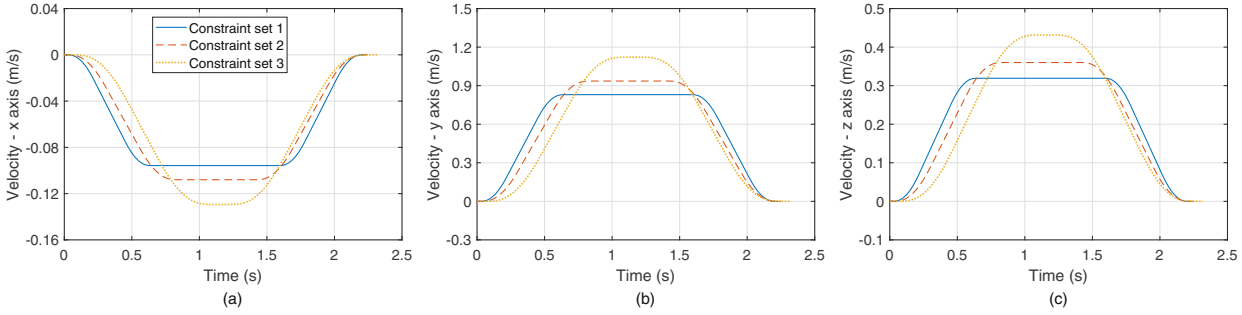


Fig. 10. End-effector velocity trajectories in the direction of (a) x-axis; (b) y-axis; (c) z-axis for task 1.

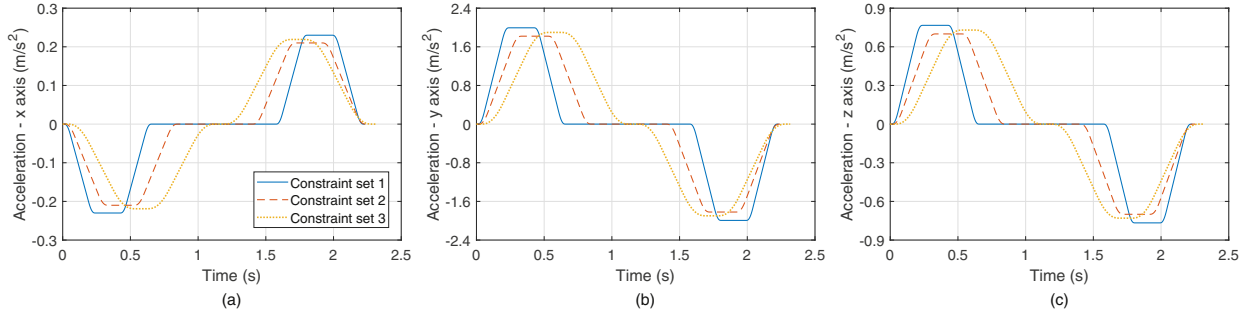


Fig. 11. End-effector acceleration trajectories in the direction of (a) x-axis; (b) y-axis; (c) z-axis for task 1.

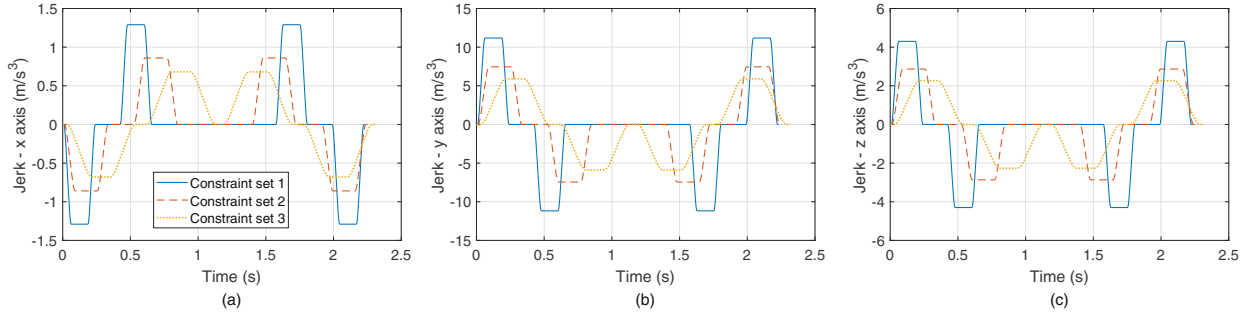


Fig. 12. End-effector jerk trajectories in the direction of (a) x-axis; (b) y-axis; (c) z-axis for task 1.

ations of the jerk or snap that the number of trajectory segments and the corresponding time periods are always the same for the trajectories along each axis.

In the meantime, Figs. 14-17 indicate that benefit from the notably smooth end-effector movement in Cartesian space, the motion curves of all robot joints always remain continuous. Analogously, the angular derivatives for the beginning and ending moments are null. The highest angular velocities, accelerations and jerks all occur at joint 1, whose angle position is desired to be varied more quickly in the first half of the path.

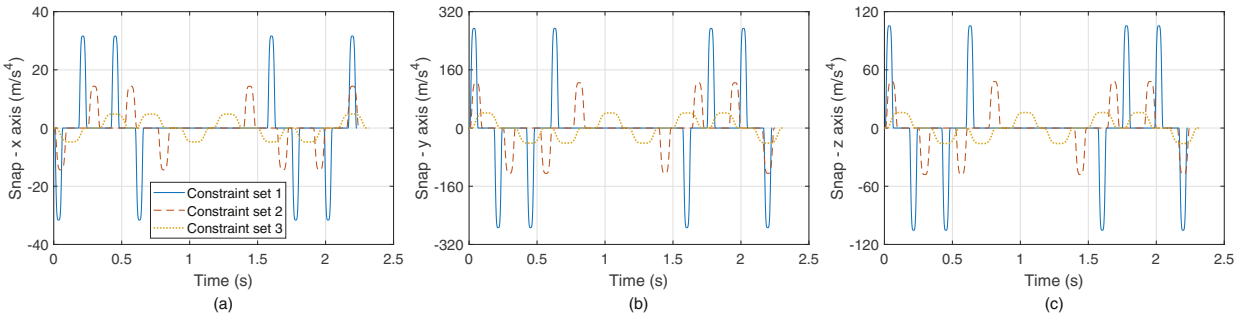


Fig. 13. End-effector snap trajectories in the direction of (a) x-axis; (b) y-axis; (c) z-axis for task 1.

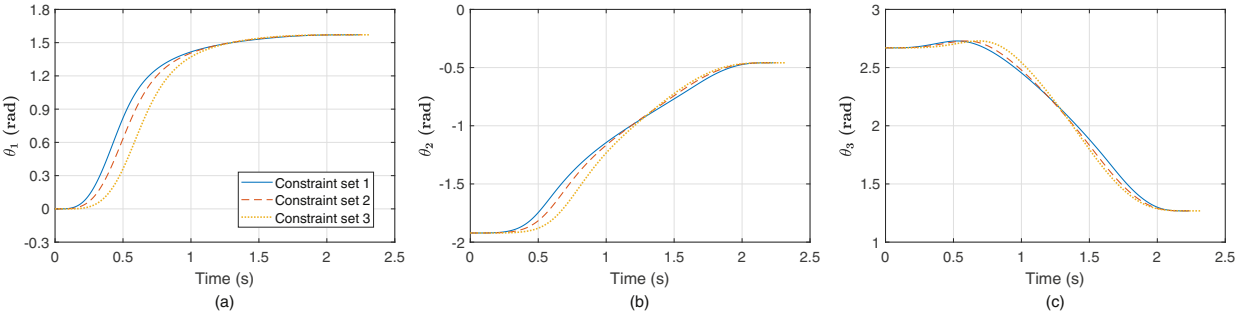


Fig. 14. Angular position trajectories of (a) joint 1; (b) joint 2; (c) joint 3 for task 1.

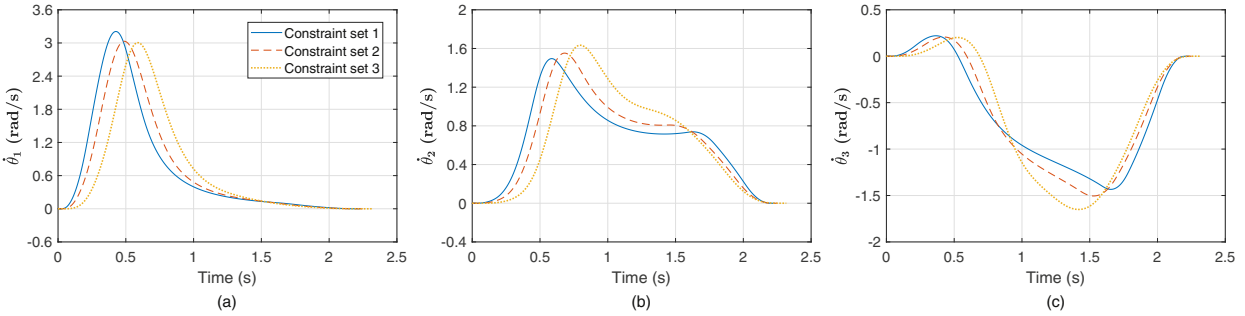


Fig. 15. Angular velocity trajectories of (a) joint 1; (b) joint 2; (c) joint 3 for task 1.

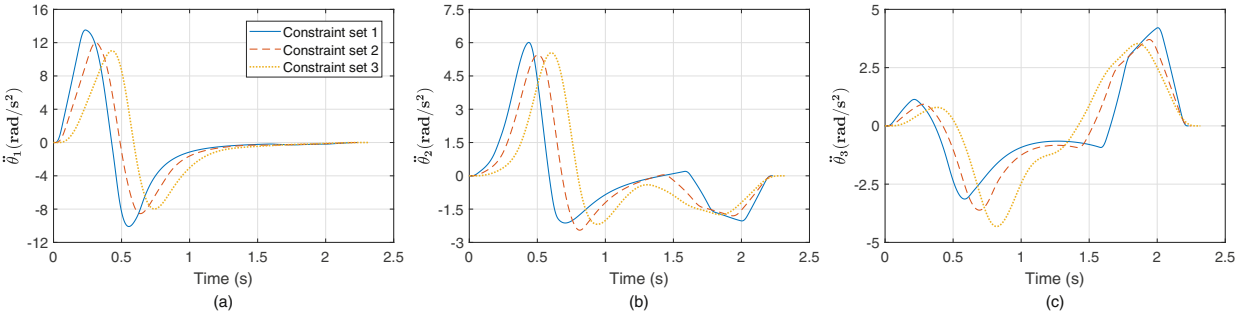


Fig. 16. Angular acceleration trajectories of (a) joint 1; (b) joint 2; (c) joint 3 for task 1.

Regarding the operating efficiency, the execution times under the former two constraint sets are respectively 2.2316 s and 2.2510 s, slightly faster than that of the polynomial trajectories (2.26214 s) in [20], whereas a longer time of 2.3179 s is demanded for the third set. This can be reasonably interpreted by the fact that among the three constraint sets, although the last one provides the highest maximum permitted values of velocity, the snap limit assigned for it has become the major restrictive factor at this time, leading to an actual peak value of velocity that is inferior to its limiting value. In effect,

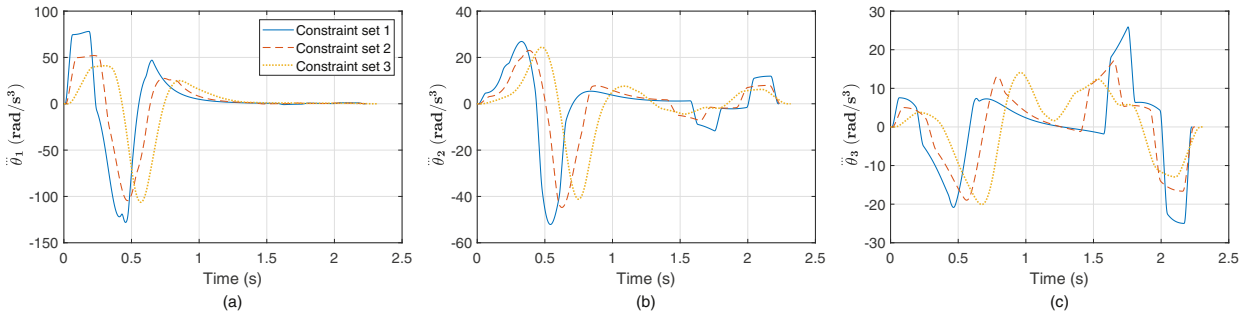


Fig. 17. Angular jerk trajectories of (a) joint 1; (b) joint 2; (c) joint 3 for task 1.

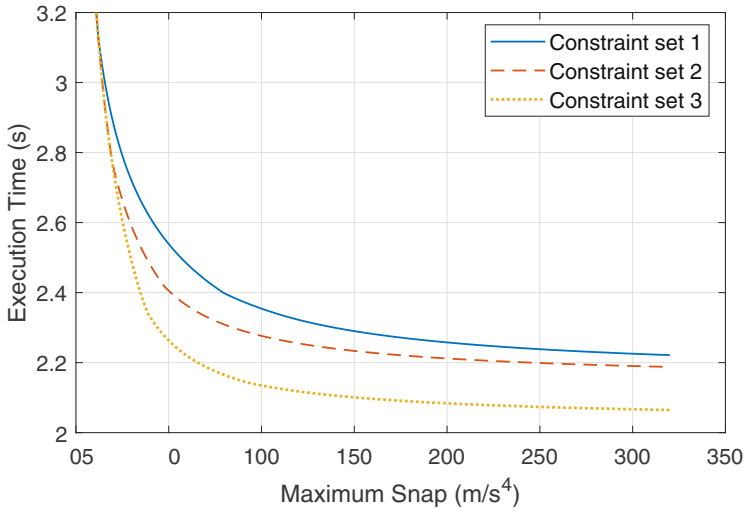


Fig. 18. Evolution of the execution time versus the maximum snap.

in case that the trajectory planning under the third constraint set is performed with the same snap limit as the former two, the required traveling times can be respectively shortened to 2.0700 s and 2.1147 s, about 8.5% and 6.5% lower than that of the polynomial trajectories, which is remarkably more efficient.

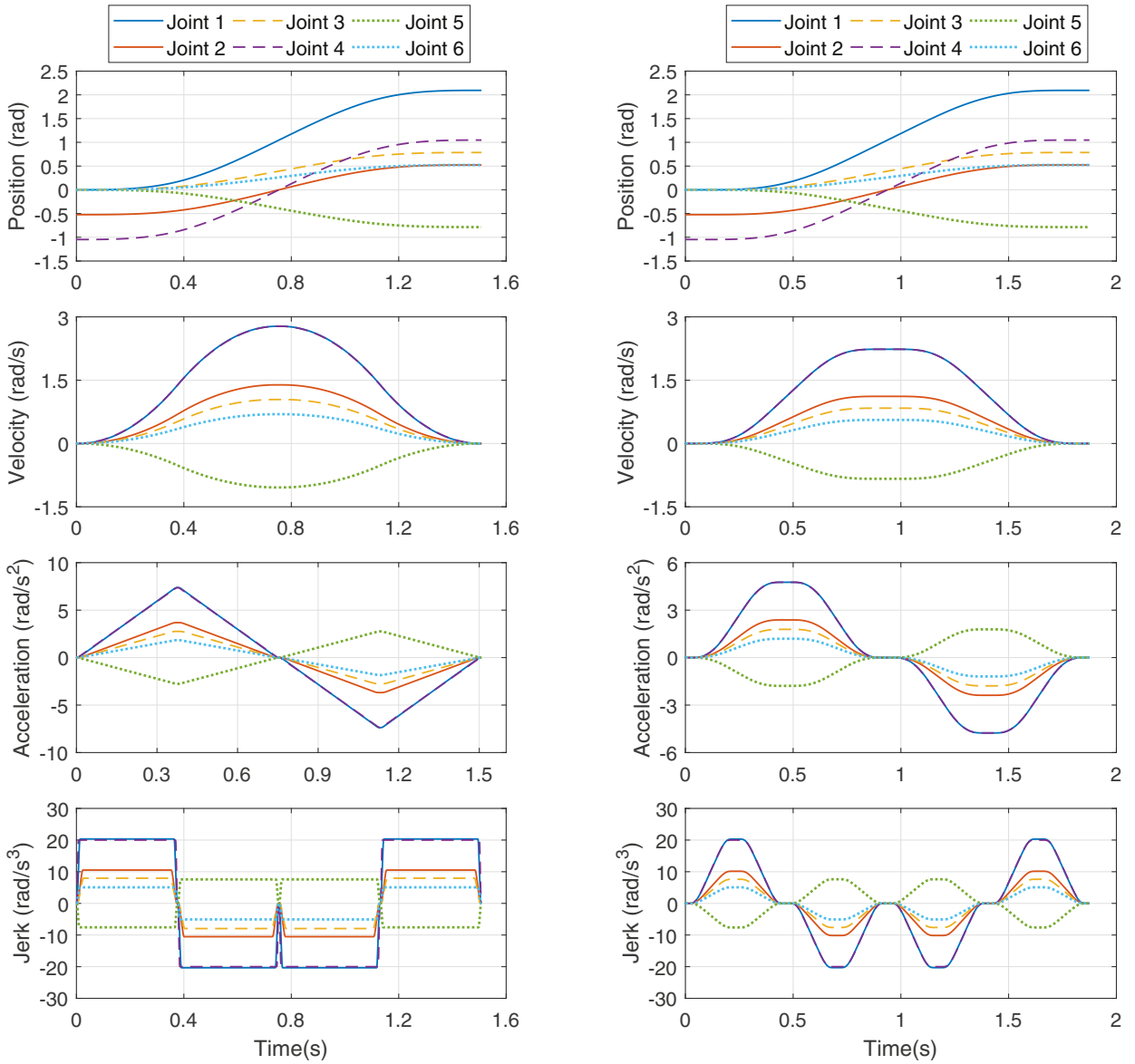
For a clear view, the evolution of the execution time with the change in the set-point value of the snap limit is evidenced in Fig. 18. All the curves completely overlap in the ultra-low snap region, as the motion time is affected only by the snap limit for a given displacement at this point. With increasing snap, the trajectory execution time first decreases fleetly and then evolves gently towards a steady-going value. This diagram can offer a reference for specifying an appropriate snap limit value subject to a given time requirement.

5.2. Example 2: 6-DOF manipulator

A pick-and-place task performed by a 6-DOF robotic manipulator, which is a typical application scenario for the proposed method, is chosen as another example to illustrate trajectory generation in joint space. Herein, the task data are taken to be identical to those in works [27] and [48]. Table 3 depicts the desired configurations along with the kinematic limits of the robot joints.

Table 3
Desired positions and kinematic limits of the robot joints for task 2.

		Joint					
		1	2	3	4	5	6
Positions (rad)	Initial point	0	$-\pi/6$	0	$-\pi/3$	0	0
	Final point	$2\pi/3$	$\pi/6$	$\pi/4$	$\pi/3$	$-\pi/4$	$\pi/6$
Kinematic constraints	Velocity (rad/s)	8	10	10	5	5	5
	Acceleration (rad/s^2)	10	12	12	8	8	8
	Jerk (rad/s^3)	30	40	40	20	20	20



(a) $S_{max} = 4000\ rad/s^4$ (b) $S_{max} = 150\ rad/s^4$

Fig. 19. Sigmoid S-curve motion profiles of the joints with (a) high snap; (b) low snap for task 2.

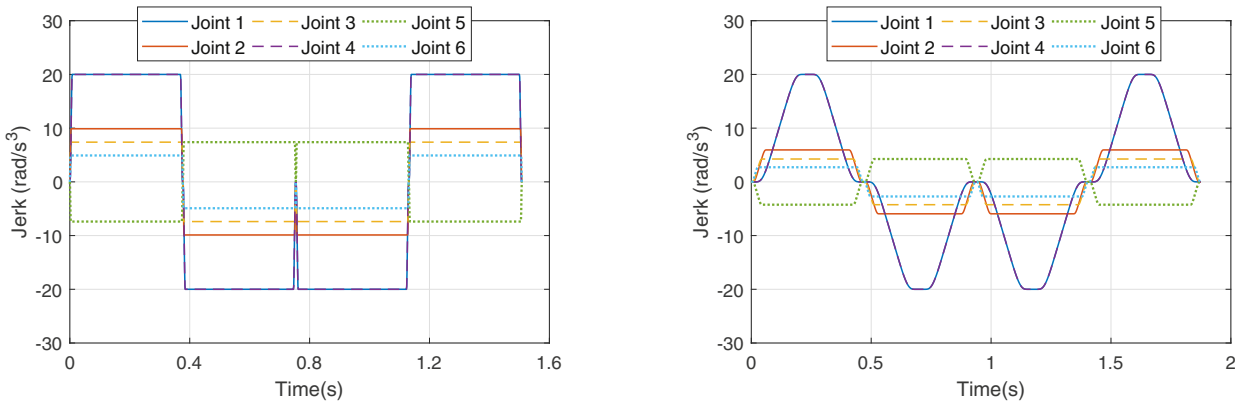
Fig. 19 presents the results achieved by the time-scaled synchronization manner with two classes of snap limits. It can be seen that in both cases the profiles of the displacement, velocity, acceleration and jerk for all joints are synchronized and smooth without any discontinuities during the whole motion while satisfying the imposed kinematic limits. Again, higher order derivatives of the trajectories are still continuous.

The comparison results of the kinematic performance with other canonical spline and S-curve methods mentioned previously are also summarized in Table 4 to illustrate the advantages of the present method. Here the single-axis third order trajectory suggested in [53] was used in conjunction with the synchronization technique provided in Section 4.1 to generate multi-axis motion profiles.

As shown in Fig. 19(a) and Table 4, if the maximum snap is set as a tremendous value (4000 rad/s^4) relative to the jerk constraints, the jerks will reach the saturation level in a comparatively short period, then the time-optimal trajectories that exhibit an analogous behavior to the 3rd order S-curve model can be reproduced by the proposed method. The overall moving time required is 1.5081 s, which is up to 40% less than that of the trigonometric S-curve trajectory in [48] and pretty close to that resulting from the 3rd order S-curve trajectory. However, the generated jerk profiles avert the abrupt changes in jerk which will lead to adverse vibrations and thus ensure a soft starting and braking.

Table 4
Comparative results of various trajectory planning methods for task 2.

Work	Trajectory model	Performance Measure		
		Execution Time (s)	Maximal Jerk Value (rad/s^3)	Continuity Class
Chettibi et al. [27]	Cubic spline	2.675	30 (Joint 1)	C_2 (Acceleration)
MU et al. [53]	Third order polynomial S-curve	1.4964	20 (Joint 1, 4)	C_2 (Acceleration)
Perumaal and Jawahar [48]	Synchronized trigonometric S-curve	2.5133	6.63 (Joint 1, 4)	C_3 (Jerk)
Present	Sigmoid S-curve	1.5081 ($S_{max} = 4000$ rad/s^4)	20.34 (Joint 1)	C_∞ (Infinity)
		1.8760 ($S_{max} = 150$ rad/s^4)	20.30 (Joint 1)	



(a) $S_{max} = 4000\ rad/s^4$ (b) $S_{max} = 150\ rad/s^4$

Fig. 20. Joint jerk profiles generated by the Min-jerk synchronization manner with (a) high snap; (b) low snap for task 2.

On the other hand, it is possible to achieve trajectories with a higher degree of smoothness provided that a relatively small magnitude of maximum snap (150 rad/s^4) is imposed (see Fig. 19(b)). In this case, the shapes of the motion profiles appear to be similar to those of the trigonometric model with more gently sloping jerk curves, yet the infinite differentiability has been extended to the entire trajectory and a savings of 25.4% in the motion duration is achievable with respect to the result in [48]. This is because one advantage of this approach is that it permits the machines to exert the largest possible actuation authority so as to acquire a minimum execution time, but without contravening any constraints. Seeing that at least one of the limit conditions for one DOF is attained, the resulting trajectories are always perceived to be optimal. For this task, the motion is conditioned by joint 4, which is the slowest among all joints since it is required to realize the longest traveling distance within the most rigorous restrictions. Actually, if the synchronization strategy introduced in Section 4.1 is applied to the sine jerk model in this example, an optimal motion time of 1.7395 s can be achieved with the maximum jerk attaining its ultimate value. Also notice that in both cases the sigmoid S-curve model outperforms the cubic spline model from [27] in terms of both speed and smoothness of the motion.

The comparative study reported above reveals that the proposed sigmoid S-curve model has the ability to incorporate the benefits of the 3rd order polynomial model and the trigonometric models in planning minimum time and jerk-continuous trajectories, when it is applied to manage the conflict between high operating speed and low mechanical vibrations with the aid of the snap limit, for which the choice of a particular value may be made depending on the specific task requirements.

If the Min-jerk synchronization method is utilized in lieu of the time-scaled synchronization method, the jerks of the joints will be minimized under the given constraints while leaving the motion duration unchanged. Fig. 20 displays the corresponding jerk profiles and Table 5 compares the two synchronization manners with regard to the peak jerk value for each joint. It is apparent that the Min-jerk synchronization method has a noticeable impact on reducing the peak jerk values relative to the other one under the condition of low snap, but this difference becomes minute in the high snap case, where the varying jerk phases account for only a tiny portion of the entire trajectory. The computation of the trajectory parameters takes an average of 0.35 ms for the time-scaled synchronization method and 2.8 ms for the Min-jerk synchronization method on a personal computer with an Intel Core i7-6700K 4.00 GHz processor and 8 GB RAM in Matlab environment, which is feasible for real-time implementation. Unlike the case in the previous example, the time synchronized trajectories for each axis generally consist of different number of segments with distinct durations. However, due to the particular limitation of the traveling distances in this task, the acceleration and velocity bounds are unreachable for all the joints, this difference is only visible when the Min-jerk synchronization manner is applied.

Table 5
Maximum jerk values for different synchronization techniques.

Joint	Maximum jerk value (rad/s^3)					
	Unsynchronized		Time-scaled synchronization		Min-jerk synchronization	
	High snap	Low snap	High snap	Low snap	High snap	Low snap
1	30	20.31	20.34 (67.8%)	20.30 (100%)	20 (66.7%)	20 (98.5%)
2	40	17.08	10.52 (26.3%)	10.15 (59.4%)	9.88 (24.7%)	5.95 (34.8%)
3	40	15.89	7.95 (19.9%)	7.61 (47.9%)	7.39 (18.5%)	4.25 (26.7%)
4	20	20	20 (100%)	20 (100%)	20 (100%)	20 (100%)
5	20	15.89	7.57 (37.9%)	7.61 (47.9%)	7.39 (37.0%)	4.25 (26.7%)
6	20	14.36	5.07 (25.4%)	5.08 (35.4%)	4.91 (24.6%)	2.72 (18.9%)

6. Experimental verification

With the aim of further validating the practical usability of the proposed approach, the experiments for the examples in last section were also conducted on a six-DOF manipulator Universal Robots UR5. The experimental setup is shown in Fig. 21 and it is composed of the robot arm itself, its control unit, and a workstation computer running an Ubuntu operative system. The computer communicates with the robot CB3 controller via the standard TCP/IP protocol. The reference position command data with the interpolation period of 1 ms were generated on the computer and then transmitted to the robot controller through the robotic operative system (ROS) framework. The actual joint and tool statuses of the robot were achieved by the Real-Time Data Exchange (RTDE) interface on 125 Hz during the trajectory execution.

For the first example, since the UR5 robot has a maximum reach of 0.85 m, the original desired coordinates in the y and z directions are halved, i.e., the robot tool performs a straight line motion from $P_0 = [0.15, 0, 0.2]$ to $P_f = [0, 0.65, 0.45]$ in Cartesian space. Additionally, the orientation of end-effector is fixed to $[\pi, 0, \pi/2]$ with the XYZ Euler angle notation. A piezoelectric CHENGTEC CL1100L accelerometer having a sensitivity of 991.5 mV/g and a range of ± 5 g has been mounted on the gripper of the robot to measure the motion-induced vibration along the y axis. The acceleration data was recorded using a MCC USB-1608 G DAQ board with the sampling frequency of 200 Hz. Given the acceleration limit $A_{max} = 2\ m/s^2$ along the axis, the execution time is 1.532 s for the 7-degree polynomial, which is considered as the optimal polynomial in [20]. The execution time of the sigmoid S-curve profile is adapted to this time by adjusting the snap while the other maximum kinematic values are restrained to be the same as those of the polynomial. Also, the trapezoidal profile originally provided by the robot controller was implemented with the same accleration limit and duration as a reference.

Fig. 22 presents the actual path of the end-effector and the corresponding measured joint profiles generated by the robot using the proposed approach. The results indicate that the robot is able to follow the planned path accurately while all the joint profiles are featured by fine smoothness. The vibrations measured by the accelerometer with the three techniques are evinced in Fig. 23. It can be noticed that the trapezoidal velocity profile leads to a significantly higher vibration level near the endpoints due to the jumps in acceleration than the other two profiles with continuous jerk profile. As summarized in Table 6, the sigmoid S-curve profile achieves an improvement on the reduced amplitude of residual vibration of 66% related

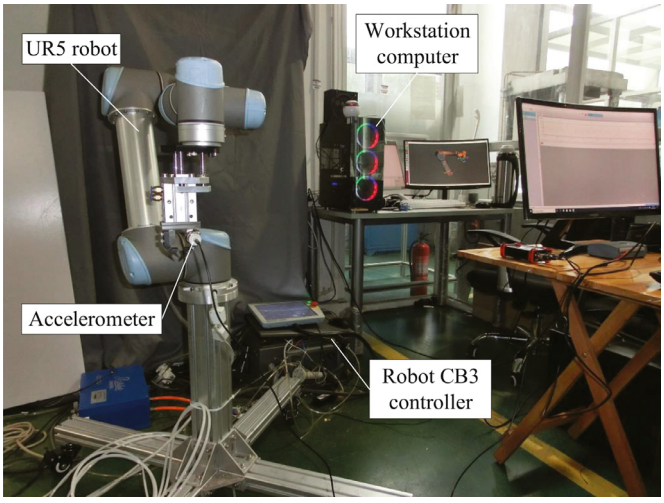


Fig. 21. Experimental setup.

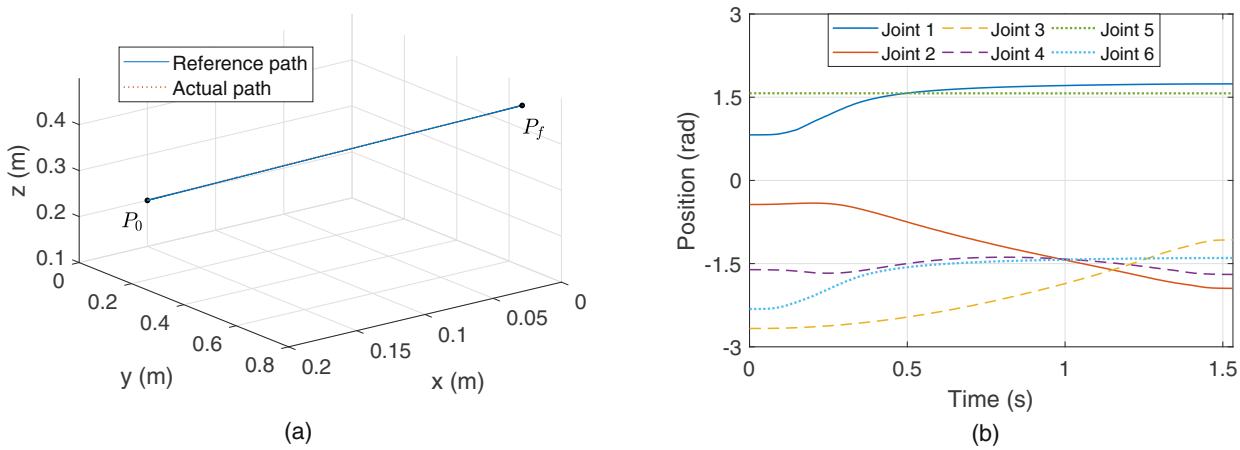


Fig. 22. Experimental results for task 1: (a) actual path; (b) measured joint position profiles.

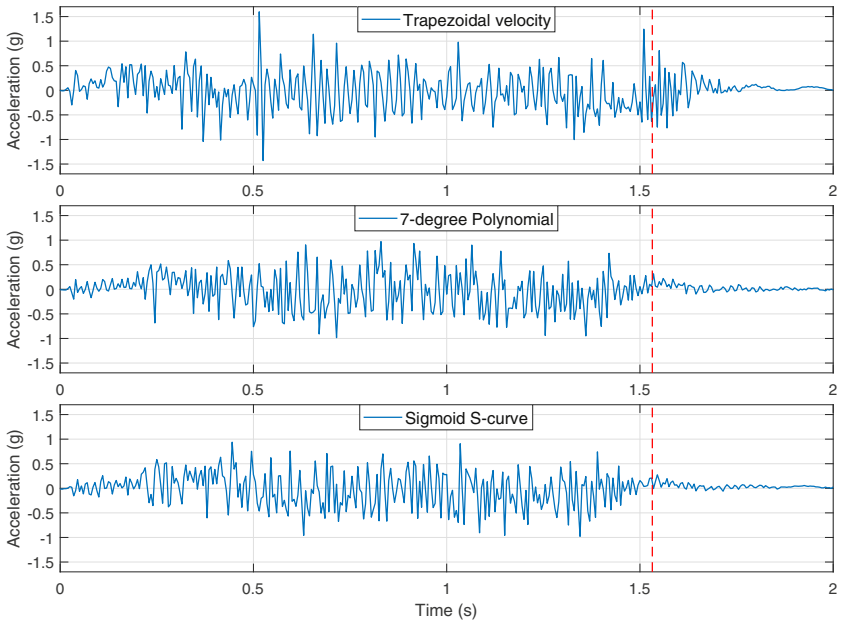


Fig. 23. Measured vibrations for task 1.

Table 6
Maximal amplitudes of residual vibration and settling time in the y axis for task 1.

	Trapezoidal velocity	Polynomial of 7-degree	Sigmoid S-curve
Residual vibration (g)	0.808 (100%)	0.315 (38.99%)	0.275 (34.03%)
Settling time (s)	1.71 (+11.62%)	1.59 (+3.79%)	1.55 (+1.17%)

to the trapezoidal velocity trajectory and of 12.7% related to the 7-degree polynomial trajectory. The settling time for which the vibration has attenuated within a specified error band (0.2 g) is also given in this table.

The results for the second example are illustrated in Fig. 24. Fig. 24(b) shows the simulated versus the measured position profile of the joint 4 when the snap limit is set to 150 rad/s^4, suggesting a satisfactory agreement between simulated and experimental results. The observations for other joints are very similar and are not presented here for space reasons. The third order S-curve profile was also executed with the same motion time by using the time-scaled synchronization manner for comparison (Fig. 24(a)). The tracking errors between the planned and actual positions for the two profiles are plotted in Fig. 24(c). It can be seen that the position errors tend to increase during the acceleration phase and revert to a tiny value less than 0.005° at the end of the trajectory. The maximal position tracking error of the sigmoid S-curve trajectory is reduced by about 15.8% to 1.49° compared to that of the third order one (1.77°). The experimental results clearly verify

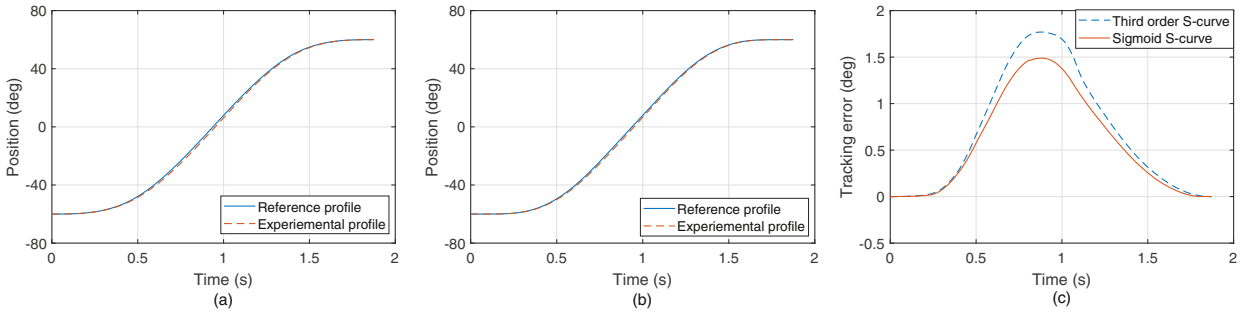


Fig. 24. Experimental results for task 2: (a) measured third order S-curve trajectory for joint 4; (b) measured sigmoid S-curve trajectory for joint 4; (c) tracking error for joint 4.

the validity of the proposed method in enhancing the dynamic positioning performance and vibration suppression of the system, benefiting from the great continuity of the generated trajectory.

7. Conclusions

This paper has introduced a novel methodology for generating extremely smooth and time optimal S-curve trajectories for robotics and manufacturing systems. Due to the modified sigmoid function adopted for the jerk profile, a crucial property of this approach is that all the derivatives of the trajectory vanish at the endpoints, thus guaranteeing an infinite order of continuity throughout the motion, which has a positive effect on eliminating vibrations and improving the positioning accuracy. As an alternative to conventional S-curve motion profiles, this method enables an enhancement in efficiency relative to the trigonometric S-curve models owing to the fact that the acceleration and jerk can be maintained at saturated levels whilst the increase in complexity is more manageable compared with the high order polynomial S-curve models.

The expressions of the motion profiles at pivotal time instants were derived by means of their geometrical symmetry to work out the closed-form solutions of time parameters, which are always reliable and precise under various scenarios. Furthermore, all the trajectories were produced without any violation of the actuator physical limitations by making the best use of them. Thus the method has the ability to strike a controllable compromise among the execution time, motion smoothness and computing complexity by varying the snap limit. Specifically, a high snap limit results in a short execution time comparable to that of the third order trajectory, while preserving a continuously differentiable jerk profile. For the planning of a straight line motion in Cartesian space, the resulting motion profiles can exhibit behaviors similar to those of polynomial profiles of different orders under corresponding kinematic limits, but different from the latter that only support the specification of acceleration limits, each motion constraint is allowed to be designated independently. The simulation and experimental results demonstrated the validity of the presented approach in yielding synchronized and jerk-continuous motion trajectories in different planning spaces for multi-DOF mechanical systems.

Future work will be devoted to extending this S-curve model to an asymmetric case with arbitrary boundary conditions for its applicability to more elaborate applications necessitating a complex spatial path comprising multiple segments. Additionally, the dynamics may further be considered to allow better capacity utilization. Finally, the collision avoidance strategy in complex working environment is also an important issue worthy of exploration.

Declarations of interest

None.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (51775332, 51675329, 51675342) and National Key Scientific Instruments and Equipment Development Program of China (2016YFF0101602, 2016YFC0104104), Shanghai Committee of Science and Technology (16441906000, 16XD1425000), and State Key Laboratory of Smart Manufacturing for Special Vehicles and Transmission System (GZ2016KF001).

References

[1] B. Tondu, S.A. Bazaz, The three-cubic method: an optimal online robot joint trajectory generator under velocity, acceleration, and wandering constraints, Int. J. Robotics Res. 18 (1999) 893–901.
[2] K.M. Lynch, Modern Robotics-Mechanics, Planning, and Control: Video supplements and software, 2017.
[3] S. Macfarlane, E.A. Croft, Jerk-bounded manipulator trajectory planning: design for real-time applications, IEEE Trans. Robotics Automation 19 (2003) 42–52.
[4] J.R.G. Martínez, J.R. Reséndiz, M.Á.M. Prado, E.E.C. Miguel, Assessment of jerk performance s-curve and trapezoidal velocity profiles, in: Engineering Congress (CONIIN), 2017 XIII International, IEEE, 2017, pp. 1–7.

[5] J. Wu, J. Wang, L. Wang, T. Li, Dynamics and control of a planar 3-DOF parallel manipulator with actuation redundancy, Mech. Mach. Theory 44 (2009) 835–849.
[6] J. Wu, G. Yu, Y. Gao, L. Wang, Mechatronics modeling and vibration analysis of a 2-DOF parallel manipulator in a 5-DOF hybrid machine tool, Mech. Mach. Theory 121 (2018) 430–445.
[7] M.H. Korayem, V. Azimirad, H. Vatanjou, A. Korayem, Maximum load determination of nonholonomic mobile manipulator using hierarchical optimal control, Robotica 30 (2012) 53–65.
[8] M. Korayem, A. Nikoobin, Maximum payload path planning for redundant manipulator using indirect solution of optimal control problem, Int. J. Adv. Manufact. Technol. 44 (2009) 725.
[9] M. Korayem, R.A. Esfeden, S. Nekoo, Path planning algorithm in wheeled mobile manipulators based on motion of arms, J. Mech. Sci. Technol. 29 (2015) 1753–1763.
[10] A. Gasparetto, V. Zanotto, A technique for time-jerk optimal planning of robot trajectories, Robotics Comput.-Integr. Manufact. 24 (2008) 415–426.
[11] H. Liu, X. Lai, W. Wu, Time-optimal and jerk-continuous trajectory planning for robot manipulators with kinematic constraints, Robotics Comput.-Integr. Manufact. 29 (2013) 309–317.
[12] K.J. Kyriakopoulos, G.N. Saridis, Minimum jerk path generation, Robotics and Automation, 1988, in: Proceedings., 1988 IEEE International Conference on, IEEE, 1988, pp. 364–369.
[13] Q. Zhang, S. Li, J. Guo, Smooth time-optimal tool trajectory generation for CNC manufacturing systems, J. Manufact. Syst. 31 (2012) 280–287.
[14] V. Zanotto, A. Gasparetto, A. Lanzutti, P. Boscariol, R. Vidoni, Experimental validation of minimum time-jerk algorithms for industrial robots, J. Intell. Robotic Syst. 64 (2011) 197–219.
[15] H. Ghariblu, M.H. Korayem, Trajectory optimization of flexible mobile manipulators, Robotica 24 (2006) 333–335.
[16] M. Korayem, M. Irani, S.R. Nekoo, Load maximization of flexible joint mechanical manipulator using nonlinear optimal controller, Acta Astronautica 69 (2011) 458–469.
[17] J. Angeles, in: Trajectory Planning: Pick-and-Place Operations, Fundamentals of Robotic Mechanical Systems, Springer, 2014, pp. 255–279.
[18] R.A. Osornio-Rios, R. de Jesús Romero-Troncoso, G. Herrera-Ruiz, R. Castañeda-Miranda, FPGA implementation of higher degree polynomial acceleration profiles for peak jerk reduction in servomotors, Robotics Comput.-Integrated Manufact. 25 (2009) 379–392.
[19] H. Wu, D. Sun, High precision control in PTP trajectory planning for nonlinear systems using on high-degree polynomial and cuckoo search, Optimal Control Appl. Methods 40 (2019) 43–54.
[20] M. Boryga, A. Graboś, Planning of manipulator motion trajectory with higher-degree polynomials use, Mech. Mach. theory 44 (2009) 1400–1419.
[21] A. Machmudah, S. Parman, A. Zainuddin, S. Chacko, Polynomial joint angle arm robot motion planning in complex geometrical obstacles, Appl. Soft Comput. 13 (2013) 1099–1109.
[22] B. Siciliano, L. Sciavicco, L. Villani, G. Oriolo, Robotics: modelling, Planning and Control, Springer Science & Business Media, 2010.
[23] A. Kirecci, M. Gilmartin, Improved trajectory planning using arbitrary power polynomials, Proc. Instit. Mech. Eng. 208 (1994) 3–13.
[24] R. Saravanan, S. Ramabalan, Evolutionary minimum cost trajectory planning for industrial robots, J. Intell. Robotic Syst. 52 (2008) 45–77.
[25] F.J. Abu-Dakka, I.F. Assad, R.M. Alkhdour, M. Abderahim, Statistical evaluation of an evolutionary algorithm for minimum time trajectory planning problem for industrial robots, Int. J. Adv. Manufact. Technol. 89 (2017) 389–406.
[26] H.-I. Lin, A fast and unified method to find a minimum-jerk robot joint trajectory using particle swarm optimization, J. Intell. Robotic Syst. 75 (2014) 379–392.
[27] T. Chettibi, H. Lehtihet, M. Haddad, S. Hanchi, Minimum cost trajectory planning for industrial robots, Eur. J. Mech.-A/Solids 23 (2004) 703–715.
[28] S. Lu, J. Zhao, L. Jiang, H. Liu, Solving the time-jerk optimal trajectory planning problem of a robot using augmented lagrange constrained particle swarm optimization, Math. Problems Eng. 2017 (2017).
[29] Y. Li, T. Huang, D.G. Chetwynd, An approach for smooth trajectory planning of high-speed pick-and-place parallel robots using quintic B-splines, Mech. Mach. Theory 126 (2018) 479–490.
[30] S.A. Ali, K.A.M. Annuar, M.F. Miskon, Trajectory planning for exoskeleton robot by using cubic and quintic polynomial equation, Int. J. Appl. Eng. Res. 11 (2016) 7943–7946.
[31] A. Gasparetto, V. Zanotto, A new method for smooth trajectory planning of robot manipulators, Mech. Mach. Theory 42 (2007) 455–471.
[32] J. Huang, P. Hu, K. Wu, M. Zeng, Optimal time-jerk trajectory planning for industrial robots, Mech. Mach. Theory 121 (2018) 530–544.
[33] Ü. Dinçer, M. Çevik, Improved trajectory planning of an industrial parallel mechanism by a composite polynomial consisting of Bézier curves and cubic polynomials, Mech. Mach. Theory 132 (2019) 248–263.
[34] S. Kucuk, Optimal trajectory generation algorithm for serial and parallel manipulators, Robotics Comput.-Integrated Manufact. 48 (2017) 219–232.
[35] X. Gao, Y. Mu, Y. Gao, Optimal trajectory planning for robotic manipulators using improved teaching-learning-based optimization algorithm, Ind. Robot 43 (2016) 308–316.
[36] P.H. Meckl, P.B. Arestides, Optimized s-curve motion profiles for minimum residual vibration, in: American Control Conference, 1998. Proceedings of the 1998, IEEE, 1998, pp. 2627–2631.
[37] F. Song, S. Yu, T. Chen, L.-N. Sun, Research on CNC simulation system with instruction interpretations possessed of wireless communication, J. Supercomput. 72 (2016) 2703–2719.
[38] H. Li, M. Le, Z. Gong, W. Lin, Motion profile design to reduce residual vibration of high-speed positioning stages, IEEE/ASME Trans. Mechatron. 14 (2009) 264–269.
[39] K.D. Nguyen, T.-C. Ng, I.-M. Chen, On algorithms for planning s-curve motion profiles, Int. J. Adv. Robotic Syst. 5 (2008) 11.
[40] Y. Bai, X. Chen, Z. Yang, A generic method to generate as-curve profile in commercial motion controller, ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, 2017 pp. V009T007A047-V009T007A047.
[41] K.-H. Rew, K.-S. Kim, A closed-form solution to asymmetric motion profile allowing acceleration manipulation, IEEE Trans. Industr. Electron. 57 (2010) 2499–2506.
[42] Y. Bai, X. Chen, H. Sun, Z. Yang, Time-Optimal Freeform S-Curve Profile Under Positioning Error and Robustness Constraints, IEEE/ASME Trans. Mech. 23 (2018) 1993–2003.
[43] A.Y. Lee, Y. Choi, Smooth trajectory planning methods using physical limits, Proc. Instit. Mech. Eng. Part C 229 (2015) 2127–2143.
[44] H. Li, A jerk-constrained asymmetric motion profile for high-speed motion stages to reduce residual vibration, IJCAT 53 (2016) 149–156.
[45] X. Broquere, D. Sidobre, I. Herrera-Aguilar, Soft motion trajectory planner for service manipulator robot, in: Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on, IEEE, 2008, pp. 2808–2813.
[46] R. Haschke, E. Weitnauer, H. Ritter, On-line planning of time-optimal, jerk-limited trajectories, in: Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on, IEEE, 2008, pp. 3248–3253.
[47] J. Lin, N. Somani, B. Hu, M. Rickert, A. Knoll, An Efficient and Time-Optimal Trajectory Generation Approach for Waypoints under Kinematic Constraints and Error Bounds, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2018, pp. 5869–5876.
[48] S. Perumaal, N. Jawahar, Synchronized trigonometric S-curve trajectory for jerk-bounded time-optimal pick and place operation, Int. J. Robot. Automation 27 (2012) 385.
[49] R.H. Castain, R.P. Paul, An on-line dynamic trajectory generator, Int. J. Robotics Res. 3 (1984) 68–72.
[50] T.-C. Lu, S.-L. Chen, Genetic algorithm-based S-curve acceleration and deceleration for five-axis machine tools, Int. J.Adv. Manufacturing Technol. 87 (2016) 219–232.
[51] S. Liu, An on-line reference-trajectory generator for smooth motion of impulse-controlled industrial manipulators, in: Advanced Motion Control, 2002. 7th International Workshop on, IEEE, 2002, pp. 365–370.

[52] C. Liu, Y. Chen, Combined S-curve feedrate profiling and input shaping for glass substrate transfer robot vibration suppression, Ind. Robot 45 (2018) 549–560.
[53] H. Mu, Y. Zhou, S. Yan, A. Han, Third-order trajectory planning for high accuracy point-to-point motion, Front. Electrical Electron. Eng. China 4 (2009) 83–87.
[54] L.S. Pontryagin, Mathematical Theory of Optimal Processes, Routledge, 2018.
[55] P. Lambrechts, M. Boerlage, M. Steinbuch, Trajectory planning and feedforward design for electromechanical motion systems, Control Eng. Pract. 13 (2005) 145–157.
[56] C.Y.W.H.S. Kai, L.M.W. Tianmiao, Algorithm for Smooth S-curve Feedrate Profiling Generation, Chin. J. Mech. Eng. 24 (2011) 1.
[57] L. Biagiotti, C. Melchiorri, Trajectory Planning For Automatic Machines and Robots, Springer Science & Business Media, 2008.
[58] J.-H. Chen, S.-S. Yeh, J.-T. Sun, An S-curve acceleration/deceleration design for CNC machine tools using quintic feedrate function, Comput.-Aided Design Appl. 8 (2011) 583–592.
[59] H.-b. Leng, Y.-j. Wu, X.-h. Pan, Research on cubic polynomial acceleration and deceleration control model for high speed NC machining, J. Zhejiang Univ.-SCIENCE A 9 (2008) 358–365.
[60] H. Li, Z. Gong, W. Lin, T. Lippa, Motion profile planning for reduced jerk and vibration residuals, SIMTech Tech. Rep. 8 (2007) 32–37.
[61] J.R. Rivera-Guillen, R. de Jesus Romero-Troncoso, R.A. Osornio-Rios, A. Garcia-Perez, G. Herrera-Ruiz, Design methodology for fully dynamic-controlled polynomial profiles and reduced tracking error in CNC machines, Int. J. Adv. Manufact. Technol. 51 (2010) 723–737.
[62] I. Herrera, D. Sidobre, On-line trajectory planning of robot manipulator's end effector in cartesian space using quaternions, 15th Int. Symposium on Measurement and Control in Robotics, 2005.
[63] G. Zhao, Y. Zhao, S. Wang, The acceleration/deceleration control algorithm based on trapezoid-curve jerk in CNC machining, Research Journal of Applied Sciences, Eng. Technol. 5 (2013) 1639–1645.