Robotica: (2020) volume 38, pp. 2121–2137. © Cambridge University Press 2020
doi:10.1017/S0263574719001863

Velocity Planning for Astronaut Virtual Training Robot with High-Order Dynamic Constraints

Lan Wang†, Lingjie Lin†*, Ying Chang† and Da Song†‡

†*College of Mechanical and Electrical Engineering, Harbin Engineering University, Harbin 150001, P.R. China*
E-mails: wanglan@hrbeu.edu.cn, changying0426@163.com, songda@hrbeu.edu.cn
‡*School of Mechanical Engineering, Northeast Electric Power University, Jilin 132012, P.R. China*

(Accepted December 14, 2019. First published online: February 10, 2020)

SUMMARY
In order to improve the training efficiency and establish a multi-person cooperative training simulation system, including "virtual human," in the process of virtual reality-based astronaut training, it is necessary to plan the velocity at which astronauts carry the target object. A velocity planning algorithm, combining a traditional six-stage acceleration/deceleration algorithm, based on a time-discrete model with high-order dynamic constraints, considering the elastic damping torque of the space suit, is proposed. The described algorithm is verified on MATLAB to prove its feasibility. Compared to other algorithms, the planning time of the proposed algorithm is significantly reduced.

KEYWORDS: Velocity planning; Astronaut virtual training; Dynamic constraint; Particle swarm optimization.

1. Introduction

Astronauts, working in space, do not feel the existence of gravity, but the inertial force still exists.[1,2] In the operation of carrying or installing solar panels, astronauts need to overcome the inertial force of the target object. If the target object has a small mass, its inertial force will not be too large, as long as the acceleration is not high. During the operation, astronauts can adjust the speed and force of their arm, by visual observation, to guarantee that the object will reach the target point, without collision. However, if the target object has a large mass or a large inertial force, it is difficult for the astronaut to ensure that the velocity at the target point is zero. Therefore, it is necessary to conduct astronaut simulation training on Earth.

In this paper, the velocity of the target object is planned, so as to reach the target point smoothly, at the shortest possible time, while in virtual reality (VR) training environment.[1]The astronaut training system, based on VR technology, can train astronauts in this process, effectively and efficiently. The training system is a cable-driven parallel robot with 6-degrees of freedom (DOFs). The maximum load of the robot is 300 kg and the maximum working space is $600 \times 600 \times 600$ mm. The robot can simulate the motion characteristics of large-mass objects, in space, as well as the tactile and force sense, under microgravity. This paper also provides a reference to the simulation of astronauts training system and the establishment of a "virtual human" system, enabling further a cooperative training model.

At present, the velocity planning algorithm is mainly applied to acceleration and deceleration control of computer numerical control (CNC) machine tools. There are many types of velocity planning

* Corresponding author. E-mail: linlingjie@hrbeu.edu.cn

algorithms, among which, the most commonly used, are linear acceleration/deceleration algorithms, S-curve acceleration/deceleration algorithms, and B-spline curve acceleration/deceleration algorithms.[3] Linear acceleration/deceleration algorithm is the simplest and most feasible method, but the resulting acceleration has abrupt change points, which has a significant impact on machine tools.[4] S-curve acceleration/deceleration algorithm ensures the continuity of velocity and acceleration, while it has less impact on machine tools. Hsien-I Lin[5] proposed a trajectory planning algorithm for joint robots, based on the minimum jerk principle. After discretizing the trajectory, the particle swarm algorithm (PSO) and K-means clustering are used to find the time required, by each trajectory, so as to obtain the respective jerk. However, when the time is too short, or the number of trajectory segments is high, the loop is difficult to converge, thus reducing the algorithm stability. Tzyy-Chyang Lu et al. proposed a velocity planning algorithm based on genetic algorithm[6] and one based on PSO,[7] simplifying the acceleration and velocity constraints into constants. Fusheng Liang et al.[8] used an iterative approach to gradually increase the control points of the B-spline curve from zero to a number, approximate to the constraints. Mingxing Yuan et al.[9] proposed a new forward and backward checking algorithm, which can directly find the optimal solution of each knot point, through multiple analytic equations. The above algorithms all adopted an assumption that the acceleration constraint of the point, to be planned, is a known quantity, before velocity planning. However, for this paper, the acceleration constraints can only be obtained after the initial velocity planning.

Akilan Bharathi et al.[10] proposed a feed rate optimization algorithm with high-order constraints. After discretizing the trajectory, the maximum allowed velocity was obtained, by binary search at each knot point, increasing the running time of the algorithm, thus achieving high-order dynamic constraints. Chenxi Fang et al.[11] proposed a time-optimal convex optimization algorithm. However, it is difficult to constrain the jerk, after conversion to a convex optimization problem. Due to the different objects studied, the above algorithms cannot be directly applied to the problems, described in this paper. At present, there is no velocity planning algorithm that takes human biomechanical model into consideration.

From the perspective of biomechanics, the dynamic acceleration and jerk constraints of the target object can be obtained by establishing a biomechanical model of human upper limb. After discretizing the time-continuous model, the jerk of each segment is obtained, by the improved PSO algorithm, while the near time-optimal velocity planning of the target object, with dynamic acceleration constraints, is realized.

The rest of this paper is organized as follows. In Section 2, a biomechanical model of human upper limb is established . In Section 3, the implementation process of the proposed algorithm is described. The results of a simulation example, under specific application, are included in Section 4, while a summary of the work is presented in Section 5.

2. Biomechanical Model of Human Upper Limb

2.1. Kinematics model of human upper limb

The movement of the object is mainly accomplished by the human shoulder, elbow, and wrist joints. Astronauts usually change the position and posture of the object, in one DOF at a time, without complex motion, when carrying large-mass objects in space. Therefore, the virtual training of astronauts carrying objects is regarded as horizontal push-pull movement,[1] while the object is limited to one-dimensional linear motion, in the sagittal plane. Each joint is simplified into a revolute pair, for analysis purposes, while a push-pull object model is established, as shown in Fig. 1.

The difference between the left and right arms of the person is not obvious, so in terms of mechanical properties, it can be considered that the left and right arms of human are symmetrical and bear the load on average. Under this premise, the push-pull object model, shown in Fig. 1, is disassembled, to establish the single-arm model, shown in Fig. 2, where l_1 is the length of the upper arm, l_2 is the length of the forearm, l_p is the distance between the projection of the human wrist and shoulder joint on the coronal plane, θ_1 is the angle between the upper arm and the coronal plane of human body, θ_2 is the angle of elbow, and F_l is the inertial force of the object.

Consider θ_1 and θ_2 as generalized coordinates, T_1 and T_2 as generalized forces, while the Lagrange dynamics equation of the right arm can be obtained:

$$T_1 = H_{11}\ddot{\theta}_1 - H_{12}\ddot{\theta}_2 + h_{122}\dot{\theta}_2^2 - 2h_{122}\dot{\theta}_1\dot{\theta}_2 + F_l[l_1\cos\theta_1 - l_2\cos(\theta_1 - \theta_2)] \quad (1)$$

https://doi.org/10.1017/S0263574719001863 Published online by Cambridge University Press

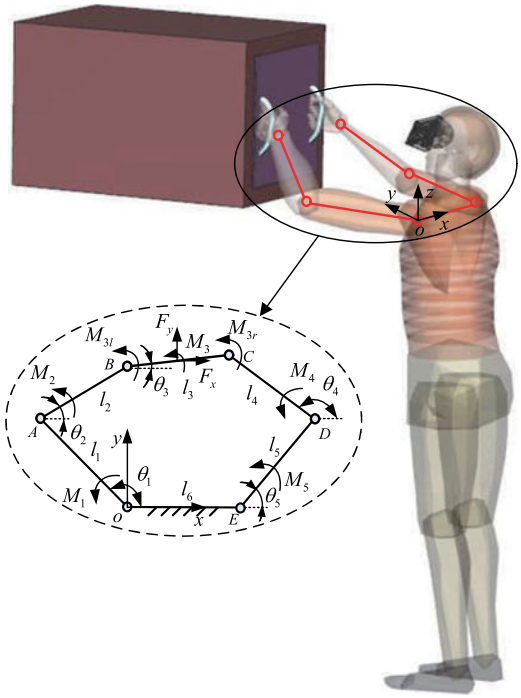

Fig. 1. Push-pull object model.

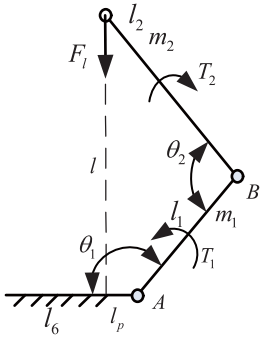

Fig. 2. Single-arm model.

$$T_2 = -H_{21}\ddot{\theta}_1 + H_{22}\ddot{\theta}_2 - h_{221}\dot{\theta}_1^2 + F_l l_2 \cos(\theta_1 - \theta_2) \tag{2}$$

where

$$\begin{cases} H_{11} = m_1 {l_{c1}}^2 + J_1 + J_2 + m_2\left({l_1}^2 + {l_{c2}}^2 - 2l_1 l_{c2}\cos\theta_2\right) \\ H_{22} = m_2 {l_{c2}}^2 + J_2 \\ H_{12} = H_{21} = m_2\left({l_{c2}}^2 - l_1 l_{c2}\cos\theta_2\right) + J_2 \\ h_{112} = h_{122} = -h_{221} = -m_2 l_1 l_{c2}\sin\theta_2 \end{cases}$$

where m_1, m_2 are the mass of upper arm and forearm, respectively; l_{c1}, l_{c2} are the distance from the centroid of the upper arm to the shoulder joint and the distance from the centroid of the forearm to the elbow joint, respectively; J_1, J_2 are the moments of inertia of the upper arm and the forearm, respectively.

According to the force analysis of the target object, the inertial force can be obtained, without a rotational motion of the object, under the assumption that the force of both arms is uniform:

$$F_l = \frac{1}{2}Ma \tag{3}$$

https://doi.org/10.1017/S0263574719001863 Published online by Cambridge University Press

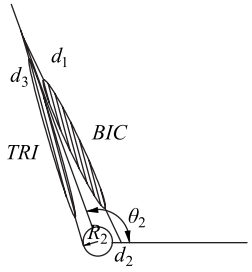

Fig. 3. The skeletal muscle model of elbow joint.

According to the geometric relationship, the distance between the projection of the human wrist and shoulder joint, on the coronal plane l_p, can be derived as:

$$l_p = l_2 \cos(\theta_1 - \theta_2) - l_1 \cos\theta_1 = \frac{1}{2}(l_6 - l_3) \tag{4}$$

The acceleration of the object is generated by the shoulder joint torque T_1 and the elbow joint torque T_2. The maximum acceleration a_{1max} of the object, determined by the shoulder and the maximum acceleration a_{2max} of the object, determined by the elbow joints, can be obtained by substituting Eqs. (3) and (4) into Eq. (1) and Eq. (2), respectively.

$$a_{1max} = \frac{2\left(T_1 - H_{11}\ddot{\theta}_1 + H_{12}\ddot{\theta}_2 - h_{122}\dot{\theta}_2^2 + 2h_{112}\dot{\theta}_1\dot{\theta}_2\right)}{Ml_p} \tag{5}$$

$$a_{2max} = \frac{2\left(T_2 + H_{21}\ddot{\theta}_1 - H_{22}\ddot{\theta}_2 + h_{221}\dot{\theta}_1^2\right)}{Ml_2 \cos(\theta_1 - \theta_2)} \tag{6}$$

The maximum acceleration of the object can be derived as:

$$a_{max} = \min(a_{1max}, a_{2max}) \tag{7}$$

During the motion of the human push-pull object, the angular velocity, the angular acceleration, the mass of the upper arm of human body, and the moment of inertia are negligible values, compared to the mass of the object, so the angular velocity and angular acceleration terms, in Eqs. (5) and (6), can be ignored.

The maximum acceleration that an object can generate is determined by the lowest value of a_{1max} and a_{2max}. In the actual training equipment, the value of l_2 is one order of magnitude higher than that of l_p. Therefore, the maximum acceleration of the object is mainly limited by the elbow joint torque T_2. By solving the elbow joint torque, the acceleration constraint of the object can be obtained.

2.2. Establishment of the elbow joint model

The muscles of the elbow joint are divided into the flexor group and the extensor group. The most important muscle in the elbow flexor group is the biceps brachii, and the most important muscle in the elbow extensor group is the triceps brachii. For the convenience of modeling, the biceps and triceps are considered equivalent to single-head muscles, while the equivalent position of the upper end of the muscle is determined by the equivalent length.

Muscle model was established by Hill and Zajac.[12] The muscle strength is related to the length of the muscle fiber l_M and the contraction speed of the muscle fiber v_M, that is, the muscle force can be described as:

$$F_f = F(l_M, v_M) \tag{8}$$

The skeletal muscle model of the elbow joint is shown in Fig. 3. d_1 and d_2 are the distances from the two stop points of the biceps brachii to the rotating center of the elbow joint, d_3 is the distance from the upper stop point of the triceps brachii to the rotating center of the elbow joint, while θ_2 refers to the angle of the elbow joint. From an anatomical point of view, the triceps are distributed

https://doi.org/10.1017/S0263574719001863 Published online by Cambridge University Press

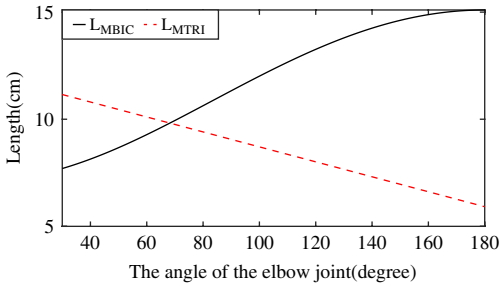

Fig. 4. The relationship between the muscle fiber lengths and the elbow joint angle: The black solid line and the red dashed line show the muscle fiber lengths of biceps and triceps, respectively.

along the outer envelope surface of the elbow joint, which is simplified to a spherical surface with a radius R_2.

In the process of muscle contraction, the muscle tension line can be approximated by a straight line. According to the geometric relationship, illustrated in Fig. 3, and the pennation angle of muscle, the muscle fiber length of biceps brachii l_{MBIC} can be derived as:

$$l_{MBIC} = \frac{\sqrt{d_1^2 + d_2^2 - 2d_1 d_2 \cos\theta_2} - l_{TBIC}}{\cos\varphi_{BIC}} \tag{9}$$

where l_{TBIC} refers to the sum of tendon lengths at the ends of the biceps brachii muscle fibers, and φ_{BIC} is the pennation angle of the biceps brachii.

In the process of muscle contraction, the tendon length can be regarded as a constant, because it changes slightly. The pennation angle also hardly changes, while its value remains generally below 20°, so its influence can be ignored.

The muscle fiber length of triceps brachii is

$$l_{MTRI} = L_{CEopt} + \frac{(\theta_0 - \theta_2)R_2}{\cos\varphi_{TRI}} \tag{10}$$

where θ_0 refers to the joint angle of elbow joint in the state of maximum isometric contraction, L_{CEopt} is ideal muscle fiber length, and φ_{TRI} is the pennation angle of triceps brachii.

According to Eqs. (9) and (10), the muscle fiber lengths of biceps brachii and triceps brachii during elbow joint bending are shown in Fig. 4. They are basically the same with the measurement results in ref. [13].

The method proposed in ref. [14] is used to obtain dynamic biceps brachii muscle force arm as:

$$d_{BIC} = \frac{dl_{MBIC}}{d_{\theta_2}} \tag{11}$$

The triceps brachii muscle force arm can be derived as:

$$d_{TRI} = \frac{dl_{MTRI}}{d_{\theta_2}} = -\frac{R_2}{\cos\varphi_{TRI}} \tag{12}$$

If the influence of pennation angle is ignored, d_{TRI} can be regarded as a constant, during the motion, to simplify the model.

The total torque of elbow joint T_2 (the torque, to make elbow joint abduct, is positive):

$$T_2 = T_f - (F_{fBIC}d_{BIC} + F_{fTRI}d_{TRI}) \tag{13}$$

where T_f is the elbow joint damping torque on the space suit, whose value is determined, as described in ref. [15]. According to [15], the damping characteristics of the elbow joint on the space suit are similar to those of the torsion spring. To simplify the calculation, the elbow joint damping torque measured, in [15], is replaced by Eq. (14).

$$T_f = k\left[\frac{180}{\pi}(\pi - \theta_2) - 30\right] \tag{14}$$

https://doi.org/10.1017/S0263574719001863 Published online by Cambridge University Press

where, $\begin{cases} k = \frac{1}{6} & \frac{\pi}{3} \leqslant \theta_2 \leqslant \frac{5\pi}{6} \\ k = \frac{1}{2} & \frac{5\pi}{6} \leqslant \theta_2 \leqslant \pi \end{cases}$

After the damping torque of the spacesuit is considered equivalent to the torsion spring, the human-space suit-object system can be equivalent to a mass-spring system. The output torque of the astronaut elbow joint must not only overcome the inertial force of the object but also overcome the spring force.

By substituting Eqs. (8)–(12) and (14) into Eq. (13), the functional relationship between the total torque of the elbow joint and the angle of the elbow joint can be obtained as:

$$T_2 = T\left(\theta_2, \dot{\theta}_2\right) \tag{15}$$

3. Six-Stage Velocity Planning Algorithm Based on Time-Discrete System

The maximum acceleration of the object can be obtained by substituting Eq. (15) into Eqs. (6) and (7).

$$a_{max} = a_{max}\left(\theta_1, \theta_2, \dot{\theta}_2\right) \tag{16}$$

Since the motion of the push-pull object is considered as one-dimensional linear motion, with single DOF, the geometric relationship, in Fig. 2, can illustrate that the angle of elbow joint θ_2 has a certain constraint relationship with the angle of shoulder joint θ_1, as shown in Eqs. (17) and (18).

$$l_1 \cos\theta_1 - l_2 \cos(\theta_2 - \theta_1) = l_p \tag{17}$$

$$l_1 \sin\theta_1 + l_2 \sin(\theta_2 - \theta_1) = l \tag{18}$$

Eq. (16) can be simplified to

$$a_{max} = a_{max}\left(\theta_2, \dot{\theta}_2\right) = a_{max}\left(l, \dot{l}\right) \tag{19}$$

It is evident that the maximum acceleration constraint of the object is related to the position and velocity, at the current moment. If, for example, the velocity, at the current moment, is unknown, the maximum acceleration constraint of the object cannot be derived.

In the seven-stage S-curve acceleration/deceleration algorithm, the object accelerates, at maximum acceleration, during the uniform acceleration stage. According to Eq. (19), the distance, along which the object has been moved, in that stage, can be obtained at any time.

$$l(t) = \iint_{t_1}^{t} a_{max}\left(l(t), \dot{l}(t)\right) dt + l_0 \quad (t > t_1) \tag{20}$$

where t_1 is the start time of the uniform acceleration stage, l_0 is the distance from the object to the coronal plane, at the start time of the uniform acceleration stage.

It is difficult to find the analytical solution of Eq. (20). Therefore, based on the seven-stage acceleration/deceleration algorithm and the motion law of the object, being carried by astronauts, a near time-optimal six-stage acceleration/deceleration algorithm, based on the time-discrete model, is proposed in this paper.

3.1. The existence analysis of uniform motion stage

In the seven-stage S-curve acceleration/deceleration algorithm, the existence condition of the uniform motion stage is that the object reaches the maximum speed, which depends on the maximum contraction speed of the biceps brachii and triceps brachii. The maximum contraction velocity of muscle fibers is about $10L_{CEopt}$. The maximum velocity of the target object, determined by the biceps brachii, can be obtained from the geometric relationship in Figs. 2 and 3 follows:

$$V_{MAXBIC} = \frac{l_1 l_2 \sqrt{d_1^2 + d_2^2 - 2d_1 d_2 \cos\theta_2}}{d_1 d_2 \sqrt{l_1^2 + l_2^2 - 2l_1 l_2 \cos\theta_2}} \sqrt{1 - \frac{l_p^2}{l_1^2 + l_2^2 - 2l_1 l_2 \cos\theta_2}} v_{maxBIC} \tag{21}$$

where v_{maxBIC} is the maximum contraction velocity of the biceps brachii.

https://doi.org/10.1017/S0263574719001863 Published online by Cambridge University Press

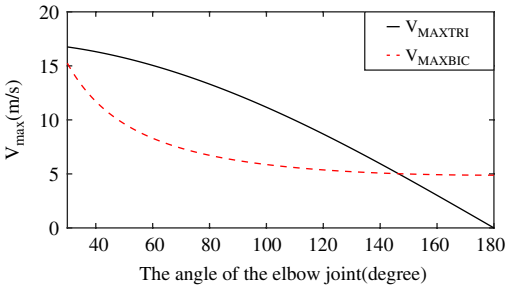

Fig. 5. The relationship between the maximum velocity of the target object and the elbow joint angle: The black solid line and the red dashed line show the maximum velocity of the target object determined by the triceps and the biceps, respectively.

The maximum velocity of the target object is determined by the triceps brachii as follows:

$$V_{MAXTRI} = \frac{l_1 l_2 \sin\theta_2}{\sqrt{l_1^2 + l_2^2 - 2l_1 l_2 \cos\theta_2 R_2}} \sqrt{1 - \frac{l_p^2}{l_1^2 + l_2^2 - 2l_1 l_2 \cos\theta_2}} v_{maxTRI} \tag{22}$$

where v_{maxTRI} is the maximum contraction velocity of the triceps brachii. The maximum velocity of the target object is defined as:

$$V_{MAX} = \min(V_{MAXBIC}, V_{MAXTRI})$$

The relationship between the maximum velocity of the target object and the elbow joint angle obtained by Eqs. (21) and (22) is shown in Fig. 5 where the angle of the elbow joint is limited within the range [30°, 180°].

Figure 5 shows that the actual velocity of the target object is difficult to reach the maximum velocity. When the elbow joint angle is 180°, the maximum velocity, determined by the triceps brachii, is zero. In fact, the velocity, after planning, is also zero. Therefore, during the process of carrying the object, the velocity of the target object cannot reach the maximum velocity; that is, the uniform motion stage does not exist. On this basis, the seven-stage acceleration/deceleration model can be simplified into a six-stage model.

3.2. Acceleration law analysis in carrying objects

In this paper, a typical push object process is taken as an example to analyze the acceleration law of the object. In this demonstration, the distance from the object to the coronal plane, at the initial moment, is 0.3 m and the target distance is 0.55 m. The parameters of the human body and muscle are set as shown in Table I, while the other structural parameters of the biceps brachii and triceps brachii are set according to ref. [16], as shown in Table II, where the multi-head muscle is equivalent to the single-head muscle. PCSA, as listed in Table II, is the physiological cross-sectional area of muscle.

The muscle activation of biceps brachii and triceps brachii, the output torque of elbow joint, and the elbow joint damping torque of the space suit can all be obtained by establishing a muscle simulation model and executing the corresponding algorithm, which takes longer time with Matlab, as shown in Fig. 6. The jerk, acceleration, and velocity of the target object, as well as the distance, from the target object to the coronal plane are shown in Fig. 7.

According to Figs. 6 and 7, the motion of each stage is analyzed as follows:

- t0–t2: the first stage of the seven-stage model, where the acceleration is positive and the absolute value of the acceleration is high;
- t2–t3: the uniform acceleration stage of the seven-stage model, where the jerk is not zero, due to the nonlinearity of the upper limb model and the variation of the damping torque of the space suit;
- t3–t5: the third stage of the seven-stage model, where the object accelerates at a reduced acceleration;

https://doi.org/10.1017/S0263574719001863 Published online by Cambridge University Press

Table I. The parameters of the human body and muscle

The length of the upper arm	0.28m
Maximum normalized muscle elongation	1.5
Active force–velocity curve factor	0.25
Maximum stress of muscle fiber	0.6MPa
The outer surface radius of the elbow joint, R_2	0.02m
Offset distance, l_p	0
The distance from the upper stop point of the biceps to the rotation center	0.35m
The length of the forearm	0.29m
Maximum passive muscle tension strain	0.6
Fiber volume of biceps brachii	$0.12^{-3}m^3$
Fiber volume of triceps brachii	$0.2^{-3}m^3$
Active force-length curve factor	0.45
Passive force-length index factor	5
The distance from the lower stop point of the biceps to the rotation center	0.04m

Table II. Structural parameters of the biceps brachii and triceps brachii

Muscle name	**PCSA(cm^2)**	L_{CEopt}**(cm)**	L_{Topt}**(cm)**
Long head of biceps	4.5	11.6	27.2
Short head of biceps	3.1	13.2	19.2
Biceps equivalent length	7.6	12.3	23.9
Long head of triceps	5.7	13.4	14.3
Lateral head of triceps	4.5	11.4	9.8
Medial head of triceps	4.5	11.4	9.1
Triceps equivalent length	14.7	12.2	11.3

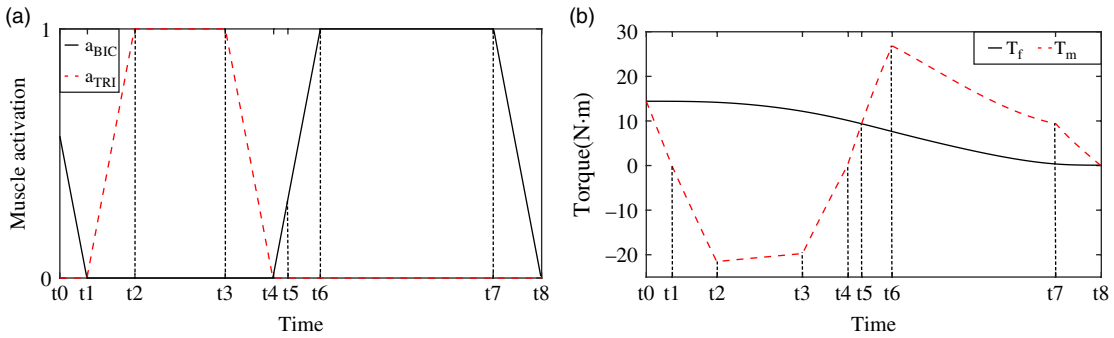

Fig. 6. Simulation results. (a) The muscle activation of biceps and triceps, in the process of pushing object, where the black solid line and the red dashed line show the muscle activation of biceps and triceps, respectively. (b) The black solid line and the red dashed line show the output torque of elbow joint and the elbow joint damping torque of the space suit, respectively.

- t5–t6: the deceleration stage of the seven-stage model, where the absolute value of acceleration increases gradually, while the acceleration is similar to that of the previous stage;
- t6–t7: the uniform deceleration stage of the seven-stage model, where, similarly, due to the nonlinearity of the upper limb model and the variation of the damping torque of the space suit, the acceleration is also not zero;
- t7–t8: the deceleration stage of the seven-stage model, where the absolute value of acceleration decreases gradually.

According to the above analysis, the damping torque of the elbow joint of the space suit has a great influence on the velocity planning. In addition to the nonlinear influence of the output torque of the human elbow joint, the acceleration constraint of the object is dynamic. It is difficult to use traditional methods for fast velocity planning, because the respective algorithm is very different from the traditional six-stage or seven-stage acceleration/deceleration algorithm. However, the acceleration

https://doi.org/10.1017/S0263574719001863 Published online by Cambridge University Press

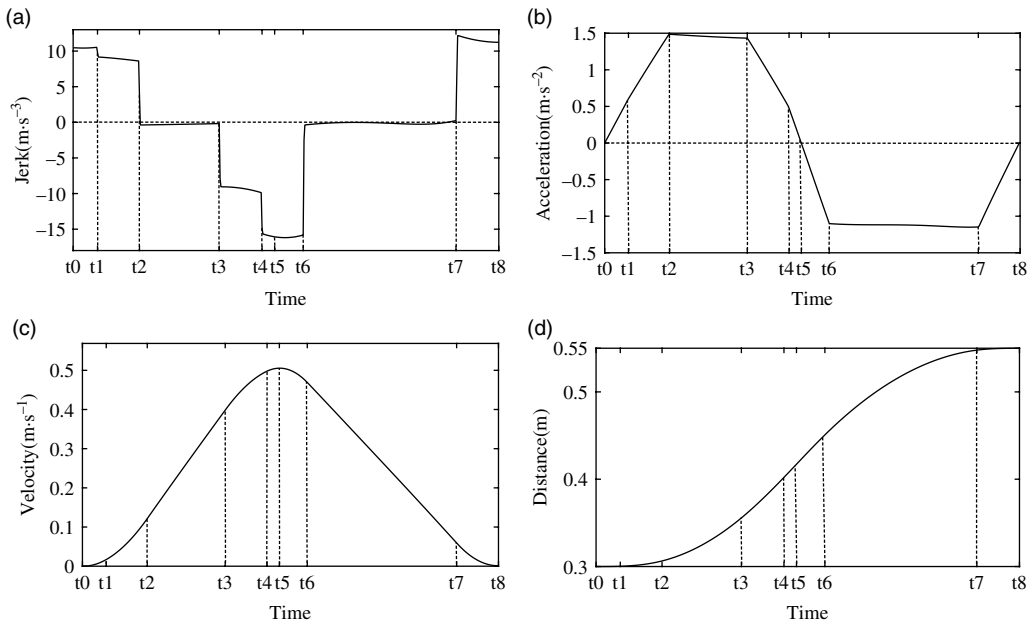

Fig. 7. Jerk, acceleration, velocity, and distance profiles. (a) Jerk profile; (b) acceleration profile; (c) velocity profile; (d) distance profile.

of the object can still be clearly divided into six stages, similar to the traditional six-stage acceleration/deceleration algorithm, in the process of motion. On this basis, the algorithm in this paper is proposed.

3.3. Time discretization

Let the duration of each stage of the six-stage acceleration/deceleration algorithm be T_1, T_2, T_3, T_4, T_5, T_6, then the total time duration is:

$$T_t = \sum_{i=1}^{6} T_i \tag{23}$$

The time is evenly divided into N segments on $t \in (0, T_t)$, while each segment is T_t/N. There are $N+1$ knot points u_k $(k = 0, 1, 2, \ldots, N)$, while the acceleration, velocity, and position of each point are a_k, v_k, l_k, respectively. The jerk, acceleration, velocity, and position, in the time from $(i-1)$ T_t/N to iT_t/N $(i = 1, 2, \ldots, N)$, are J_i, $a_i(t)$, $v_i(t)$, and $l_i(t)$, as shown in Eqs. (24), (25), and (26), respectively.

$$a_i(t) = a_{i-1} + \int_0^t J_i dt = a_{i-1} + J_i t, \ \ t \in \left(0, \frac{T_t}{N}\right] \tag{24}$$

$$v_i(t) = v_{i-1} + \int_0^t a_i(t) dt = v_{i-1} + a_{i-1} t + \frac{1}{2} J_i t^2, \ \ t \in \left(0, \frac{T_t}{N}\right] \tag{25}$$

$$l_i(t) = l_{i-1} + \int_0^t v_i(t) dt = l_{i-1} + v_{i-1} t + \frac{1}{2} a_{i-1} t^2 + \frac{1}{6} J_i t^3, \ \ t \in \left(0, \frac{T_t}{N}\right] \tag{26}$$

The acceleration, velocity, and position of the object, at each knot point, can be obtained by substituting $t = T_t/N$ into Eqs. (24), (25), and (26), as follows:

$$a_k = a_{k-1} + J_k \frac{T_t}{N} \tag{27}$$

$$v_i(t) = v_{k-1} + a_{i-1} \frac{T_t}{N} + \frac{1}{2} J_k \left(\frac{T_t}{N}\right)^2 \tag{28}$$

$$l_i(t) = l_{k-1} + v_{k-1} \frac{T_t}{N} + \frac{1}{2} a_{k-1} \left(\frac{T_t}{N}\right)^2 + \frac{1}{6} J_k \left(\frac{T_t}{N}\right)^3 \tag{29}$$

https://doi.org/10.1017/S0263574719001863 Published online by Cambridge University Press

The acceleration, velocity, and position of the object at each knot point, expressed by initial state, can be obtained by successive iterations. So the acceleration, velocity, and position of the object at the end of time can be obtained as shown in Eqs. (30), (31), and (32), respectively.

$$a_N = a_0 + \sum_{i=1}^{N} J_i \frac{T_t}{N} = a_d \tag{30}$$

$$v_N = v_0 + a_0 T_t + \sum_{i=0}^{N-1} \left(\frac{1}{2} + i\right) J_{N-i} \left(\frac{T_t}{N}\right)^2 = v_d \tag{31}$$

$$l_N = l_0 + v_0 T_t + \frac{1}{2} a_0 T_t^2 + \sum_{i=0}^{N-1} \left(\frac{1}{6} + \frac{i+i^2}{2}\right) J_{N-i} \left(\frac{T_t}{N}\right)^3 = l_d \tag{32}$$

At the same time, Eqs. (23), (30), (31), and (32), $J_1, J_2, ..., J_N, T_1, T_2, ..., T_6, T_t$ are variables, while the number of variables is $N + 7$. $T_t, J_{N-2}, J_{N-1}, J_N$ can be expressed by $J_1, J_2, ..., J_{N-3}, T_1, T_2, ..., T_6$, if $J_1, J_2, ..., J_{N-3}, T_1, T_2, ..., T_6$ are regarded as known quantities.

$$\begin{aligned} T_t &= \sum_{i=1}^{6} T_i \\ J_{N-2} &= \Sigma - \Psi + \frac{1}{3}\Lambda \\ J_{N-1} &= -2\Sigma + 3\Psi - \frac{7}{6}\Lambda \\ J_N &= \Sigma - 2\Psi + \frac{11}{6}\Lambda \end{aligned}$$

where

$$\Sigma = \left(l_d - l_0 - v_0 T_t - \frac{1}{2} a_0 T_t^2\right) \left(\frac{N}{T_t}\right)^3 - \sum_{i=3}^{N-1} \left(\frac{1}{6} + \frac{i+i^2}{2}\right) J_{N-i}$$

$$\Psi = (v_d - v_0 - a_0 T_t) \left(\frac{N}{T_t}\right)^2 - \sum_{i=3}^{N-1} \left(\frac{1}{2} + i\right) J_{N-i}$$

$$\Lambda = (a_0 - a_d) \frac{N}{T_t} - \sum_{i=1}^{N-3} J_i$$

Therefore, given $J_1, J_2, ..., J_{N-3}, T_1, T_2, ..., T_6$ is a set of solutions satisfying the equations, while the number of solutions is infinite. It is necessary to find the time-optimal solution, under the given constraints.

The optimal problem is discretized, under dynamic constraints, into the following static optimal problem:

$$\begin{aligned} &\text{Min} \sum_{i=1}^{6} T_i \\ s.t. \quad &a_0 = v_0 = a_N = v_N = 0;\ l_0 = y_0;\ l_N = y_d; \\ &a_k \leqslant a_{kmax}(l_k, v_k, a_{BICmax}, a_{TRImax});\ a_k \geqslant a_{kmin}(l_k, v_k, a_{BICmin}, a_{TRImin}); \\ &|J_k| \leqslant J_{max};\ |a_k| \leqslant a_{glove};\ k = 1, 2, ..., N. \end{aligned}$$

where a_{glove} is the maximum acceleration of the object, allowed by the spacesuit glove, so as not to be torn, a_{kmax} and a_{kmin} are the maximum and the minimum allowable acceleration of the object, respectively, determined by the torque of the elbow joint, according to Eq. (19). a_d is the acceleration of the object, at the target position, v_d is the velocity of the object, at the target position. The values of a_{BIC} and a_{TRI} are shown as follows: $a_{BICmax} = 0,\ a_{TRImax} = 1,\ a_{BICmin} = 1,\ a_{BICmin} = 0$.

https://doi.org/10.1017/S0263574719001863 Published online by Cambridge University Press

3.4. Parameter optimization based on PSO

For the optimization of the variables $J_1, J_2, ..., J_{N-3}, T_1, T_2, ..., T_6$, the improved PSO algorithm is adopted in this paper. The above analysis shows that the dimension of the search space is $N+3$ in solving this problem. The first six dimensions of the particle represent the time of six stages, while the last $N-3$ dimensions represent the jerk of the object, in the first $N-3$ segments. The position constraint of the search space is set as follows.

In the PSO algorithm, for solving this problem, the range of the last $N-3$ dimensions, representing the object jerk, is determined by the first six dimensions representing the time of each stage. In the optimization process of the PSO algorithm, the range of the last $N-3$ dimensions is dynamically adjusted. Since $T_1, T_2, ..., T_6$ may not be an integer multiple of T_t/N, each stage may not contain integer discrete time T_t/N, so $T_1, T_2, ..., T_6$ need to be rounded. The rounding process is as follows:

$$N_i = \text{round}\left(\frac{T_i}{T_t}N\right),\ i = 1, 2, 3, 4, 5, 6 \tag{33}$$

$$N_5 = N - \sum_{\substack{i=1 \\ i\neq 5}}^{6} N_i \tag{34}$$

where round(x) is a rounding function.

Because rounding function is used, the remainder would continuously accumulate up to a certain stage. In this paper, the remainder is accumulated in the fifth stage, because it takes a long time and the value of jerk is low in this stage. Even if there is a deviation, it will not have a great impact on the final result.

In summary, the PSO is carried out in the $(N+3)$-dimensional search space to solve this problem. The swarm is composed of m particles, $Swarm = \left\{x_1^{(k)}, x_2^{(k)}, ..., x_m^{(k)}\right\}$. The position vector of the particle i, in the search space, at time k:$x_i^{(k)} = \left\{x_{i1}^{(k)}, x_{i2}^{(k)}, ..., x_{id}^{(k)}, x_{i(N+3)}^{(k)}\right\}$, $i = 1, 2, ..., m$. The first six dimensions $x_{i1}^{(k)} - x_{i6}^{(k)}$ represent the time of six stages, ranging from 0 to T_{max}, while the last $N-3$ dimensions $x_{i7}^{(k)} - x_{i(N+3)}^{(k)}$ represent the jerk of the object in the first $N-3$ segments, ranging from J_{min} to J_{max}. The values of J_{min} and J_{max} are shown as follows:

$$J_{min} = \begin{cases} J_{mins1}, & 6 < d \leqslant 6 + \sum_{i=1}^{1} N_i \\ J_{minsj}, & 6 + \sum_{i=1}^{j-1} N_i < d \leqslant 6 + \sum_{i=1}^{j} \\ J_{mins6}, & 6 + \sum_{i=1}^{5} < d \leqslant N + 3 \end{cases}$$

$$J_{max} = \begin{cases} J_{maxs1}, & 6 < d \leqslant 6 + \sum_{i=1}^{1} N_i \\ J_{maxsj}, & 6 + \sum_{i=1}^{j-1} N_i < d \leqslant 6 + \sum_{i=1}^{j} \\ J_{maxs6}, & 6 + \sum_{i=1}^{5} < d \leqslant N + 3 \end{cases}$$

where $j = 2, 3, 4, 5$, N_i is the number of segments in each stage according to Eqs. (34) and (34).

If the updated position vector exceeds the search space, the nearest neighborhood is used to ensure that the jerk of each stage is within the range of values. The nearest method is defined as:

If $x_i^{(k)} \notin S$(S is the search space), reset $x_i^{(k)} = x'$, where x' is satisfied that $\forall x \in S,\ dist\left(x_i^{(k)}, x'\right) \leqslant dist\left(x_i^{(k)}, x\right)$, that is, x' is the closest position to the particle $x_i^{(k)}$, in the search space.

3.5. Higher order constraints processing

In constrained optimization problems, it is very important to deal with constrained conditions. The constraints in this paper are divided into equality constraint, inequality constraint, and upper/lower boundary constraint. The equality constraint can be expressed by Eqs. (30)–(31). Regarding inequality constraints, the current constraints processing techniques can be divided into the following categories: penalty function method,[7,17] multi-object method,[18] and other methods.[19–22]

https://doi.org/10.1017/S0263574719001863 Published online by Cambridge University Press

There are two kinds of inequality constraints in this paper. The first type is high-order constraints like J_N, while the second one is the acceleration of the object, at each knot point a_k. In this paper, the penalty function method is used to deal with the constraints, while the fitness function $f(X)$, with penalty terms, is reconstructed by adding penalty terms to the objective function, as follows:

$$f(X) = \sum_{i=1}^{6} T_i + C_1(X) + C_2(X)$$

where $C_1(X)$, $C_2(X)$ are the penalty terms who do not satisfy the first and the second kind of constraints, respectively. Their definitions are as follows: $C_1(X) = \sum_{k=N-2}^{N} c_{1k}$, where c_{1k} is the penalty function for J_{N-2}, J_{N-1}, J_N as follows:

$$c_{1k} = \begin{cases} \exp[p_1(|J_i| - |J_{max}|)], & iter \leqslant 0.3ger \\ \inf, & |J_i| > J_{max},\ iter > 0.3ger \\ 0, & \text{else} \end{cases}$$

where *iter* is the current iterative order, *ger* is the total iteration number, and p_1 is the penalty factor.

$$C_2(X) = \sum_{k=1}^{N} c_{2k}$$

where c_{2k} is the penalty function for acceleration, at each knot point a_k, as follows:

$$c_{2k} = \begin{cases} \exp[p_2(a_k - a_{kmax})], & a_k > a_{kmax},\ iter \leqslant 0.3ger \\ \exp[p_2(a_{kmin} - a_k)], & a_k < a_{kmin},\ iter \leqslant 0.3ger \\ \inf, & a_k > a_{kmax} | a_k < a_{kmin},\ iter > 0.3ger \\ 0, & \text{else} \end{cases}$$

The penalty function, constructed in this paper, has the following characteristics: in the initial stages of evolution, there are few feasible solutions that meet the constraints, while most of them exhibit low fitness. At this time, for the infeasible solution, the approach of increasing the penalty term is adopted. If the reconfigurable fitness of the infeasible solution is still lower than the fitness of the feasible solution, the infeasible solution is retained. In this process, since the construction of the penalty term adopts the exponential function, even if the infeasible solution does not meet the constraint, it is close to the feasible region boundary. In the subsequent evolution process, it is easy to enter the feasible region and become a feasible solution for the reserved infeasible solution. At the end of evolution, almost all particles enter the feasible region. At this time, the fitness of the feasible solution must be lower than that of the infeasible solution. The penalty term of the fitness function fails and the algorithm gradually converges to the optimal solution.

The flowchart of the PSO algorithm, designed in this paper, is shown in Fig. 8.

4. Simulation and Comparisons

Simulations are conducted with Matlab R2018 to verify the algorithm. The simulation platform is an Intel 3.7 GHz computer with 16 GB RAM. The simulation parameters are listed in Table III. The range of jerk in each stage is shown in Table IV, while the maximum acceleration of the object, allowed by the spacesuit glove not being torn, a_{glove} is 2m/s^2.

The jerk, acceleration, and velocity of the object, the distance from the target object to the coronal plane, as obtained by the algorithm are shown in Fig. 9.

The acceleration and its limits, in this process, are plotted in the same graph, as shown in Fig. 10, to verify the effect of the constraint. It is evident that the actual acceleration can be closer to the acceleration limits.

To establish a "virtual person" system, it is necessary to obtain the value of the muscle activation in the process. In order to observe the changes in muscle activation, during this process, the muscle activation of the biceps and triceps is obtained, as shown in Fig. 11(a) and the muscle forces of the biceps and triceps are shown in Fig. 11(b). During the whole motion, the muscle force is stable without sudden changes.

https://doi.org/10.1017/S0263574719001863 Published online by Cambridge University Press

Table III. Simulation parameters

Object mass, M	200 kg
Initial position of the object, y_0	0.3 m
Target position of the object, y_d	0.55 m
Total iteration number, ger	100
Number of segments, N	50
Penalty factor, p_1	0.05
Penalty factor, p_2	1

Table IV. The range of jerk in each stage

	s_1	s_2	s_3	s_4	s_5	s_6
J_{mins*}	3	−0.5	−5	−5	−0.5	3
J_{maxs*}	5	0.2	−3	−3	0.2	5

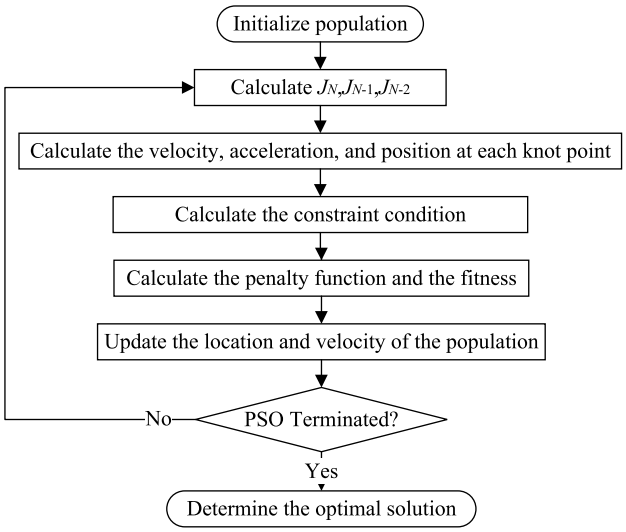

Fig. 8. The flowchart of the PSO algorithm.

The jerk, acceleration, velocity, and distance obtained by applying the standard PSO algorithm to the discrete static optimal problem are shown in Fig. 12. Comparing Figs. 12 to 11 shows that, although the standard PSO algorithm can also realize the optimization of discrete static optimal problems, the jerk profile has many sudden change points. The acceleration cannot increase rapidly, so the final planning time is 1.6 s longer than that of the algorithm presented in this paper.

The algorithm, presented in ref. [7], is adopted and modified to become suitable for the problem applied in this paper. The constraints of acceleration are modified according to those in this paper. The actual acceleration and acceleration limits are plotted in the same graph, as shown in Fig. 13.

Comparing Figs. 10, 11, 12, and 13, shows that the algorithm in this paper can easily reach the boundary of acceleration limit, resulting in shorter planning time. Compared to the algorithm in ref. [7], the planning time of this algorithm is reduced by 6.5%.

In order to verify the stability of the algorithm, presented in this paper, 20 simulation tests were performed, with $N = 30$, 40, 50, and 60, respectively. The average CPU computing time, the average planning time, the standard deviation of planning time, and the shortest planning time are listed in Table V, compared to the standard PSO algorithm and the algorithm in ref. [7].

In the case of $N = 50$, 20 simulation tests were performed, using the algorithm in this paper and in ref. [7], with different mass of the object. The results are shown in Table VI.

Table V shows that, when the number of segments is larger, meaning that it is the closer to the real continuous curve, the higher the feasibility of planning a shorter planning time is more feasible. Compared to the case of $N = 30$, the average planning time and the shortest planning time are

https://doi.org/10.1017/S0263574719001863 Published online by Cambridge University Press

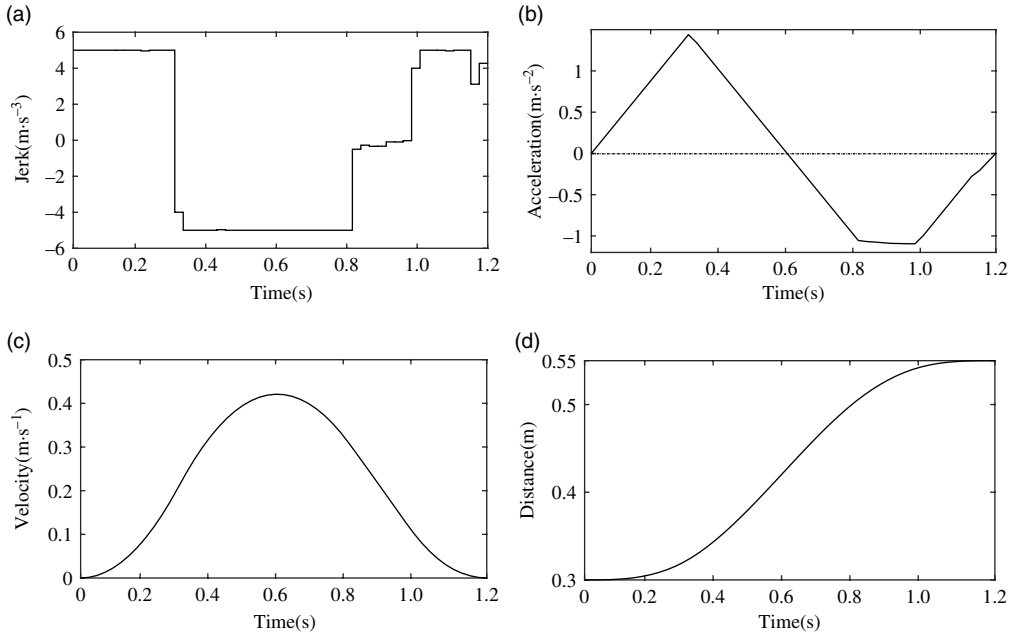

Fig. 9. Planning results of the algorithm: (a) jerk profile; (b) acceleration profile; (c) velocity profile; (d) distance profile.

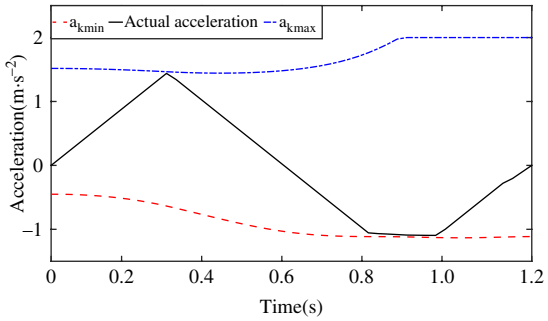

Fig. 10. Acceleration constraints: The blue dotted line and the red dashed line show the minimum acceleration constraint and maximum acceleration constraint, respectively, while the black solid line shows the actual acceleration.

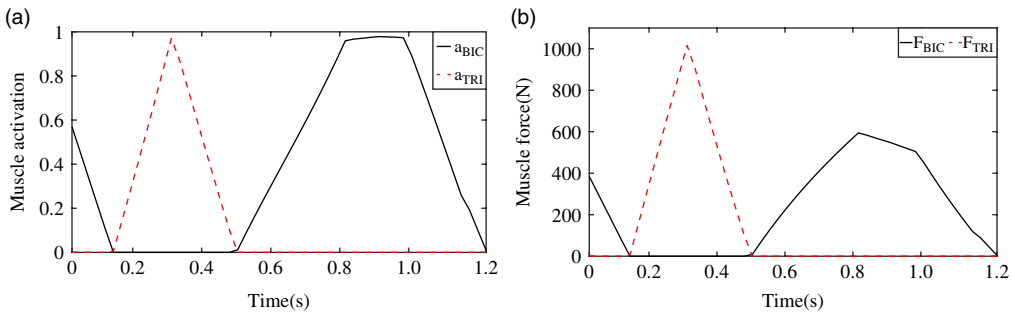

Fig. 11. Muscle activation and muscle force in the process. (a) The black solid line and the red dashed line show the muscle activation of biceps and triceps, respectively. (b) The black solid line and the red dashed line show the muscle forces of biceps and triceps, respectively.

reduced by 1.6% and 1.8%, respectively, in the case of $N = 60$. However, if the number of segments is high, it is easier for the PSO algorithm to converge to the local optimal solution, so the algorithm is unstable with a larger standard deviation. There is no feasible solution to be found, because of

https://doi.org/10.1017/S0263574719001863 Published online by Cambridge University Press

Table V. Algorithm stability verification and comparisons

Algorithm type	Number of segments	Average CPU computing time	Average planning time	Standard deviation of planning time	The shortest planning time
Algorithm in this paper	$N = 30$	3.66	1.2328	0.0038	1.2205
	$N = 40$	4.76	1.2202	0.0040	1.2083
	$N = 50$	5.95	1.2106	0.0047	1.1994
	$N = 60$	7.11	1.2109	0.0096	1.1988
Standard PSO	$N = 40$	3.93	2.8062	0.4368	2.0718
Algorithm in [7]	$N = 5$	2.09	1.2830	0.0007	1.2826

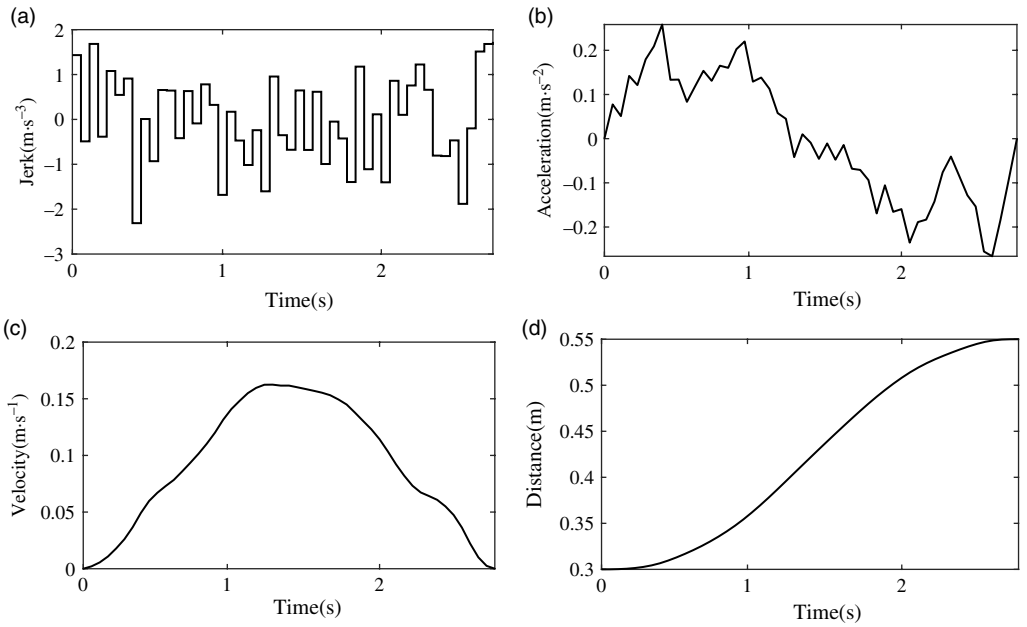

Fig. 12. Planning results of the standard algorithm. (a) jerk profile; (b) acceleration profile; (c) velocity profile; (d) distance profile.

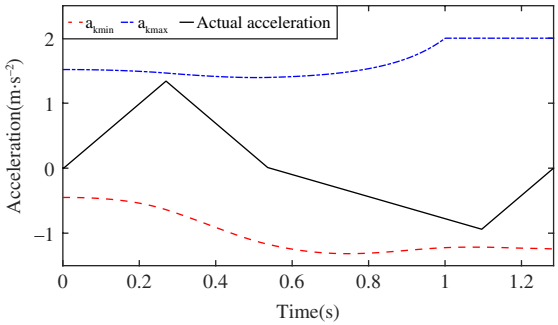

Fig. 13. Acceleration constraints.

too many constraints, which leads to the failure of solving. In the process of experiment, it is found that the result of the algorithm will eventually converge to an infeasible solution under the current parameters with $N = 60$. When the number of segments tends to infinity, the final planning time is the shortest, but the probability of converging to the feasible solution is close to zero, at the longest CPU computing time. Therefore, the number of segments should be selected, keeping some balance in mind. It can be known, from repeated tests, that $N = 40$ to 50 is a reasonable parameter range. The trajectory is divided into only five segments in ref. [7], so the algorithm stability is higher, but longer

https://doi.org/10.1017/S0263574719001863 Published online by Cambridge University Press

Table VI. Planning results with different object mass

M	Algorithm type	Average planning time	Standard deviation of planning time	The shortest planning time
50	Algorithm in this paper	1.1402	0.0059	1.1250
	Algorithm in ref. [7]	1.1787	0.0020	1.1777
100	Algorithm in this paper	1.1416	0.0055	1.1251
	Algorithm in ref. [7]	1.2241	0.0002	1.2241
150	Algorithm in this paper	1.1412	0.0087	1.1251
	Algorithm in ref. [7]	1.2577	0.0014	1.2577
200	Algorithm in this paper	1.2106	0.0047	1.1994
	Algorithm in ref. [7]	1.2830	0.0007	1.2826
250	Algorithm in this paper	1.2308	0.0086	1.2205
	Algorithm in ref. [7]	1.3214	0.0009	1.3203
300	Algorithm in this paper	1.2509	0.0124	1.2693
	Algorithm in ref. [7]	1.3836	0.0033	1.3801

planning time is required. Especially when the planning distance is long or the mass of the object is large, the planning time of this algorithm is significantly different from that of the algorithm in this paper. The presented application, in this paper, does not require the algorithm to be real-time, while the optimal solution can be obtained, after repeatedly running the algorithm.

Table VI shows that, when the mass of the object is small, the results, obtained by the algorithm in this paper, differ less from those of the algorithm in ref. [7]. The average planning time and the shortest planning time with $M = 50$ kg are reduced by 3.8% and 3.6%, respectively, compared to the algorithm in ref. [7], but by 6.7% and 8.0%, with $M = 300$ kg. When the mass is small, the maximum acceleration of the object, determined by the joint torque, is high, while the acceleration constraint depends on the maximum acceleration of the object, allowed by the spacesuit glove not being torn. The algorithm is degraded to the velocity planning algorithm, under static constraints. There is no uniform acceleration stage, because the acceleration is difficult to reach the maximum allowable value at a short planning distance. The planning result of the algorithm, in this paper, is similar to the five-stage model. When the mass of the object is large, the results obtained by the algorithm, in this paper, are quite different from those presented in ref. [7]. The reason is that the maximum allowable acceleration of the object, determined by the joint torque, is low, while the acceleration constraints determined by the joint torque are dynamic. In addition, the acceleration of the object can reach the maximum allowable value. Therefore, the planning result of the presented algorithm, in this paper, is similar to the six-stage model.

5. Conclusion

The motion law of object, carried by astronauts wearing spacesuits, is analyzed, in this paper, from the perspective of biomechanics. Based on the analysis, the improved PSO algorithm is used to optimize the jerk of each segment, by discretizing the time-continuous model, while the near time-optimal velocity planning of mass-spring system is realized, with dynamic acceleration constraints. The optimal number of segments is determined based on repeated simulation. Simulation results show that the planning time of this algorithm is shorter than the planning time of other algorithm, in the case of large objects. The algorithm of this paper exhibits practical application feasibility.

Further research includes:

1. At present, only one-dimensional linear motion of the carried object is studied. In actual conditions, the object has not only a translational motion in one direction but also a rotational motion around the coordinate axis. A next research issue is to analyze the more complex process, involving rotational motion.
2. In the operation process, the time is not the only factor to be considered. The energy consumed by astronauts, in the process of operation, should also be taken into account. Another research

https://doi.org/10.1017/S0263574719001863 Published online by Cambridge University Press

point is to consider both operation time and energy consumed, in a multi-objective optimization process.
3. There are still improvement prospects, regarding the poor stability of the PSO algorithm.
4. The parameters of the current muscle model are generic and may not be applicable to every astronaut. In future research, the algorithm will continuously adjust the planning results of the generic algorithm, according to the sEMG, during the training process, so as to provide individual time-optimal velocity planning.

Acknowledgements

This work was supported by the National Natural Science Foundation of China, under Grant 61773007.

References

1. D. Song, L. X. Zhang and B. J. Wang, "The control strategy of flexible cable driven force interactive robot," *Robot* **40**(2), 440–447 (2018).
2. D. Song, L. X. Zhang and F. Xue, "Configuration optimization and a tension distribution algorithm for cable-driven parallel robots," *IEEE Access* **6**, 33928–33940 (2018).
3. J. Z. Han and W. H. Chen, "Velocity control algorithm in glass polishing based on the cubic NURBS curve," *Proc. Inst. Mech. Eng. Part C-J. Mech. Eng. Sci.* **232**(4), 685–696 (2018).
4. M. S. Tsai, S. K. Wu and H. W. Huang, "Study on Acceleration Deceleration Feedrate Planning for Multi-Block Line Segments Using Estimated Contour Error Formulation," **In**: *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics* (2014) pp. 489–493.
5. H. I. Lin, "A fast and unified method to find a minimum-Jerk Robot joint trajectory using particle swarm optimization," *J. Intel. & Robot. Syst.* **75**, 379–392 (2014).
6. T. C. Lu and S. L. Chen, "Genetic Algorithm-Based S-curve Acceleration and Deceleration for Five-Axis Machine Tools," *Int. J. Adv. Manuf. Technol.* **87**(1), 219–232 (2016).
7. T. C. Lu, S. L. Chen and E. C. Yang, "Near time-optimal S-curve velocity planning for multiple line segments under axis constraints," *IEEE Trans. Ind. Electron.* **65**(12), 9582–9592 (2018).
8. F. S. Liang, J. Zhao and S. J. Ji, "An iterative feed rate scheduling method with confined high-order constraints in parametric interpolation," *Int. J. Adv. Manuf. Technol.* **92**(5), 2001–2015 (2017).
9. M. X. Yuan, Z. Chen, B. Yao and X. C. Zhu, "Time optimal contouring control of industrial biaxial gantry: A highly efficient analytical solution of trajectory planning," *IEEE-ASME Trans. Mechatron.* **22**(1), 247–257 (2017).
10. A. Bharathi and J. Y. Dong, "Feedrate Optimization for Smooth Minimum-Time Trajectory Generation with Higher Order Constraints," *Int. J. Adv. Manuf. Technol.* **82**, 1029–1040 (2016).
11. C. X. Fang, H. Zhang and P. Q. Ye, "Convex optimization approach in time-optimal feed-rate planning for CNC," *Comput. Integr. Manuf. Syst.* **21**(1), 187–194 (2015).
12. F. E. Zajac, "Muscle and tendon: Properties, models, scaling, and application to biomechanics and motor control," *Crit. Rev. Biomed. Eng.* **17**(4), 359–411 (1989).
13. M. Murray and D. Wendy, "Variation of muscle moment arms with elbow and forearm position," *J. Biomech.* **28**(5), 513–525 (1995).
14. A. Storace and B. Wolf, "Functional analysis of the role of the finger tendons," *J. Biomech.* **12**(8), 575–578 (1979).
15. J. D. Zhao, Y. W. Liu, H. G. Cai and H. Liu, "An Improved Algorithm of Measuring Extravehicular Mobility Unit (EMU) Spacesuit Joint Damping Parameters for the Old Passive Robot System," **In**: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (2008) pp. 77–82.
16. Katherine R. S. Holzbaur, W. M. Murray and S. L. Delp, "A model of the upper extremity for simulating Musculoskeletal surgery and analyzing neuromuscular control," *Ann. Biomed. Eng.* **33**(6), 829–840 (2005).
17. H. Garg, "A hybrid PSO-GA algorithm for constrained optimization problems," *Appl. Math. Comput.* **274**, 292–305 (2016).
18. Q. Ding, H. Chen, C. Wang, A. P. Jiang and W. W. Lin, "Research of PSO Algorithm with Variable Constraints in Process System," **In**: *Proceedings of the 10th World Congress on Intelligent Control and Automation* (2012) pp. 544–548.
19. G. Gao, C. Sun, J. Zeng and S. D. Xue, "A Constraint Approximation Assisted PSO for Computationally Expensive Constrained Problems," **In**: *Proceeding of the 11th World Congress on Intelligent Control and Automation* (2014) pp. 1354–1359.
20. M. Kohler, L. Forero, M. Vellasco, R. Tanscheit and M. A. Pacheco, "PSO Plus: A Nonlinear Constraints-Handling Particle Swarm Optimization," **In**: *Proceeding of the 2016 IEEE Congress on Evolutionary Computation* (2016) pp. 2518–2523.
21. K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm NSGA-II," *IEEE Trans. Evol. Comput.* **6**(2), 182–197 (2002).
22. N. Gionfra, G. Sandou, H. Siguerdidjane, D. Faille and P. Loevenbruck, "Wind farm distributed PSO-based control for constrained power generation maximization," *Renew. Energy* **133**, 103–117 (2019).