Robotica: (2020) volume 38, pp. 719–731. © Cambridge University Press 2019
doi:10.1017/S0263574719001000

Research on Attitude Interpolation and Tracking Control Based on Improved Orientation Vector SLERP Method

Mingjie Dong†, Guodong Yao‡, Jianfeng Li†* and Leiyu Zhang†

†*College of Mechanical Engineering and Applied Electronics Technology, Beijing University of Technology, Beijing, People's Republic of China. E-mails: dongmj@bjut.edu.cn; zhangleiyu@bjut.edu.cn*
‡*Institute of Electrical Engineering, Chinese Academy of Sciences, Beijing, People's Republic of China. E-mail: jellyyao@126.com*

(Accepted May 25, 2019. First published online: July 2, 2019)

SUMMARY
In order to make the end of the three-axis platform follow the control command and achieve stable control of the end attitude, an improved orientation vector spherical linear interpolation (SLERP) method is proposed for the requirements, which specifically handles the position of the gimbal lock, so that the platform can move smoothly around the gimbal lock position. A three-axis platform with a camera at the end is set up for the validity of the proposed algorithm. At first, an adaptive speed measurement method based on incremental encoder is introduced, which can automatically adapt to high and low speed, and estimate the ultra-low speed to realize the speed measurement of large dynamic range, and this is used for the motion control of the three-axis platform. Then, the SLERP method for the quaternion interpolation on the starting and ending attitudes represented in quaternion is introduced in detail, and it is continuously improved in response to its existing problems for the platform. Finally, an orientation vector SLERP method is proposed, which uses viscosity factor and rejection factor to adjust the algorithm near the platform's gimbal lock position. A tracking experiment was designed using the red ball as the following target detected by the designed target tracking algorithm using the camera, which verified the effectiveness of the attitude tracking control based on the proposed improved orientation vector SLERP.

KEYWORDS: Improved SLERP; Gimbal lock; Adaptive speed measurement; Motion control; Attitude tracking.

1. Introduction

Many robotics applications require smooth orientation planning, that is, interpolation or approximation of a frame orientation through prescribed configurations such that the angular velocity and its time derivatives are smooth.[1] Nowadays, the demand for rehabilitation robots for people with neurological disorders is rapidly increasing with an aging population,[2] followed by the high demand for their orientation path planning and motion control. The performance of robots has been improved along with the need of all kinds of applications of rehabilitation robots[3,4] that generally have multiple axes of rotation. And the smoothness of orientation movement and the precision of interpolation control are really important.[5] Meanwhile, the motion control of

* Corresponding author. E-mail: lijianfeng@bjut.edu.cn

the robots needs to simultaneously generate the trajectory of the position and the attitude of the end operator.[6]

Many researchers have studied the trajectory generation and interpolation problems of robots. A trajectory generation based on B-spline has been enhanced with a RC-type (Repetitive Control) mechanism that modifies the control points defining the spline in order to nullify the tracking error at the desired points by considering cyclic tasks, which are quite common in the industrial and robotics field.[7] Reference [8] incorporated the sensor motion into the optimization by interpolating each acquired point cloud between its starting and ending position to realize the pose interpolation SLAM (Simultaneous Localization And Mapping) for large maps using moving 3D sensors by considering a continuous time trajectory which is defined as the linear interpolation of a discrete set of control poses in SE3.

However, the position interpolation methods are not fit for attitude interpolation of the robots for the reason that the attitude interpolation belongs to SO(3).

In general, Euler angles are employed for the orientation interpolation of robots because of its minimum representation of orientation with three angles.[9] However, the orientation interpolation using Euler angles can result in singular point (gimbal lock), which can cause the malfunction of robots with systematic errors and lead to undesirable results because of the nonlinearity of the three Euler angles.[10] Reference [11] clearly points out that Euler angle is not satisfactory because of the gimbal lock in attitude interpolation. The ranges of the three angles are usually limited to avoid the singular points which can result in limited area of the orientation movement of robot. Except for the interpolation of Euler angels, there have been interpolation methods of axis-angle, quaternion, and rotation matrices, and their advantages and disadvantages are discussed in refs. [12, 13].

Among them, the quaternion is widely used in end attitude planning of robots because of its ability to efficiently perform smooth interpolation and to effectively avoid gimbal locks. SHOEMAKE[14] extended the properties and expressions of Bezier curves to quaternion spaces and proposed the unit quaternion attitude interpolation algorithm, spherical linear interpolation (SLERP), and spherical and quadrangle (SQUAD). SLERP describes a path between two quaternions along the shortest connection with a constant angular velocity. Therefore, because of its simplicity, it is the standard for the path generation between two orientations and is widely used. However, ref. [10] proves that SLERP only has CO continuity, and SQUAD only has C1 continuity. Reference [15] proposed two auxiliary formulas in S^3 space to solve the control points accurately and obtained the C2 continuous attitude curve by interpolation. To avoid the singularity using Euler angles and the non-smooth movement using SLERP, Kong et al.[16] present a novel interpolation method based on unit quaternion which constructs the orientation movement through interpolation the angle of rotation and results in smooth and controllable velocity and acceleration of orientation movement combined with sine-jerk trajectory planning which could generate a C3 continuous angle curve. Jiang et al.[17] propose a point-to-point dynamic trajectory planning technique for reaching a series of poses with a 6-DOFs cable-suspended parallel robot, with each trajectory segment designed to have zero translational and rotational velocity at its endpoints, and SLERP is used to produce singularity-free, optimally interpolated rotational trajectory segments.

In this paper, we propose an improved orientation vector SLERP method, which specifically handles the position of the gimbal lock using two correction factors, so that the platform can move smoothly around the gimbal lock position. The platform we designed for the verification of the proposed algorithm is as in Fig. 1, with its schematic diagram of the mechanical structure shown in Fig. 2. It is a three-axis platform consisting of three attitude measurement units, three motion units driven by three DC motors, a microcomputer, and a camera mounted on the end of the platform, that is the third axis. The function to be implemented is to make the end of the three-axis platform follow the control command and achieve stable control of the end attitude.

The paper is organized as follows: Section 2 demonstrates the adaptive velocity and position measurement using incremental encoders. Section 3 presents the attitude coordinate transformation, orientation vector SLERP and its improvements in detail. Design of the target tracking algorithm and tracking algorithm based on the proposed interpolation algorithm are displayed and discussed in Section 4, while conclusion is drawn in Section 5.

https://doi.org/10.1017/S0263574719001000 Published online by Cambridge University Press

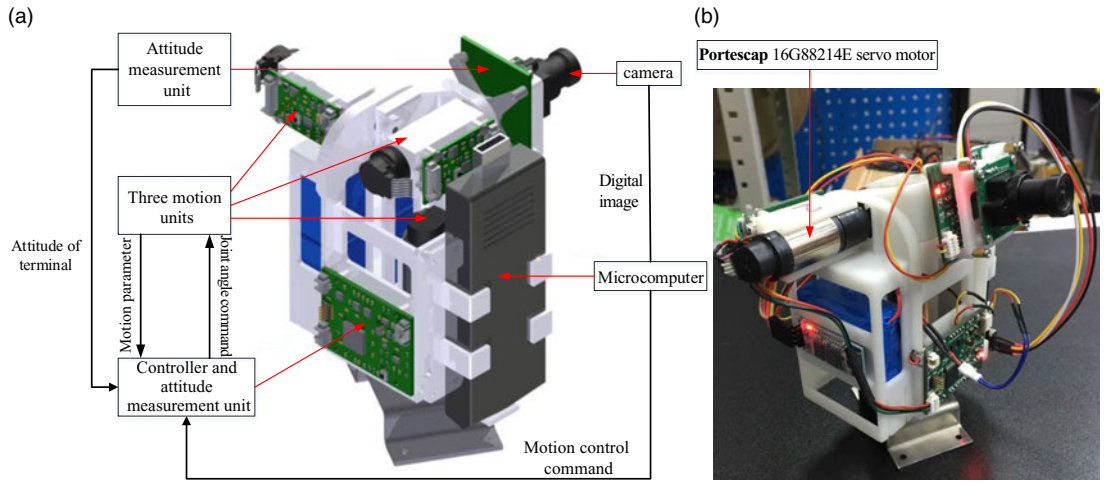

Fig. 1. Schematic diagram and prototype of the three-axis motion platform. (a) Schematic diagram. (b) Prototype of the system.

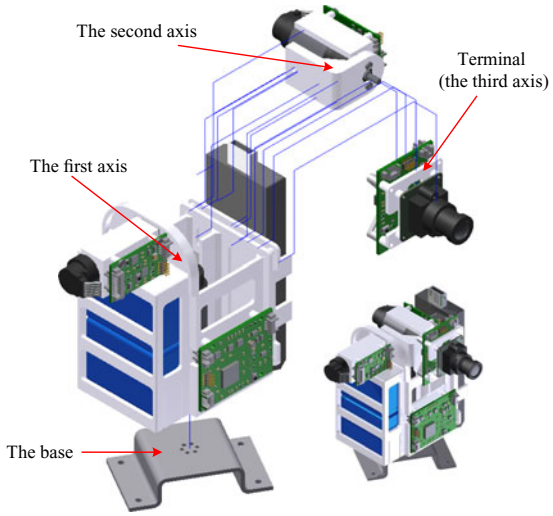

Fig. 2. Schematic diagram of the mechanical structure.

2. Adaptive Velocity and Position Measurement

Velocity and position measurements are the basis for target tracking control. For each rotation axis of the designed three-axis motion platform that is driven by a motor with an incremental encoder, we propose an adaptive velocity and position measurements method based on incremental encoders. The adaptive here mainly refers to the adaptation of speed measurement between low speed and high speed. Meanwhile, the positions can also be obtained along with the measurement of velocity.

For velocity measurement with incremental encoders, the most basic measurement strategy is to measure the period and frequency. The period measurement method is suitable for low-speed measurement while the method of measuring frequency is suitable for high-speed measurement. For many applications, the dynamic range of speed is large, and it is necessary to combine the two methods together in order to obtain good measurement accuracy no matter at high speed or low speed. The speed measurement method introduced here is an adaptive measurement method combining period measurement and pulse number measurement, and especially, the speed estimation algorithm at particularly low speed is introduced here.

The basic idea of the velocity measurement is to simultaneously measure the number of pulses, N, generated by the encoders and the exact time interval of the N pulses in one sampling window. As in Fig. 3, the velocity measurement and update are performed in every sampling window time during the velocity measurement. And the number of generated pulses N and the exact time of the pulse edge Δt

https://doi.org/10.1017/S0263574719001000 Published online by Cambridge University Press

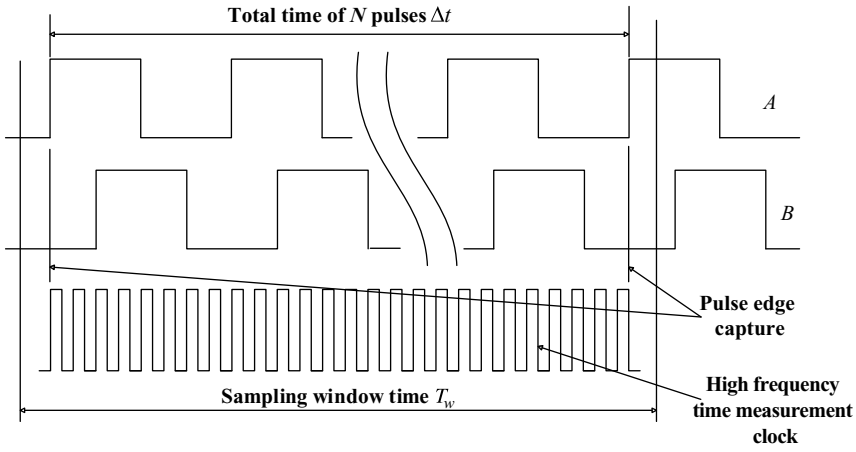

Fig. 3. Velocity measurement of encoders.

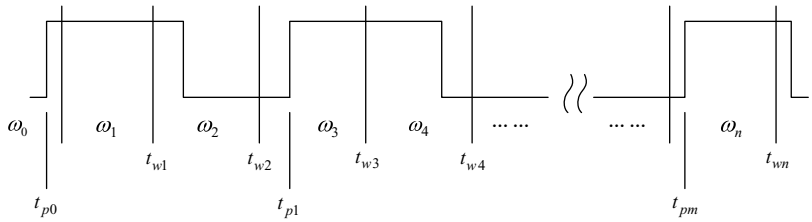

Fig. 4. Example of velocity sampling.

are simultaneously recorded, with accurate velocity measured by calculation of $\frac{N}{\Delta t}$. The improvement of this method compared to the frequency measurement method is that it measures the exact time of the pulse edge. And the time interval used in the velocity calculation is the real time of generating N pulses, instead of the sampling window time in frequency measurement method. However, when the speed is very low, such as when there is only one pulse in one sampling window time, this measurement method is changed into a period measurement method to maintain the measurement accuracy, while the velocity measured by the frequency method will jump between $\frac{0}{T_w}$, $\frac{1}{T_w}$ and $\frac{2}{T_w}$ with its great fluctuation of the velocity and its inaccuracy. Therefore, an adaptive velocity estimation algorithm is designed for velocity lower than one pulse per sampling period, so that the velocity measurement value can be updated for every sampling period.

As shown in Fig. 4, let us assume that the time at which the valid pulse edge is last captured before time 0 is t_{p0}, the value of the measured or estimated velocity in the last sample window is ω_0. Then the velocity estimation in the next sampling window can be divided into two cases. In the first case, the valid pulse edge is detected during this sampling period and the detected edge time is t_{p1}, then the value of velocity estimation in the current sampling period is $\omega_1 = \frac{N}{t_{p1}-t_{p0}}$, where, N is the total number of pulses from the edge of the time t_{p0} to the detected edge. The second case is that no valid pulse edge is detected during this sampling period. Assuming that the end time of the sampling window is t_{w1}, it can be inferred that the absolute value of the average velocity from t_{p0} to t_{w1} is less than $\frac{1}{t_{w1}-t_{p0}}$. And the velocity is used to compare with the absolute value of velocity $|\omega_0|$ in the previous sampling period. If the velocity is less than $|\omega_0|$, then it is used as the velocity estimation for the current sampling period with its sign same as the previous velocity estimate $\omega_1 = \frac{\omega_0}{|\omega_0|}\frac{1}{t_{w1}-t_{p0}}$, otherwise, $\omega_1 = \omega_0$.

The velocity estimation according to the above method does not need to consider when to switch the size of sampling window. The estimation result obtained at high velocity is $\frac{N}{\Delta t}$, while the velocity estimation at low velocity is equivalent to gradually elongate the sampling window and to make an upper bound estimation of the velocity when there is no valid pulse. According to this method, if the velocity has been reduced to zero, the actually estimated velocity is a small value but not zero.

https://doi.org/10.1017/S0263574719001000 Published online by Cambridge University Press

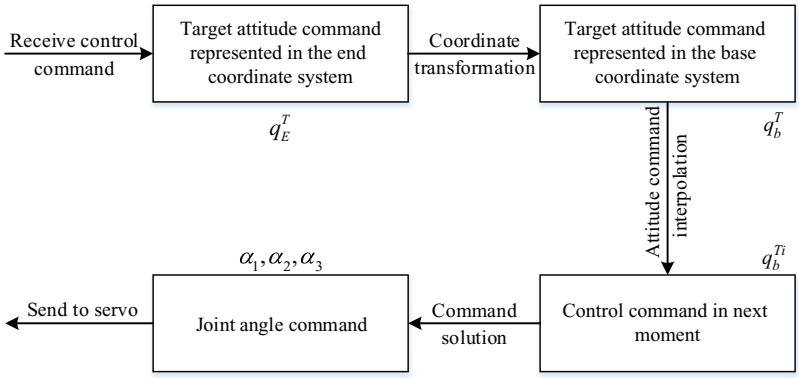

Fig. 5. Example of velocity sampling.

Therefore, a zero velocity threshold can be set, and the estimated velocity is directly set to zero when the velocity is less than the threshold.

Taking the case shown in Fig. 4 as an example, assuming that the velocity direction does not change and only the rising edge of the pulse is taken as a valid pulse edge. Then the velocity estimation after time 0 can be performed according to (1).

$$\begin{cases} \omega_1 = \begin{cases} \frac{1}{t_{w1}-t_{p0}} \left(\frac{1}{t_{w1}-t_{p0}} < \omega_0 \right) \\ \omega_0 \left(\frac{1}{t_{w1}-t_{p0}} \geq \omega_0 \right) \end{cases} & \omega_2 = \begin{cases} \frac{1}{t_{w2}-t_{p0}} \left(\frac{1}{t_{w2}-t_{p0}} < \omega_1 \right) \\ \omega_1 \left(\frac{1}{t_{w2}-t_{p0}} \geq \omega_1 \right) \end{cases} \\ \omega_3 = \frac{1}{t_{p1}-t_{p0}} & \omega_4 = \begin{cases} \frac{1}{t_{w4}-t_{p1}} \left(\frac{1}{t_{w4}-t_{p1}} < \omega_3 \right) \\ \omega_3 \left(\frac{1}{t_{w4}-t_{p1}} \geq \omega_3 \right) \end{cases} \end{cases} \tag{1}$$

3. Attitude Interpolation Algorithms

The control objective is to compensate the influence of the platform base's attitude change on the terminal attitude through the three-axis simultaneous motion control. If not specified, the "target attitude" here refers to the terminal attitude relative to the platform base when the attitude stability problem is not discussed. However, the attitude is relative to the local geographic coordinate system when discussing the attitude stability problem. Similarly, the "current attitude" also refers to the current attitude of the terminal.

3.1. Attitude coordinate transformation from end coordinate system to base coordinate system

Three-axis simultaneous motion control of the platform mainly includes three problems. The first one is to convert the control command represented in the end coordinate system into the control command represented in the base coordinate system of the platform, and the second is to implement the interpolation of the control command, while the third is to solve the control command represented by the quaternion after interpolation as the joint angle command, as shown in Fig. 5.

When the platform works, the controller receives the motion control command represented in the end coordinate system, which can be regarded as the terminal attitude in the current end coordinate system, and can also be considered as the attitude increment from the current end coordinate system. However, the attitude command interpolation and the joint angle command solution need to be expressed in the base coordinate system, so the coordinate system transformation of the command is required. Here, we define a function *sum*() to represent an equivalent rotation for a series of consecutive rotations, and it is used to calculate the representation of the target attitude in the base coordinate system. And its corresponding quaternion representation is $\mathbf{q_b^T} = \mathbf{q_b^E q_E^T}$, where, $\mathbf{q_b^T}$ is target attitude's quaternion representation in the base coordinate system, $\mathbf{q_b^E}$ is the attitude of the current terminal relative to the base, while $\mathbf{q_E^T}$ is the received attitude control command.

After the above calculations, the target attitude can be obtained. However, the target attitude and the current attitude may have a large difference, and the end attitude trajectory may be quite different

https://doi.org/10.1017/S0263574719001000 Published online by Cambridge University Press

from the expected one if the joint angle command is directly solved by the target attitude. Therefore, the attitude command needs to be interpolated to improve the control performance.

3.2. Attitude interpolation and joint angle solution

The desired control effect is to make the end of the platform reach the target attitude as quickly and directly as possible, and to immediately move from the current attitude to the new target attitude every time a new control command arrives with its previously unexecuted commands discarded directly. Therefore, the control algorithm performs a "point-to-point" motion interpolation from the current attitude directly to the target attitude every time a new control command is received.

For the interpolation of attitudes, the most direct method is to use the SLERP method for quaternion interpolation on the starting and ending attitudes represented in quaternion, as in (2).

$$Slerp\left(\mathbf{q_1}, \mathbf{q_2}, t\right) = \frac{\sin\left(\left(1-t\right)\alpha\right)}{\sin\alpha}\mathbf{q_1} + \frac{\sin\left(t\alpha\right)}{\sin\alpha}\mathbf{q_2}, \tag{2}$$

where $\mathbf{q_1}$ and $\mathbf{q_2}$ are the starting and ending attitudes, respectively, $\mathbf{q_1} = \mathbf{q_b^E}$ and $\mathbf{q_2} = \mathbf{q_b^T}$. t is the parameter for controlling the interpolation position, with its value $t = \frac{i}{upper\left(\frac{\alpha}{\alpha_{\max}}\right)}, \left(i = 1, 2, \cdots, upper\left(\frac{\alpha}{\alpha_{\max}}\right)\right)$, and the function $upper\left(\right)$ here makes its value round up to an integer, while α is the angle between the starting and ending quaternion vectors with $\cos\alpha = \mathbf{q_1} \cdot \mathbf{q_2}$, $\alpha_{\max}$ is the maximum angle of the quaternion vector corresponding to the allowed adjacent interpolation points.

After interpolating to obtain the attitude command and solving the joint angle command, the calculation can be directly performed according to the transformation relationship of the quaternion to the Euler angle, as shown in (3).

$$\begin{bmatrix}\alpha_3\\ \alpha_2\\ \alpha_1\end{bmatrix} = \begin{bmatrix} a\tan 2\left(ab+cd, \frac{1}{2}-\left(b^2+c^2\right)\right)\\ a\tan 2\left(ac-bd, \sqrt{\left(ab+cd\right)^2+\left(\frac{1}{2}-\left(b^2+c^2\right)\right)^2}\right)\\ a\tan 2\left(ad+bc, \frac{1}{2}-\left(c^2+d^2\right)\right)\end{bmatrix}, \tag{3}$$

where α_1, α_2, and α_3 are the angles of the first, second, and third axes of the platform correspond to the yaw angle, pitch angle, and roll angle of the Euler angles, respectively.

The attitude trajectory obtained by interpolation according to the above method directly interpolates the end attitude. However, a further analysis shows that this interpolation method has some problems in the attitude control of the platform. As we install the camera at the end of the platform with the camera's optical axis coinciding with the end axis. Considering such control requirements, we need to control the end of the platform to keep a specific target in the center of the camera's field of view (FOV). The orientation of the camera's optical axis determines the target's distance from the center of the FOV. And the angle of the camera around its optical axis determines the position of the target around the center of the FOV. The direction of the optical axis is only related to the yaw and pitch angles of the terminal, and the angle of the camera around the optical axis is just the roll angle of the terminal. Under such control requirements, the roll angle of the terminal is relatively independent of the yaw angle and pitch angle. When interpolating according to the method described above, the trajectory obtained by interpolation is affected by three Euler angles at the same time, that is, for the same initial attitude, the end attitude has only two interpolations with different roll angles, and the trajectories of the three obtained Euler angles are different, which is not what we expected, as shown in Fig. 6.

In addition, the left part of Fig. 6 draws a trajectory on the unit ball by a vector oriented along the end rotation axis, that is, a trajectory of the optical axis of the camera is drawn on the unit ball. As we can see that the trajectory obtained by interpolation has a large gap with the desired trajectory.

After the above analysis, the direct interpolation of the attitude represented by the quaternion is not applicable to our requirement, mainly because this method interpolates the end attitude as a whole. However, we hope to directly control the orientation of the camera's optical axis. To solve this problem, we separate the roll angle from the other two Euler angles to interpolate separately. For the reason that the yaw angle and pitch angle determine the orientation of the optical axis, we can

https://doi.org/10.1017/S0263574719001000 Published online by Cambridge University Press

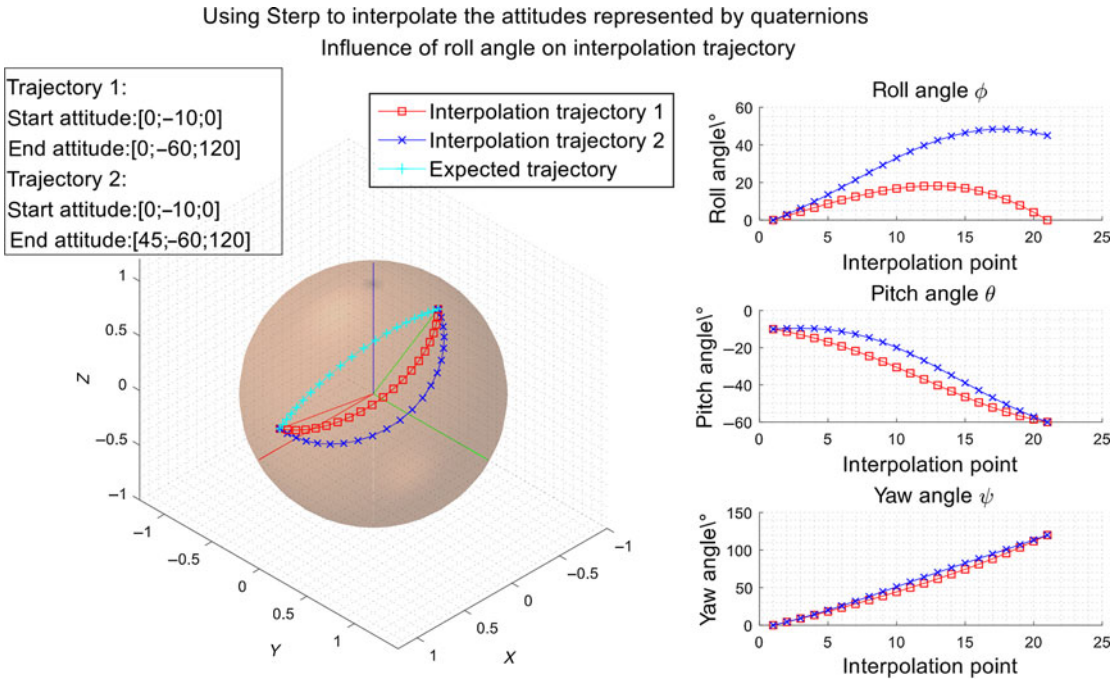

Fig. 6. Influence of roll angle on interpolation trajectory.

perform SLERP on the unit vector along the direction of starting and ending optical axis, so that the trajectory that the optical axis traverses on the unit ball is the desired trajectory. Since the roll angle does not participate in the above interpolation, uniform linear interpolation of the roll angle can be performed independently.

3.3. Orientation vector SLERP method

The method of interpolating the vector along the optical axis with the SLERP method is similar to the method of interpolating the quaternion, except that the starting and ending vectors are three-dimensional vectors determined by the starting and ending yaw and pitch angles, respectively. As in (4), we use orientation vector to represent the vector along the end axis, $\mathbf{v_s}$ and $\mathbf{v_e}$ represent the starting orientation vector and the ending orientation vector, respectively.

$$slerp\,(\mathbf{v_s}, \mathbf{v_e}, t) = \frac{\sin\left((1-t)\,\beta\right)}{\sin\beta}\mathbf{v_s} + \frac{\sin\left(t\beta\right)}{\sin\beta}\mathbf{v_e} \tag{4}$$

Assuming that the Euler angles of the starting and ending attitudes are expressed as $\mathbf{E_s} = \left[\phi_s\ \theta_s\ \psi_s\right]^T$ and $\mathbf{E_e} = \left[\phi_e\ \theta_e\ \psi_e\right]^T$, respectively, where ψ is the yaw angle, θ is the pitch angle, and ϕ is the roll angle. Then, we can get $\mathbf{v_s} = \left[\mathrm{c}\psi_s\mathrm{c}\theta_s\ \mathrm{s}\psi_s\mathrm{c}\theta_s\ -\mathrm{s}\theta_s\right]^T$, $\mathbf{v_e} = \left[\mathrm{c}\psi_e\mathrm{c}\theta_e\ \mathrm{s}\psi_e\mathrm{c}\theta_e\ -\mathrm{s}\theta_e\right]^T$, here, "c" is the abbreviation of cosine and "s" is the abbreviation of sinusoidal. t is the parameter that controls the interpolation position, with its value $t = \frac{i}{upper\left(\frac{\beta}{\beta_{\max}}\right)}$, $\left(i = 1, 2, \cdots, upper\left(\frac{\beta}{\beta_{\max}}\right)\right)$, and the function *upper* () here makes its value round up to an integer. β is the angle between $\mathbf{v_s}$ and $\mathbf{v_e}$, $\cos\beta = \mathbf{v_s} \cdot \mathbf{v_e}$, $\beta_{\max}$ is the maximum angle of the allowed adjacent orientation vector.

This method can well control the orientation of the end axis and separate the interpolation of the roll angle from the interpolation of the other two Euler angles at the same time, solving the problem caused by the direct interpolation of the attitude represented in quaternion.

However, this method still has a serious problem. When the orientation vector is close to the gimbal lock position of the platform, the yaw angle corresponding to the trajectory obtained by the above interpolation algorithm may change drastically, as shown in Fig. 7, the three trajectories are all obtained by the above interpolation method. Their starting orientations are the same and the ending

https://doi.org/10.1017/S0263574719001000 Published online by Cambridge University Press

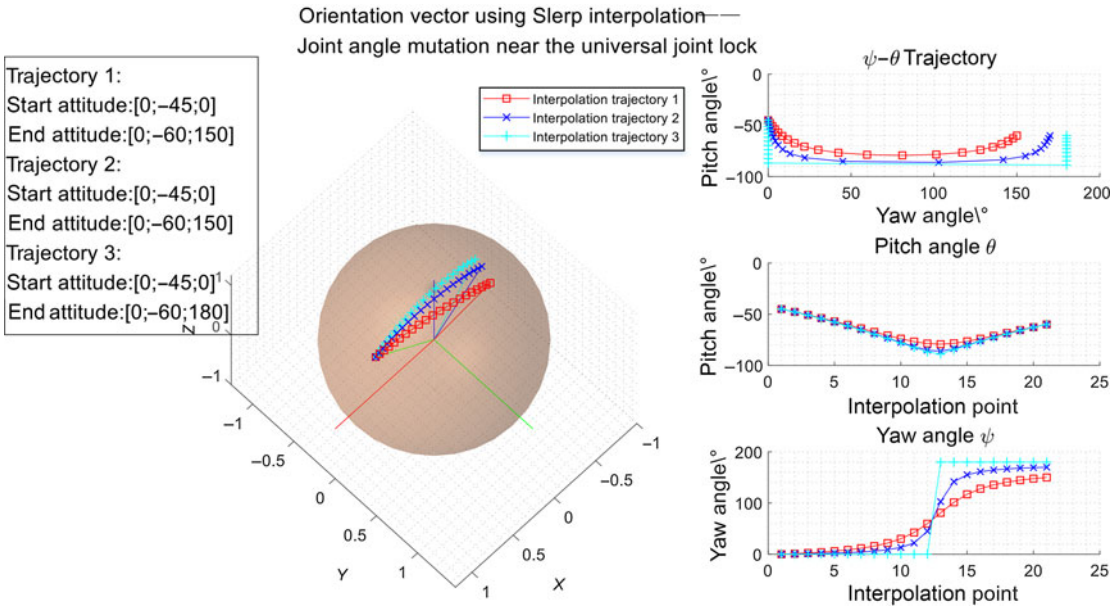

Fig. 7. Joint angle mutation near the gimbal lock.

orientations are close to the position of the gimbal lock. As can be seen from Fig. 7, the yaw angle between adjacent interpolation points becomes larger as the trajectory approaches the gimbal lock position. And the third trajectory passes directly from the gimbal lock position, at which time the yaw angle changes greatly.

In order to solve the above problem, we propose an improved orientation vector SLERP method. This method introduces two correction factors to adjust the algorithm near the platform's gimbal lock based on the above-mentioned orientation vector SLERP method.

3.4. SLERP based on improved orientation vector

The two correction factors mentioned above are defined as viscosity factor μ_t and rejection factor μ_r, and therein, μ_t makes the sampling point dense when the trajectory passes near the gimbal lock, while μ_r makes the trajectory travel away from the position that is near to the gimbal lock. The improved algorithm uses an iterative method for calculation with each iteration performed in three steps. The first step is the calculation of μ_t and step length, the second step is to perform SLERP and normalization, and the third step is to calculate μ_r and to perform "rejection" correction.

In order to calculate the two correction factors, we need to estimate the closeness of the trajectory to the position of the gimbal lock at first. For the platform in this paper, the direction of the gimbal lock corresponding to the end of the platform is the Z-axis direction of the base coordinate system, which is the axis represented by the blue straight lines in Figs. 6 and 7, with the red and green lines are the X-axis and Y-axis, respectively. The cosine of the angle between the orientation vector and the Z-axis is used to reflect the extent to which the trajectory is close to the Z-axis.

The calculation for the i-th iteration of $\mu_{t,i}$ is as shown in (5), where a and b are the two adjustment parameters. a determines the maximum viscosity, that is, the maximum multiple of the density increase of the sampling point. b determines the influence range of the viscous effect, and the viscosity factor μ_t will not work when the angle between the orientation vector and the Z-axis is greater than b. $\langle \mathbf{v_{i-1}}, \mathbf{Z} \rangle$ is the angle between the orientation vector obtained from the previous iteration and the Z-axis.

$$\mu_{t,i} = \begin{cases} 1 & (|\cos\langle \mathbf{v_{i-1}}, \mathbf{Z}\rangle| < |\cos b|) \\ \frac{1-\cos b}{1-a\cos b+(a-1)\cos\langle \mathbf{v_{i-1}}, \mathbf{Z}\rangle} & (|\cos\langle \mathbf{v_{i-1}}, \mathbf{Z}\rangle| \geq |\cos b|) \end{cases} \tag{5}$$

The step length t_i of the i-th iteration is affected by $\mu_{t,i}$, which can result in the viscous effect. The calculation of t_i is as in (6), where, Δt is the normal step length, and t_i varies from 0 to 1 in all iteration steps.

https://doi.org/10.1017/S0263574719001000 Published online by Cambridge University Press

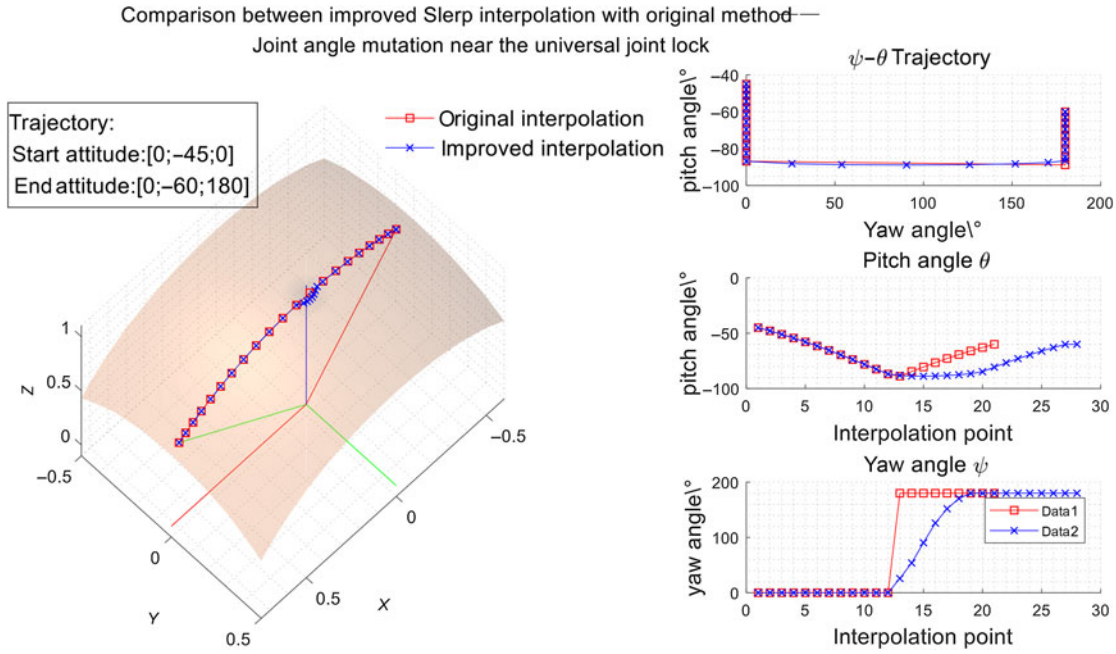

Fig. 8. Comparison between improved SLERP with original method.

$$t_i = t_{i-1} + \mu_{t,i}\Delta t \tag{6}$$

The second step of the iteration is to calculate the normal process of SLERP, using the above (4) with $t = t_i$. And the interpolation results need to be normalized to get $\mathbf{v_{slp,i}}$.

The third step of the iteration is to calculate the rejection factor $\mu_{r,i}$, as in (7), where, c and d are the two adjustment parameters. c determines the maximum degree of rejection, d determines the extent of exclusion, and $\langle \mathbf{v_{slp,i}}, \mathbf{Z} \rangle$ is the angle between the $\mathbf{v_{slp,i}}$ and the Z-axis.

$$\mu_{r,i} = \begin{cases} 0 & \left(\left|\cos\langle \mathbf{v_{slp,i}}, \mathbf{Z}\rangle\right| < |\cos d|\right) \\ c \cdot \frac{\cos\langle \mathbf{v_{slp,i}}, \mathbf{Z}\rangle - \cos d}{1 - \cos d} & \left(\left|\cos\langle \mathbf{v_{slp,i}}, \mathbf{Z}\rangle\right| \geq |\cos d|\right) \end{cases} \tag{7}$$

Finally, the rejection correction is performed to obtain the result v_i of this iteration according to (8), where, $\mathbf{n}$ is the unit normal vector of the plane formed by the starting and ending orientation vector with its direction away from the Z-axis.

$$v_i = \frac{\mathbf{v}_{slp,i} + \mu_{r,i}\mathbf{n}}{\left\|\mathbf{v}_{slp,i} + \mu_{r,i}\mathbf{n}\right\|} \tag{8}$$

$$\mathbf{n} = \pm\frac{\mathbf{v}_s \times \mathbf{v}_e}{\|\mathbf{v}_s \times \mathbf{v}_e\|} \tag{9}$$

The interpolation result of the improved algorithm compared with the original algorithm is as shown in Fig. 8, with the values of the four control parameters being $a = 6$, $b = 4$, $c = 0.02$, and $d = 3$, respectively.

It can be seen from Fig. 8 that the trajectory will bypass the gimbal lock position to avoid a sudden change of the yaw angle when the interpolation trajectory passes near to the gimbal lock. In addition, from the trajectory in $\psi - \theta$ plane, the sampling points near the gimbal lock become dense, which is good to perform motion control.

4. Target Tracking Algorithm and Experiments

In order to verify the effectiveness of the proposed three-axis simultaneous motion control based on the improved orientation vector SLERP algorithm, we design a trajectory tracking experiment to

https://doi.org/10.1017/S0263574719001000 Published online by Cambridge University Press

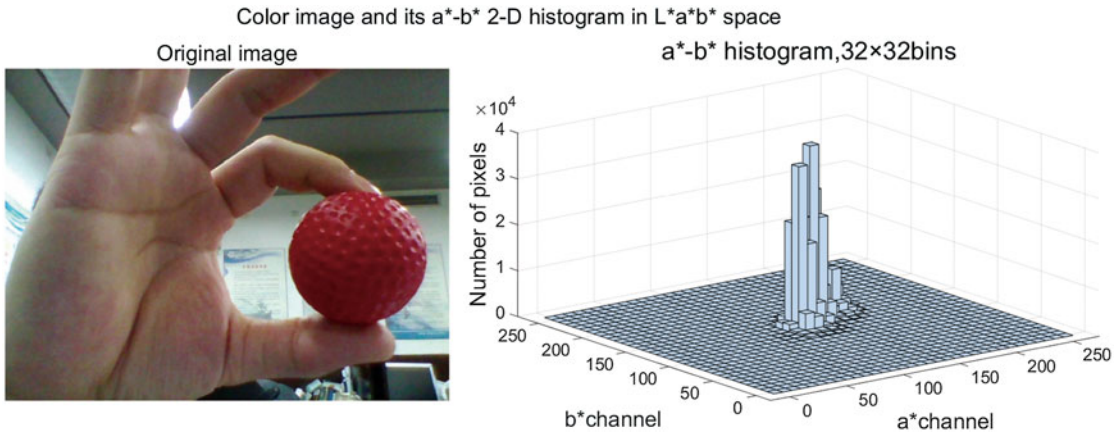

Fig. 9. Color image and its a*-b* 2D histogram.

detect a red ball through the camera installed at the end of the platform. And the platform is controlled to keep the detected target in the center of the FOV according to the detection results.

4.1. Design of the target tracking algorithm

The target tracking algorithm we design uses color as the feature of target recognition, and is implemented in the L*a*b* color space, where, channel L* represents luminosity, channel a* represents the change from green to red, and channel b* represents the change from yellow to blue. Since channel L* varies with luminosity, we only use channel a* and b* to define a color feature here. Generally, a* and b* take positive and negative symmetry values in the L*a*b* space. So, we add them an offset to map their values to a range of 0–255.

Here, we use the color histogram as the feature of the target, and a 2D histogram is needed to represent the characteristics of the target since two channels are used at the same time. Assuming that there is a $m \times n$ color image I whose channel a* and channel b* are I_a and I_b , respectively. Then, its a*-b* histogram can be calculated according to (10), where, $H_{i,j}$ is the (i, j)-th element in the histogram, *Count* () indicates the number of pixels that satisfies the condition, Δa and Δb are the two channels' widths of the data set in the histogram, respectively.

$$H_{i,j} = Count\left(I\left(x, y\right) | i * \Delta a \leq I_a\left(x, y\right) < (i+1) * \Delta a, j * \Delta b \leq I_b\left(x, y\right) < (j+1) * \Delta b\right) \quad (10)$$

As shown in Fig. 9, the left side is the original image of the red ball that needs to be recognized and tracked, and the right side is its a*-b* histogram in the L*a*b* color space. The histogram itself can be regarded as a probability density map of a random variable in its value space, so we can use the color histogram as a representation of the target features.

In order to track the target, first we need to generate a target histogram H_{obj}, which can be generated by an image containing only the object to be detected according to (10). Each element in H_{obj} can be regarded as the likelihood that the color range corresponding to this element appears in the target.

After obtaining the histogram H_{obj} representing the target feature, we need to determine the possibility that the pixel in the new image belongs to the specific target, in order to find this specific target in a new image. And this possibility can be replaced by the corresponding value of the color of the pixel in the target histogram H_{obj}. Figure 10 is the result of the inverse mapping from the target histogram to the image. The left image is the original one with the selected target area marked with rectangle, the middle image is the 2D histogram of the selected target area, and the right one is a probability density map obtained by inversely mapping each pixel in the original image from the target histogram, and the brighter the pixel is, the higher the likelihood that this pixel belongs to the target is.

In practical applications, we detect a new image of the target through the camera. After obtaining the probability density map of the target in one image, we can know where the target is most likely to be.

https://doi.org/10.1017/S0263574719001000 Published online by Cambridge University Press

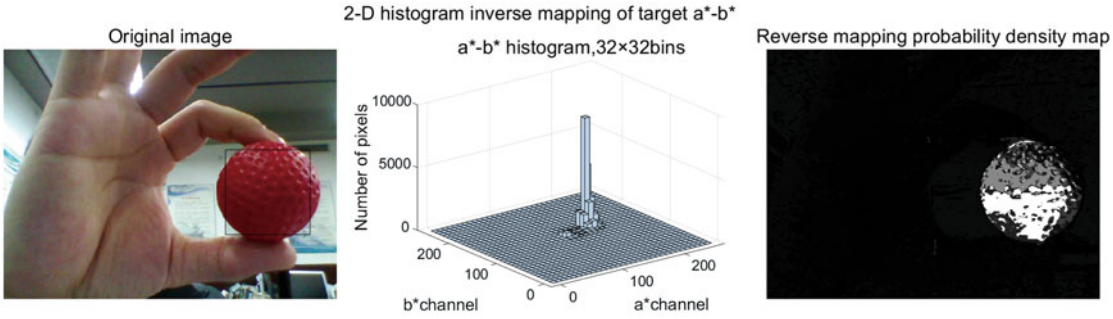

Fig. 10. Reverse mapping from target histogram to image.

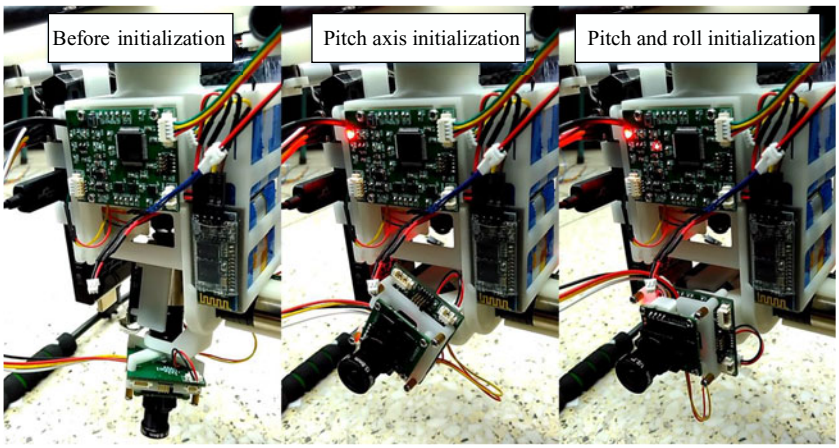

Fig. 11. Process of the initialization at the beginning of the experiment.

4.2. Tracking experiments based on three-axis simultaneous motion control

In order to verify the effectiveness of the three-axis simultaneous motion control based on improved orientation vector SLERP designed in this paper, a tracking experiment was designed using the red ball as the following target detected by the above target tracking algorithm. At the beginning of the experiment, the three-axis motion platform needs to be initialized for better tracking performance, which means that the three axes need to move to their specified initial attitudes, and in this designed platform, the initialization is mainly for the second and third axes. Here, we determine the initial attitudes of the platform with the attitude measurement value of the attitude measurement units installed at the end of the platform and the first axis of the platform. The pitch axis and the roll axis are sequentially returned to the initial attitude during the initialization, with its process as shown in Fig. 11.

During the experiment, the target attitudes were directly sent to the motion controller, and the motion controller performs attitude interpolation and controls the end of the platform to the specified attitude. The commands sent in the experiment were absolute commands relative to the base with its starting Euler angle $\left[0^\circ\ -45^\circ\ 0^\circ\right]^T$. Then, we sent the control command $\left[0^\circ\ -45^\circ\ 179^\circ\right]^T$, and finally sent the control command $\left[0^\circ\ -45^\circ\ 0^\circ\right]^T$ to make the platform back to its starting position.

The trajectory generated by the above-mentioned attitude commands is very close to the position of the gimbal lock, thus the performance of the proposed interpolation algorithm in the vicinity of the gimbal lock can also be verified. The tracking process based on the improved orientation vector SLERP is shown in Fig. 12.

The joint angle trajectory generated by real-time interpolation during the trajectory-following process is shown in Fig. 13, while the attitude trajectory in the $\psi - \theta$ plane is shown in Fig. 14.

It can be seen from the above experimental results that the trajectory obtained by interpolation has no big mutation and the control process is relatively smooth although the trajectory is very close to the gimbal lock position of the platform. This experiment proves the validity of the designed improved orientation vector SLERP algorithm. At the same time, the effectiveness of the three-axis simultaneous motion control strategy is also verified.

https://doi.org/10.1017/S0263574719001000 Published online by Cambridge University Press

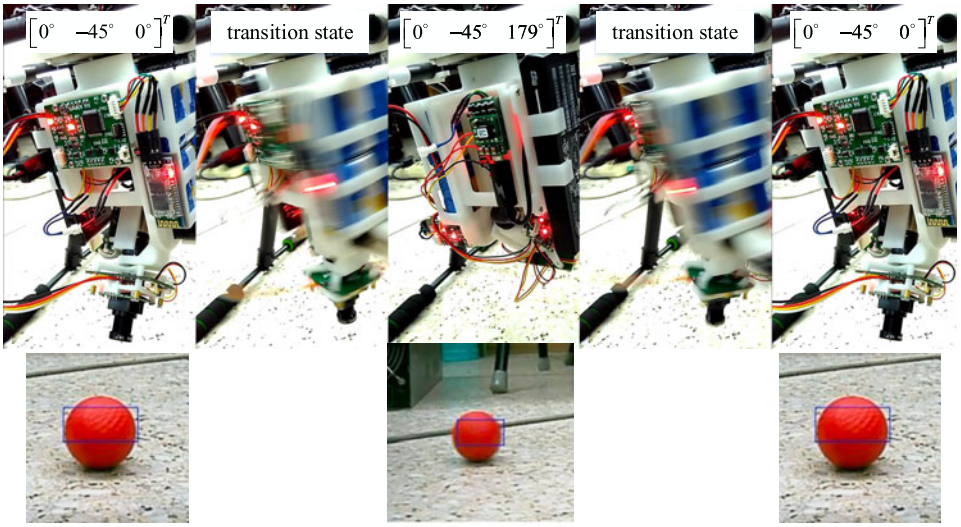

Fig. 12. Tracking process based on proposed improved SLERP.

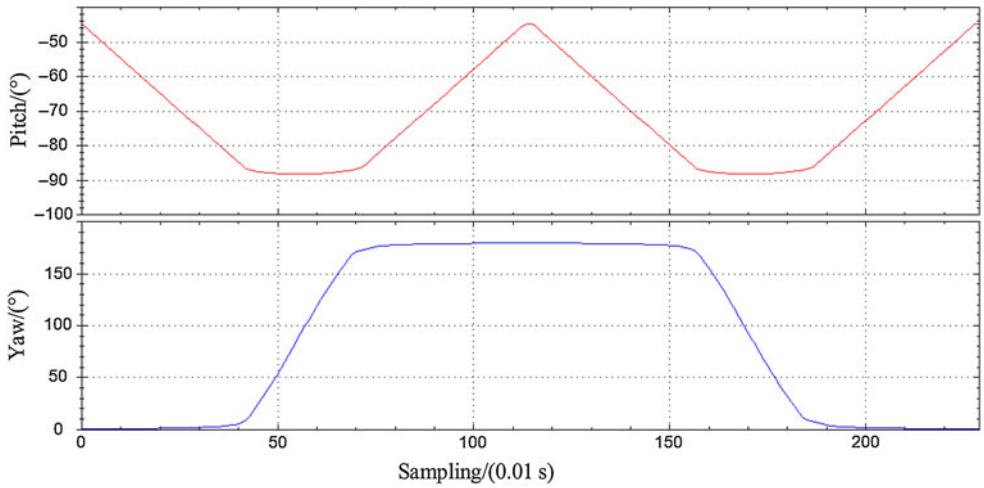

Fig. 13. Curve of pitch angle and yaw angle during track-following process.

5. Conclusion

In order to realize the attitude tracking control of the three-axis platform, the paper proposes an improved orientation vector SLERP method which uses two correction factors to adjust the algorithm near the platform's gimbal lock position, so that the platform can move smoothly around the gimbal lock position. At first, an adaptive speed measurement method based on incremental encoder is introduced, which can automatically adapt to high speed and low speed, and estimate the ultra-low speed to realize the speed measurement of large dynamic range. Then, the SLERP method for the quaternion interpolation is introduced and improved in detail. Based on the joint angle mutation near the gimbal lock, an orientation vector SLERP method is proposed which uses viscosity factor and rejection factor to adjust the algorithm near the platform's gimbal lock position. Finally, a tracking experiment was designed using the red ball as the following target detected by the designed target tracking algorithm, which verified the effectiveness of the attitude tracking control of the three-axis platform based on the proposed improved orientation vector SLERP.

In future research, the attitude interpolation and tracking control algorithm of the three-axis motion platform will be further optimized, and will be transplanted to our upper limb rehabilitation robot to further improve its performance during the active and passive rehabilitation training of the upper limb rehabilitation robot.

https://doi.org/10.1017/S0263574719001000 Published online by Cambridge University Press

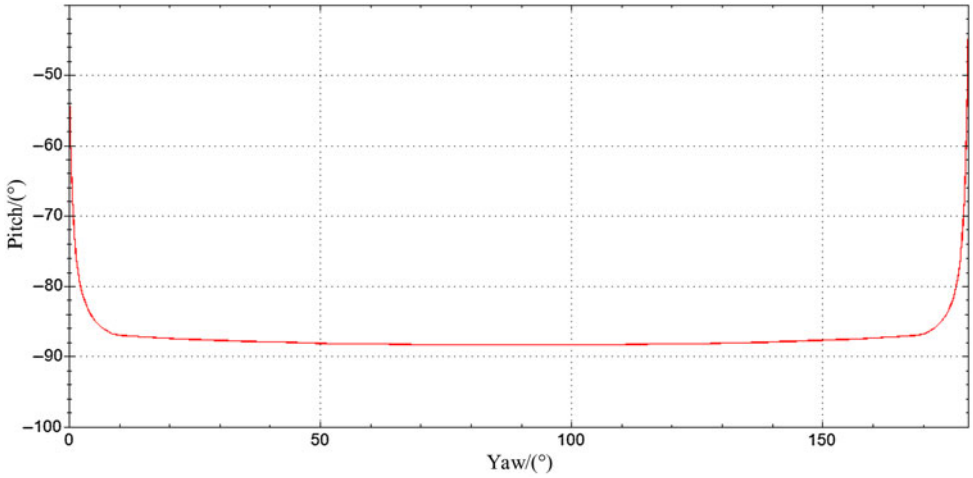

Fig. 14. Attitude trajectory in the $\psi - \theta$ plane.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grants No. 51675008 and 51705007), in part funded by the Beijing Natural Science Foundation (Grants No. 17L20019 and 3171001).

References

1. M. Neubauer and A. Muller, "Smooth Orientation Path Planning with Quaternions Using B-splines," *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2015) pp. 2087–2092.
2. H. Lee and N. Hogan, "Energetic passivity of the human ankle joint," *IEEE Trans. Neural Syst. Rehabil. Eng.* **24**(12), 1416–1425 (2016). A Publication of the IEEE Engineering in Medicine & Biology Society.
3. S. Hussain, P. K. Jamwal and M. H. Ghayesh, "State-of-the-art robotic devices for ankle rehabilitation: Mechanism and control review," *Proc. Inst. Mech. Eng. Part H: J. Eng. Med.* **231**(12), 1224–1234 (2017).
4. A. Niyetkaliyev, S. Hussain, M. Ghayesh and G. Alici, "Review on design and control aspects of robotic shoulder rehabilitation orthoses," *IEEE Trans. Hum.-Mach. Syst.* **47**(6), 1134–1145 (2017).
5. K. Or, M. Tomura, A. Schmitz, S. Funabashi and S. Sugano, "Position-Force Combination Control with Passive Flexibility for Versatile In-hand Manipulation Based on Posture Interpolation," *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2016).
6. B. Fang, F. Sun, H. Liu and G. Di, "A novel data glove using inertial and magnetic sensors for motion capture and robotic arm-hand teleoperation," *Ind. Rob.* **44**(2), 155–165 (2017).
7. L. Biagiotti, L. Moriello and C. Melchiorri, "A repetitive Control Scheme for Industrial Robots Based on B-spline Trajectories," *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2015) pp. 5417–5422.
8. S. Ceriani, C. Sanchez, P. Taddei, E. Wolfart and V. Sequeira, "Pose Interpolation Slam for Large Maps Using Moving 3D Sensors," *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2015) pp. 750–757.
9. S. W. Chang, Y. T. Chiang and F. R. Chang, "Slerp-Based Optimal Triad Algorithm," *Proceedings of Sice Conference 2010* (2010) pp. 331–335.
10. A. Jin-Su, C. Won-Jee and J. Chang-Doo, "Realization of orientation interpolation of 6-axis articulated robot using quaternion," *J. Cent. South Univ.* **19**(12), 3407–3414 (2012).
11. J. Park, "Interpolation and Tracking of Rigid Body Orientations," *International Conference on Control Automation and Systems* (2010) pp. 668–673.
12. E. B. Dam, M. Koch and M. Lillholm, Quaternions, Interpolation and Animation *Technical Report* (1998).
13. B. Siciliano and O. Khatib, *Springer Handbook of Robotics*, vol. 56, no. 8 (Springer, Berlin, Heidelberg, 2007) pp. 987–1008.
14. S. Ken, "Animating rotation with quaternion curves," *ACM SIGGRAPH Comput. Graphics* **19**(3), 245–254 (1985).
15. W. Ge, Z. Huang and G. Wang, *Interpolating Solid Orientations with a C 2-Continuous B-Spline Quaternion Curve* (Springer, Berlin, Heidelberg, 2007).
16. M. X. Kong, C. Ji, Z. S. Chen and R. F. Li, "Application of Orientation Interpolation of Robot Using Unit Quaternion," *IEEE International Conference on Information and Automation* (2014) pp. 384–389.
17. X. Jiang, E. Barnett and C. Gosselin, "Dynamic point-to-point trajectory planning beyond the static workspace for six-dof cable-suspended parallel robots," *IEEE Trans. Rob.* **34**(3), 781–793 (2018).