

Research Article

Smooth and Accurate Trajectory Planning for Industrial Robots

Youdong Chen, Ling Li, and Xudong Ji

School of Mechanical Engineering and Automation, Beihang University, Beijing, 100191, China

Correspondence should be addressed to Ling Li; liuyunmu@126.com

Received 18 February 2014; Accepted 25 March 2014; Published 10 April 2014

Academic Editor: Yong Tao

Copyright © 2014 Youdong Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes a smooth and accurate trajectory planning for industrial robots using geodesics. The workspace of a robot is split into positional and orientational parts. A Riemannian metric is given on each space such that the desired motion is a geodesic for the given metric. By regarding joint variables as local coordinates of the position space and the orientation space, Cartesian trajectories are represented by joint trajectories. A smooth and accurate motion of the robot end-effector and smooth joint trajectories corresponding to the motion can be obtained by calculating geodesics on the position space and the orientation space. To demonstrate the effectiveness of the proposed method, simulation experiments are conducted using the PUMA 560 robot.

1. Introduction

Industrial robots are used in many different industrial applications, like handling, machining, arc welding, and painting. In some cases, such as machining, arc welding, and painting, it is important that a robot end-effector moves accurately and in a smooth continuous manner along a specified trajectory. For example, in a machining process a robot end-effector may be required to move a tool for cutting a three-dimensional surface smoothly.

Olabi et al. presented a method of trajectory planning adapted for continuous machining by robots [1]. In this method, a smooth motion law was generated by means of a parametric speed interpolator. Ning et al. proposed a method for trajectory generation based on dynamic movement primitives [2]. They generated smooth trajectories which satisfied position and velocity boundary conditions at start and end points with high precision. Boryga and Graboś presented a planning mode of manipulator motion trajectory with higher-degree polynomials [3]. Gasparetto and Zanotto developed a method for smooth trajectory planning of robot manipulators [4]. They worked out an objective function containing a term proportional to the integral of the squared jerk (to ensure that the trajectory is smooth) and the second

term, proportional to the total execution time. Liu et al. presented a smooth trajectory planning method which was designed as a combination of the planning with multidegree splines in the Cartesian space and multidegree B-splines in the joint space [5]. Yang and Sukkarieh proposed an analytical continuous-curvature path-smoothing algorithm which was based upon parametric cubic Bézier curves [6].

However, the trajectory planning methods are often using interpolating functions, and the interpolating is performed either in the joint space or the Cartesian space. In joint space schemes, in order to obtain accurate trajectories, planning can be done by interpolating a sequence of via points, but the motion actually performed by the robot end-effector between these via points is not easily foreseeable. On the other hand, because of the nonlinearity of the inverse kinematics, a smooth trajectory in the Cartesian space may cause a nonsmooth joint motion when the interpolated data in the Cartesian space are mapped into the joint space and consequently a nonsmooth and inaccurate motion in the Cartesian space.

Since the end of the 20th century, geodesics have been introduced into robotics for path planning. Geodesic, which is the necessary condition of the shortest length between two points on a Riemannian manifold, is intrinsic and has

no relation with coordinates [7]. The joints of a robot are regarded as linearly dependent in the geodesic method [8].

Rodnay and Rimon visualized the dynamic properties of a 2DOFs (degrees of freedom) robot in the three-dimensional Euclidean space [9]. They introduced geodesics on the dynamic surface to represent trajectories when the robot system made free motions. Using the notions of the Riemannian metric and the covariant derivative, Žefran et al. focused on the generation of smooth three-dimensional rigid body motions which were Cartesian space schemes in their paper [7]. Tchoń and Duleba addressed on inverting singular kinematics [10]. An advantage of their method was that the workspace locations generated by the method lay on a straight line. However, the method was based upon the so-called generalized Newton algorithm and the straight line was planned point by point using the inverse kinematics. Selig and Ovseevitch sought to combine the control and the trajectory planning and manipulate robots along helical trajectories [11]. They applied the Lie algebra and the Lie group which often made the planning abstruse, abstract, and not easy to put into practice. Zhang et al. regarded the arc length and the kinetic energy as Riemannian metrics to achieve the shortest path and the invariant kinetic energy trajectories, respectively [8, 12]. But when they used the arc length as the Riemannian metric, they actually could not plan trajectories of robots with more than 3 DOFs and the orientation problem had not been tackled.

In order to overcome the above-mentioned problems, this paper proposes a smooth and accurate trajectory planning using geodesics. The workspace of a robot is split into positional and orientational parts. Riemannian metrics are then given on the two spaces such that the desired motions are geodesics for the given metrics. It allows them to generate trajectories that include the position and the orientation. By regarding joint variables as local coordinates of the position space and the orientation space and calculating geodesics on them, Cartesian trajectories are represented by joint trajectories. Because of the geometric characteristics of geodesics, Cartesian trajectories and corresponding joint trajectories are both smooth and accurate.

As mentioned above, the distance length and kinematic energy were regarded as Riemannian metrics, respectively, in previous studies. The proposed method splits the workspace into positional and orientational parts and uses two Riemannian metrics: the distance metric and the orientation metric. The shortest paths of a robot with more than 3 DOFs and orientation trajectories can be obtained using geodesics, while these are not considered adequately in the previous works. Traditional methods often require a constant solution to the inverse kinematics. The interpolating is performed either in the joint space or the Cartesian space, and the joint scheme and the Cartesian scheme are often separate. The constant solution to the inverse kinematics and the separation of joint schemes and Cartesian schemes may cause the planned trajectories nonsmooth or inaccurate. The immediate output of the geodesic method which selects joint variables as local coordinates is joint trajectories, and the Cartesian trajectory obtained by the forward kinematics is

exactly the planning geodesic path. The geodesic method requires only the inverse kinematic solutions of endpoints.

2. Background of Geodesic

To begin with, we provide a brief background on geodesics.

2.1. Manifold and Local Coordinate. A manifold M^n is a Hausdorff topological space for which each point p has a neighborhood $U \subset M^n$ homeomorphic to an open subset of the Euclidean space R^n [13]. If $\varphi : U \rightarrow \varphi(U) \subset R^n$ ($\varphi(U)$ is open) is the homeomorphism and denotes $x^i(p) \doteq (\varphi(p))^i$, we say that $(U; x^i)$ is a local coordinate system of M and x^i ($i = 1, \dots, n$) are local coordinates.

2.2. Riemannian Manifold. Let (M^n, g) be a Riemannian manifold; that is, M^n is an n -dimensional differentiable manifold and g is a Riemannian metric. A Riemannian metric g is a symmetric, positive definite quadratic form. Let $\{\theta_i\}_{i=1}^n$ be local coordinates in a neighborhood U of a point of M . The metric may be written in local coordinates as

$$g = \sum_{i,j=1}^n g_{ij} d\theta_i d\theta_j, \quad (1)$$

where $g_{ij} \doteq g(\partial/\partial\theta_i, \partial/\partial\theta_j)$ and d, ∂ refer to the differential and the partial differential, respectively [14, 15].

2.3. Geodesic. Let $\gamma : I \rightarrow M$ be a smooth curve. In a local coordinate system, it can be written as

$$\theta_i = \theta_i(t), \quad t \in I, \quad i = 1, 2, \dots, n. \quad (2)$$

We say that γ is a geodesic if and only if it satisfies the geodesic equations

$$\frac{d^2\theta_i}{dt^2} + \sum_{k,j=1}^n \Gamma_{kj}^i \frac{d\theta_k}{dt} \frac{d\theta_j}{dt} = 0, \quad i = 1, 2, \dots, n. \quad (3)$$

The Christoffel symbols Γ_{kj}^i are given in terms of the Riemannian metric by

$$\Gamma_{kj}^i = \frac{1}{2} \sum_{m=1}^n g^{mi} \left(\frac{\partial g_{km}}{\partial \theta_j} + \frac{\partial g_{jm}}{\partial \theta_k} - \frac{\partial g_{kj}}{\partial \theta_m} \right), \quad (4)$$

where g^{mi} is the element of the inverse Riemannian metric coefficient matrix [16].

3. Smooth and Accurate Trajectory Planning Using Geodesics

Once the link frames have been attached and the corresponding link parameters have been found, the kinematical equations are straightforwardly determined. Assume that the T matrix of a robot end-effector is

$$T = \begin{pmatrix} R & p \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (5)$$

where $n = o \times a$, $o \cdot o = 1$, $a \cdot a = 1$, $o \cdot a = 0$ [17].

3.1. Local Coordinates and the Selection Mode. To combine joint space schemes and Cartesian space schemes, a relationship between the joint space and the Cartesian space should be established. A nature idea is to select the set of all joint variables as a coordinate system of the Cartesian space. The forward kinematics gives a description from the joint space to the Cartesian space. Specifically, given a set of joint angles, the position and the orientation of the tool frame relative to the base frame are definite. Conversely, given a coordinate which represents the position and the orientation of the tool frame relative to the base frame, the joint coordinates may be indefinite, because the inverse solution is not always unique or even existent. There is often no one-to-one mapping between the Cartesian space and the joint space. For this reason, the set of all joint variables cannot be regarded as a coordinate system of the Cartesian space.

Although joint variables cannot be regarded as coordinates of the Cartesian space, they can be selected as local coordinates. The workspace of a robot is split into its positional and orientational parts. Some joint variables are selected as local coordinates of the position space, and others are regarded as local coordinates of the orientation space.

3.2. Trajectory Planning for Linear Motions Using Geodesics. A line is the only geodesic in the Euclidean space with the distance metric. The position space can be viewed as a Euclidean space and the distance metric of the position space is

$$g_d = (dp)^2, \quad (6)$$

where dp is the derivative of the position vector $p = (p_x, p_y, p_z)$. A manipulator trajectory planning approach in [8] was proposed by using this metric. But the position is only related to no more than three variables. For a robot whose DOFs are more than three, the trajectory planning approach using the metric g_d only makes part of joints move while movements of others are unknown or still.

Normally, the orientation of the tool frame along a path is defined by the task and cannot be freely selected. On the orientation space, define the Riemannian metric g_o as follows:

$$g_o = (dn)^2 + (do)^2 + (da)^2, \quad (7)$$

where dn , do , and da are the derivatives of the orientation vectors $n = (n_x, n_y, n_z)$, $o = (o_x, o_y, o_z)$, and $a = (a_x, a_y, a_z)$, respectively.

Both the dimensions of the position space and the orientation space are no more than three. Trajectories of a robot with no more than six DOFs can be planned. If a robot has more than six DOFs, an extra metric which focuses on task constraints or dynamic optimizations should be constructed to get redundant joint trajectories. The flow graph of the trajectory planning using geodesics is shown in Figure 1.

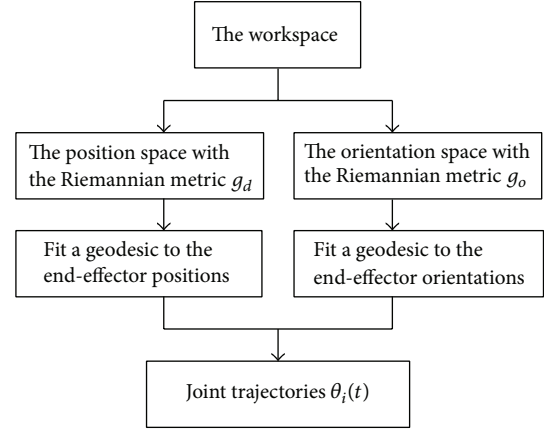


FIGURE 1: A flowchart of the trajectory planning using geodesics.

3.3. The Evaluation Criterion of the Riemannian Metric Selection. The key point of the geodesic method is finding a suitable metric given the desired trajectory. First of all, a metric should be a symmetric, positive definite quadratic form. Secondly, a metric should be a measure of distance, energy, or other aspects that depend on the desired motions. Geodesic is the necessary condition of the shortest length which is measured by the metric. For example, g_d is a distance metric which measures the distance and a geodesic corresponding to g_d is the shortest path. Finally, the relationship between the amount of a robot DOFs and the dimension of the Riemannian manifold corresponding to the selected metric should be focused on. Often, little attention is devoted to the Riemannian manifold. When the amount of robot DOFs is more than the dimension of the Riemannian manifold, something goes wrong. To avoid a constant solution to the inverse kinematics, a geodesic is desired to be represented by joint trajectories, which can be implemented by choosing the joint variables as local coordinates. The amount of the DOFs should be no more than the dimension of the Riemannian manifold. If not, two or more Riemannian manifolds and their metrics should be constructed, which provides the possibility of multiobjective optimization.

3.4. Computational Considerations. Generally, when a Riemannian metric is given, put it in the form of the coefficient matrix

$$g = (d\theta_1 \ d\theta_2 \ \cdots \ d\theta_n) G \begin{pmatrix} d\theta_1 \\ d\theta_2 \\ \vdots \\ d\theta_n \end{pmatrix}, \quad (8)$$

where $G = (g_{ij})_{n \times n}$ is the coefficient matrix of the Riemannian metric and θ_i ($i = 1, \dots, n$) are joint positions which can be linear or angular. With the coefficient matrix G , we can calculate the Christoffel symbols Γ_{kj}^i by (4). Then the geodesic equations are obtained with the form as (3).

With the Riemannian metrics g_d and g_o , two sets of geodesic equations with the form as (3) are obtained. These equations and boundary conditions constitute simultaneous differential equations. The solutions of the simultaneous differential equations give the joint trajectories which represent the desired linear motion of the robot end-effector.

In many practical systems, the calculating time is a consideration. The computational efficiency is mainly related to the solution to geodesic equations which are second-order nonlinear ordinary differential equations. The second-order differential equations can be converted to a first-order system. The differential equations are approximated by finite difference equations which are algebraic equations. Unlike initial value problems, a boundary value problem probably does not have a unique solution. Because of this, an initial guess for the desired solution should be provided [18].

For example, the function `bvp4c` in MATLAB solves two-point boundary value problems for ordinary differential equations [19]. `bvp4c` produces a solution that is continuous and has a continuous first derivative [20]. The solution is fourth-order accurate uniformly. Six-order accuracy can be got by a sixth-order solver `bvp6c` which not only supplies greater accuracy, but also performs more efficiently [21]. One also can write a C++ program to get the solution of the geodesic equations with high calculation speed and high accuracy.

3.5. Trajectory Planning for the PUMA 560 Robot Using Geodesics. To illustrate the whole trajectory planning procedure using geodesics, the PUMA 560 robot is taken for example. The PUMA 560 robot and its link parameters are shown in Figure 2 and Table 1. The T matrix of the robot end-effector is with the form as (5), where

$$\begin{aligned} n_x &= c_1 [c_{23} (c_4 c_5 c_6 - s_4 s_6) - s_{23} s_5 c_6] \\ &\quad + s_1 (s_4 c_5 c_6 + c_4 s_6), \\ n_y &= s_1 [c_{23} (c_4 c_5 c_6 - s_4 s_6) - s_{23} s_5 c_6] \\ &\quad - c_1 (s_4 c_5 c_6 + c_4 s_6), \\ n_z &= -s_{23} (c_4 c_5 c_6 - s_4 s_6) - c_{23} s_5 c_6, \\ o_x &= c_1 [c_{23} (-c_4 c_5 s_6 - s_4 c_6) + s_{23} s_5 s_6] \\ &\quad + s_1 (-s_4 c_5 s_6 + c_4 c_6), \\ o_y &= s_1 [c_{23} (-c_4 c_5 s_6 - s_4 c_6) + s_{23} s_5 s_6] \\ &\quad - c_1 (-s_4 c_5 s_6 + c_4 c_6), \\ o_z &= -s_{23} (-c_4 c_5 s_6 - s_4 c_6) + c_{23} s_5 s_6, \\ a_x &= -c_1 (c_{23} c_4 s_5 + s_{23} c_5) - s_1 s_4 s_5, \\ a_y &= -s_1 (c_{23} c_4 s_5 + s_{23} c_5) + c_1 s_4 s_5, \end{aligned}$$

TABLE 1: Link parameters of the PUMA 560 robot.

i	α_{i-1}	a_{i-1}	d_i	θ_i	Parameter values (mm)
1	0	0	0	θ_1	$a_2 = 431.8$
2	$-\frac{\pi}{2}$	0	d_2	θ_2	$a_3 = 20.32$
3	0	a_2	0	θ_3	$d_2 = 149.09$
4	$-\frac{\pi}{2}$	a_3	d_4	θ_4	$d_4 = 433.07$
5	$\frac{\pi}{2}$	0	0	θ_5	
6	$-\frac{\pi}{2}$	0	0	θ_6	

$$a_z = s_{23} c_4 s_5 - c_{23} c_5,$$

$$p_x = c_1 [a_2 c_2 + a_3 c_{23} - d_4 s_{23}] - d_2 s_1,$$

$$p_y = s_1 [a_2 c_2 + a_3 c_{23} - d_4 s_{23}] + d_2 c_1,$$

$$p_z = -a_3 s_{23} - a_2 s_2 - d_4 c_{23}.$$

(9)

The position $p = (p_x, p_y, p_z)$ depends on θ_1, θ_2 , and θ_3 which are selected as local coordinates of the position space. To control the position, we need to control the three joint variables. For a linear motion of the end-effector, we just need to construct a Riemannian metric and calculate the geodesics on the position space. The Riemannian metric is given by

$$e = (dp)^2 = (d\theta_1 \ d\theta_2 \ d\theta_3) E \begin{pmatrix} d\theta_1 \\ d\theta_2 \\ d\theta_3 \end{pmatrix}, \quad (10)$$

where dp is the derivative of the position vector p , $E = (e_{ij})_{3 \times 3}$ is the coefficient matrix corresponding to e , and

$$\begin{aligned} e_{11} &= (s_1 d_3 - c_1 (c_2 a_2 + c_{23} a_3 - s_{23} d_4))^2 \\ &\quad + (c_1 d_2 + s_1 (c_2 a_2 + c_{23} a_3 - s_{23} d_4))^2, \\ e_{12} &= d_{21} = d_2 (s_2 a_2 + s_{23} a_3 + c_{23} d_4), \\ e_{13} &= d_{31} = d_2 (s_{23} a_3 + c_{23} d_4), \\ e_{22} &= a_2^2 + a_3^2 + d_4^2 + 2a_2 (c_3 a_3 - s_3 d_4), \\ e_{23} &= d_{32} = a_3^2 + d_4^2 + a_2 (c_3 a_3 - s_3 d_4), \\ e_{33} &= a_3^2 + d_4^2. \end{aligned} \quad (11)$$

The corresponding geodesic is determined by (3) and (4). Equation (3) corresponding to the Riemannian metric e is a 2-order differential equation group with 3 unknowns (θ_1, θ_2 , and θ_3). The geodesic is identified by given initial conditions or boundary conditions.

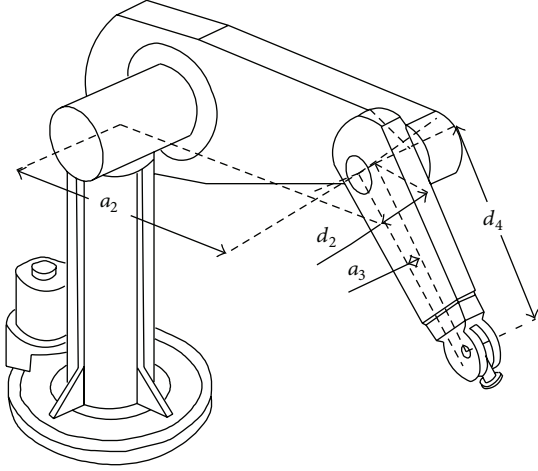


FIGURE 2: The PUMA 560 robot.

When the values of θ_1 , θ_2 , and θ_3 are obtained, the orientation depends on θ_4 , θ_5 , and θ_6 which are selected as local coordinates of the orientation space. Define the Riemannian metric h of the orientation space as follows:

$$h = (dn)^2 + (do)^2 + (da)^2$$

$$= \begin{pmatrix} d\theta_4 & d\theta_5 & d\theta_6 \end{pmatrix} H \begin{pmatrix} d\theta_4 \\ d\theta_5 \\ d\theta_6 \end{pmatrix}, \quad (12)$$

where

$$H = (h_{ij})_{3 \times 3} = \begin{pmatrix} 2 & 0 & 2c_5 \\ 0 & 2 & 0 \\ 2c_5 & 0 & 2 \end{pmatrix} \quad (13)$$

is the coefficient matrix corresponding to h .

Equation (3) corresponding to the Riemannian metric h is a 2-order differential equation group. With the Riemannian metrics e and h , the trajectories can be obtained by solving the geodesic equations under boundary conditions.

For example, when the robot moves linearly from the initial point A to the end point B , the coordinates of A and B are shown in Tables 2 and 3. The robot performs a linear motion from the initial point to the end point according to this planning. The trajectory of the end-effector determined by the geodesic method is shown in Figure 3(a). The velocity, acceleration, and jerk of the end-effector are shown in Figure 3(b). The orientation of the robot end-effector is shown in Figure 4. The joint trajectories are shown in Figure 5. As it can be seen, not only the position trajectories but also the orientation trajectories of the end-effector are smooth. The joint trajectories and their derivatives (velocity, acceleration, and jerk, resp.) are also smooth.

TABLE 2: The coordinates of the end points in the Cartesian space.

	A	B
p	(-67, 138, -347) (mm)	(-128, 100, -200) (mm)
n	(-0.6, 0.2, -0.774597)	(-0.4, 0.5, -0.768115)
o	(0.675603, -0.391868, -0.6245)	(0.699733, -0.374666, -0.608276)
a	(-0.428439, -0.89802, 0.1)	(-0.591924, -0.780785, -0.2)

TABLE 3: The coordinates of the end points in the joint space.

	A	B
t (s)	0	5
θ_1 (rad)	0.214251	0.499461
θ_2 (rad)	0.514124	0.549039
θ_3 (rad)	0.787751	1.12713
θ_4 (rad)	-1.25336	-2.20054
θ_5 (rad)	0.975085	0.520477
θ_6 (rad)	0.490851	1.50993

3.6. Results of the Simulations. The method described in this paper has been tested in simulations for the PUMA 560 robot. The results of the simulations declare that the motion of the robot end-effector is accurate and the trajectories and their derivatives are smooth as shown in Figure 3. The orientation of the end-effector is smooth as shown in Figure 4. The trajectories of the joints and their derivatives are smooth as shown in Figure 5. That is, the method can be used to plan smooth and accurate motions of end-effectors and corresponding smooth joint trajectories for industrial robots.

4. Conclusion and Future Work

A smooth and accurate trajectory planning for industrial robots using geodesics is presented in this paper. The main idea is to select a suitable metric for which a geodesic is the desired motion. The workspace of a robot is split into positional and orientational parts. A Riemannian metric is given on each space such that the desired motion is a geodesic. The geodesic equations are solved numerically and the results are used to control the robot. Simulations of this method using the PUMA 560 robot are given.

The method applies to robots with no more than 6 DOFs and focuses on linear motions. Future work will be directed towards constructing appropriate metrics to plan trajectories of robots with more than six or redundant DOFs and achieve geodesic planning of circular and other curve motions.

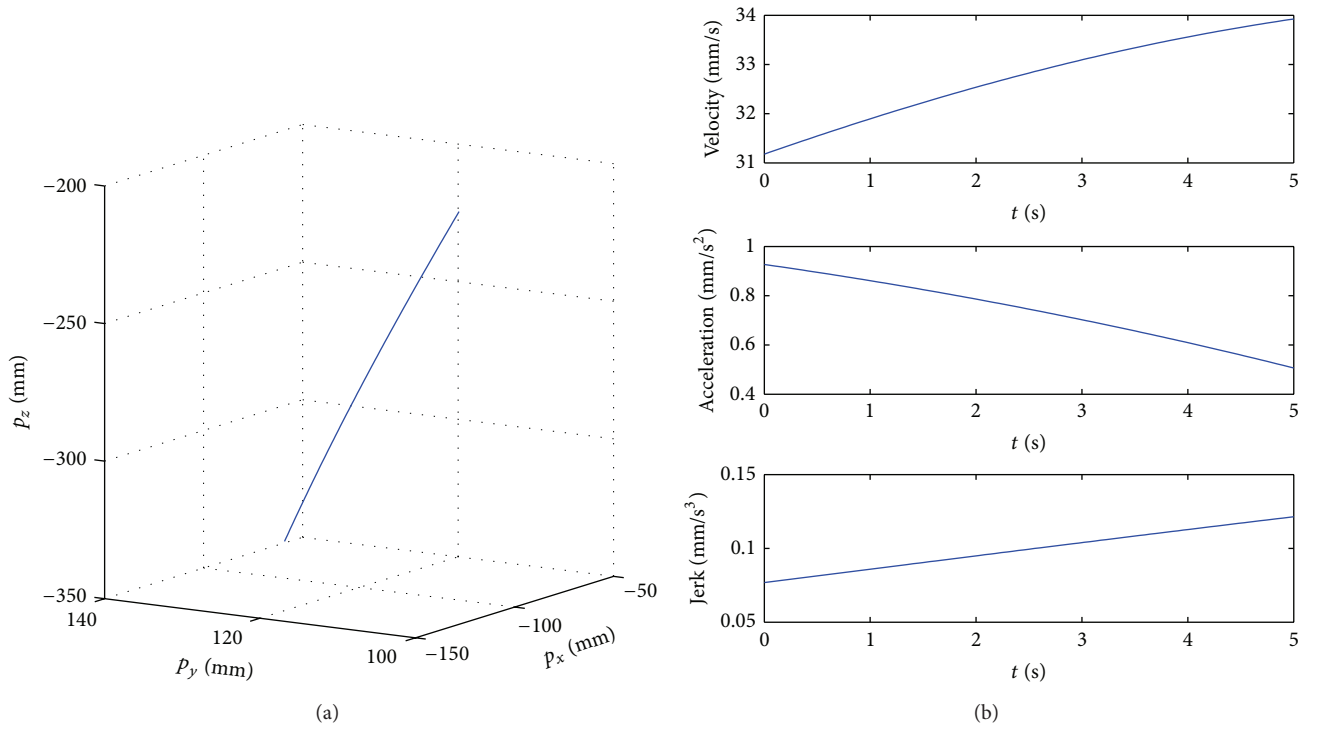


FIGURE 3: The trajectories of the end-effector according to the linear motion: (a) the positions and (b) the velocities, accelerations, and jerks.

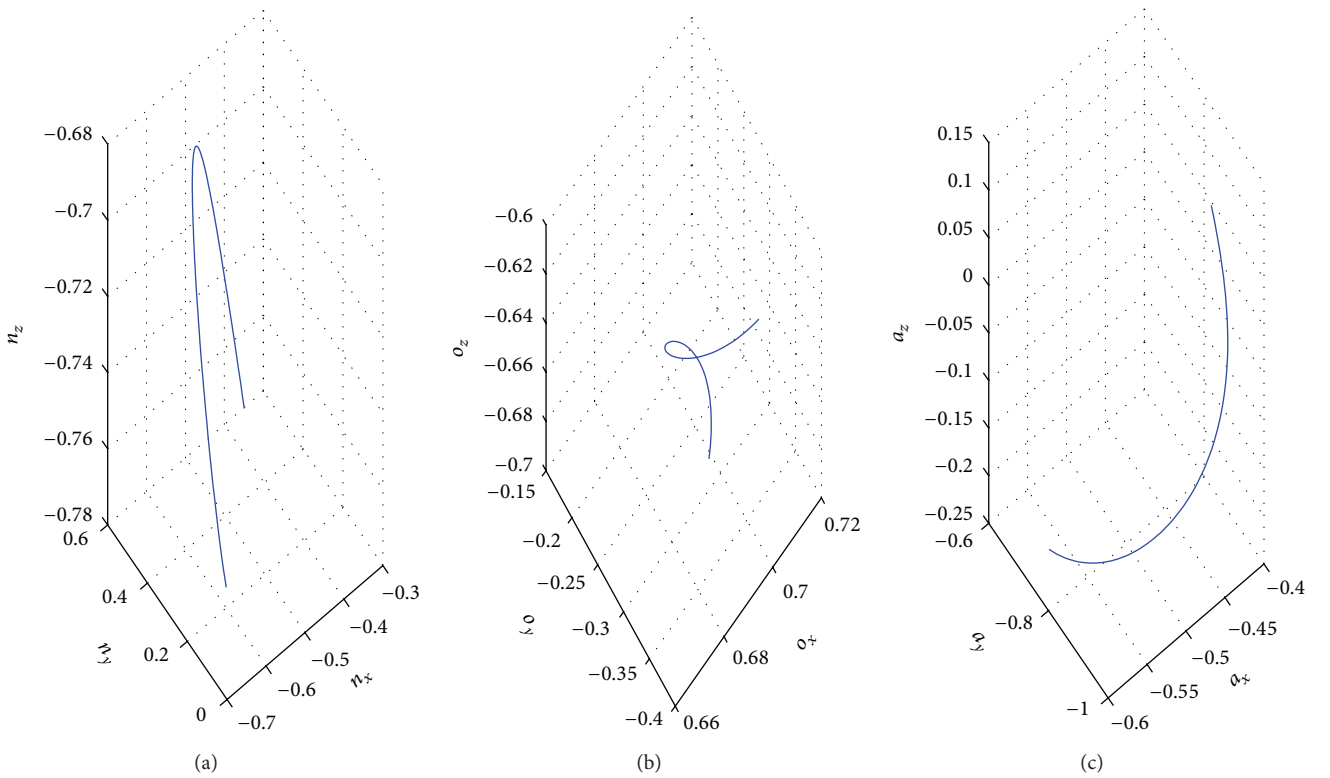


FIGURE 4: The orientation of the end-effector.

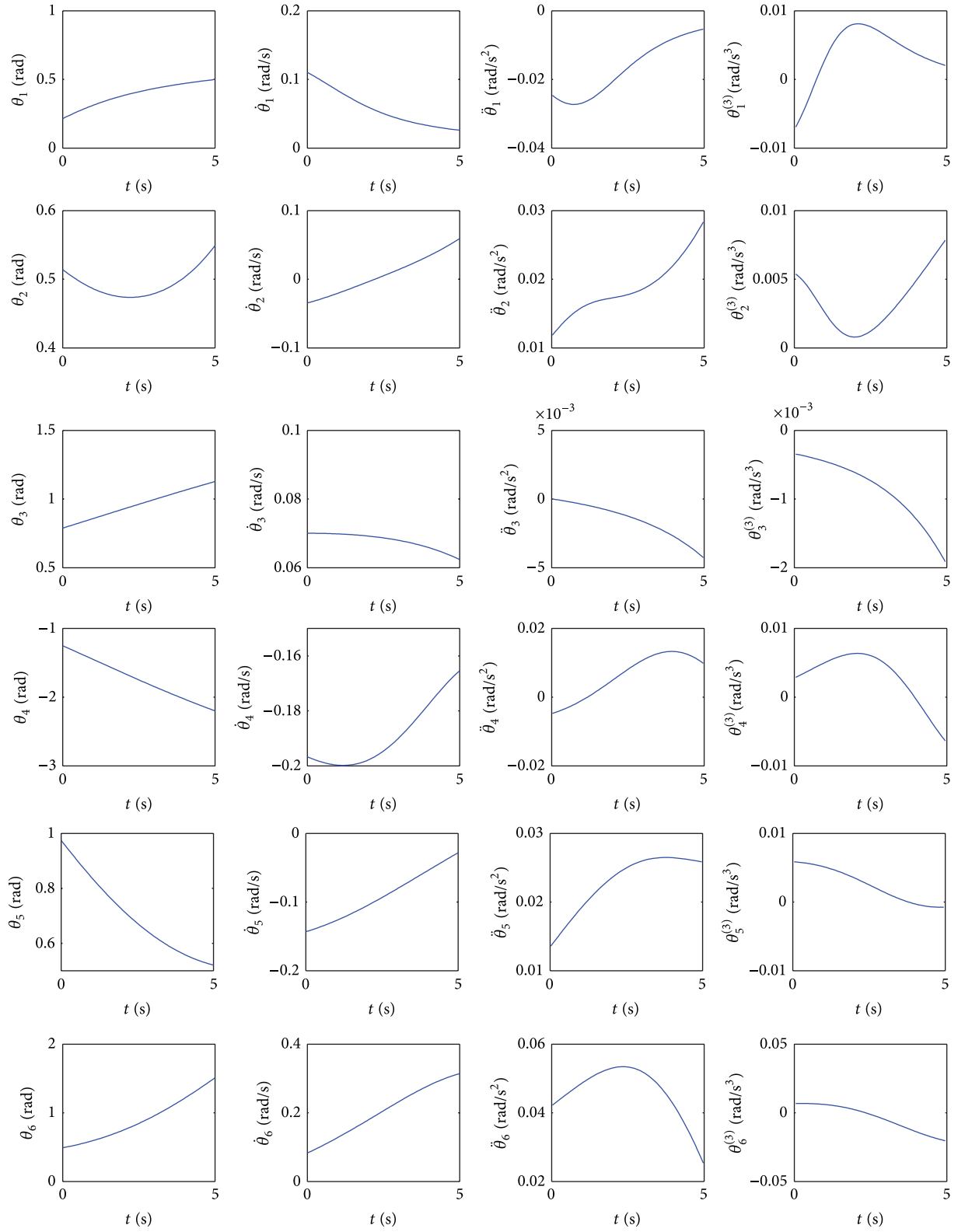


FIGURE 5: The joint trajectories of the PUMA 560 robot.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the National key Technology R&D Program of China under Grant no. 2012BAF06B01.

References

- [1] A. Olabi, R. Béarée, O. Gibaru, and M. Damak, "Feedrate planning for machining with industrial six-axis robots," *Control Engineering Practice*, vol. 18, no. 5, pp. 471–482, 2010.
- [2] K. Ning, T. Kulvicius, M. Tamosiunaite, and F. Worgotter, "Accurate position and velocity control for trajectories based on dynamic movement primitives," in *Proceedings of the IEEE International Conference Robotics and Automation*, pp. 5006–5011, May 2011.
- [3] M. Boryga and A. Graboś, "Planning of manipulator motion trajectory with higher-degree polynomials use," *Mechanism and Machine Theory*, vol. 44, no. 7, pp. 1400–1419, 2009.
- [4] A. Gasparetto and V. Zanutto, "A new method for smooth trajectory planning of robot manipulators," *Mechanism and Machine Theory*, vol. 42, no. 4, pp. 455–471, 2007.
- [5] H. Liu, X. Lai, and W. Wu, "Time-optimal and Jerk-continuous Trajectory Planning for Robot Manipulators with Kinematic Constraints," *Robotics and Computer-Integrated Manufacturing*, vol. 29, pp. 309–317, 2013.
- [6] K. Yang and S. Sukkarieh, "An analytical continuous-curvature path-smoothing algorithm," *IEEE Transactions on Robotics*, vol. 26, no. 3, pp. 561–568, 2010.
- [7] M. Žefran, V. Kumar, and C. B. Croke, "On the generation of smooth three-dimensional rigid body motions," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 4, pp. 576–589, 1998.
- [8] L. Zhang, C. Zhou, and D. Wang, "Manipulator trajectory planning using geodesic method, robotic welding," in *Intelligence and Automation*, vol. 362 of *Lecture Notes in Control and Information Sciences*, pp. 517–528, 2007.
- [9] G. Rodnay and E. Rimon, "Isometric visualization of configuration spaces of two-degrees-of-freedom mechanisms," *Mechanism and Machine Theory*, vol. 36, no. 4, pp. 523–545, 2001.
- [10] K. Tchoń and I. Duleba, "On inverting singular kinematics and geodesic trajectory generation for robot manipulators," *Journal of Intelligent & Robotic Systems*, vol. 8, no. 3, pp. 325–359, 1993.
- [11] J. M. Selig and A. I. Ovseevitch, "Manipulating robots along helical trajectories," *Robotica*, vol. 14, no. 3, pp. 261–267, 1996.
- [12] Z. Liandong and Z. Changjiu, "Robot optimal trajectory planning based on geodesics," in *Proceedings of the IEEE International Conference on Control and Automation (ICCA '07)*, pp. 2433–2436, Guangzhou, China, June 2007.
- [13] F. W. Warner, *Foundations of Differentiable Manifolds and Lie Groups*, Chapter 1, Springer, New York, NY, USA, 1983.
- [14] S. Gallot, D. Hulin, and J. Lafontaine, *Riemannian Geometry*, Chapter 1, Springer, Berlin, Germany, 2004.
- [15] B. Chow, P. Lu, and L. Ni, *Hamilton's Ricci Flow Graduate Studies in Mathematics*, vol. 77, Chapter 1, Science Publishers, 2006.
- [16] M. Spivak, *A Comprehensive Introduction to Differential Geometry*, vol. 1, Chapter 3, Chapter 9, Publish or Perish Press, Berkeley, Calif, USA, 1979.
- [17] J. J. Craig, *Introduction To Robots: Mechanics and Control*, Chapter 3, Pearson/Prentice Hall, 3rd edition, 2004.
- [18] U. M. Ascher, R. M. M. Mattheij, and R. D. Russell, *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*, SIAM, Philadelphia, Pa, USA, 1995.
- [19] D. J. Higham and H. J. Higham, *Matlab Guide*, SIAM, Philadelphia, Pa, USA, 2005.
- [20] L. F. Shampine, M. Reichelt, and J. Kierzenka, "Ordinary Differential Equations in Matlab withbvp4c," http://www.mathworks.com/matlabcentral/answers/uploaded_files/1113/bvp_paper.pdf.
- [21] N. Hale and D. R. Moore, "A sixth-order extension to the MATLAB package bvp4c of J. Kierzenka and L. Shampine," Tech. Rep., Oxford University Computing Laboratory, 2008.