Robotics and Computer-Integrated Manufacturing 29 (2013) 309–317

Contents lists available at SciVerse ScienceDirect

Robotics and Computer-Integrated Manufacturing

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

Time-optimal and jerk-continuous trajectory planning for robot manipulators with kinematic constraints

Huashan Liu [a,*], Xiaobo Lai [b], Wenxiang Wu [c]

[a] College of Information Science and Technology, Donghua University, Shanghai 201620, China
[b] College of Information Technology, Zhejiang Chinese Medical University, Hangzhou 310053,China
[c] State Key Laboratory of Fluid Power Transmission and Control, Zhejiang University, Hangzhou 310027, China

ARTICLE INFO

Article history:
Received 8 August 2011
Received in revised form
12 August 2012
Accepted 30 August 2012
Available online 2 October 2012

Keywords:
Robot
Trajectory planning
Spline
B-spline
Time optimization

ABSTRACT

In this paper a high smooth trajectory planning method is presented to improve the practical performance of tracking control for robot manipulators. The strategy is designed as a combination of the planning with multi-degree splines in Cartesian space and multi-degree B-splines in joint space. Following implementation, under the premise of precisely passing the via-points required, the cubic spline is used in Cartesian space planning to make either the velocities or the accelerations at the initial and ending moments controllable for the end effector. While the septuple B-spline is applied in joint space planning to make the velocities, accelerations and jerks bounded and continuous, with the initial and ending values of them configurable. In the meantime, minimum-time optimization problem is also discussed. Experimental results show that, the proposed approach is an effective solution to trajectory planning, with ensuring a both smooth and efficiency tracking performance with fluent movement for the robot manipulators.

1. Introduction

As known, the trajectory planning in Cartesian space (operating space or task space) [1–4] is intuitive and easy to observe the motion trail and attitude of the end effector of the robot, especially when there are obstacles to avoid in the operating space. However, such method usually fails to sidestep the problems caused by kinematic singularities. The second method is to carry out the trajectory planning in joint space [5–8], it provides an approach to non-singularity workspace for the robot manipulators, the joint trajectories can be obtained by means of interpolating functions which meet the imposed kinematic and dynamic constraints [9]. Also, it ensures that the end effector of the robot passes through the via-points and would be easier to adjust the trajectory for the system controller in contrast to the former method, but do not guarantee the definite path due to the non-linear relationship between the trajectories in Cartesian space and those in joint space [10,11].

With a review of the most representative interpolating functions in the trajectory planning, polynomials are widely adopted [12]. Generally, when higher accuracy is required for the interpolation, higher degree polynomials will be applied, which probably causes Runge's phenomenon and unstableness of convergence [13]. To eliminate this negative factor, piecewise polynomials are involved in many practical situations. Piecewise line, as a special function of this kind, gains a more satisfactory convergence property, however, it results in non-differentiable points at some internal knots. Therefore, the piecewise polynomials should be at least twice differentiable to guarantee the continuity at every internal knot. Splines, as a class of special functions defined piecewise by multi-order polynomials, are popular curves in tackling interpolating problems with the simplicity of construction, accuracy of evaluation and capacity to approximate complex shapes [14–16]. They are often preferred to polynomials because it yields similar results, even when using low-degree polynomials, while avoiding Runge's phenomenon for higher degrees.

Actually, the motion control system of the robot manipulators acts on the joints, so the smoothness of the joint trajectories would be more important than that of the Cartesian trajectory of the end effector. Aiming to create smooth enough joint trajectories passing through all the internal knots, some typical works with applying different kinds of curves in the planning algorithm are contributed, but most of them fail to obtain satisfactory local support property (if one knot of the curve changes, it only effects the local trajectory besides the knot, without re-computing the entire trajectory) of the trajectories [15–20]. B-splines, characterized by good local support, are invoked in some literatures to obtain both local support property and satisfactory smoothness of the trajectories. Specially, Thompson and Patel [21] developed a planning method for constructing joint trajectories by using B-splines, but it aimed at approximating rather than interpolating

* Corresponding author. Fax: +86 21 67792353.
E-mail address: watson683@163.com (H. Liu).

0736-5845/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.rcim.2012.08.002

the desired sequences of the discrete joint trajectories. Saravanan, Ramabalan, and Balamurugan [22] presented an evolutional theory based methodology for optimal trajectory planning using uniform cubic B-splines, where the robot joint accelerations and jerks of the resulting trajectories can be restricted within their limiting values. Gasparetto and Zanotto [9] applied quintic B-spline in the interpolation to generate smooth joint trajectories, the proposed method enables one to impose kinematic constraints, expressed as upper bounds on the absolute values of velocity, acceleration and jerk of the robot joint. By the same authors [19,20], an objective function composed of two terms (one is proportional to the execution time and the other is proportional to the integral of the squared jerk) is minimized during the cubic splines and quintic B-splines based planning to get the optimal trajectory. It's worth mentioning that, to consider the kinematic constraints in the on-line trajectory generation problem [23], Kröger discussed cases of constant [24] and non-constant [25] kinematic motion constraints imposed on the robots in depth.

High jerk of the robot joint can heavily excite the resonance frequencies of the body structure, creating vibrations, and slow down the tracking speed, as well as affect the tracking precision, so keeping the absolute value of jerk in a relative small bounded area is vital. Furthermore, if the continuity of the jerk is guaranteed, the flexible impact created by the joint actuator will be small. With respect to the smoothing techniques that could be found in previous related studies, the method described in this work ensures that the trajectory in Cartesian space is twice continuous differentiable, while in the joint space, the velocity, acceleration and jerk are all continuous and bounded. Moreover, the initial and the ending value of the velocity, acceleration and jerk of each robot joint can be configured almost arbitrarily as needed. Moreover, considering both smooth performance and efficiency execution, optimization for minimum-time trajectory tracking is also presented.

The rest of this paper is organized as follows. Section 2 details the trajectory planning method in Cartesian space by splines. In Section 3, the general formula of multi-degree B-splines is discussed, based on it, the trajectory planning in joint space is presented. In Section 4, minimum-time optimization is presented. Then, experimental results on a PUMA 560 structured 6 DOF serial robot with revolute joints are shown in Section 5. Finally, some conclusions are given in Section 6.

2. Trajectory planning in Cartesian space

Given the discrete sequence of interpolation samples: $w_i = f(t_i)$, $i=0, 1, \ldots, n$, where $0 \le a = t_0 < t_1 < t_2 < \cdots < t_n = b$. If there exists $s(t)$ with n piecewise polynomials as

$$s(t) = \begin{cases} s_1(t) & t \in [t_0, t_1] \\ s_2(t) & t \in [t_1, t_2] \\ \vdots & \\ s_n(t) & t \in [t_{n-1}, t_n] \end{cases}$$

$$s_j(t) = u_{k,\ j} t^k + u_{k-1,\ j} t^{k-1} \ldots + u_{1,\ j} t + u_{0,j} \quad j = 1,2,\ldots,n \tag{1}$$

where $u_{k,\ j}, u_{k-1,\ j}, \ldots, u_{0,\ j}$ are constant coefficients, k is the degree of the polynomial $s_j(t)$, that satisfies

(i) the interpolating property, $s(t_i) = w_i = f(t_i)$;
(ii) the curves to join up, $s_j(t_j) = s_j + 1(t_j)$;
(iii) $(k-1)$ times continuous differentiable, $s_j'(t_j) = s_{j+1}'(t_j)$, $s_j''(t_j) = s_{j+1}''(t_j), \ldots, s_j^{(k-1)}(t_j) = s_{j+1}^{(k-1)}(t_j)$.

Then, $s(t)$ is a spline interpolating function of degree k for the discrete sequence $w_i = f(t_i)$ [26].

In mathematics, a spline is a special function defined piecewise by polynomials. In engineering applications, spline interpolation is often preferred to polynomial interpolation because the interpolation error can be made small even when using low-degree polynomials for the spline. As a positive result of this feature, spline interpolation avoids the problem of Runge's phenomenon which occurs when using high degree polynomials.

Aiming to enhance the smooth movement performance of the robot manipulators, we invoke cubic splines in the interpolation to obtain a twice continuous differentiable trajectory in Cartesian space, i.e., the acceleration along the coordinate x (or y, z) for time t varies continuously. Accordingly, the expression of function $s_j(t)$ can be written as

$$s_j(t) = u_{3,j} t^3 + u_{2,j} t^2 + u_{1,j} t + u_{0,j} \quad j = 1, 2, \ldots, n \tag{2}$$

Noting that $s_j''(t)$is a first-order polynomial on the closed interval $[t_{j-1}, t_j]$, we suppose the value of $s_j''(t)$at the ends of this interval are known: $s''(t_{j-1}) = M_{j-1}, s''(t_j) = M_j$, then

$$s_j''(t) = \frac{(t_j - t)M_{j-1} + (t - t_{j-1})M_j}{h_j} \tag{3}$$

where $h_j = t_j - t_{j-1}$.

By calculating the integrals of (3), we obtain the general expression for the cubic spline at any time t in $[t_{j-1}, t_j]$:

$$s_j(t) = \frac{(t_j-t)^3 M_{j-1} + (t-t_{j-1})^3 M_j + (t_j-t)(6w_{j-1} - M_{j-1}h_j^2) + (t-t_{j-1})(6w_j - M_j h_j^2)}{6h_j} \tag{4}$$

Eq. (4) indicates that there are $(n+1)$ unknown variables $(M_0, M_1, \ldots, M_n)$ for $s(t)$, to get the complete expression of the spline, we need to construct $(n+1)$ independent equations for $M_0, M_1, \ldots, M_n$.

Specially, using condition (iii), we get

$$\mu_j M_{j-1} + 2M_j + \lambda_j M_{j+1} = \gamma_j \quad j = 1,2,\ldots,n-1 \tag{5}$$

where $\mu_j = h_j/(h_j + h_{j+1})$, $\lambda_j = 1 - \mu_j$, $\gamma_j = 6[(w_{j+1} - w_j)/h_{j+1} - (w_j - w_{j-1})/h_j]/(h_j + h_{j+1})$.

For the n cubic polynomials comprising $s(t)$, there are $(n-1)$ interior knots, giving us $(n-1)$ equations in the form of (5). To solve the $(n+1)$ unknown variables, we still require two other constraint conditions, which can be imposed upon the problem for different reasons.

Case 1. $s'(t_0) = w_n'$, $s'(t_n) = w_n'$.

The significance of these two constraint conditions on strategy of motion control in Cartesian space is: we can set the initial and ending value of velocity along the desired trajectory according to the actual needs. Generally, we set $w_0' = w_n' = 0$.

By such restrictions, we get two equations as

$$2M_0 + M_1 = \gamma_0,\ M_{n-1} + 2M_n = \gamma_n \tag{6}$$

where $\gamma_0 = 6[(w_1 - w_0) - h_1 w_0']/h_1^2$, $\gamma_n = 6[h_n w_n' - (w_n - w_{n-1})]/h_n^2$.

From (5) and (6), we can obtain the $(n+1)$ order linear equations as

$$\begin{bmatrix} 2 & 1 & & & & \\ \mu_1 & 2 & \lambda_1 & & & \\ & \mu_2 & 2 & \lambda_2 & & \\ & & \ddots & \ddots & & \\ & & & \mu_{n-1} & 2 & \lambda_{n-1} \\ & & & & 1 & 2 \end{bmatrix} \begin{bmatrix} M_0 \\ M_1 \\ M_2 \\ \vdots \\ M_{n-1} \\ M_n \end{bmatrix} = \begin{bmatrix} \gamma_0 \\ \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_{n-1} \\ \gamma_n \end{bmatrix} \tag{7}$$

where μ_i, λ_i and $\gamma_i (i = 0, 1, \ldots, n)$ are the known quantities as referred in (5) and (6), M_i are the unknown variables.