논문 2008-45SC-4-9

매끄럽지 않게 연결된 두 곡선에 대해 제한된 곡률로 부드럽게 연결할 수 있는 천이 궤적 생성 방법

(A Path-level Smooth Transition Method with Curvature Bound between Non-smoothly Connected Paths)

최윤종*, 박부견**

(Yun Jong Choi and Poo Gyeon Park)

요 약

연속적인 경로 사이를 부드러운 곡선으로 잇기 위해서 기존의 로봇 제어기들은 일반적으로 연속적인 경로를 시간 축에서 합성하는 방법을 사용해 왔다. 하지만 이런 방법은 다음과 같은 두 가지 단점을 내재하고 있다. 천이 경로의 형태가 연접하게 생성될 수 없다는 점과 천이하는 동안 속력을 제어할 수 없다는 점이 그것이다. 이러한 문제점들을 극복하기 위해서 본 논문은 매끄럽지 않게 연결된 두 경로들을 부드럽게 잇기 위해 곡률이 제한된 새로운 천이 궤적 생성 방법을 제시하고자 한다. 실험 결과는 기존의 방법들보다 천이 궤적이 더 부드럽게 생성되는 것을 보여주며, 또한 보장된 곡률의 제한 수준은 0.02 ~ 1 임을 보여준다.

Abstract

For a smooth transition between consecutive paths, conventional robot controllers usually generate a transition trajectory by blending consecutive paths in a time coordinate. However, this has two inherent drawbacks: the shape of a transition path cannot be designed coherently and the speed during transition is uncontrollable. To overcome these problems, this paper provides a path-level, rather than trajectory-level, smooth transition method with the curvature bound between non-smoothly connected paths. The experiment results show that the resultant transition trajectory is more smoothly connected than the conventional methods and the curvature is closely limited to the desired bound within the guaranteed level (0.02 ~ 1).

Keywords : curvature bound, parametric interpolation, transition path, trajectory blending, smoothly connected paths

I. Introduction

To improve the performance of a robot, it is preferred to reduce the task cycle without degrading the quality of the task. Since a general robotic task consists of a sequence of motions that may not be smoothly connected, generating a smooth transition trajectory has been an important subject.

In conventional manipulators, a tool path is usually described with a line segment or a circular-arc. Thus, discontinuities in velocity and acceleration may generally exist during switching of motions. If no transition is applied, the switching of consecutive motions generally entails the jumps in velocity and acceleration at the switching instant. For a smooth transition, several methods have been proposed, which perform in time coordinates. One of the methods is to connect trajectories of consecutive paths with a smooth polynomial that spans an

* 학생회원, ** 평생회원-교신저자, 포항공과대학교 (POSTECH)

※ 본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음 (IITA-2008-C1090-0801-0004)

접수일자: 2007년11월1일, 수정완료일: 2008년7월7일

interval, the so-called *transition window*[1]. This method is intuitive and easily implementable but is very difficult to control the spatial shape of a transition trajectory. In other methods[2~4], a transition trajectory has been generated by blending, where a velocity profile or a position profile of the transition trajectory is made by blending the original ones. With this method, the spatial shape of a transition trajectory can be controlled to some degree by changing the parameters of a blending function. However, since the spatial shape is intricately involved not only with the blending function but also with the original paths, it is still difficult to obtain a delicate shape of a transition path. Moreover, since blending of trajectories is performed in a *time-coordinate*, an undesired speed change occurs during the transition period. Thus, these methods are unacceptable for tasks that require a precise control of speed, for example, carrying a vessel filled with some fluid. Recently, the researches[5~6] have presented a motion planning primitive to be used for the iterative steering of vision-based autonomous vehicles by a quintic spline. However, the vehicle model and the path planning are handled only on the two-dimensional space, not on the higher-dimensional space, where the robot systems are usually working.

To overcome these drawbacks, this paper proposes a new curvature-bounded smooth transition method, which handles these in the *path level*, where a time-law is ignored and only the geometric information is considered. Since a trajectory can be decomposed into a path and a speed profile, where the former describes a pure geometric information of a motion and the latter details how fast a manipulator moves along the path, it is natural that imposing geometrical constraints upon a trajectory is more effectively conducted in the path level. Moreover, since acceleration is proportional to the curvature of a path while the manipulator moves at a constant speed, a constant speed of a transition trajectory can be achieved with a bounded acceleration by independently generating a transition path and a speed profile once the curvature of the transition path is bounded to a limit. Since research on a speed profile generation has been conducted and many of useful results are found in the literature [7~10], this paper does not address the problem of speed profile generation but focuses on the generation of a transition path.

The proposed method generates a quintic spline transition path whose curvature is nearly bounded to a desired value, which is performed through the followings. First, a temporary quintic spline transition path is generated by computing an appropriate transition window size and tangential vectors at the boundary of the window so that the generated path resembles a circular arc with the desired curvature. Next, the transition window is adjusted for satisfying the desired curvature limit, and thus the transition path is completed by recomputing the coefficient of the quintic spline with the adjusted transition window. The simulation results show that the proposed method generates much smoother transition trajectory than the conventional blending method while keeping the desired curvature bound.

The paper is organized as follows. Section II discusses the preliminaries and the previous works. Section III explains the new method, *curvature-bounded smooth transition trajectory*, in details. Section IV presents some experimental results. Finally, section V concludes the paper with summarization.

II. The previous works

1. Preliminaries : Path and trajectory

Recently developed industrial robot controllers support a variety of parametric interpolation to directly handle a parametric path without segmenting a desired path into a number of linear blocks. Parametric expressions of paths have been well established and are used in most of modern CAD systems for an accurate description of a geometrical shape. A superior performance of parametric interpolations is backed up with the report that the accuracy and efficiency of machining can be improved through parametric interpolation[11~13].

We explain the difference of a path and a trajectory by quoting their definitions[14]. A *path* denotes the locus of points, which the manipulator has to follow in the execution of the assigned motion; a path is then a pure geometric description of motion. On the other hand, a *trajectory* is a path on which a time law is specified, for instance in terms of velocities and/or accelerations at each point. A *parametric path* is a path expressed in a parametric equation, which may include expressions for orientations as well as positions. For example, a line segment form P_0 to P_1 can be described in the following expression.

$$P(u) = P_0 + (P_1 - P_0)u, \text{ where } u \in [0,1]. \tag{1}$$

Some examples of parametric paths extensively used in industrial robot systems and CAD systems are B-splines, NURBS's (Non-Uniform Rational B-Splines), and cubic splines.

A trajectory of a manipulator moving along a given parametric path $P(u) \in C^2$ can be computed with the chain-rule. If we denote the position, the velocity, and the acceleration as $p(t), v(t)$ and $a(t)$, respectively, then the trajectory can be expressed as follows:

$$\begin{aligned} p(t) &= P(u(t)), \\ v(t) &= P'(u(t))\dot{u}(t), \\ a(t) &= P''(u(t))\dot{u}^2(t) + P'(u(t))\ddot{u}(t), \end{aligned} \tag{2}$$

where

$$P'(u(t)) = \frac{\partial P(u(t))}{\partial u(t)},\ P''(u(t)) = \frac{\partial^2 P(u(t))}{\partial u^2(t)},$$
$$\dot{u}(t) = \frac{du(t)}{dt},\ \ddot{u}(t) = \frac{d^2u(t)}{d^2t}$$

Using the time-derivatives of the parameter $u(t)$ in the previous work[15],

$$\begin{aligned} \dot{u}(t) &= \frac{s_p(t)}{|P'(u(t))|}, \\ \ddot{u}(t) &= \frac{\dot{s}_p(t)}{|P'(u(t))|} - s_p^2(t)\frac{P'(u(t)) \cdot P''(u(t))}{|P'(u(t))|^4} \end{aligned} \tag{9}$$

then, we have

$$\begin{aligned} p(t) &= P(u(t)), \\ v(t) &= s_p(t)\frac{P'(u(t))}{|P'(u(t))|}, \\ a(t) &= \dot{s}_p(t)\mathrm{T}(t) + s_p^2(t)\kappa(t)\mathrm{N}(t), \end{aligned} \tag{4}$$

where $s_p(t)$ is the speed, $\kappa(t)$ is the curvature, $\mathrm{T}(t)$ is unit tangent vector, and $\mathrm{N}(t)$ is unit normal vector.

$$\begin{aligned} \mathrm{T}(t) &= P'(u(t))/|P'(u(t))|, \\ \mathrm{N}(t) &= \Lambda(t)/|\Lambda(t)|, \end{aligned} \tag{5}$$

where

$$\begin{aligned} \Lambda(t) &= P''(u(t)) - \mathrm{T}(t)(\mathrm{T}(t) \cdot P''(u(t))), \\ \kappa(t) &= \frac{|P'(u(t)) \times P''(u(t))|}{|P'(u(t))|^3}. \end{aligned}$$

Calculating $p(t), v(t)$ and $a(t)$ requires the value of parameter u at a specified time t. The relation between time and parameter can be expressed as an integral equation with the traveling length $L(t)$ as follows:

$$L(t) = \int_0^t s_p(r)dr = \int_{u(0)}^{u(t)} |P'(u)|du. \tag{6}$$

Unfortunately, $u(t)$ for the given $L(t)$ can be numerically computed but not explicitly obtained, for the general parametric curve $P(u)$ except for special curves such as straight lines and circular arcs, that is, the exact computation of arc-length is impossible to achieve a general parametric curve. The conventional interpolation algorithms resolve this through real-time numerical integration of the parameter or by treatment of the curve as the sequence of small blocks with a known length-to-parameter relation. In our paper, we resolve this with the sampled-parametric interpolation (SPI) method[17].

2. Conventional techniques for a transition trajectory generation

As shown in Fig.1 suppose that a manipulator is moving along a particular path x_1 and at time t_s, it switches to a second path x_2. To ensure a smooth movement over the two paths without stopping, an

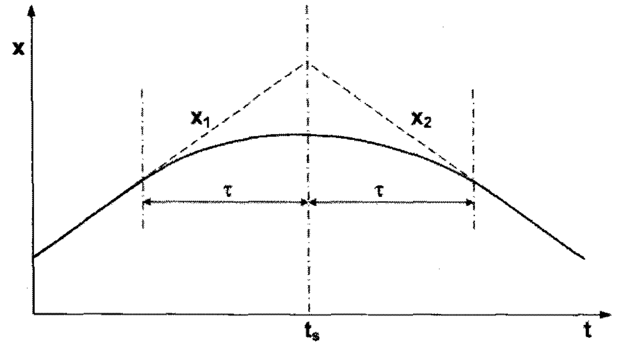


그림 1. 경로 천이 윈도우. 점선은 원래의 선을 실선은 천이 윈도우를 사용하여 재구성된 최종 연결곡선을 나타냄

Fig. 1. Illustration of a transition trajectory in one dimension. The dashed lines indicate original paths and the solid curve indicates the final connected curve.

appropriate transition trajectory is required. If no transition is applied, then the switching between paths at t_s will generally create a discontinuity in acceleration and velocity. Several techniques have been proposed for handling this problem, which can be classified into two groups. We present a simple review for a better understanding of our results. The following convention will be used in this section.

$$\begin{aligned} v_a &= v(t_s-\tau), \\ v_b &= v(t_s+\tau), \\ s &= \frac{t-t_s+\tau}{2\tau}, \end{aligned} \tag{7}$$

where v_a, v_b and s are the velocity of starting and ending a transition, and the normalized time at the specified transition trajectory.

2.1. Approaches based on a connecting polynomial

This approach is to connect the trajectories of the consecutive paths with a smooth polynomial that spans an interval $t \in [t_s-\tau, t_s+\tau]$, which is often called a *transition window*[1]. This involves determination of an appropriate transition time 2τ and formulation of the connecting polynomial, for which quintic polynomials are often used. A simple way to estimate the necessary transition time is to divide the magnitude of velocity change by some appropriately chosen reference acceleration a_r

$$2\tau = \frac{|v_b - v_a|}{a_r}. \tag{8}$$

If the transition of orientation is also considered, then one may compute separate transition times for the translational and rotational components and then take the maximum. The six coefficients of a quintic polynomial can be obtained from the six boundary conditions on the position, the velocity, and the acceleration at $t_s-\tau$ and $t_s+\tau$.

2.2. Approaches based on a blending function

Although the above method is easily implementable, it is very difficult to control the spatial shape of a transition trajectory. Some methods based on a blending function enable more flexibility in the shaping of transition trajectories[2~4], where transition trajectories are generated by blending velocity profiles or position profiles. For a concise formulation, we use a normalized time coordinate s rather than t and then the transition occurs during the interval $s \in [0,1]$. By employing a normalized blending function $f'(s) \in [0,1]$, the position, the velocity, and the acceleration profiles are obtained as follows:

$$\begin{aligned} p(s) &= p(0) + v_a 2\tau s + (v_b - v_a)2\tau f(s), \\ v(s) &= v_a + (v_b - v_a) f'(s), \\ a(s) &= (v_b - v_a)\frac{df'(s)}{ds}\frac{1}{2\tau}. \end{aligned} \tag{9}$$

If the maximum allowed acceleration is specified, then the transition time may be determined using the following equation.

$$2\tau = \frac{(v_b - v_a)}{|a|_{\max}} \left. \frac{df'(s)}{ds} \right|_{s=\frac{1}{2}}. \tag{10}$$

There are several simple choices available for blending functions and examples of these are $f'(s) = s$ for linear blending, $f'(s) = -2s^3 + 3s^2$ for cubic polynomial blending, and $f'(s) = \sin^2(\frac{\pi}{2}s)$ for cycloidal blending. These blending methods will be compared to our proposed method in section Ⅳ(Experimental results). Herein, it is verified that the proposed method makes a smoother transition trajectory than any other blending function methods.

Remark 1: A more sophisticated blending method is available in [4], where the control of the spatial shape of a transition trajectory is more flexible. However, the coherence of the resultant spatial shapes cannot be guaranteed because the spatial shape is intricately involved not only with blending functions and its parameters but also with the crossing situation of the original paths. In addition, since blending of trajectories is performed in a time coordinate, an undesired speed change can occur during transition period.

Ⅲ. Main results: Path-level smooth transition

As explained in the previous section, since conventional methods handle the transition trajectory problem in the trajectory level where a time-law is involved, the undesired change of speed is almost inevitable and the spatial shape of the transition trajectory is hardly controllable. Thus, the conventional methods are not acceptable for the tasks that require precise control of speed, for example, carrying a vessel filled with some fluid.

This problem can be effectively solved if the time-law is ignored. If the transition problem is handled in the path level rather than the trajectory level, then the loci of transition trajectory can be more easily shaped to the desirable geometric curve. In addition, an appropriate speed profile of the transition trajectory can be generated independently of a transition path when the curvature of the path is under a limit.

Throughout the paper, we assume that the translational motion is dominant and thus the rotational motion will be trackable once the translational motion is trackable. Handling of the case where the rotational motion is dominant, requires another measure for smoothness of the rotational path, as well as a curvature bound for that of the translational path. Therefore, this case is not considered in our paper, and will be studied in future works.

1. Problem formulation

Suppose that a manipulator is required to move across a pair of non-smoothly connected parametric paths $P_1(u)$ and $P_2(u)$ without a speed change, where the range of parameter u is $[0,\bar{u}]$ and $\bar{u}$ is a constant value. For a smooth transition, one has to design a smooth transition path to avoid discontinuity in velocity and acceleration at the switching point between the two paths. In addition, the shorter length of the transition path is more suitable because it is for reducing task cycles of the manipulator.

We denote such a transition path as $Q(s)$ for $s \in [0,\bar{s}]$ and also denote the parameter of $P_1(u)$ at the beginning point of transition as u_b and the parameter of $P_2(u)$ at the ending point of transition as u_e. Without loss of generality, let us assume that the manipulator moves at the unit speed $s_p(t)=1$. An optimization problem is obtained for the construction of a transition path $Q(s)$ that satisfies the prescribed constraints.

Problem 1: Minimize the arc-length $\int_0^{\bar{s}}|Q'(s)|\,ds$

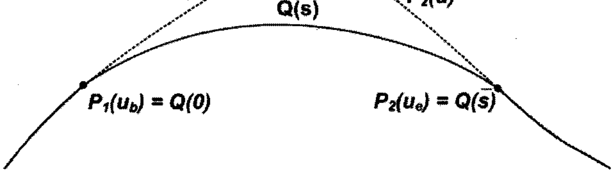


그림 2. 천이 경로 $Q(s)$

Fig. 2. Illustration of a transition path $Q(s)$, which is depicted as a blue solid line. The dashed curves indicate non-smoothly connected original paths $P_1(u)$, $P_2(u)$ and the solid curves indicate actual paths.

under the curvature constraint

$$\kappa_M \geq \max_{0 \leq s \leq \bar{s}} \frac{|Q'(s) \times Q''(s)|}{|Q'(s)|^3}, \tag{11}$$

and the boundary conditions

$$\begin{aligned} &Q(0) = P_1(u_b), \quad Q(\bar{s}) = P_2(u_e), \\ &Q'(0) = \frac{P_1'(u_b)}{|P_1'(u_b)|}, \ Q'(\bar{s}) = \frac{P_2'(u_e)}{|P_2'(u_e)|}, \\ &A(Q, 0) = A(P_1, u_b), \ A(Q, \bar{s}) = A(P_2, u_e), \end{aligned} \tag{12}$$

where

$$A(P, u) = \frac{P''(u)}{|P'(u)|^2} - \frac{P'(u)(P'(u) \cdot P''(u))}{|P'(u)|^4}, \tag{13}$$

since, substitution of (3) into (2) yields

$$\begin{aligned} a(t) &= A(P, u) \\ &= P''(u) \frac{s_p(t)^2}{|P'(u)|^2} \ ) \\ &+ P'(u) \left(\frac{\dot{s}_p(t)}{|P'(u)|} - s_p^2(t) \frac{P'(u) \cdot P''(u)}{|P'(u)|^4} \right), \end{aligned} \tag{14}$$

and $s_p(t) = 1$, $\dot{s}_p(t) = 0$.

2. The proposed path-level smooth transition method with the curvature bound

Problem 1 is so complicated that it cannot be solved analytically because (a) no structure on $Q(s)$ is given, (b) both the computations for the arc-length and the maximum curvature of a general parametric path $Q(s)$ are not analytic, and finally (c) no analytic formula for the determination of u_b and u_e can be found due to the previous reasons. Thus, we apply a heuristic approach for this problem.

For (a), we suppose $Q(s)$ to be a quintic spline. This is because it is the simplest polynomial that can satisfy six of the boundary conditions.

$$Q(s) = As^5 + Bs^4 + Cs^3 + Ds^2 + Es + F, \tag{15}$$

where $s \in [0, \bar{s}]$, and $\bar{s}$ is a constant value.

The decision of u_b and u_e is a quite complex problem because it involves the computation of the arc-length of $Q(s)$. Thus for (b) and (c), it needs to be assumed that the arc-length from the beginning point of transition, $P_1(u_b)$, to the crossing point, $P_1(\bar{u})$, between P_1 and P_2 is the same as the arc-length from the crossing point, $P_2(0)$, to the ending point of transition, $P_2(u_e)$. The arc-length is denoted as λ, which will correspond to a half of a transition window size τ in a conventional technique. By minimizing the transition window size 2λ instead of the arc-length, we can obtain u_b and u_e while avoiding a heavy computation for arc-length calculation.

However, it is still not easy to solve the optimization problem of λ and the boundary conditions of $Q(s)$. The proposed method solves these problems by fitting $Q(s)$ to a circular arc with the desired curvature κ_M. The decision procedure consists of three steps with the following assumptions.

Assumption:

A1. The transition path $Q(s)$ is a quintic spline defined by (15).

A2. The arc-length from $P_1(u_b)$ to $P_1(\bar{u})$ is the same as the arc-length from $P_2(0)$ to $P_2(u_e)$.

A3. $P_1(u)$ and $P_2(u)$ are nearly straight lines and $Q(s)$ can fit into a circular arc of radius $1/\kappa_M$ in the vicinity of the crossing point.

Step1: Choose initial u_b and u_e

In this step, we shall calculate the initial u_b and u_e with λ, which can be computed as the distance from the crossing point to the point of contact between $P_1(u)$ and $Q(s)$ as follows, which is depicted in Fig. 3.

$$\lambda = R/\tan\left(\frac{1}{2}\arccos\left(\frac{P_1'(\bar{u}) \cdot P_2'(0)}{|P_1'(\bar{u})||P_2'(0)|}\right)\right) \tag{16}$$

The initial u_b and u_e can be computed using the algorithm introduced in [17] with λ obtained by the above equation.

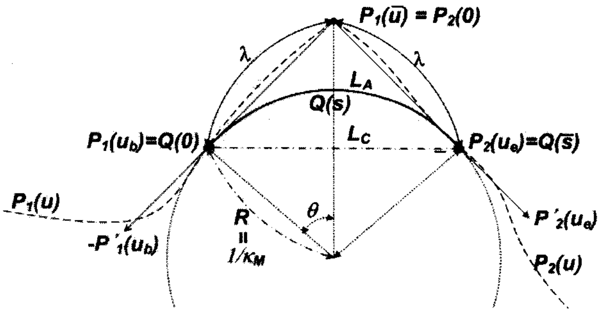


그림 3. 부드러운 천이 궤적 $Q(s)$
Fig. 3. The smooth transition trajectory $Q(s)$.

Step2 : Find $Q(s)$

This step determines the parameter range of $Q(s)$ with the boundary conditions and computes the coefficients of $Q(s)$. The range of parameter and the magnitude of the first derivatives at the boundary points are computed from the chordal length between the two boundary points with the assumption 3, which osculate at the boundary points. In Fig. 3, we can obtain L_A as the arc-length of $Q(s)$, which can be approximated with a Taylor series approximation as follows:

$$\begin{aligned} L_A &= \frac{2\alpha\theta}{\kappa_M} = \alpha\theta\frac{L_c}{\sin(\theta)}, \\ &\simeq \alpha L_c\left(1.0 + \frac{1}{6}\theta^2 + \frac{7}{360}\theta^4 + \frac{31}{15120}\theta^6\right), \end{aligned} \tag{17}$$

where

$$L_c = |P_1(u_b) - P_2(u_e)|, \tag{18}$$

$$\theta = \frac{1}{2}\left(\pi - \arccos\left(\frac{-P_1'(u_b) \bullet P_2'(u_e)}{|P_1'(u_b)||P_2'(u_e)|}\right)\right), \tag{19}$$

and α is an adjustment parameter for $Q(s)$ not to violate the curvature constraint. Users may choose an acceptable value of α from $[1.0, 1.3]$.

For $Q(s)$ under the assumption 3, the parameter range $\bar{s}$ is 2θ, which is simply $L_A\kappa_M/\alpha$ by (17). Therefore, we can find $Q(s)$ with the following boundary conditions, which can be determined as

$$\begin{aligned} &Q(0) = P_1(u_b), \quad Q(\bar{s}) = P_2(u_e), \\ &Q'(0) = \frac{\alpha}{\kappa_M}\frac{P_1'(u_b)}{|P_1'(u_b)|}, \; Q'(\bar{s}) = \frac{\alpha}{\kappa_M}\frac{P_2'(u_e)}{|P_2'(u_e)|}, \\ &Q''(0) = \frac{\alpha^2}{\kappa_M^2}A(P_1, u_b), \; Q''(\bar{s}) = \frac{\alpha^2}{\kappa_M^2}A(P_2, u_e), \end{aligned} \tag{20}$$

since

$$s_p(t) = \frac{L_A}{\bar{s}} = \frac{L_A}{L_A k_M/\alpha} = \frac{\alpha}{k_M}(constant), \dot{s}_p(t) = 0.$$

Step3 : Adjust λ

The final step scales λ up until the maximum curvature of $Q(s)$ is up to the desired curvature bound κ_M, and recalculates the coefficients of $Q(s)$ by repeating *Step2* again. As shown in Fig. 5, the maximum curvature of $Q(s)$ appears in the neighborhood at the midpoint, *i.e.* $\left[\frac{\bar{s}}{2} - \delta, \frac{\bar{s}}{2} + \delta\right]$ and δ is a small value, hence we divide the interval 2δ into small intervals, where we choose the maximum value among the results calculated in the small intervals instead of the entire region $s \in [0, \bar{s}]$ as

$$\begin{aligned} &\max_{0 \le s \le \bar{s}} \frac{|Q'(s) \times Q''(s)|}{|Q'(s)|^3} \\ &\cong \max_{\frac{\bar{s}}{2} - \delta \le s \le \frac{\bar{s}}{2} + \delta} \frac{|Q'(s) \times Q''(s)|}{|Q'(s)|^3} \\ &= K_{\max} \le k_{M,} \end{aligned} \tag{21}$$

where δ is a user-defined parameter, which is chosen as $\bar{s}/10$ in our paper.

The curvature of a circle is inversely proportional to the radius, which is proportional to λ in our case and we introduce a new value λ^* instead of λ to compensate a non-zero curvature at the boundary of transition with additionally concerning the curvatures of $P_1(u)$ and $P_2(u)$ at the transition boundary,

$$\lambda^* = \frac{\lambda}{\kappa_M}\left(K_{\max} + \frac{1}{2}K(P_1, u_b) + \frac{1}{2}K(P_2, u_e)\right), \tag{22}$$

where $K(P, u) = \frac{|P'(u) \times P''(u)|}{|P'(u)|^3}$, namely, the curvature of P at u. If $K_{\max} \cong \kappa_M$ and both $P_1(u)$ and $P_2(u)$ are nearly straight lines in the vicinity of u_b and u_e, then $\lambda^* \cong \lambda$.

Consequently, we calculate u_b^* and u_e^* corresponding to λ^* using the algorithm in [17] and thus we finally obtain the resulting smooth transition trajectory $Q^{*}(s)$ by repeating *Step2.*

Ⅳ. Experimental results

In this section, we demonstrate the performance of the proposed path-level smooth transition method with the curvature bound, which is tested on generating trajectories for a SCARA 600-arm robot. Three consecutive parametric paths, namely, two cubic splines and a line, are constructed for these tests. The control points of two splines are (X_0, X_1, X_2) at parameters $\{0, 1.5, 3.4\}$ and (X_2, X_3, X_4, X_5) at $\{0, 0.63, 1.98, 3.18\}$. The line segment spans from X_5 to X_6. The positions and roll angles of each point are

$$
\begin{aligned}
X_0 &= (200, 300, -50, 0),\\
X_1 &= (210, 450, -50, -\pi/2),\\
X_2 &= (400, 440, -50, -\pi),\\
X_3 &= (380, 380, -50, -\pi/2),\\
X_4 &= (480, 290, -50, 0),\\
X_5 &= (400, 200, -50, \pi/2),\\
X_6 &= (300, 280, -50, \pi),
\end{aligned}
$$

which are marked to square boxes in Fig. 4. All the position vectors are chosen from a planar space for a clear observation of transition path shapes.

Two experiments have been performed to verity the generation of the curvature-bounded smooth transition trajectory and to compare conventional and proposed methods.

1. The curvature-bounded smooth transition trajectory

In the first experiment, the proposed method is applied under several different situations of the desired curvature $\kappa_M = 1.0, 0.1, 0.05, 0.033$ and 0.02, where the adjustment parameter α is set to 1.05 in all these cases. The parameters for a speed profile are $V = 100mm/s$, $A = D = 1000mm/s^2$ and $J = 10000mm/s^2$, where V, A, D and J denote a speed limit, an acceleration limit, a deceleration limit and a jerk limit, respectively. The spatial loci depicted in Fig. 4 shows how smoothly a transition shape can be adjusted. Also, the curvature fluctuation depicted in Fig. 5 describes how well a curvature-bounded trajectory can be generated according to κ_M, where in Fig. 5 the case of $\kappa_M = 1$ is omitted, but its curvature still bounds to κ_M.

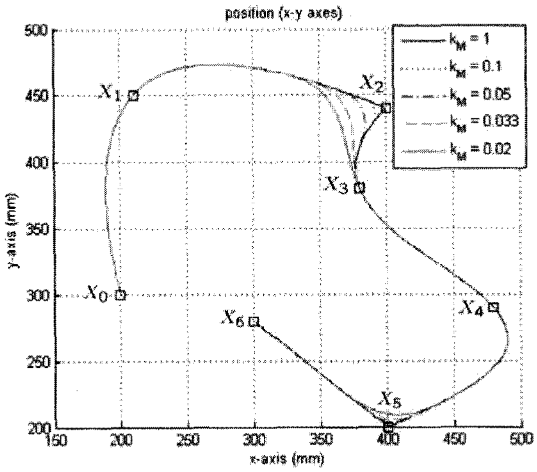


그림 4. 곡률에 따른 경로의 궤적

Fig. 4. Spatial loci of the trajectories according to curvature bounds.

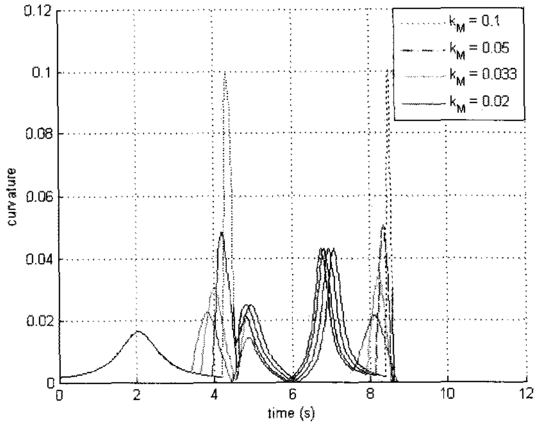


그림 5. 궤적에 따른 곡률의 변화

Fig. 5. Curvature fluctuation along the trajectories. The color (not solid block) lines depict the curvature variation in the transition paths.

Unfortunately, in all conditions, the above results are not always satisfied. Herein, we remark that the guaranteed bound of curvature for the experimenting system (SCARA) is $0.02 \sim 1.0$. If κ_M is larger than 1.0, then the interval of transition becomes very short and the SCARA robot hardly runs along the resulting trajectory. If it is smaller than 0.02, then the smooth trajectory can be made but the shape of the resultant trajectory becomes far apart from the origin circle of our proposed algorithm, *i.e.* when the assumption in our paper is broken, the bound of curvature is not guaranteed. This comes from the fact that the system performance is associated with the dynamics and the control of the system.

2. The curvature-bounded smooth transition trajectory

In the second experiment, the proposed method is compared to two conventional transition methods (t *linear and third order polynomial blending method*) [3] under the condition ($V=100mm/s$), where the adjustment parameter α and the desired curvature κ_M are set to 1.05 and 0.05.

Fig. 6, 7 and 8 depict the actual cartesian trajectories, and speed profiles at link 1 and link 2 for three methods, respectively, while tables 1 and 2 summarizes the statistics of the results of each links. As shown in the figures, the proposed method generates smoother transition trajectories than the linear and the third polynomial methods and thus the SCARA robot can follow the reference trajectory along the transition very well.

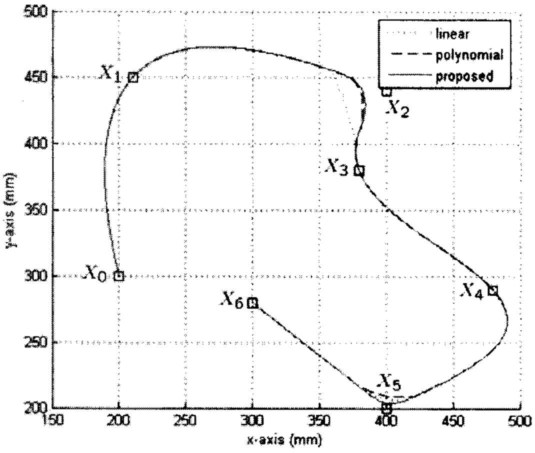


그림 6. 세 가지 방법을 각각 적용하여 얻은 경로의 궤적

Fig. 6. Spatial loci of the trajectories for three methods.

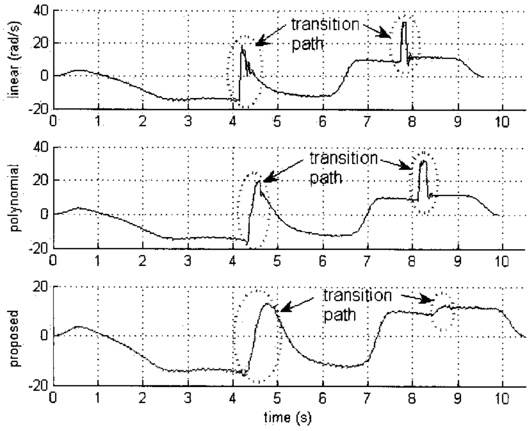


그림 7. 세가지 방법을 적용하여 얻은 링크 1 (스카라 A 축) 의 속도 프로파일

Fig. 7. Speed profiles at link 1 (SCARA A arm) for three methods. Solid and dotted lines represent the reference and the actual speed, respectively.

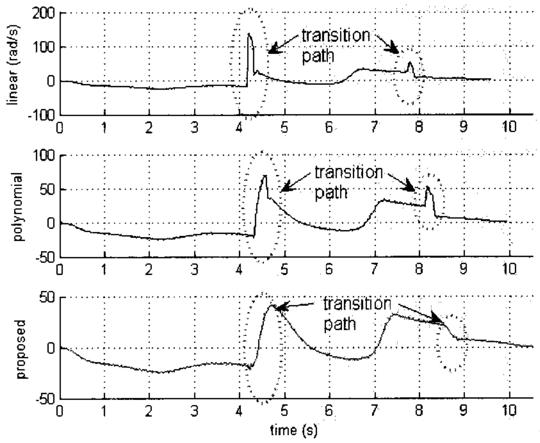


그림 8. 세가지 방법을 적용하여 얻은 링크 2 (스카라 B 축) 의 속도 프로파일

Fig. 8. Speed profiles at link 1 (SCARA B arm) for three methods. Solid and dotted lines represent the reference and the actual speed, respectively.

표 1. 링크1,2 (스카라 A,B 축) 에 대해서 세가지 방법에 대한 속도 참조 값과 실제 속도 값과의 에러 비교 (속도제한 $100mm/s$)

Table 1. The comparison of the performance for three methods at each link with the speed limit ($100mm/s$).

link	three method	speed error	
		peak (rad/s)	rms (rad/s)
link 1 (A arm)	linear polynomial proposed	5.1503 6.4848 1.2219	8.9430 8.4654 4.0480
link 2 (B arm)	linear polynomial proposed	40.3669 10.6638 3.269	45.9076 21.1175 13.8060

표 2. 링크1,2 (스카라 A,B 축) 에 대해서 세가지 방법에 대한 위치 참조 값과 실제 위치 값과의 에러 비교 (속도제한 $100mm/s$)

Table 2. The comparison of the performance for three methods at each link with the speed limit ($100mm/s$).

link	three method	position error	
		peak (rad/s)	rms (rad/s)
link 1 (A arm)	linear	0.4217	0.9416
	polynomial	0.3624	0.9900
	proposed	0.0530	0.1455
link 2 (B arm)	linear	2.4816	4.5148
	polynomial	0.8162	2.4968
	proposed	0.0706	0.3014

Remark 2: In [3], there are three blending methods : linear, third order polynomial and cycloidal. Since the performance of cycloidal method is nearly the same as that of the third order polynomial, we only compare the proposed method to the linear and the third order polynomial.

The experiments show that the transition trajectories are successfully generated and thus curvature is nearly bounded to a desired limit within the guaranteed bound. Moreover, trajectories are much more smoothly connected than the other methods. Therefore, the proposed method is expected to play a suitable safety role in motion execution of consecutive overall parametric paths.

Ⅴ. Conclusions

So far, we have considered the trajectory generation problem for smooth transition between non-smoothly connected parametric paths. Since conventional methods generate a transition trajectory in a time-coordinate, they were accompanied by two fundamental drawbacks: a slowdown of speed during transition and incoherent geometric shapes of resultant transition trajectories. To overcome these drawbacks, we presented a new path-level, rather than trajectory-level, transition method with the curvature bound. The experiments showed that the proposed method made much more smoothly connected trajectory than the other methods, where the curvature was closely bounded to a desired limit within the guaranteed bound. Therefore, it is expected that the proposed method can provide an efficient tool for fast execution of consecutive motions.

References

[1] R. P. Paul, *Robot Manipulators: Mathematics, Programming, and Control*, MIT Press, Massachusets, 1981.

[2] R.H. Taylor, "Planning and execution of straight line manipulator trajectories", *IBM J. Res. Develop.*, Vol. 23, no. 4, pp.454-436, July 1979.

[3] R. Volpe, "Task space velocity blending for real-time trajectory generation", *Proc. of IEEE Conf. on Robo. Automa.*, Vol. 2, pp. 680-687, May 1993.

[4] J. Lloyd, and V. Hayward, "Trajectory generation for sensor-driven and time-varying tasks", *Int. J. Robo. Res.*, Vol. 12, no. 4, pp. 380-393, 1993.

[5] C. Guarino Lo Bianco, and A. Piazzi, "Optimal trajectory planning with quintic G^2-splines", *Proc. IEEE Intelli. Vehi. Sym.*, pp.620-625, Oct. 2000.

[6] A. Piazzi, C. Guarino Lo Bianco, M. Bertozzi, A. Fascioli, and A. Broggi, "Quintic G2-splines for the iterative steering of vision based autonomous vehicles", *IEEE Trans. on Intelli. Trans. Sys.*, Vol. 3, no. 1, pp.27-36, March 2002.

[7] K. Erkorkmaz, and Y. Altintas, "High speed CNC system design. Part I: jerk limited trajectory generation and quintic spline interpolation", *Int. J. Mach. Tools Manuf.*, Vol. 44, no. 9, pp. 1323-1345, July 2001.

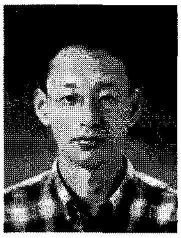
[8] S. Macfarlane, and E.A. Croft. "Jerk-bounded manipulator trajectory planning: Design for real-time applications", *IEEE Trans. on Robot. Automa.*, Vol. 19, no. 1, pp. 42-52, Feb. 2003.

[9] S.H. Nam, and M.Y. Yang. "A study on a generalized parametric interpolator with real-time jerk-limited acceleration", *Computer-Aided Design*, Vol. 36, no. 1, pp. 27-36, Jan. 2004.

[10] S.Y. Jeong, Y.J. Choi, P. Park, and S.G. Choi. "Jerk limited velocity profile generation for high speed industrial robot trajectories", *IFAC World*

Cong., Prague, Czech Rep., July 2005.
[11] M. Shpitalni, Y. Koren, and C.C. Lo, "Realtime curve interpolators", *Computer-Aided Design*, Vol. 26, no. 11, pp. 832-838, Nov. 1994.
[12] D.C.H. Yang, and T. Kong, "Parametric interpolator versus linear interpolator for precision CNC machining", *Computer-Aided Design*, Vol. 26, no. 3, pp. 225-234, March 1994.
[13] X. Zhiming, C. Jincheng, and F. Zhengjin. "Performance evaluation of a real-time interpolation algorithm for NURBS curves", *Int. J. Adv. Manuf. Tech.*, Vol. 20, no. 4, pp. 270-276, Aug. 2002.
[14] L. Sciavicco L, and B. Siciliano, *Modeling and Control of Robot Manipulators*, Springer, London, 2000.
[15] R.T. Farouki, and Y.F. Tsai, "Exact taylor series coefficients for variable-feedrate CNC curve interpolators", *Computer-Aided Design*, Vol. 33, no. 2, pp. 155-165, Feb. 2001.
[16] D.T. *Greenwood. Principles of Dynamics*, Prentice-Hall, New Jersey, 1980.
[17] S.Y. Jeong, Y.J. Choi, and P. Park, "Parametric interpolation using sampled data", *Computer-Aided Design*, Vol. 38, no. 1, pp. 39-47, Jan. 2006.

저자소개



최 윤 종(학생회원)
2001년 경북대학교 전자공학과 학사 졸업
2002년~포항공과대학교 전자전기공학과 석박사 통합과정
<주관심분야 : 로보틱스, 제어 및 신호처리>

박 부 견(평생회원)
1988년 서울대학교 제어계측공학과 학사 졸업
1990년 서울대학교 제어계측공학과 석사 졸업
1995년 Stanford University 전기공학과 박사 졸업
1996년~포항공과대학교 전자과 교수
<주관심분야 : 강인제어, 예측제어, 퍼지, SMC, 통신 및 신호처리>