INFORMATICA, 2006, Vol. 17, No. 1, 69–84

Velocity Distribution Profile for Robot Arm Motion Using Rational Frenet–Serret Curves

Reza RAVANI
Department of Computer Engineering, Science & Research Campus
Azad University, Tehran, Iran
e-mail: rezaravani@hotmail.com

Ali MEGHDARI
Department of Mechanical Engineering
Sharif University of Technology, Tehran, Iran
e-mail: meghdari@sharif.edu

Received: December 2004

Abstract. The aim of this paper is to demonstrate that the techniques of Computer Aided Geometric Design such as spatial rational curves and surfaces could be applied to Kinematics, Computer Animation and Robotics. For this purpose we represent a method which utilizes a special class of rational curves called Rational Frenet–Serret (RF) curves for robot trajectory planning. RF curves distinguished by the property that the motion of their Frenet–Serret frame is rational. We describe an algorithm for interpolation of positions by a rational Frenet–Serret motion. Further more we present an algorithm for tracking the constructed RF motion to achieve the desired velocity distribution profile of robot arm.

Key words: spatial curves, rational Frenet–Serret curves, velocity profile, Frenet–Serret frames.

1. Introduction

In recent years, it has been realized that the methods of Computer Aided Geometric Design (CAGD) provide elegant tools for various tasks in Computer Graphics, Robotics and Kinematics, especially for the design of rigid body motions. One of the first contributions to this research area was the spherical generalization of the de Casteljau algorithm introduced by Shoemake (1985) in order to interpolate the orientations of a moving object. Among others, Ge and Ravani (1991; 1994) and Park and Ravani (1995) presented methods for constructing so-called Bézier motions by generalizing the subdivision algorithm of Bernstein-Bézier curves. Furthermore, this concept has been extended to other areas such as spatial kinematics or robotics by introducing so-called rational spline motions. Rational spline motions are characterized by the property that the trajectories of the points of the moving object are rational spline curves, i.e., the trajectories are NURBS (non-uniform rational B-spline) curves (Farin, 1993).

NURBS curves and surfaces became an industrial standard (STEP) for the data exchange between CAD systems. The main advantage of this approach is the data conformity to state of the art CAD systems, which allows straightforward data transfer from CAM to CAD systems. Therefore, the programming and control of robots could be done by using of CAD data. In general, these data specify the desired trajectory of end-effector but not the orientation. In our approach by using Frenet–Serret Frame we also specify the desired orientation of end-effector to perform specific tasks such as arc welding, spray painting and scanning of surfaces with robot equipment.

Rational motions can be said to be the direct generalization of rational curves to kinematics. They are defined by the property, that the trajectories of the points of the moving object(s) are (piecewise) rational curves. Therefore we can apply the algorithms of CAGD directly to these curves.

Using this approach the design of a Cartesian motion of an end-effector is usually done by specifying a set of key control configurations which are interpolated or approximated. In interpolation, the curve passes through each control point and in approximation the curve only passes through the end-points. The other control points exert a “pull”. So the intermediate points in approximation simply have some influence on the shape of the curve.

The first who applied rational motions to motion design were Ge and Ravani (1991, 1994). Their interpolation algorithm is based on rational dual quaternion curves. Another contribution has been given by Johnstone *et al.* (1995) who used normalized rational quaternion curves in order to interpolate orientations of a moving object for animation.

In this paper we discuss a special class of rational motions called Rational Frenet–Serret (RF) (Ravani and Meghdari, 2004a; Ravani and Meghdari, 2004b) and apply this type of rational motions to robot trajectory planning. In application requiring control of the orientation of a rigid body, as its center of mass executes a given path, alignment of body’s principal axes with the Frenet–Serret frame at each point appear to be the solution. For this purpose we derive a general formula for RF curves and by using this representation the motion of end-effector in 3D space will be achieved.

Our work also analyzes the RF motion of robot arm considering the desired velocity of tool center point (TCP) and the angular velocity of end-effector which user specifies and computes the points which the motion should be accelerated or slowed down to realize the velocity profile. We show that the arc length of this motion can be approximated with sufficient accuracy in real time to achieve the desired distribution velocity. This enables us to increase the flexibility and continuity of RF motions.

This paper is organized as follows: first we are going to introduce basic notations and review some fundamentals in spatial kinematics and the theory of rational motions. In Section 3 we review the concept of Frenet–Serret Frame and investigate and attain a few properties of RF curves and derive a general representation formula for RF curves. Next in Section 4 we present the motion tracking algorithm for RF curves by computing the intermediate positions and corresponding velocities to achieve the velocity profile. In Section 5 we give a detailed procedure for application of RF curves in robot trajectory planning. Finally, we finish with concluded remarks and future works.

2. Basic Notations

In the sequel we describe the points p in 3-space with the help of homogeneous coordinates $p = (p_0, p_1, p_2, p_3)^T \in R^4 \backslash \{(0,0,0,0)^T\}$. For points not at infinity, i.e., $p_0 \neq 0$, the corresponding inhomogeneous Cartesian coordinates are $\bar{p} = (\bar{p}_1, \bar{p}_2, \bar{p}_3)^T \in R^3$ of the every point p from $\bar{p}_i = p_i/p_0$ where $i = 1, 2, 3$. The homogeneous coordinate vectors p and λp describe the same point for any constant factor $\lambda \neq 0$. By analogously we are going to use homogeneous plane coordinates P^* for the description of planes. A point P lies in the plane P^* if $< P, P^* >= 0$, where $<>$ denote the canonical scalar product in R^3 or R^4.

Consider two coinciding coordinate system in Euclidean 3D-space, the fixed coordinate system E^3 (“world coordinate”) and the moving coordinate system $\overset{\frown}{E}{}^3$. Both coordinate systems are associated with right-handed Cartesian coordinates frames. Frame is an affine extension of a basis: Requires a vector basis plus a point O (the origin): $F = (\vec{v}_1, \vec{v}_2, ..., \vec{v}_n, O)$. Points can be described in either coordinate system. We denote the fixed coordinates of a point by p or $\bar{p}$, and the moving coordinates by $\hat{p}$ or $\hat{\bar{p}}$ respectively. In order to convert moving coordinates into fixed coordinates we have to apply the coordinate transformation that maps $\hat{E}^3$ intoE^3. Using homogeneous coordinates, this transformation can be represented by a 4×4 matrix:

$$M = \left[\begin{array}{ccc|c} 0 & 0 & 0 & v_0 \\ \hline r_{1,1} & r_{1,2} & r_{1,3} & v_1 \\ r_{2,1} & r_{2,2} & r_{2,3} & v_2 \\ r_{3,1} & r_{3,2} & r_{3,3} & v_3 \end{array}\right] = \left[\begin{array}{ccc|c} 0 & 0 & 0 & v_0 \\ \hline & & & v_1 \\ & R & & v_2 \\ & & & v_3 \end{array}\right], \quad v_0 \neq 0, \tag{1}$$

where the 3×3 submatrix $R = (r_{i,j})_{i,j=1,2,3}$ satisfies the orthogonality conditions

$$R.R^T = v_0^2 I_3 \quad \text{and} \quad \det(R) > 0.$$

A continuous one-parameter set of positions of $\overset{\frown}{E}{}^3$ defines a motion $M = M(t)$ where the parameter t is assumed to be the time. If all element functions of $M = M(t)$ are polynomials of degree n, the motion M is said to be a rational motion of degree n. In this case, all points trajectories $p(t) = M(t).\ \overset{\frown}{p}$ are rational curves of general degree n. Applying dual quaternion calculus one can prove that every rational motion of degree n can be written

$$M = \left[\begin{array}{ccc|c} 0 & 0 & 0 & \bar{v}_0(d_0^2 + d_1^2 + d_2^2 + d_3^2) \\ \hline & & & v_1 \\ & \bar{v}_0 D & & v_2 \\ & & & v_3 \end{array}\right] \tag{2}$$

with the dual quaternion $D = (d_0, d)^T$ and

$$D = \begin{pmatrix} d_0^2 + d_1^2 & -2d_0d_3 & 2d_0d_2 \\ -d_2^2 - d_3^2 & +2d_1d_2 & +2d_1d_3 \\ 2d_0d_3 & d_0^2 - d_1^2 & -2d_0d_1 \\ +2d_1d_2 & +d_2^2 - d_3^2 & +2d_2d_3 \\ -2d_0d_2 & 2d_0d_1 & d_0^2 - d_1^2 \\ +2d_1d_3 & +2d_2d_3 & -d_2^2 + d_3^2 \end{pmatrix}. \tag{3}$$

Here $\bar{v}_0, v = (v_1, v_2, v_3)^T$ and D are polynomials of degree $n - 2k$, n and k, respectively, where $0 \leqslant 2k \leqslant n$. The four parameters $d_0, \ldots, d_3$ are Euler's parameters of the rotational part D of the motion (Bottema, 1979).

The design of rational motions is done most efficiently by computing polynomials $\bar{v}_0, v, D$ that match certain prescribed constraints. The B-Spline representation of M can be obtained by inserting them into (2). In this case the motion M is expressed by the help of Bernstein polynomials of degree n:

$$M(t) = \sum_{i=0}^{n} B_i^n(t) A_i, \tag{4}$$

where with Bernstein polynomials $B_i^n(t) = \binom{n}{i}(1-t)^{n-i}t^i$ and A_i are the control points.

Considering $\vec{v}(t) = (1/v_0(t))(v_1(t)v_2(t)v_3(t))^T$ as the translational vector of the trajectory $M(t)$, the velocity of TCP with respect to t results from

$$\vec{v}(t) = -v_0/v_0^2 + \vec{v}'/v_0 \tag{5}$$

the prime denotes the derivative with respect to parameter t.

Note that Eq. 3 is the classical representation of rotation matrix $R(t)$ with help of Euler parameters. As an abbreviation, the four Euler parameters of the rotation matrix $R(t)$ are collected in four-dimensional Euler Vector:

$$\tilde{d}(t) = \big(d_0(t)d_1(t)d_2(t)d_3(t)\big)^T. \tag{6}$$

The first derivative $\tilde{d}'(t) = (d_0'(t)d_1'(t)d_2'(t)d_3'(t))^T$ of this vector with respect to t called the Euler velocity of the motion. The corresponding angular velocity of the end-effector is

$$\vec{\omega}(t) = (2/\tilde{d}^T\tilde{d})(\vec{d} \times \vec{d}' - d_0'\vec{d} + d_0\vec{d}') \tag{7}$$

with $\vec{d}(t) = (d_1(t)d_2(t)d_3(t))^T$.

In Section 4 we will apply a reparameterization $t = t(\tau)$ of the motion in order to realize the desired distribution of the velocity for RF motions.

3. Rational Frenet–Serret Motion

We discuss a special class of rational motions for which the moving object stays fixed with respect to the Frenet–Serret frame of a given curve. We present the concept of Frenet–Serret frame in CAGD and derive the general representation formula for RF curves. We finish this section by implementation the algorithm with output results.

3.1. *Frenet–Serret Frame*

At each point of a regular space curve, the Frenet–Serret frame defines an orthonormal basis of vectors in R^3 aligned with the local intrinsic curve geometry (see Fig. 1). The elements of this basis are the curve tangent vector, normal vector and binormal vector. Without loss of generality we may assume that describers the point path of the origin of the moving coordinate system. Let us further assume that is a twisted curve without inflection points and with proper parameterization, i.e., $||\dot{c} \times \ddot{c}|| \neq 0$, $||\dot{c}|| \neq 0$ under these assumptions, the motion of the Frenet–Serret frame is given by

$$F = \left(\begin{array}{ccc|c} 0 & 0 & 0 & 1 \\ \hline t & n & b & c \end{array}\right) \tag{8}$$

with

$$t = \frac{\dot{c}}{||\dot{c}||}, \quad n = \frac{(\dot{c} \times \ddot{c})}{||\dot{c} \times \ddot{c}||} \times t, \quad b = t \times n. \tag{9}$$

In other word we can express the Frenet–Serret formula by

$$\begin{bmatrix} \dot{t} \\ \dot{n} \\ \dot{b} \end{bmatrix} = \begin{bmatrix} 0 & \kappa & 0 \\ -\kappa & 0 & \tau \\ 0 & \tau & 0 \end{bmatrix} \begin{bmatrix} t \\ n \\ b \end{bmatrix}, \tag{10}$$

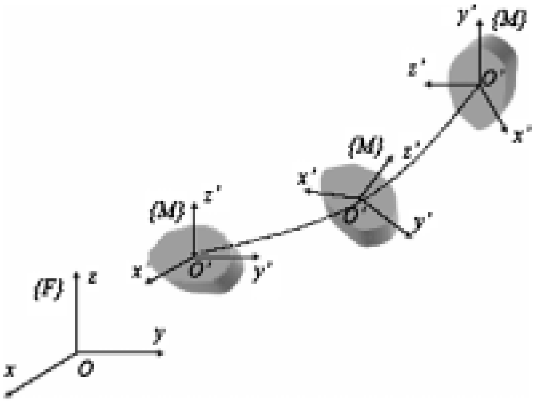


Fig. 1. The fixed frame and the moving frame attached to the rigid body.

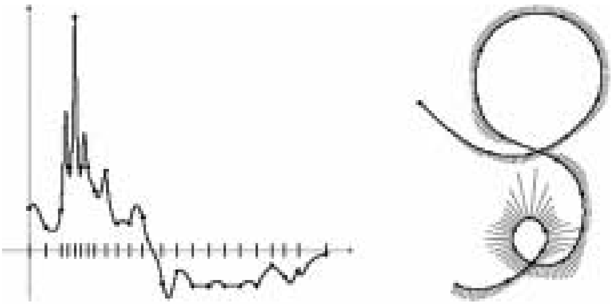


Fig. 2. The curvature plot (left) and a porcupine plot of a cubic B-spline curve (right).

where κ is curvature (see Fig. 2) and τ is torsion: a plane curve is completely determined by a single real valued function, the *curvature*, and a space curve is completely determined by two real valued functions, the *curvature* and *torsion*.

3.2. *RF Motion Construction*

In the sequel we will call a motion of type (8) the Frenet–Serret motion of $c(t)$. Taking (8) under consideration it is obvious that the Frenet–Serret motion of a rational curve in general will not be rational. We therefore want to characterize those curves $c(t)$ whose Frenet–Serret motion (8) is rational. Without loss of generality we may assume that $c(t)$ has sufficient differentiability.

At each point of the curve with $c' \times c'' \neq 0$, the osculating, normal and rectifying planes are spanned by the pair of vectors, (n, b) and (b, t) respectively.

DEFINITION 1. A curve whose Frenet–Serret motion is rational is called a Rational Frenet–Serret curve (RF curve).

Theorem 2. *A curve $c(t)$ with nonvanishing curvature $\kappa(t)$ is an RF curve if and only if $c(t)$ is rational and has rational speed $v(t) = \|\dot{c}(t)\|$ and rational curvature $\kappa(t)$.*

Proof. The unit vector, the principal normal vector and the binormal vector of $c(t)$ satisfy the Frenet–Serret equations:

$$\frac{\mathrm{d}}{\mathrm{d}s}\dot{t} = \kappa n, \quad \frac{\mathrm{d}}{\mathrm{d}s}\dot{n} = -\kappa t + \tau b, \quad \frac{\mathrm{d}}{\mathrm{d}s}\dot{b} = -\tau n, \tag{11}$$

where s denotes the arc length of $c(t)$. Rewriting (11) with respect to arbitrary curve parameter t immediately proves the claimed conditions.

A curve with rational speed $v(t)$ is called a rational Pythagorean Hodograph (PH) curve. Such curves has been investigated in a sequence of papers by Farouki and Sakkalis citer10,r11, Farouki and Pottmann (1996) and Pottmann (1995). Obviously every RF curve is a rational PH curve. On the other hand every planar rational PH curves has rational curvature, which simply yields: A planar curve $c(t)$ is an RF curve if and only if it is a traditional PH curve. In space however, there exist rational PH curves with nonrational curvature. If κ is rational we immediately obtain that the curve defined by the unit tangent vector of $c(t)$ has to be a rational PH curve as well.

For presenting a general formula for RF curves, we consider a rational curve $c(t)$ and its tangent surface Φ which is enveloped by the set of osculating planes of $c(t)$. The curve $c(t)$ is called the line of regression of Φ. Using homogeneous plane coordinates for Φ we obtain the so-called the dual representation $c^*(t)$ of $c(t)$ (see, e.g., (Pottmann, 1994)), which reads

$$c^* = \big(- < b, c >,\ b\big)^T. \tag{12}$$

Instead of computing a general representation formula of the point set directly, we focus on the dual form (12) of an RF curve.

Theorem 3. *Let $c(t)$ be an RF curve and $c^*(t)$ is dual representation. Then there exist three pairs (a, b), (e, f) and (g, h) of polynomials such that*

$$C^* = \begin{pmatrix} g(x_0^2 + x_1^2 + x_2^2) \\ 2hx_0x_1 \\ 2hx_0x_2 \\ h(x_0^2 - x_1^2 - x_2^2) \end{pmatrix}, \qquad \begin{aligned} x_0 &= (a^2 + b^2)(\dot{a}b - a\dot{b})f^2, \\ x_1 &= \frac{1}{2}(a^2 - b^2)(\dot{e}f - e\dot{f}) - (a\dot{a} - b\dot{b})ef, \\ x_2 &= (\dot{a}b - a\dot{b})ef - ab(\dot{e}f - e\dot{f}). \end{aligned} \tag{13}$$

Proof. According to Farouki and Pottmann (1996) each planner rational PH curve with homogenous coordinates $(x_0, x_1, x_2)^T$ has a representation of the type (13). Eq. 13 finally results from $c^* = (g, hb)^T$ by applying the stereographic projection onto the unit sphere.

A parametric representation of the point set $c(t)$ can be obtained by calculating the intersection point of $c^*(t)$ and the first and second derivative plane $\dot{c}^*(t), \ddot{c}^*(t)$. This yield

$$c(t) = c^*(t) \times \dot{c}^*(t) \times \ddot{c}^*(t). \tag{14}$$

Theorem 3 together with (14) provides a straightforward method for designing RF curves. In Fig. 3 you see RF curves for designing a Cartesian motion of an end-effector.

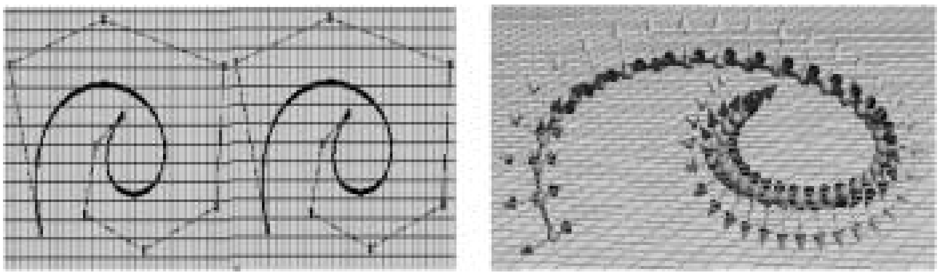


Fig. 3. Control points and rational b-spline curve (left) and equivalent RF curve with Frenet–Serret frame (right).

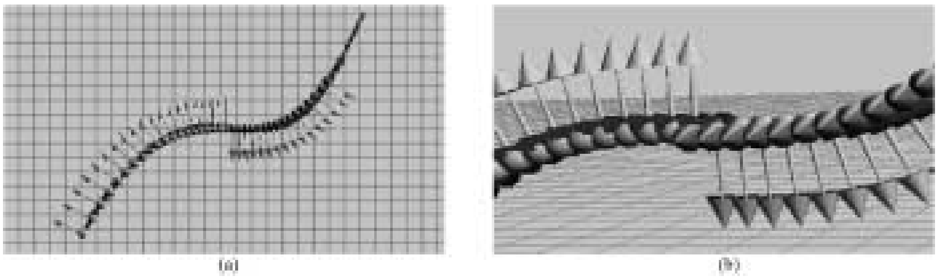


Fig. 4. (a) Control points and rational b-spline curve with inflection points (right view). (b) Perspective and zoom view in inflection point which direction of frame changes.

In Fig. 4 RF curves with some inflection points is shown. In this case the orientation of Frenet–Serret frame does not remain constant based on the curvature changes. For resolving this issue we should develop rotation minimizing frame on RF curves to minimize the angular velocity which is a different contribution of authors.

4. Tracking RF Motion

In this section by applying reparameterization of motion parameter, we will achieve the desired distribution of the velocity. For the ease of presentation we concentrate on trapezoidal velocity profile which is a piecewise constant and linear velocity distribution. This method easily can be applied to more general velocity profiles. In our scheme we do not consider robot dynamics because the limitation factor in the robot velocity is the process (such as arc welding, clueing, spray painting) and robot dynamics has minor effects and importance.

4.1. *Reparameterization of Motion*

For tracking the constructed RF motion by the end-effector of the robot, we have to generate a sequence of positions $m(t^{(k)}), m(t^{(k+1)}), \dots$ according to a certain time cycle $\Delta\tau$ for each segment of the path. We consider the segment end positions $p_1, \dots, p_N$ are specified by user. In Fig. 5 we show an example for the tracking of RF motion with 7 taught points. In our method using the motion parameter t as the time would cause an undesired distribution of velocity along the path in general. Therefore we have to apply an appropriate repararneterization $t = t(\tau)$ (which is assumed to be continuously differentiable) relating the motion parameter t to the time τ. The reparameterization function $t(\tau)$ is monotonically increasing and it satisfies $t(0) = t_0$, i.e., the start point P_1 corresponds to the time $\tau = 0$. In our discussion the velocities and accelerations with respect to the real time parameter τwill be marked by an asterisk*.

Resulting from the reparameterization, the absolute value $v^*(\tau)$ of the velocity of the TCP is

$$v^*(\tau) = \left\| \frac{\mathrm{d}}{\mathrm{d}\tau} m(t(\tau)) \right\| = \left\| m'(t(\tau)) \right\| . \dot{t}(\tau), \tag{15}$$

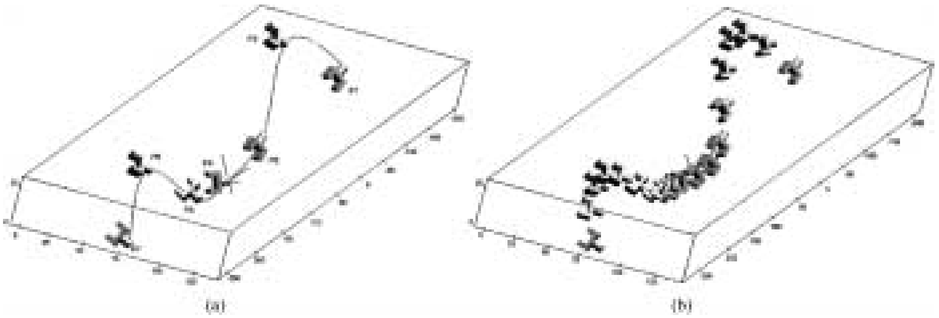


Fig. 5. (a) 7 taught positions in space, (b) constructed motion of end-effector based on the taught points.

where the dot indicates the derivative with respect to the time τ. Similarly, the absolute value $\omega^*(\tau)$ of the angular velocity of the end-effector is

$$\omega^*(\tau) = \left\|\vec{\omega}(t(\tau))\right\|.\dot{t}(\tau), \tag{16}$$

where the vector $\vec{\omega}(t)$ is defined as in the (7).

The reparameterization has to produce an almost trapezoidal velocity distribution: segments with constant speed are joined by segments with linearly varying speed distribution (e.g., Fig. 8b).

The user of the robot can either specify the desired velocity of the TCP or the desired angular velocity of the end-effector. For most segments the user should specify the velocity of the TCP because this is the more intuitive measure of the robot speed. But the velocity control by the angular velocity has to be used for motion segments with a constant TCP when only the end-effector rotates around its TCP's Frenet–Serret frame. Analogously, the speed control by the velocity of the TCP has to he used for motion segments with constant rotational part. In order to observe if the user specified speed are realistic we check one segment in advance. We do not perform global check to keep the local property of the tracking algorithm.

Consider ith segment $t_i \leqslant t(\tau) < t_{i+1}$ of the RF motion $i = 1, \ldots, N-1$. If the motion speed is controlled by the TCP, then the user specifies the desired absolute values $v^*_{i,seg} > 0$ and $v^*_{i+1,loc} \geqslant 0$ of the velocity of the TCP in the internal part of the segment and at the segment end point P_{i+1}. Otherwise, the user specifies the desired absolute values $\omega^*_{i,seg} > 0$ and $\omega^*_{i+1,loc} \geqslant 0$ of the angular velocity of the end-effector in the internal part of the segment and at the segment end point P_{i+1}, respectively. If the user does not specify a value for one of the mentioned velocities the scheme keeps the previous values.

We denote by $t^{(k)}$ the parameter value after the k th time cycle, $t^{(k)} = t(k.\Delta\tau)$. In addition to the specified velocities and angular velocities, the user should choose the maximal and minimal parameter step size $\Delta t_{\max}$ and $\Delta t_{\min}$ which are upper and lower bounds of the differences $t^{(k+1)} - t^{(k)}$ between adjacent parameter values. With smaller bounds the tracking algorithm passes more intermediate positions, so it increases the accuracy of algorithm. Moreover the user have to choose the maximal absolute values $a^*_{\max}$

and $\alpha^*_{\max}$ of the tangential components of the acceleration and of the angular acceleration. We compute the sequence of positions such that the velocities always satisfy

$$\left|\frac{\mathrm{d}}{\mathrm{d}\tau}v^*(\tau)\right| \leqslant a^*_{\max} \quad \text{and} \quad \left|\frac{\mathrm{d}}{\mathrm{d}\tau}\omega^*(\tau)\right| \leqslant \alpha^*_{\max}. \tag{17}$$

The values of $a^*_{\max}$ and $\alpha^*_{\max}$ are related to the actuator acceleration limits of the robot joints dynamics.

In order to realize the desired distribution of the velocities, we have to determine the positions where the robot motion must be accelerated or slowed down. As we consider only the tangential components of the acceleration of the TCP and of the angular acceleration of the end-effector, these positions depend only on the arc length of the trajectories. Unfortunately it is generally impossible to compute the exact arc length of a RF curve. As numerical integrations are too expensive for real-time calculations we use lower bounds for the arc length.

4.2. *Velocity Distribution Profile*

We give an outline algorithm for one time cycle of path generation. The parameter value $t^{(k)}$ and the corresponding position of the end-effector $m(t^{(k)})$ are assumed to be known. According to the specified robot speed we should compute the next parameter value $t^{(k+1)}$ and the resulting position of the end-effector $m(t^{(k+1)})$.

For this purpose first we estimate the real velocity $v^*_{k-1,real}$ of the TCP in the previous time cycle:

$$v^*_{k-1,real} = (1/\Delta\tau).\left\|m(t^{(k)}) - m(t^{(k-1)})\right\|. \tag{18}$$

For the first time cycle $(k = 0)$ we set$v^*_{-1,real} = 0$.

The desired velocity of TCP$v^*_{k,spec}$, in the next time cycle is determined based on the user-specified velocities $v^*_{i,seg}$ and $v^*_{i+1,loc}$. If the distance of $m(t^{(k)})$ to the segment end point P_{i+1} is small enough, then the motion must be accelerated or slowed down in order to realize the specified velocity $v^*_{i+1,loc}$ at P_{i+1} (Fig. 6). More precisely, if the inequality$v^*_{k-1,real} > v^*_{i+1,loc}$ and $\|m(t^{(k)}) - P_{i+1}\| \leqslant 1/2a^*_{\max}((v^*_{k-1,real})^2 - (v^*_{i+1,loc})^2)$ hold, then we should slow down the motion. Hence, we set

$$v^*_{k,spec} = \max\left\{v^*_{k-1,real} - a^*_{\max}.\Delta\tau, v^*_{i+1,loc}\right\}. \tag{19}$$

But, if the inequality $v^*_{k-1,real} < v^*_{i+1,loc}$ and $\|m(t^{(k)}) - P_{i+1}\| \leqslant 1/2a^*_{\max}((v^*_{i+1,loc})^2 - (v^*_{k-1,real})^2)$ hold, then we should accelerate the motion:

$$v^*_{k,spec} = \min\left\{v^*_{k-1,real} + a^*_{\max}.\Delta\tau, v^*_{i+1,loc}\right\}. \tag{20}$$

These computations are based on the fact that the distance $\|m(t^{(k)}) - P_{i+1}\|$ is the lower bound for the arc length of the trajectory of the TCP between the points $m(t^{(k)})$ and

$m(t_{i+1}) = P_{i+1}$ (see Fig. 6). In order to obtain tighter lower bounds for the arc length, one may inscribe a polygon to the spline curve to approximate the arc length of trajectory with sufficient accuracy.

If $v^*_{k,spec}$ neither results from (19) nor (20), then we try to realize the desired velocity $v^*_{i,seg}$ which has been specified by the user for the internal part of the segment. If $v^*_{k-1,real} > v^*_{i,seg}$ holds, then we slow down the robot motion,

$$v^*_{k,spec} = \max\left\{v^*_{k-1,real} - a^*_{\max}.\Delta\tau, v^*_{i,seg}\right\}, \tag{21}$$

otherwise we accelerate the motion,

$$v^*_{k,spec} = \min\left\{v^*_{k-1,real} + a^*_{\max}.\Delta\tau, v^*_{i,seg}\right\}. \tag{22}$$

For computing the next position first we should derive the next parameter value $t^{(k+1)}$ and based on this parameter we compute the corresponding position of the end-effector.

The parameter value $t^{(k+1)}$ should satisfy

$$\left\|m(t^{(k+1)}) - m(t^{(k)})\right\| = \Delta\tau.v^*_{k,spec}. \tag{23}$$

This non-linear equation for $t^{(k+1)}$ is solved approximately by use of Regula falsi method. Let $\Delta t^{(k)} = t^{(k+1)} - t^{(k)}$. As an initial guess for this difference we choose

$$\Delta t^{(k)}_0 = \min\left\{\Delta\tau.\frac{v^*_{k,spec}}{\|m'(t^{(k)})\|}, \Delta t_{\max}\right\}. \tag{24}$$

Then we apply one step of the Regula falsi method:

$$\Delta t^{(k)}_{j+1} = \min\left\{\frac{\Delta t^{(k)}_j.\Delta\tau.v^*_{k,spec}}{\|m(t^{(k)}) - m(t^{(k)} + \Delta t^{(k)}_j)\|}, \Delta t_{\max}\right\} \quad (j = 0, 1, 2). \tag{25}$$

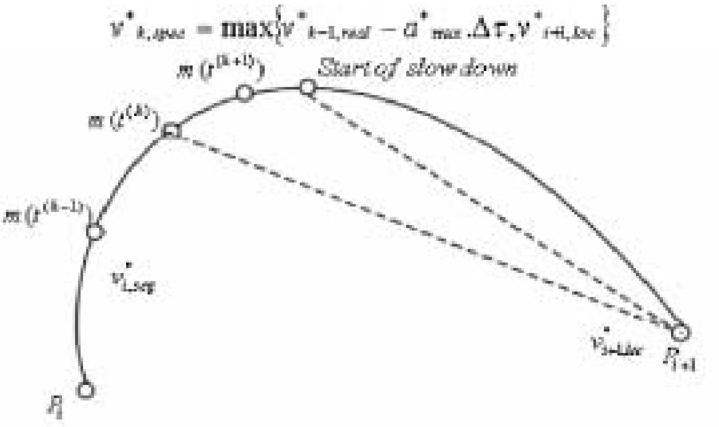


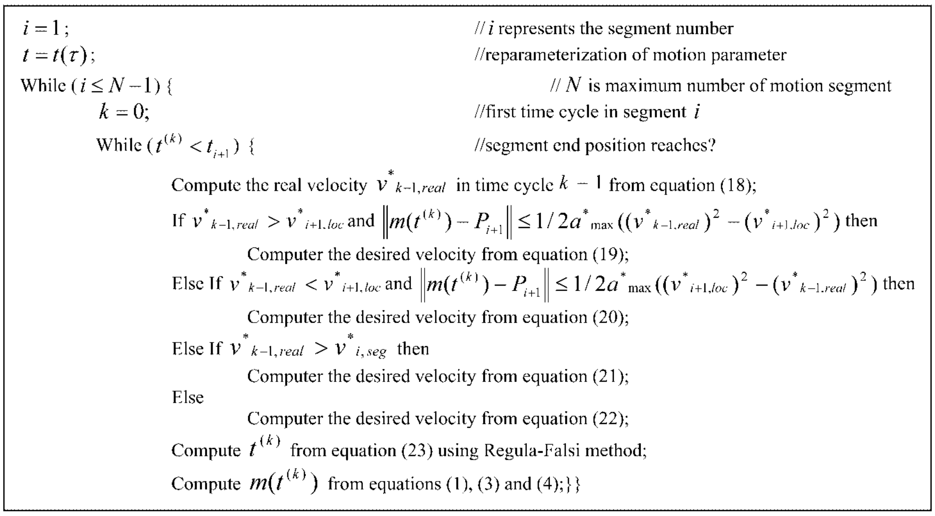
Fig. 6. Tracking on segment between P_i and P_{i+1}.

The accuracy should be sufficient for robot control; otherwise we may iterate the Regula falsi. We compute the next parameter value

$$t^{(k+1)} = \min\left\{t^{(k)} + \max\left\{\Delta t_3^{(k)}, \Delta t_{\min}\right\},\ t_N\right\}. \tag{26}$$

If $t^{(k+1)} \geqslant t_{i+1}$ holds, then we have to increment the current segment number i. The tracking of RF motion is completed for $i = N - 1$. Now we compute the position $m(t^{(k)})$ of the TCP and the corresponding rotation matrix $R(t^{(k)})$ from Eqs. 1, 3 and 4 (for more information see (Hoschek and Lasser, 1993)).

If the control of the robot speed is based on the specified angular velocities $\omega^*_{i,seg}$ and $\omega^*_{i+1,loc}$ of the end-effetor, then the next position of the end-efector results from an algorithm which is completely analogous to the previous one. Instead of the motion $m(t^{(k)})$ and its derivative $m'(t^{(k)})$ of the TCP, one has to use the normalized Euler vec-



```
$i = 1$;                                          // $i$ represents the segment number
$t = t(\tau)$;                                    //reparameterization of motion parameter
While ($i \le N - 1$) {                           // $N$ is maximum number of motion segment
        $k = 0$;                                  //first time cycle in segment $i$
        While ($t^{(k)} < t_{i+1}$) {             //segment end position reaches?
                Compute the real velocity $v^*_{k-1,real}$ in time cycle $k - 1$ from equation (18);
                If $v^*_{k-1,real} > v^*_{i+1,loc}$ and $\left\|m(t^{(k)}) - P_{i+1}\right\| \le 1/2a^*_{\max}((v^*_{k-1,real})^2 - (v^*_{i+1,loc})^2)$ then
                        Computer the desired velocity from equation (19);
                Else If $v^*_{k-1,real} < v^*_{i+1,loc}$ and $\left\|m(t^{(k)}) - P_{i+1}\right\| \le 1/2a^*_{\max}((v^*_{i+1,loc})^2 - (v^*_{k-1,real})^2)$ then
                        Computer the desired velocity from equation (20);
                Else If $v^*_{k-1,real} > v^*_{i,seg}$ then
                        Computer the desired velocity from equation (21);
                Else
                        Computer the desired velocity from equation (22);
                Compute $t^{(k)}$ from equation (23) using Regula-Falsi method;
                Compute $m(t^{(k)})$ from equations (1), (3) and (4);}}
```

Fig. 7. Pseudo code for tracking motion algorithm.

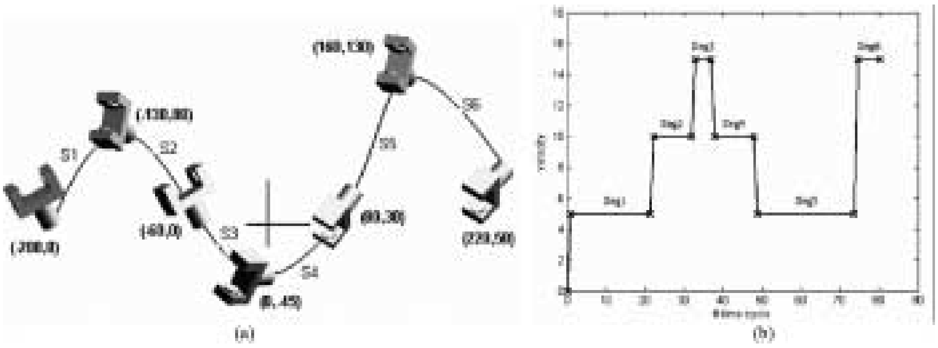


Fig. 8. (a) Motion of robot arm based on the desired positions. (b) Velocity profile in different motion segments.

tor $(\|\tilde{d}(t^{(k)})\|)^{-1}.\tilde{d}(t^{(k)})$ and its derivative with respect to$t$. Additionally, the distance $\|m(t^{(k)}) - P_{i+1}\|$ between two adjacent positions of the TCP must be replaced by the angle $2\arccos((\|\tilde{d}(t^{(k)})\|)^{-1}.\tilde{d}(t^{(k)})^T\tilde{q}_{i+1})$ of the corresponding rotation $R(t^{(k)})$.

Outline algorithm for tracking method has been presented in Fig. 7. In Fig. 8.a, seven taught positions have been drawn. These 6 motion segments should be tracked by different velocities of TCP. We show the velocity distribution profile in Fig. 8.b

5. RF Curves Application

In the following we want to develop a procedure for the automated design of rational Frenet–Serret motions that describe the Cartesian space trajectory of a robot's end- effector. We assume that the motion is implicitly described by the task level considerations, such that the trajectory of the tool center point is a given curve $c(t)$ and the direction of the z-axis of the hand coordinate system is described by a vector field $n(t)$. Furthermore we assume that n is normalized, i.e., $\|n(t)\| = 1\ \forall t$. In arc welding, for example, $c(t)$ would be the seam and $n(t)$ a vector field related to the relative positions of the electrode with respect to the bead of weld. In applications that deal with the scanning of surfaces such as in aircraft inspection on the other hand, $c(t)$ would be a known curve on the surface that has to be scanned and $n(t)$ the corresponding unit normal vector to the surface.

First we observe that $c(t)$ has to be a RF curve if we want to achieve a rational robot trajectory. It is therefore necessary to approximate the tool center point path by a RF curve $c(t)$. Then the orientation of the vectors in Frenet–Serret frame specify the exact path and direction of end-effector. For this reason we employ the following procedure:

1. Specifying a set of key control configurations which are interpolated (for end points) and approximated (for other control points or intermediate points). End points have a special concept in robot trajectory planning. It shows the start and goal positions.
2. Designing a rational Frenet–Serret curve based on (14).
3. Achieving a spatial frame (Frenet–Serret frame) for each point on a trajectory. By means of this frame the direction of tool center point of a robot in space could be

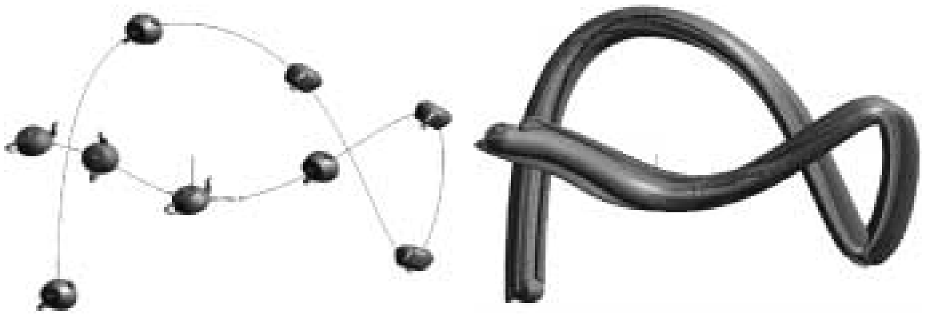


Fig. 9. Spatial rational motion of degree 9 (left); continuous motion of teapot along a designed trajectory (right).

easily determined. In fact each orthogonal vector (t, n, b) has a special meaning (subsection 3.1).

In addition to robot end effector motion design, RF curves have another important application in computer animation specially in key frame interpolation to compute a smooth motion which interpolates or approximates the given positions, such that chosen feature points of the moving system run on smooth paths (Fig. 9).

6. Conclusion and Future Works

In this paper, we have used RF curves to design robot trajectories in the Cartesian space and have obtained the velocity profile for robot arm motion considering user requirements for desired velocity and positions. Furthermore, the paper showed in various examples that RF curves can be applied successfully to any design algorithm that is based on a spine curve. We presented this method have several advantages compared to the traditional methods including: ease of programming (less teach points are necessary), flexible robot motion design considering rational frames and faster optimization of robot program.

We finish this paper by pointing to some topics for further research:

- Obstacle avoidance in trajectory planning with RF curves.
- Taking the optimization of RF motions with robot dynamics into account to minimize time or energy functions.
- Advanced CAD/CAM interfaces. By applying the rational motion techniques it is possible to use more sophisticated geometric models.

References

Bottema, R. (1979). *Theoretical Kinematics*. North Holland Publishing Company, Amsterdam, New York, Oxford.

Farin, G. (1993). *Curves and Surfaces for Computer Aided Geometric Design*, 3ed edition. Academic Press, Boston.

Farin, G., J. Hoschek and M.-S. Kim (2002). Pythagorean-hodograph curves. In *Handbook of Computer Aided Geometric Design*, North Holland. pp. 405–427.

Farouki, R.T., and T. Sakkalis (1994). Pythagorean-hodograph space curves. *Advanced in Computational Mathematics*, **2**, 41–66.

Farouki, R.T., and H. Pottmann (1996). Polynomial and rational Pythagorean-hodograph curves reconciled. In G. Mullineux (Ed.), *The Mathematics of Surfaces*, VI. Oxford University Press. pp. 355–378.

Johnstone, J., and J. Williams (1995). Rational control of orientation for animation. In *Proc. of Graphics Interface, Canadian Human Computer Communication Society*, Toronto. pp. 179–186.

Ge, Q., and B. Ravani (1991). Computer aided geometric design of motion interpolants. In *Proc. ASME Design Automation Conference*, DE-vol. 32-2. pp. 33–41.

Ge, Q., and B. Ravani (1994). Geometric construction of Bézier motions. *ASME Journal of Mechanical Design*, **116**, 749–755.

Hoschek, J., and D. Lasser (1993). *Fundamental of Computer-Aided Geometric Design*. AK Peters, Wellesley, MA.

Park, F., and B. Ravani (1995). Bézier curve on Riemannian manifolds and lie groups with kinematics alications. *ASME Journal of Mechanical Design*, 36–40.

Pottmann, H. (1994). Applications of the dual Bezier representation of rational curves and surfaces. In P.J. Laurent *et al.* (Eds.), *Curves and Surfaces in Geometric Design*. AK Peters, Boston. pp. 377–384.

Pottmann, H. (1995). Curve design with rational Pythagorean-hodograph curves. *Advances in Computational Mathematics*, **3**, 147–170.

Ravani, R., and A. Meghdari (2004a). Robot trajectory planning using rational Frenet–Serret curves. In *7th Biennial ASME Conference Engineering Systems Design and Analysis*, ESDA2004.

Ravani, R., and A. Meghdari (2004b). Spatial rational motions based on rational Frenet–Serret curves. In *IEEE International Conference on Systems, Man and Cybernetics*, vol. 5. pp. 4456–4461.

Shoemake, (1985). Animation rotation with quaternion curves. *ACM Siggraph*, **19**, 245–254.

R. Ravani graduated as a computer engineer-hardware in 1996 from Shahid Beheshti University in Iran. After that he gained the master in 1999 from Azad University, Science and Research campus and also his PHD in robotics field in 2005. His main research interests are in robotics, kinematics and computer-aided geometric design. He specially works on the robot path planning methods by using geometric models.

A. Meghdari received his PhD in mechanical engineering from the University of New Mexico in 1987. He then joined the robotics group of the Los Alamos National Laboratory (LANL) as a research collaborator. In 1988, he joined the Sharif University of Technology in Tehran where he currently holds the rank of full-professor of mechanical engineering. He has performed extensive research in the areas of robotics dynamics, flexible manipulators kinematics/dynamics, and dynamic modeling of biomechanical systems. He is currently a professor and director of Center of Excellence in Design, Robotics&Automation, and a vice-president of Academic Affairs of the Sharif University of Technology. He is also a fellow of the American Society of Mechanical Engineers.

Racionaliųjų Frenet–Serret kreivių panaudojimas roboto rankos judesio planavimui

Reza RAVANI, Ali MEGHDARI

Šio straipsnio tikslas yra pademonstruoti, kad kompiuterinis geometrinis projektavimas panaudojant erdvines racionaliąsias kreives ir paviršius gali būti panaudotas kinematikoje, kompiuterinėje animacijoje ir robotikoje. Šiam tikslui pristatomas racionaliąsias Frenet–Serret kreives naudojantis roboto trajektorijos planavimo metodas. Šios kreivės išsiskiria tuo, kad jų Frenet–Serret rėmai yra racionalūs. Straipsnyje apibūdinamas pozicijų interpoliavimo racionaliuoju Frenet–Serret judėjimu algoritmas. Taip pat pristatomas tokio judėjimo sekimo algoritmas roboto rankos norimo judėjimo pasiskirstymo užtikrinimui.