Babol Noshirvani University of Technology

10th Iranian Conference on Manufacturing Engineering
ICME 2010
March 1st - 3rd 2010
Babol Noshirvani University of Technology

Society of Manufacturing Engineering of Iran

Development of a Pythagorean-hodograph interpolator for high speed CNC machining

Behnam Moetakef Imani *, 1 and Amirmohammad Ghandehariun[2]

1. Associate Professor, Mechanical Engineering Department, Ferdowsi University of Mashhad
2. M.Sc. Student, Mechanical Engineering Department, Ferdowsi University of Mashhad

Abstract

Methodologies for parametric interpolation for NURBS curves have been proposed in the past. However, the errors caused by the approximate nature of the NURBS interpolator were often neglected. This paper proposes an integrated look-ahead algorithm which combines the NURBS interpolation with the exact Pythagorean-hodograph (PH) interpolation to take into account geometric and interpolator approximation errors simultaneously. The algorithm consists of four different modules: a sharp corner detection module, a PH construction module, a jerk-limited module, and an interpolation module. Simulations and experiments are performed to validate the proposed algorithm. It is shown that the developed interpolator improves tracking and contour accuracies significantly compared to adaptive-feedrate and curvature-feedrate algorithms.

Keywords: Numerical Control – Parametric Interpolation – PH Curve

1. Introduction

The function of the real-time interpolator in a computer numerical control (CNC) machine is to convert prescribed tool path and feedrate data into reference points for each sampling interval of the servo system. The closed-loop position and speed control can be achieved through comparing the actual machine position, measured by encoders on motor axes, with the reference point.

Parametric curves are extensively being applied to a wide range of industries such as automotive, aerospace and dies/molds. There are different representations for parametric curves, mainly including Bezier, B-spline, cubic spline and non-uniform rational B-spline (NURBS). Among these representations, NURBS have gained wide popularity to gradually become the industry standard [1]. Modern CNC machines not only provide linear/circular interpolations, but also offer parametric interpolations. Some researchers have shown that parametric interpolations can reduce feedrate fluctuations and chord errors and shorten machining time in comparison with linear/circular interpolations [2].

Accurate feedrate performance issues become especially important in the context of high-speed machining, where one requires extreme feed acceleration/deceleration rates, and maintenance of high speeds. In addition, failure of the interpolator to properly maintain the commanded feedrate may incur tool chatter or breakage [3]. Interpolators for general B-spline/NURBS curves typically rely on Taylor series expansions to compute parameter values of successive reference points. Such schemes inevitably incur truncation errors, by omission of higher-order terms [3].

To overcome this problem, the tool path can be described in terms of the Pythagorean-hodograph (PH) curves [4]. The intrinsic algebraic structure of PH curves facilitates a closed-form reduction of the interpolation integral, yielding real-time CNC interpolator algorithms for constant or variable feedrates that are remarkably accurate, robust and flexible [5].

Many interpolation methods for parametric curves have been developed. Yeh et al. [6] developed a parametric curve interpolator using a Taylor series expansion algorithm. Farouki et al. [3] derived the exact Taylor series coefficients for variable-feedrate interpolators up to the cubic terms. Tsai et al. [5] proposed an algorithm for parametric interpolation with time-dependent feedrates along PH curves. Yeh et al. [7] proposed an adaptive-feedrate interpolation algorithm. The feedrate was adjusted based on curvatures and confined chord errors. Tsai et al. [2] proposed an integrated look-ahead dynamics-based algorithm for interpolation along NURBS curves which considered geometric and servo errors simultaneously.

In this paper, an integrated look-ahead Pythagorean hodograph interpolation algorithm for NURBS curves is proposed. The algorithm consists of a sharp corner detection module, a PH construction module, a jerk-limited module and an interpolation module to take into account chord errors, feedrate fluctuations, and interpolation errors on sharp corners simultaneously. With the proposed algorithm, a higher feedrate profile can be planned to achieve better contour accuracy and shorten machining time. Simulation and experiments are performed to validate the proposed algorithm.

2. NURBS interpolator algorithm

Suppose $C(u)$ represents a NURBS curve and is given by [8]:

$$C(u)=\frac{\sum_{i=0}^{n} N_{i,p}(u)P_iW_i}{\sum_{i=0}^{n} N_{i,p}(u)W_i} \tag{1}$$

where P_i represents the control point, W_i is the weight of P_i, $n+1$ is the number of the control points, and p is the degree of NURBS. $N_{i,p}(u)$ is the B-spline basis function, and can be calculated using the recursive formulas given as follows:

$$N_{i,0}(u)=\begin{cases}1 & u_i \le u < u_{i+1}\\ 0 & otherwise\end{cases} \tag{2}$$

$$N_{i,p}(u)=\frac{u-u_i}{u_{i+p}-u_i}N_{i,p-1}(u)+\frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}}N_{i+1,p-1}(u) \tag{3}$$

where $\{u_0,...,u_{n+p+2}\}$ represents the knot vectors and u is the interpolation parameter.

To implement NURBS interpolation, a second-order approximation interpolation algorithm is adopted here. Using the Taylor series expansion method, the curve approximation up to the second derivative is given as follows:

$$u_{k+1}=u_k+\left.\frac{du}{dt}\right|_{t=t_k}\Delta t+\left.\frac{d^2u}{dt^2}\right|_{t=t_k}\frac{(\Delta t)^2}{2!} \tag{4}$$

Substituting the derivatives into the Eq. (4), the second-order Taylor expansion can be expressed as [3]:

$$u_{k+1}=u_k+\frac{V_kT_s}{|C'(u_k)|}+\frac{T_s^2}{2}\left\{\frac{A_k}{|C'(u_k)|}-\frac{V_k^2[C'(u_k).C''(u_k)]}{|C'(u_k)|^4}\right\} \tag{5}$$

where V_k, A_k, T_s , $C'(u_k)$ and $C''(u_k)$ are the feedrate, the acceleration, the sampling time, and the first and second derivatives of the NURBS curve, respectively. Since computing $C(u_k)$, $C'(u_k)$ and $C''(u_k)$ by the De Boor algorithm can improve computational performance significantly compared to the recursive basis function [2], it is used in real-time implementation.

3. Pythagorean-hodograph interpolator algorithm

A polynomial PH curve $r(\xi) = (x(\xi), y(\xi))$ is defined by its parametric hodograph [5]:

$$x'(\xi) = u^2(\xi) - v^2(\xi) \quad y'(\xi) = 2u(\xi)v(\xi) \tag{6}$$

These forms guarantee that $x'(\xi)$ and $y'(\xi)$ are elements of a Pythagorean triple of polynomials-they satisfy

$$x'^2(\xi) + y'^2(\xi) = \sigma^2(\xi) \tag{7}$$

where

$$\sigma(\xi) = |r'(\xi)| = u^2(\xi) + v^2(\xi) = \frac{ds}{d\xi} \tag{8}$$

is the parametric speed of $r(\xi)$. A PH quintic is obtained by substituting Bernstein-form quadratic polynomials

$$\begin{aligned} u(\xi) &= u_0(1-\xi)^2 + 2u_1(1-\xi)\xi + u_2\xi^2 \\ v(\xi) &= v_0(1-\xi)^2 + 2v_1(1-\xi)\xi + v_2\xi^2 \end{aligned} \tag{9}$$

into Eq. (6) and integrating [5].
Considering a time-dependent feedrate function $V(t)$ with the indefinite integral $F(t)$ imposed on the PH curve $r(\xi)$, the interpolation equation yields to the solution of the relation

$$s(\xi) = F(t) \tag{10}$$

where $s(\xi)$ is the arc length [5]. For a sampling interval Δt, the real-time CNC interpolator must compute the parameter values $\xi_1, \xi_2, \ldots$ of reference points at times $\Delta t, 2\Delta t, \ldots$These values are roots of the polynomial equations

$$s(\xi_k) = F(k\Delta t) \quad k = 1,2,\ldots \tag{11}$$

The above equations can be solved to obtain ξ_k using a few Newton-Raphson iterations [5],

$$\xi_k^{(r+1)} = \xi_k^{(r)} - \frac{s(\xi_k^{(r)}) - F(k\Delta t)}{\sigma(\xi_k^{(r)})}; r = 0,1,\ldots \tag{12}$$

with starting approximation $\xi_k^{(0)} = \xi_{k-1}$.

4. The integrated look-ahead interpolation

4.1. System architecture

The integrated look-ahead algorithm is stated in this section. This algorithm is utilized to implement CNC control tasks. The controller consists of three main programs: a CNC interpreter, a reference point generator, and a motion controller. The CNC interpreter reads NC strings from NC files to generate and store NC blocks in the NC FIFO (First-In-First-Out) memory. The reference point generator generates consecutive reference points

based on the proposed algorithm which will take into account the chord errors, curvatures, feedrate fluctuations and jerk limits. The motion controller will then control the X-Y table based on the generated reference points.
The proposed algorithm consists of four different modules: a sharp corner detection module, a PH curve construction module, a jerk limited feedrate planning module and an interpolation module. The sharp corner detection module looks ahead NC blocks to identify sharp corners of the curve. Then the curve is divided into small segments according to sharp corners. The length of each segment is calculated and stored in the buffer. The PH curve construction module approximates the segment on the NURBS curve that is found to be a sharp corner with a PH curve. The jerk-limited module plans the feedrate profile of each segment based on constraints on chord errors, feedrate, acceleration/deceleration, and jerk. The interpolation module will then generate successive reference points using the NURBS or PH interpolation equations. Finally, the motion controller performs real-time control on the X-Y table. Details of algorithms for each module are given in the following sections.

4.2. The sharp corner detection module

The sharp corner detection module plays an important role in the look-ahead algorithm. In this study, a sharp corner is defined as the feedrate sensitive zone at which the feedrate should be reduced to maintain contour accuracy. There are two criteria in identifying sharp corners. The first criterion is that the derivative of the curvature at a sharp corner is equal to zero; it is given as:

$$\left.\frac{d\kappa(u)}{du}\right|_{u=u_k} = 0 \tag{13}$$

where $\kappa(u)$ is the curvature given as:

$$\kappa(u) = \frac{\left|\frac{dC}{du} \times \frac{d^2C}{du^2}\right|}{\left|\frac{dC}{du}\right|^3} \tag{14}$$

This criterion can be illustrated by the curve shown in Figure 1, where the five points marked as A, B, C, D, and E are identified as sharp corners.
The first criterion is not sufficient in determining sharp corners. Indeed, feedrate effects should be included in determining sharp corners [2]. Therefore, the second criterion in identifying sharp corner is that the curvature at the sharp corner zone should exceed the threshold value κ_{th}, which is defined as:

$$\kappa_{th} = \frac{A_{\max}}{V_{\max}^2} \tag{15}$$

where $A_{\max}$ is the maximum acceleration limit and $V_{\max}$ is the given feedrate in the NC block. In other words, the second criterion examines whether the centripetal accelerations of local max/min points exceed the maximum acceleration limit [2]. The region on the curve where this value is exceeded is identified as the sharp corner zone, and it should be replaced with a PH curve and the feedrate should be supervised.

4.3. PH construction module

Feedrate fluctuations caused by the approximate nature of the NURBS interpolation equation in sharp corner regions will reduce the contour accuracy and surface quality. Because of the possibility of analytic reduction of the interpolation integral for PH curves, approximating the sharp corner area on the NURBS curve with a PH curve will decrease feedrate fluctuations.

In order to approximate an area on the NURBS curve by a PH quintic, Hermite data of the segment, namely the start and end points and derivatives, should be available. Using the complex representation for a planar PH curve [9],

$$r(\xi) = x(\xi) + iy(\xi) \tag{16}$$

where ξ is a real parameter, the complex polynomial $w(\xi) = u(\xi) + iv(\xi)$ will be defined so that

$$r'(\xi) = w^2(\xi) \tag{17}$$

Considering r_0, r_1, d_0 and d_1 as complex representations of the start point, end point, start derivative and end derivative, respectively, $w(\xi)$ can then be expressed in Bernstein form [9]

$$w(\xi) = w_0(1-\xi)^2 + 2w_1(1-\xi)\xi + w_2\xi^2 \tag{18}$$

where

$$w_0 = \pm\sqrt{d_0} \tag{19}$$

$$w_2 = \pm\sqrt{d_1} \tag{20}$$

$$w_1 = \frac{1}{4}\left[-3(w_0 + w_2) \pm \sqrt{120(r_1 - r_0) - 15w_0^2 - 15w_2^2 + 10w_0w_2}\right] \tag{21}$$

Figure 2 shows PH approximation of sharp corners of the curve in Figure 1.

4.4. The jerk-limited module

The first task of the jerk-limited module is to obtain the feedrate at sharp corners. To achieve this task, the proposed algorithm combines the adaptive-feedrate interpolation scheme [7] with the curvature-based feedrate interpolation algorithm [10]. The algorithm for determining the feedrate at sharp corners is given as:

$$V_{sp}(u_k) = \min\{V_{af}(u_k), V_{cf}(u_k)\} \tag{22}$$

where $V_{af}(u_k)$ and $V_{cf}(u_k)$ are given as follows:

$$V_{af}(u_k) = \frac{2}{T_s}\sqrt{\frac{2\delta}{\kappa(u_k)} - \delta^2} \tag{23}$$

$$V_{cf}(u_k) = \frac{\kappa^*}{\kappa(u_k) + \kappa^*} V_{\max} \tag{24}$$

Here, V_{af}, V_{cf}, $V_{\max}$ are adaptive-feedrate, curvature-feedrate, and feedrate commands, respectively. δ and κ are the chord tolerance and the curvature of a NURBS curve, respectively. $\bar{\kappa}$ maintains the derivative continuity of curvature-feedrate in Eq. (24).

Having obtained the feedrate at sharp corners, the second task of the jerk-limited module is to plan the feedrate profile of each segment such that the constraints on feedrate, acceleration/deceleration, and jerk are satisfied. The bell-shape feedrate profile [11] is adopted here. The interpolation module will then use Eq. (5) and Eq. (11) to generate successive reference points.

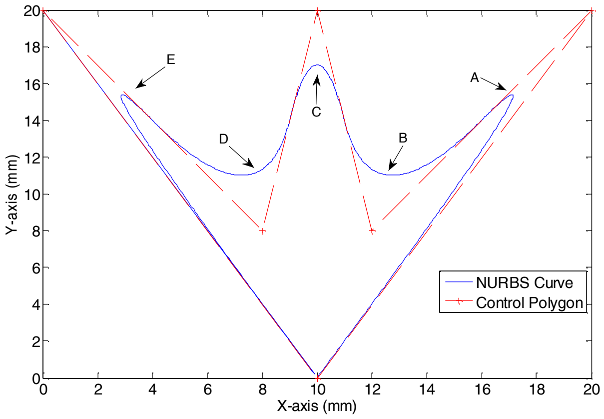


Figure 1. NURBS representation of a contour.

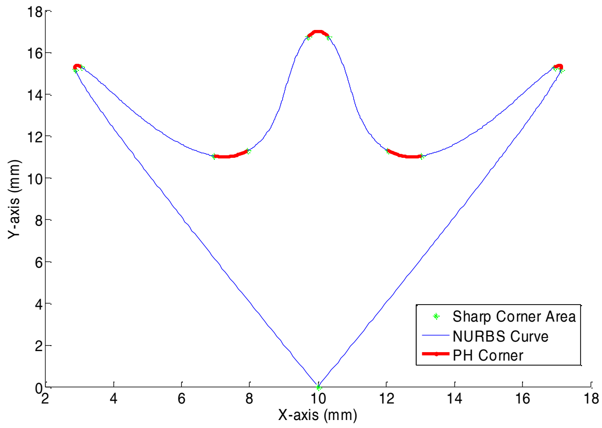


Figure 2. NURBS contour with PH approximation on sharp corners.

5. Numerical simulation and experimental verification

To evaluate the proposed interpolation algorithm, the NURBS curve shown in Figure 1 is used as benchmark. Parameters of the interpolator for numerical simulations and experiments are as listed in Table 1.

5.1. Numerical simulation

In order to simulate the errors generated in the servo control system, the block diagram shown in Figure 3 is used. The transfer function between the tracking errors on each axis and the feedrate is found to be:

$$\frac{E(s)}{V(s)} = \frac{a_4 s^4 + a_3 s^3 + (a_2 - b_2)s^2 + (a_1 - b_1)s + (a_0 - b_0)}{a_4 s^5 + a_3 s^4 + a_2 s^3 + a_1 s^2 + a_0 s} \qquad (25)$$

Parameters of Eq. (25) are listed in Table 2.Simulations are conducted to compare the performance among adaptive-feedrate [7], curvature-feedrate [10], and the proposed interpolation algorithms. Performances of the three interpolation algorithms are listed in Table 3. It is clear that the proposed algorithm reduces the contour error significantly.

5.1. Experimental results

Figure 4 shows the hardware setup, which includes a homemade X-Y table driven by two AC servomotors. The resolution of the encoders is 2500 pulses/rev. The PC interface utilized a motion control card[1] to send the command pulses and receive encoder feedback

[1] Advantech PCI-1240 Motion Control Card

signals. The interpolator was implemented using MATLAB and Visual Basic.NET programming on a PC platform.

Table 1. Parameters of the interpolator for numerical simulations and experiments [2].

Maximum acceleration	$A_{\max}$	$2450 mm/s^2$
Maximum jerk	$J_{\max}$	$5\times10^4 mm/s^3$
Chord error limitation	δ	$1\mu m$
Reference curvature	$\bar{\kappa}$	$1mm^{-1}$

Table 2. Parameters of the transfer function between the tracking errors on each axis and the feedrate

Parameter	X-axis	Y-axis
a_0	1.938×10^9	1.904×10^9
a_1	3.538×10^7	3.496×10^7
a_2	2.135×10^5	2.120×10^5
a_3	6.984×10^2	6.948×10^2
a_4	1.00	1.00
b_0	1.938×10^9	1.904×10^9
b_1	3.476×10^7	3.435×10^7
b_2	1.471×10^5	1.466×10^5

Table 3. Numerical simulation of performance comparisons among three different interpolation algorithms.

Interpolation algorithm	Tracking error (μm)				Contour error (μm)			Time (s)
	X-axis		Y-axis					
	Max	RMS	Max	RMS	Max	RMS	Mean	
Adaptive-feedrate	191.636	29.066	320.153	54.976	144.505	27.759	13.040	1.041
Curvature-feedrate	88.434	14.582	177.214	29.683	46.625	12.069	7.522	1.273
Integrated look-ahead	31.918	9.724	55.052	20.273	22.979	6.718	3.746	1.886

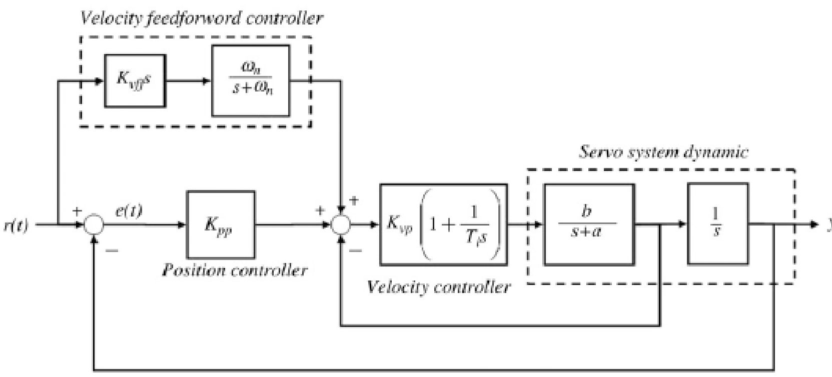


Figure 3. Block diagram of the servo control system [2].

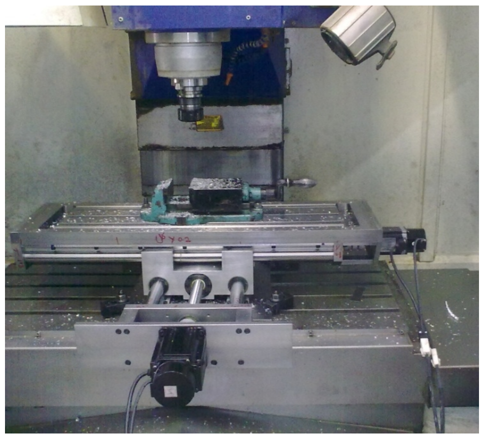


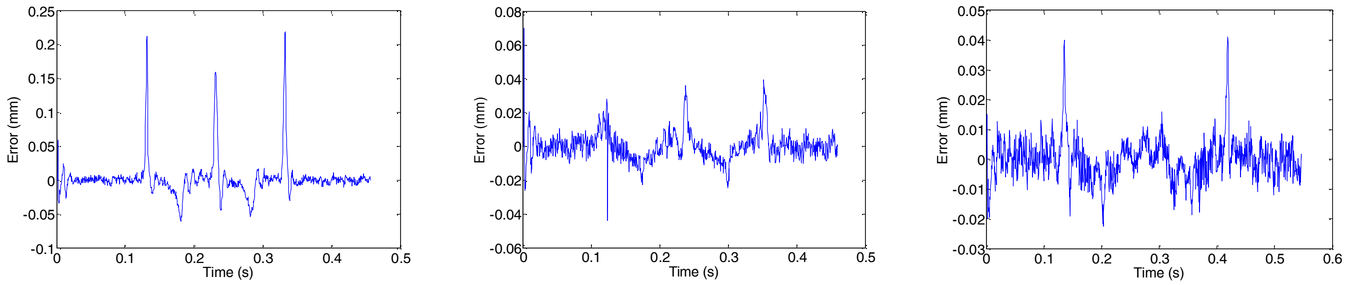
Figure 4. Hardware used in experiment.

The NURBS curve (Figure 1) is tested under feedrate command equal to 3500 mm/min. Contour error comparisons among adaptive-feedrate, curvature-feedrate and the proposed algorithms are shown in Figure 5. It is clear that the proposed algorithm can significantly improve contour accuracy, especially at sharp corners.

6. Conclusion

A novel NURBS interpolator was proposed in this paper. Sharp corners of the curve are identified not only by the derivative of the curvature but also by the feedrate criterion. A PH curve is then constructed on the corner to reduce feedrate fluctuations. The curve is divided into small segments and a smooth jerk-limited profile is planned on each segment.

Simulations were performed to validate the proposed algorithm. Experiments show that the integrated look-ahead algorithm can increase contour accuracy significantly compared with adaptive-feedrate and curvature-feedrate interpolation algorithms. This demonstrates the effectiveness of the proposed algorithm.



(a) (b) (c)

Figure 5. comparison of contour error for different algorithms (maximum feedrate 3500 mm/min). (a) Adaptive-feedrate; (b) Curvature-feedrate; (c) Integrated look-ahead.

References

1. Liu X., Ahmad F., Yamazaki K., Mori M., “Adaptive interpolation scheme for NURBS curves with the integration of machining dynamics”, “International Journal of Machine Tools and Manufacture”, Vol. 45, pp. 433-444, 2005.
2. Tsai M.S., Nien H.W., Yau H.T., “Development of an integrated look-ahead dynamics-based NURBS interpolator for high precision machinery”, “Computer-Aided Design”, Vol. 40, pp. 554-566, 2008.
3. Farouki R.T., Tsai Y.F., “Exact Taylor series coefficients for variable-feedrate CNC curve interpolators”, “Computer-Aided Design”, Vol. 33, pp. 155-165, 2001.
4. Farouki R.T., Sakkalis T., “Pythagorean hodographs”, “IBM Journal of Research and Development”, Vol. 34, pp. 736-752, 1990.
5. Tsai Y.F., Farouki R.T., Feldman, B., “Performance analysis of CNC interpolators for time-dependent feedrates along PH curves”, “Computer Aided Geometric Design”, Vol. 18, pp. 245-265, 2001.
6. Yeh S.S., Hsu P.L., “The speed-controlled interpolator for machining parametric curves”, “Computer-Aided Design”, Vol. 31, pp. 349-357, 1999.
7. Yeh S.S., Hsu P.L., “Adaptive-feedrate interpolation for parametric curves with a confined chord error”, “Computer-Aided Design”, Vol. 34, pp. 229-237, 2002.
8. Piegl L., Tiller W., The NURBS Book, 2nd Ed., New York: Springer, (1997).
9. Moon H.P., Farouki R.T., Choi H.I., “Construction and shape analysis of PH quintic Hermite interpolants”, “Computer Aided Geometric Design”, Vol. 18, pp. 93-115, 2001.
10. Zhiming X., Jincheng C., Zhengjin F., “Performance evaluation of a real-time interpolation algorithm for NURBS curves”, “International Journal of Advanced Manufacturing Technology”, Vol. 20, pp. 270-276, 2002.
11. Cheng C.W., Tsai M.C., “Real-time variable feed rate NURBS curve interpolator for CNC machining”, “International Journal of Advanced Manufacturing Technology”, Vol. 23, pp. 865-873, 2004.