A Hybrid Parametric Interpolator for Complete Motion Profile Generation

Tom Kong[1] and Daniel C. H. Yang[1*]

[1]University of California - Los Angeles
[*]Corresponding author: dyang@seas.ucla.edu

ABSTRACT

Complete motion profile for parametric curves is difficult to be planned because analytic solution of exact arc length does not exist. This paper proposes a hybrid parametric interpolation method, which takes advantage on the fact that the exact curve dimension in parametric domain is naturally available. This hybrid parametric interpolator is a combination of both speed and parameter convolutions. During accelerating and cruising operation, this hybrid parametric interpolator operates under speed convolution for motion profile generation; meanwhile, it monitors the remaining deceleration distance in parameter domain at every exampling time. When motion needs to stop or need to perform a smooth segment transition, parameter convolution technique will take over according a switch mechanism. As a result, this hybrid parametric interpolator enables the generation of entire motion profile that delivers adequate tracking speed, accurate tracking position, precise end position, smooth curve-segment transition, low motion jerks.

Keywords: hybrid parametric interpolator, complete motion profile, jerk-limited motion, parametric curves.

1. INTRODUCTION

Parametric curves such as Bezier curves, B-splines, and NURBS are widely used in modern CAD/CAM tools. All motion controllers equipped to modern computer-controlled multi-axis machines, like CNC machines, CMM machines, robots, and motion simulators need interpolators to interpolating these parametric curves. Besides, motion interpolators are also needed in many advanced applications in CAD animation for movies or video games.

The main function of an interpolator is to provide control computer with coordinate multiple-axis motion inputs to track given parametric curves, representing either position or orientation or both, at desired speed, or feedrate. A modern interpolator is an interpolation methodology and algorithm that is converted into a set of computer code. A good interpolation algorithm should provide the motion control with complete coordinated motion profile that delivers adequate tracking speed, accurate tracking position, precise end position, smooth curve-segment transition, low motion jerks, etc.

Several interpolators designated for interpolating parametric curves have been reported. They include the direct approximation method [4],[9],[13-14]; the feedback iteration method [7]; and the implicit techniques [11]. However, due to the absent of the information of exact arc length in the parametric curves that used in most modern CAD systems, today, existing parametric curve interpolating algorithms are only able to yield partial satisfactory results for many industrial applications, particularly, when tasks demand high speed and accuracy.

Some research attempts were performed to solve the problem on arch length estimation. For instance, Wang and Yang [12] and Erkorkmaz and Altintas [2] design quintic spline curves that are nearly arc-length parametrized. However, Besides, Pythagorean- Hodograph curve [3], proposed by Farouki and Shan, offers essentially exact solutions to real-time interpolation of 2D curved paths at fixed or varying feedrate. To make use of the algorithms presented in reports [2-3],[12] major changes on current commercial CAD packages are needed.

Recently, Nam and Yang [8] presented a recursive trajectory generation method that estimates an admissible path increment and determines the initiation of the final deceleration stage according to the distance left to travel estimated at every sampling time, resulting in exact feedrate trajectory generation through jerk-limited acceleration profiles for the parametric curves. However, this method requires heavy computational load because consecutive tool path is determined recursively at every sampling period. Newton-Raphson method is used to iterate numerical solution. Moreover, arc-length is still calculated numerically.

In this study, a hybrid parametric interpolator design will be introduced. It takes advantage on the fact that the exact curve dimension in parametric domain is naturally available. This hybrid parametric interpolator is a combination of both speed and parameter convolutions. During accelerating and cruising operation, this hybrid parametric interpolator operates under speed convolution for motion profile generation. When motion needs to stop or need to perform a smooth segment transition, parameter convolution technique will take over according a switch mechanism. As a result, this hybrid parametric interpolator enables the generation of entire motion profile that delivers adequate tracking speed, accurate tracking position, precise end position, smooth curve-segment transition, low motion jerks.

This paper is organized in five sections. Some existing theory and algorithm for motion interpolator design are summarized in Sections 2, 3, and 4. In section 5, our hybrid parametric interpolator is presented. Some discussion and remarks are presented in Section 6.

2. MOTION INTERPOLATOR FOR PARAMETRIC CURVES

As mentioned, motion control interpolators for parametric curves have been studied for many years. Fig. 1 shows a typical 2D parametric curve P(u) and P(u) = [x(u) y(u)]. The theory and algorithm presented in this paper can be applied to P(u) of any dimensions. For example a six degrees-of-freedom machine can use this interpolator to generate motion control references for tracking a curve like P(u) = [x(u) y(u) z(u) θ_1(u) θ_2(u) θ_3(u)]. The parameter u is defined within [U_s, U_e], where U_s and U_e stand for start and end parameter values. Usually u is normalized to within range [0. 1].

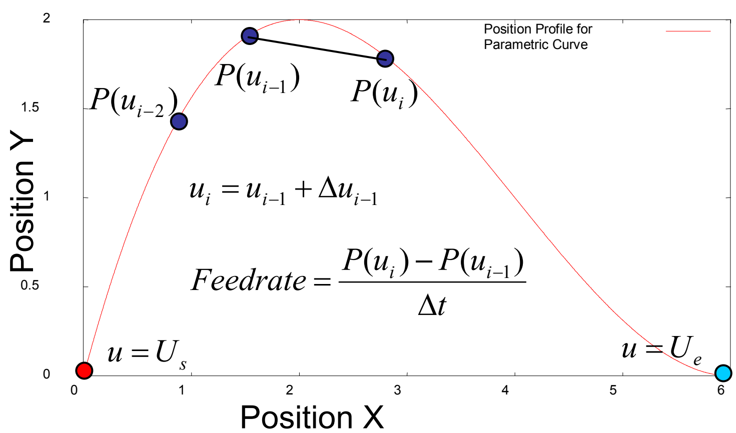

Fig. 1. Parametric Free Form Curve

The theory of a motion control interpolator can be depicted in Fig. 2, as three functional blocks. These functions are 1. speed smoothing for overall motion planning, 2. time/parameter conversion for calculating parameter increment at each sampling time, and 3. position reference generation for coordinated motion control.

As shown in the Fig. 2, the block "speed smoother" is to carry out the function of speed smoothing, i.e., to generate the entire speed profile, so that the motion jerks during the periods of acceleration to or deceleration from the desired cruising speed are limited to within acceptable ranges. Similarly the main function of a "time/parameter converter" (or T/P converter) is to take the sampling period, Δt, smoothed reference velocity, V(u) and the curve, P(u) as inputs and convert Δt into Δu_i according to Eqn. (1). Then within each sampling period update u_i, the current parametric position, by adding the calculated Δu_i. Finally, via the "reference generator", the reference position S_i at each sampling time for servo control inputs is computed by substituting u_i into the parametric curve P(u).

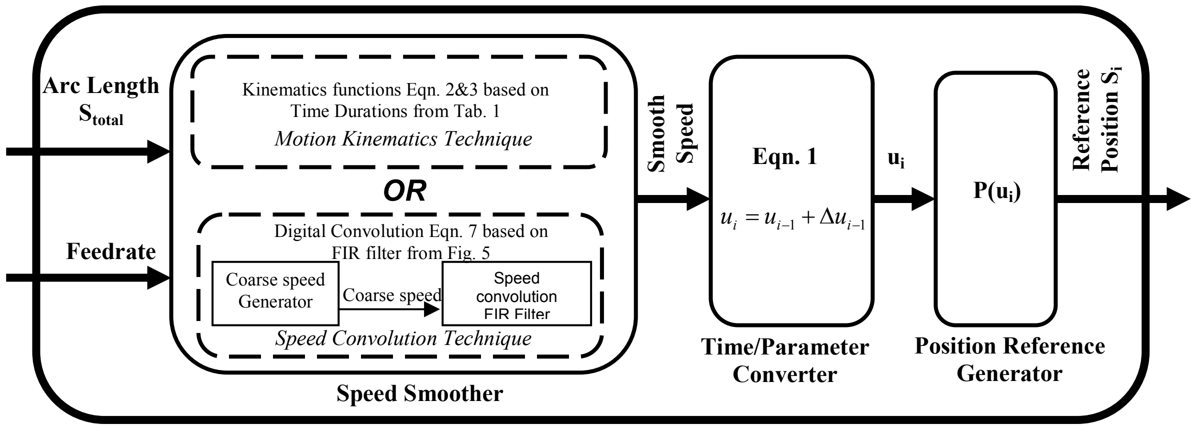

Fig. 2. Parametric Speed Interpolator

3. INTERPOLATION THEORY ON TIME/PARAMETER CONVERSION

A general theory for the sampling time/parameter conversion function of a motion interpolator for parametric curves was developed by Chou and Yang [1]. On the basis of this theory, Huang and Yang [4] and Yang and Kong [13] developed algorithms for computer implementation. From their reports, if a parametric curve is represented as P(u), and the desired tracking speed is V(u), then the differential parameter/time relationship can be constructed by a second order Taylor approximation as:

$$u_i = u_{i-1} + \frac{V(u_{i-1})}{\left|\frac{dP(u)}{du}\right|_{u=u_{i-1}}}\Delta t + \left[\frac{A(u_{i-1})}{\left|\frac{dP(u)}{du}\right|_{u=u_{i-1}}} - \frac{V^2(u_{i-1})\left[\frac{dx}{du}\frac{d^2t}{du^2}+\frac{dy}{du}\frac{d^2t}{du^2}+\frac{dz}{du}\frac{d^2t}{du^2}\right]_{u=u_{i-1}}}{2\left|\frac{dP(u)}{du}\right|^4_{u=u_{i-1}}}\Delta t^2\right] + E(\Delta t^3) \tag{1}$$

where u_i is the u value at sampling time *i* and $E(\Delta t^3)$ is the residual error of the second order approximation. The time/parameter conversion described by Eqn. (1) is a general one and can be applied to any parametric curves.

4. INTERPOLATION METHODS ON SPEED SMOOTHING

Currently there are three major speed smoothing techniques for the generation of the entire motion profile within motion interpolators. They are 1. by using motion kinematics and 2. by using speed convolution, and 3. by using parameter convolution. These three techniques are presented in Section 4.1, 4.2 and 4.3, respectively.

4.1 Kinematics Based Interpolator Speed Smoothing

Kinematics functions technique is one of most commonly used motion planning technique. The motion profile consists of seven piecewise regions as indicated in the first column of Tab. 1, which requires a heavy computational process to calculate the time duration of seven regions. In terms of definitions, T_k is the reference time, J_k is the jerk, A_k is the acceleration, V_k is the velocity and S_k is the position. The subscript k is the associate piecewise regions. The associate kinematics functions for acceleration A_k and velocity V_k are listed in Eqn. (2) and Eqn. (3) respectively.

$$A_k(t) = \int_{T_{k-1}}^{t} J_k d\tau = tJ_k - T_{k-1}J_k + A_{k-1} \tag{2}$$

$$V_k(t) = \int_{T_{k-1}}^{t}\int_{T_{k-1}}^{\beta} J_k d\alpha + A_{k-} d\beta = \frac{J_k t^2}{2} + (A_{k-1} - T_{k-1}J_k)t + \frac{J_k (T_{k-1})^2}{2} - A_{k-1}T_{k-1} + V_{k-1} \tag{3}$$

Provided that initial and final conditions are given, time duration of these seven regions can be obtained. The results are summarized in the second column of Tab. 1, where

$$\tau_{acl} = \frac{2(V_0 - V_3) * (2A_{\max} + A_0)}{2(A_{\max})^2(1+J_{pct}) + 2A_0 A_{\max} J_{pct} - (A_0)^2(1-J_{pct})} \tag{4}$$

$$\tau_{dec} = \frac{2(V_4 - V_7) * (2D_{\max} + A_4)}{2(D_{\max})^2(1+J_{pct}) + 2A_4 D_{\max} J_{pct} - (A_4)^2(1-J_{pct})} \tag{5}$$

The resultant speed V_k from Eqn. (2) and acceleration A_k from Eqn. (3) can then be forwarded to the P/T converter of Eqn. (1), as shown in Fig. 1.

1 Change To Max. Acceleration	$\tau_1 = \frac{(A_0 + A_{\max})(J_{pct}-1)\tau_{acl}}{2A_{\max} + A_0}$
2 Stay At Max. Acceleration	$\tau_2 = J_{pct}\tau_{acl}$
3 Change To Zero Acceleration	$\tau_3 = \frac{A_{\max}(J_{pct}-1)\tau_{acl}}{2A_{\max} + A_0}$
4 Stay At Zero Acceleration	$\tau_4 = \frac{S_{total} - S_{acl} - S_{dec}}{V_4}$
5 Change To Max. Deceleration	$\tau_5 = \frac{(A_4 + D_{\max})(J_{pct}-1)\tau_{dec}}{2D_{\max} + A_4}$
6 Stay At Max. Deceleration	$\tau_6 = J_{pct}\tau_{dec}$
7 Change to Zero Deceleration	$\tau_7 = \frac{D_{\max}(J_{pct}-1)\tau_{dec}}{2D_{\max} + A_4}$

Tab. 1. Kinematics Functions for Time Durations

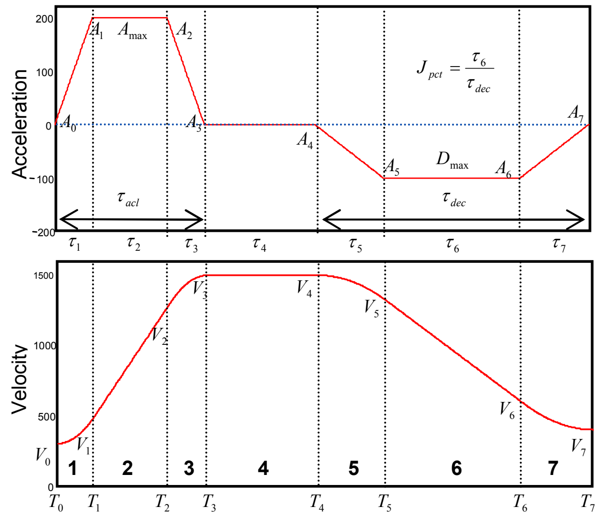

Fig. 3. Jerk-Limited Motion Profile

It is well known that by using motion kinematics for speed smoothing, the accuracy on the estimation of arc length $S_{total,}$ used in Tab. 1, has direct impact on motion planning. Since the arc length S_{total} has to be estimated numerically, this motion profile may lead to large tracking errors. For instance, when S_{total} is shorter than exact arc length, end position is not reached at zero velocity. If S_{total} is longer than exact length, end position is reached in motion, resulting in sudden stop at end position. Traditionally, kinematics based speed smoothing is suitable for interpolating straight or very simple motions. This technique becomes very difficult, if not impossible, when complex parametric curves of freeforms involve in.

4.2 Speed Convolution Based Interpolator Speed Smoothing

For the above mentioned reason, when tracking parametric curves of freeforms (including straight lines) interpolation with speed convolution technique has been developed. Several digital filter techniques on speed convolution have been proposed for generating motion profile to achieve speed smoothing. They include low pass filters, exponent filters, and finite-impulse response (FIR) filters, etc. A filter used for motion interpolation should be linear, so that the sum of the filter's outputs must converge to the sum of inputs.

Fig. 4(a) illustrates the idea of a finite impulse response (FIR) filter with a finite length, or stage, η. Value η is also referred as the FIR filter time constant. The FIR filter can be considered as a shift register whereby when a new value is input into the filter. A shift register can be represented by a series of z^{-1} operator, where $V(k) = z^{-1} V(k)$ means that $V(k) = V(k-1)$, and k denotes discrete time steps. The output of the filter is computed as the sum of all the outputs of the shift registers multiple with corresponding weight coefficients a_i, then divided by sum of weight coefficients. For motion profile planning purpose, the shape of weight coefficients a_i describes the acceleration profile. If the shape of weight coefficients is trapezoidal, the output of that FIR filter results in jerk-limited motion profile as shown in Fig. 3.

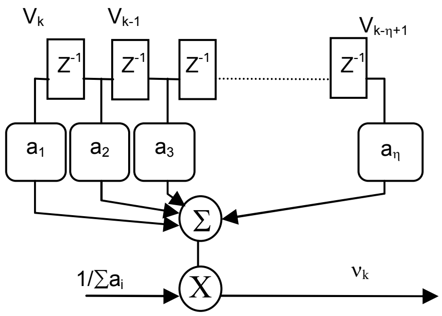

Fig. 4(a). General FIR Filter

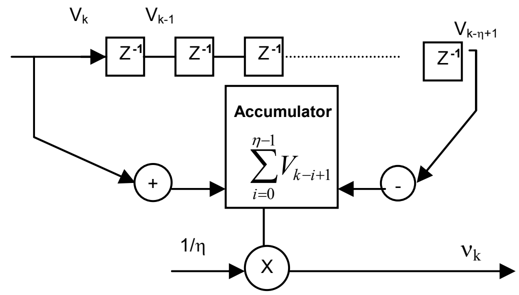

Fig. 4(b). Linear FIR Acc/Dec Processor

4.2.1 Linear speed convolution

One simplified form of FIR filter is a linear FIR filter where all weight coefficients are one. It is shown in Fig. 4(b). Suppose that at any instant k, the average of the latest η samples of a data sequence V_i, is given as Eqn. (6):

$$v_k = \frac{1}{\eta}\sum_{i=1}^{\eta} V_{k-i+1} \qquad (6)$$

This simple form of linear FIR is also known as moving average filter. A significant improvement in computational efficiency can be achieved if we perform the calculation of the mean in a recursive fashion. A recursive solution is one, which depends on a previously calculated value.

$$v_k = \frac{1}{\eta}\left[\sum_{i=0}^{\eta-1} V_{k-i+1} + V_k - V_{k-1}\right] \qquad (7)$$

Compared to simple averaging, it can be seen that we need only perform 1 division, 1 addition and 1 subtraction operation. This is always the case, regardless of the number of data points (η) we consider. However, calculating the current filtered value requires the use of accumulator, i.e. the measurement η time-steps in the past. This means that we need to store the value of $V_{k-\eta}$ which depending on the way the algorithm is coded, may require up to η storage locations.

Linear FIR filter in recursive form is the simplest digital convolution filter. However, it can only generate trapezoidal velocity profile because the acceleration function a_i is unity constant. Therefore, this motion is not jerk limited.

4.2.2 Cascade speed convolution

By cascading two of these linear FIR filters each with different finite lengths m and n, the overall filter can be used to construct a simple and yet robust acceleration/deceleration scheme for motion path planning to achieve speed smoothing and to generate jerk-limited motion profiles. It is shown in the following Fig. 5(a). By cascading, it is meant that the output of the first filter Z_k becomes the input to the second filter, with the same rules as described above applied to each filter. The overall cascaded filter is still an FIR filter as shown in Fig. 5(b).

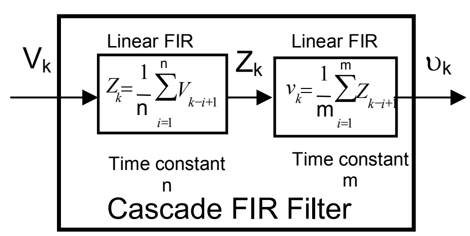

Fig. 5(a). Cascade Two Stages FIR Filter

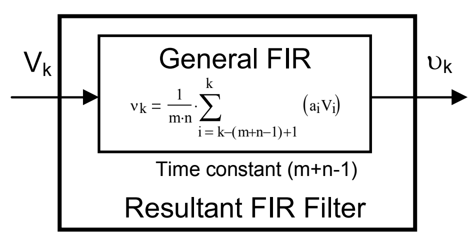

Fig. 5(b). Equivalent Combine FIR Filter

As a result, the sum n+m-1 of a two-stage cascade linear filter represents the total acceleration time, which is also the total deceleration values into the filter. The weigh coefficient a_i of resultant FIR filter is trapezoidal; it takes a total of n terms for acceleration and n terms for deceleration. Thus, the percentage of jerk limitation J_{pct} can be calculated as:

$$J_{pct} = \frac{2n}{n+m-1} \tag{8}$$

As shown in Eqn. (8), the acceleration profile in Fig. 3 can be controlled by adjusting the ratio of time constant n and m. For example, 100% jerk limitation means that n = m -1.

4.2.3 Time constant determination for two-stage cascade linear filter

The calculation of acceleration time constants can be simplified from Eqn. (4), where A_0 is zero. As Δt is the sampling period for interpolation, we get the following acceleration time constant η_a:

$$\tau_{acl} = \eta_a \Delta t = \frac{2(V_{\max} - V_0)}{A_{\max}(1 + J_{pct})} \tag{9}$$

From Eqn. (8) and (9), we can also determine the filter length for first stage linear FIR filter:

$$n = \frac{J_{pct}\eta_a}{2} \tag{10}$$

Then, the filter length of second stage linear FIR filter should be

$$m = \eta_a - n + 1 \tag{11}$$

Similar to acceleration, the deceleration time constant η_d is:

$$\tau_{dec} = \eta_d \Delta t = \frac{2(V_{\max} - V_7)}{D_{\max}(1 + J_{pct})} \tag{12}$$

4.2.4 Speed convolution based interpolator

The overall structure of a motion interpolator based on this cascade speed convolution technique is depicted in Fig. 2. As shown, the inputs, V_i, for the FIR need to be first calculated. This process is referred as the "coarse speed generation" here. Fig. 6 provides the flow chart diagram of coarse speed generation.

The arc length of a curve segment must be pre-estimated in order to determine when to stop. This interpolator's performance depends, therefore, largely on the accuracy of this arc-length estimation. Simple curves such as straight lines and circular arcs have exact arc length information available. Therefore, for those curves, interpolators equipped with this speed smoother can generate complete satisfactory motion profiles.

However, most parametric curves such as NURBS do not provide exact information of arc length. Any approximation inaccuracy of arc length results in discontinuous motion profile, which shall in turn lead to axis vibration. Thus, interpolators of speed convolution have certain limits on their performance.

4.3 Parameter Convolution Based Interpolator Speed Smoothing

Fig. 7 illustrates the concept of parameter-convolution based interpolator, which is based on filtered parameter increment (or "parametric speed") Δu_i. By taking parametric speed Δu_i as input to a digital convolution filter, also known the speed smoother, we can achieve a complete and smooth stop at target position. To calculate the input parametric increment Δu_i for the filter, the process of T/P conversion is first performed in this case. Again, Eqn. (1) is used for this purpose based on the desired constant cruising speed. As the sum of input Δu_i of the speed smoother is clearly defined in u domain with range $[U_s, U_e]$, the sum of associate Δu_i' shall be also clearly defined in $[U_s, U_e]$. As a result, motion profile is completed when u_i reaches U_e.

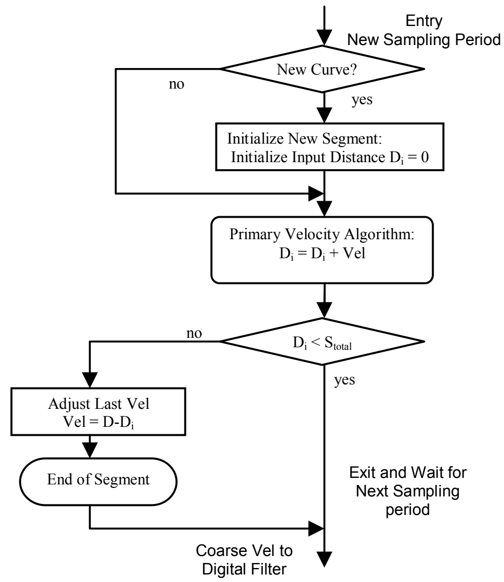

Fig. 6. Flow Chart for Coarse Speed Generation

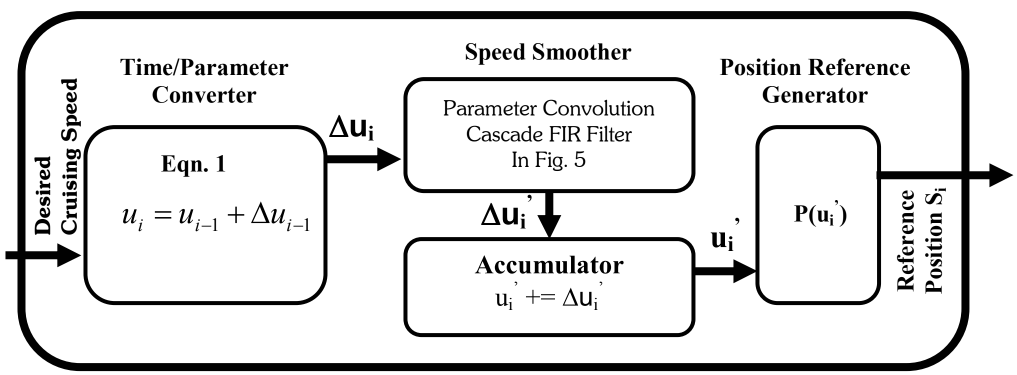

Fig. 7. Parameter Convolution Interpolation Technique

Within operational range of feedrate, this interpolation method provides a reasonable solution. Fig. 1 shows a typical parametric free-form curve profile with its curvature varying along the path. As shown in Fig. 8(a), S-curve velocity profile is achieved in tolerance limit of feedrate fluctuation.

This interpolation method is totally based on parameter u and its range. It does not require knowledge of exact arc length. In other words, unlike other existing interpolators, this interpolator can automatically stop smoothly at end positions. However, the tradeoff is that it offers no direct control of feedrate. The lag caused by FIR filter has a significant effect on federate control. As shown in Fig. 8(b), parameter convolution interpolator illustrates how feedrate fluctuation with curvature at a high-speed operation.

5. A HYBRID PARAMETRIC INTERPOLATOR

In this study we introduce a new, hybrid parametric interpolator that takes advantages of both speed and parameter convolutions. Fig. 9 shows the flow chart diagram for implementing a hybrid parametric interpolator. At any given time, only one module is selected to serve while the other one stays idle to meet operating needs.

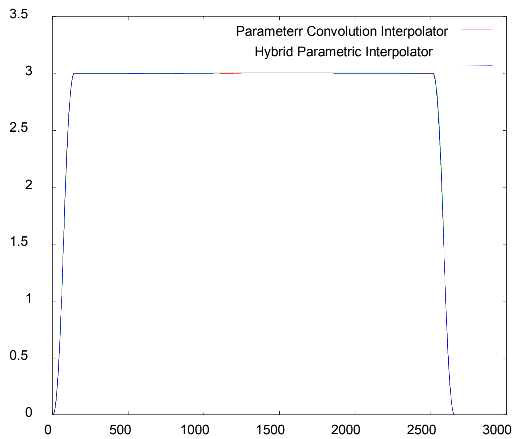

Fig. 8(a). Low Speed Feedrate Fluctuation

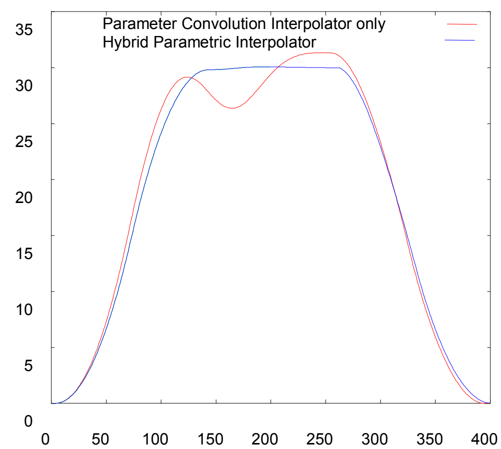

Fig. 8(b). High Speed Feedrate Fluctuation

This hybrid parametric interpolator emphases on eliminating feedrate fluctuation and enabling smooth and complete stop at exact target position. In order to achieve the first goal, parametric speed interpolation is selected until deceleration zone is reached. Digital convolution technique is used as motion profile planning. At every exampling time, the remaining deceleration distance in parameter u is monitored. A monitoring methodology is required to help determine the switch point of the interpolation mode change. And when motion needs to stop, parameter convolution interpolation will take over parametric speed interpolator at proper designated parameter u distance.

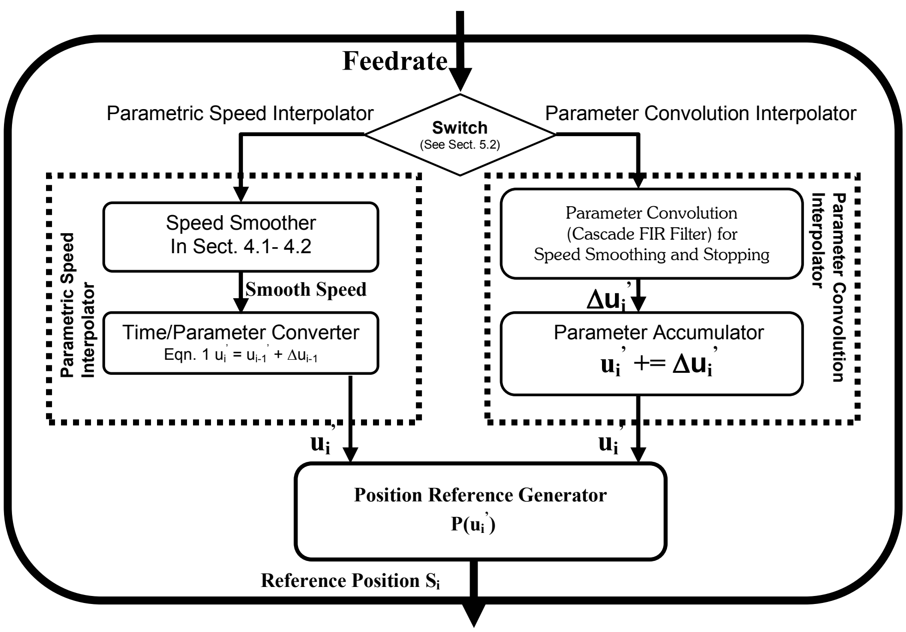

Fig. 9. Flow Chart for Hybrid Parametric Interpolator

5.1 The Switch Mechanism

First let us defines the terms U_d, the "u deceleration distance" and u_s, the "switch point." Value U_d is the distance needed for the servo motion to decelerate from the cruising speed to a complete stop ate the targeted end position where $u = U_e$. According to our two-stage cascade filter, the value U_d can be calculated by

$$U_d = \frac{(n+m)}{2} \Delta u_i \tag{13}$$

where, n and m are time constants for the first and second FIR filters, respectively. Please note that in this equation the value of Δu_i needs to be updated at each sampling period according to the output of the T/P converter (Eqn. (1)).

The switch point is defined as the u value at which the tracking motion needs to be decelerated to a stop, and

$$u_s = U_e - U_d \tag{14}$$

In each sampling period, the accumulated parameter value u_i and the switch point value u_s have to be updated and compared. When $u_i \geq u_s$, the switch mechanism turns on and the interpolator switches from speed-convolution mode to parameter-convolution mode, as illustrated in Fig. 9.

5.2 Adjustment of u_i for Smooth Switching

In order for motion to exactly stop at target end position, it is critical that we obtain the actual u deceleration distance, U_a, where

$$U_a = U_e - u_i \tag{15}$$

Note that U_a is different from U_d as differently defined according to Eqn. (13) and Eqn. (15). The corresponding parametric speed "Δu_a" can be calculated by using Eqn. (13) as

$$\Delta u_a = \frac{2U_a}{n+m} \tag{16}$$

The parametric increment Δu_i should be very close to Δu_a, yet they are slightly different. As a result, velocity during hybrid transition may become slightly discontinuous, and this is not a favorable feature.

In order to guarantee speed continuity during switching, initial parametric speed when parameter convolution turns on must be set equal to the final parametric speed Δu_i updated from the T/P converter. For our two-stage cascade linear FIR filter, the second stage filter has more influence. Therefore, this initial velocity setting requirement can be easily met by setting second stage linear FIR filter as Δu_i. Then, offsets compensation is added into first stage FIR filter. To calculate the compensation, we presume instantaneous parametric velocity in second stage filter offsets the parametric velocity needed to obtain exact u stop distance by a variable, namely Δu_{o2}.

$$\Delta u_{o2} = \Delta u_i - \Delta u_a \tag{17}$$

Then, the corresponding compensation for first linear FIR filter is calculated and satisfies the following relationship in terms of time constants in both stages.

$$\Delta u_{o1} = \Delta u_{o2} \frac{m+1}{n-1} \tag{18}$$

The compensated setting of first stage FIR filter should be Δu_s - Δu_{o1}.

The resultant velocity profile is also shown in Fig. 8(b). Our new hybrid parametric interpolator presents the satisfactory result. Hybrid parametric interpolator emphases on achieving controllable feedrate and enabling smooth and complete stop at exact target position.

6. DISCUSSION AND CONCLUDING REMARKS

In this paper, a new hybrid parametric interpolator is introduced. We believe this interpolator can achieve the following interpolating objectives for multi-axis motion control:

(1) Providing complete motion profile: This hybrid parametric interpolator does not require the exact arc length, which is not available. During deceleration zone, hybrid parametric interpolator regulates the decelerating parametric velocity based on u domain. Thus, motion profile can be completed smoothly at the end of travel.

(2) Reducing feedrate (tracking speed) fluctuation: Feedrate fluctuation could be kept low by parametric speed algorithm, where the performance relies on proper selection and usage of the order of Taylor expansion polynomial. In cases that feedrate fluctuation is a big concern, e.g., in high speed and high curvature scenarios, higher order polynomial approximation can be used to obtain better performance.

(3) Increasing position accuracy: This hybrid parametric interpolator performs motion profile planning before interpolation. It could eliminate contour error as reference position S_i is now calculated directly from $P(u_i)$. Other acceleration/deceleration methods [5] by placing digital filter after interpolation introduce path command error.
(4) Achieving time deterministic: This hybrid parametric interpolator can be classified as direct approximation method, where numerical iteration is not necessary. Therefore, the execution time is deterministic. The reliability of the system is guaranteed.
(5) Enabling on-the-fly (spontaneous) feedrate adjustment: Kinematics function technique is not a suitable technique for motion profile planning because it is very complex in calculating time durations for complete motion profile as shown in Tab. 1. Digital convolution is a very efficient technique for motion profile planning. Digital convolution technique with its low computational load should make on-the-fly feedrate adjustment possible.
(6) Providing axis speed continuity: This hybrid parametric interpolation can easily handle composite parametric curves with sharp corner. One possible easy solution is to simply stop at the curve intersections or sharp corners. In some application where continuous motion is required, blending with superposition technique of this hybrid interpolator is feasible. After deceleration zone is reached in the first curve, second curves will also start its movement. The summary of the decelerating velocity command and the accelerating speed command from both composite curves can automatically form a smooth blending transition around the sharp corner. The superposition blending technique does not require look-ahead path planning. Hybrid parametric interpolator, which is designed to smoothly stop at end of travel, will easily provide axis speed continuity.
(7) Meeting system physical constraints: Physical constraints of servo system are being met by governing the maximum allowable acceleration rate in the process of setting cascade linear FIR filter.

In summary, we believe this hybrid parametric interpolator should be a useful contribution to motion control technologies; it posses many functional advantages and could be easily implemented.

6. REFERENCES

[1] Chou J.J. and Yang, D.C.H., On the Generation of Coordinated Motion of Five-Axis CNC/CMM Machines, ASME *Journal of Engineering for Industry*, Feb. 1992, Vol. 114, pp. 15-22.

[2] Erkorkmaz K. and Altintas Y., High speed CNC system design. Part I. Jerk limited trajectory generation and quintic spline interpolation, *Int. J. Machine Tools Manufacturing.*, Vol. 41, 2001, pp 1323-1345.

[3] Farouki R. T. and Shah S., Real-time CNC interpolators for Pythagorean-Hodograph curves, *Computer-Aided Geometric Design*, Vol. 13, 1996, pp 583-600.

[4] Huang J. T., and Yang D. C. H., A generalized interpolator for command generation of parametric curves in computer controlled machines, *Japan/USA Symp Flexible Automat.,* 1993, pp393-399.

[5] Kim D. I., Jeon J. W., and Kim S., Software acceleration/deceleration methods for industrial robots and CNC machine tools, *Mechatronics,* Vol. 4, No. 1, 1994, pp 37-53.

[6] Kong, C. T., *Advanced Interpolation Technologies for Complete Profile Motion Control – Theory and Implementation*, Ph.D. Thesis, University of California, Los Angeles, CA, 2005.

[7] Lo C. C., Feedback Interpolators for CNC machine tools., *ASME Journal of Manufacturing Science and Engineering.*, Vol. 119, No. 4, 1997, pp 587-592.

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

[9] Shpitalni M., Koren Y. and Lo C. C., Realtime curve interpolators, *Computer-Aided Design*, Vol. 26, No. 11, 1994, pp 349-357.

[10] Tam H. Y., Xu H. and Tse P. W, An algorithm for the interpolation of hybrid curves, *Computer-Aided Design*, Vol. 35, 2003, pp 267-277.

[11] Tsai Y. F., Farouki R. T. and Feldman B., Performance analysis of CNC interpolators for time-dependent feedrates along PH curves, *Computer-Aided Geometric Design*, Vol. 18, 2001, pp 245-265.

[12] Wang F. C., and Yang D. C. H., Nearly arc-length parameterized quintic-spline interpolation for precision machining, *Computer-Aided Design*, Vol. 25, No. 5, 1993, pp 281-288.

[13] Yang D. C. H. and Kong T., Parametric interpolator versus linear interpolator for precision CNC machining, *Computer-Aided Design*, Vol. 26, No. 3, 1994, pp 225-233.

[14] Yeh S. S. and Hsu P. L., The speed-controlled interpolator for machining parametric curves, *Computer-Aided Design*, Vol. 31, 1999, pp 349-357.