Accurate Interpolation of Machining Tool-paths Based on FIR Filtering

Shingo Tajima
Graduate Student
tajimas@oregonstate.edu
Mechanical Engineering Department
Oregon State University
Corvallis, Oregon, USA

Burak Sencer, Ph.D (corr. Author)
Assistant Professor
Burak.sencer@oregonstate.edu
Mechanical Engineering Department
Oregon State University
Corvallis, Oregon, USA

Eiji Shamoto, Dr.Eng.
Professor
shamoto@mech.nagoya-u.ac.jp
Graduate School of Engineering
Nagoya University
Nagoya, JAPAN

Abstract

This paper presents a novel real-time (online) interpolation algorithm based on Finite Impulse Response (FIR) filters to generate smooth and accurate reference motion trajectories for machine tools and motion systems. Typically, reference tool-paths are composed of series of linear (G01) or circular (G02) segments. Basic point-to-point (P2P) feed motion can be generated by interpolating each segment with trapezoidal or S-curved velocity profile. However, smooth and accurate transitions between path segments are necessary to realize non-stop contouring motion. In this study, FIR filters are utilized, and the reference tool-path is filtered to interpolate a non-stop rapid feed motion. By using a chain of FIR filters, acceleration and jerk continuous motion profiles are generated from velocity pulse commands. A segment interpolation timing technique is developed to control the contour errors during non-stop real-time interpolation of tool-paths. Furthermore, by utilizing FIR filters for interpolation, frequency spectrum of the interpolated motion profiles is controlled. The time constant (delay) of the filter is tuned to create notches around the lightly damped vibration modes of the motion system, which allows mitigation of unwanted vibrations and thus enables delivering accurate feed motion. Simulation studies and industrial scale experimental validations are provided to illustrate effectiveness of the developed interpolation technique.

1. Introduction

Reference trajectory generation plays a key role in the computer control of machine tools and motion systems. Generated trajectories must not only describe the desired tool path accurately, but must also have smooth kinematic profiles in order to maintain high tracking accuracy, and avoid exciting natural vibration modes of the mechanical structure or servo control system. As a matter of fact, most machining tool-paths are defined in terms of series of linear (G01) segments or circular (G02) arcs [1][2]. This imposes serious limitations in terms of delivering a non-stop smooth and rapid motion for productivity, and to achieve the desired final part geometry.

There are several challenges associated with interpolating a smooth motion along these discrete tool-paths. Consider interpolation on a single path segment; feedrate (tangential velocity) profile needs to be planned with smooth acceleration and decelerations to avoid excitation of the machine tool's structural modes [3] and at the same time respect kinematic limits, i.e. torque, acceleration and jerk, of the drives [4][5][6]. Polynomial based feed profiles, such as trapezoidal velocity [1], acceleration [4] and jerk profiles [8] are well-known to the machine tool literature. They can be planned to fully exploit machine limits and generate time-optimal feed motion along predetermined paths [5][6]. However, these methods suffer from two bottlenecks. Firstly, they don't provide any quantitative means to control the frequency spectrum of the interpolated acceleration commands. In practice, the jerk limit is used to mitigate any residual vibrations [3]. Note that, tuning the jerk limit smoothens acceleration profile. But, it does not directly control the frequency spectrum. Robotics literature adapted exponential [9], trigonometric [10] or minimum jerk spline [6] based

acceleration profiles to help attenuate frequency spectrum of reference trajectories. In precision machine tool literature, input shapers (IS) [11][12] and notch filtering are utilized to filter the reference motion commands to attenuate the excitation around the lightly damped resonant frequencies of the machine. These techniques are easy to implement and robust against parameter variations [11], which makes them suitable for practice. However, input shaping distorts interpolated tool trajectories due to shaper dynamics and induces interpolation errors. Either machining velocity (feedrate) has to be lowered to reduce those errors, which is widely employed in practice; or, model-based compensation techniques that are mostly computationally costly are proposed in the literature [13].

Another bottleneck is the computational load of reference trajectory generators. As the degree and complexity of the acceleration profile becomes higher, computational cost to plan polynomial based trajectory generation increases [14]. Recent efforts are directed towards generating online, real-time suitable interpolation techniques [15][16][17]. These approaches essentially utilize a tuned dynamic system to filter and smoothen velocity/displacement commands. They are designed with a chain of integrators and cascaded feedback loops [18]. Online path smoothers can be implemented in the form of recursive difference equations. To attain time-optimal motion, nonlinear feedback elements such as saturation blocks are also introduced [19][20]. Nevertheless, unless combined with an input shaper, methods cannot control frequency spectrum of generated trajectories.

Finite Impulse Response (FIR) filters provide a computationally efficient framework for online trajectory generation. The use of FIR filtering for real-time interpolation and trajectory generation is known to the machine tool literature [22]. Chain of 1^{st} order FIR filters can be used to generate smooth reference trajectories with trapezoidal acceleration and jerk profiles [23][24]. Time constants of filters can be assigned to realize time optimal motion. Furthermore, frequency response of the filter can be tuned so that the excitation of the reference trajectory is shifted away from the resonances of the machine tool. Finally, it can be implemented as a moving average filter on modern micro-processors with minimum computational effort [23].

Although FIR filtering is an effective technique for online interpolation of reference trajectories, so far, its use is constrained in simple point-to-point (P2P) moves. If consecutive moves, e.g. linear or circular segments, are interpolated continuously without a full stop at the segment junctions, large interpolation contouring errors occur due to sudden change in the feed direction and the dynamics of the filter. Unless these contour errors are confined, the use of FIR filtering for generating uninterrupted, rapid and accurate feed motion in precision motion systems is limited. Recent literature recognized these shortcomings and proposed compensation techniques[13][24][25]. However, these techniques are either computationally expensive because they need to estimate errors through dynamic models. This greatly limits their application in real-time implementation. Or, they consider contouring errors only around junction of linear segments [24], which is not realistic since conventional machining tool-paths consist of both mixture of linear and circular segments, and transitions in-between those segments must be considered for a non-stop high-speed contouring motion.

This paper, for the first time, presents comprehensive interpolation techniques for generating uninterrupted and accurate feed motion along multi-segmented machining tool-paths based on FIR filtering. Contributions of the paper are laid out as follows. Section 2 first analyses high-order trajectory generation based on FIR filtering technique. It is followed by the introduction of accurate interpolation of linear and circular paths. A feedrate control technique is presented to control contour errors during interpolation of circular paths. Section 3 presents online interpolation of multi-segmented toolpaths based on FIR filtering. Dwell time control technique is presented to control the interpolation errors that occur during non-stop transition between linear and circular segments. Finally, Sections 4 and 5 present illustrative examples and rigorous experimental validations along complex tool-paths.

2. Online Trajectory Generation based on FIR Filtering

2.1. Generation of high order kinematic profiles

Typically, "trapezoidal acceleration" or "trapezoidal jerk" based feed profiling is employed to generate reference trajectories for high-speed and precision motion systems [4][8]. This section outlines the basic methodology to generate high-order trajectories utilizing a chain of FIR filters [23].

A 1st order FIR filter is defined in Laplace (s) domain by the following transfer function [23][26]:

$$M_i(s) = \frac{1}{T_i}\frac{1-e^{-sT_i}}{s}, \quad i = 1...N \tag{1}$$

where T_i is the time constant (delay) of the i^{th} FIR filter. Observed from Eq.(1), a FIR filter consists of an integrator (*1/s*) and a pure delay e^{-sT_i} resembling a simple moving average filter [26]. The impulse response is evaluated by taking inverse Laplace transform of Eq.(1) as:

$$m(t) = L^{-1}\left(M_i(s)\right) = \frac{u(t)-u(t-T_i)}{T_i} \text{ where } u = \begin{cases} 1, & t \geq 0 \\ 0, & t < 0 \end{cases} \tag{2}$$

and as shown in Figure 1, it becomes a simple rectangular pulse with a duration of T_i having a magnitude of $1/T_i$. This implies that for any $T_i>0$, the area underneath the impulse response is unitary. As a result, when an arbitrary signal is convolved with the FIR filter, area underneath the original signal does not alter. Furthermore, since the filter has a free integrator (*1/s*) it increases degree of the filtered (convolved) signal. This property can be used to generate high order real-time motion trajectories as follows.

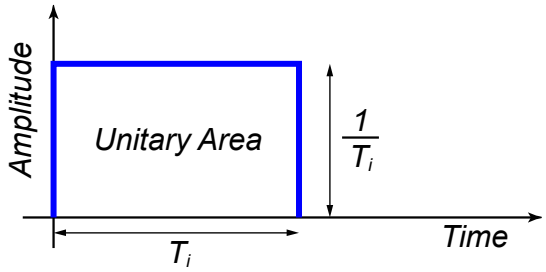

Figure 1: Impulse response of a 1st order FIR filter

Let us consider a simple linear movement for a length of *L* commanded at a velocity of *F*. This trajectory can be commanded by a rectangular pulse for a duration of $T_v=L/F$ as shown in Figure 2a. FIR filtering this rectangular velocity pulse generates the well-known trapezoidal velocity profile [1] with piecewise constant acceleration segments (See Figure 2b). Subsequently, another FIR filter can be convolved with the trapezoidal velocity profile to generate smoother trapezoidal (jerk limited) acceleration profile [4]. As outlined in Figure 2, high-order reference kinematic profiles can be generated by filtering a reference velocity pulse though chain (series) of FIR filters [23][24]. Finally, the resultant velocity profile is integrated to obtain reference displacement profile as shown in Figure 2c.

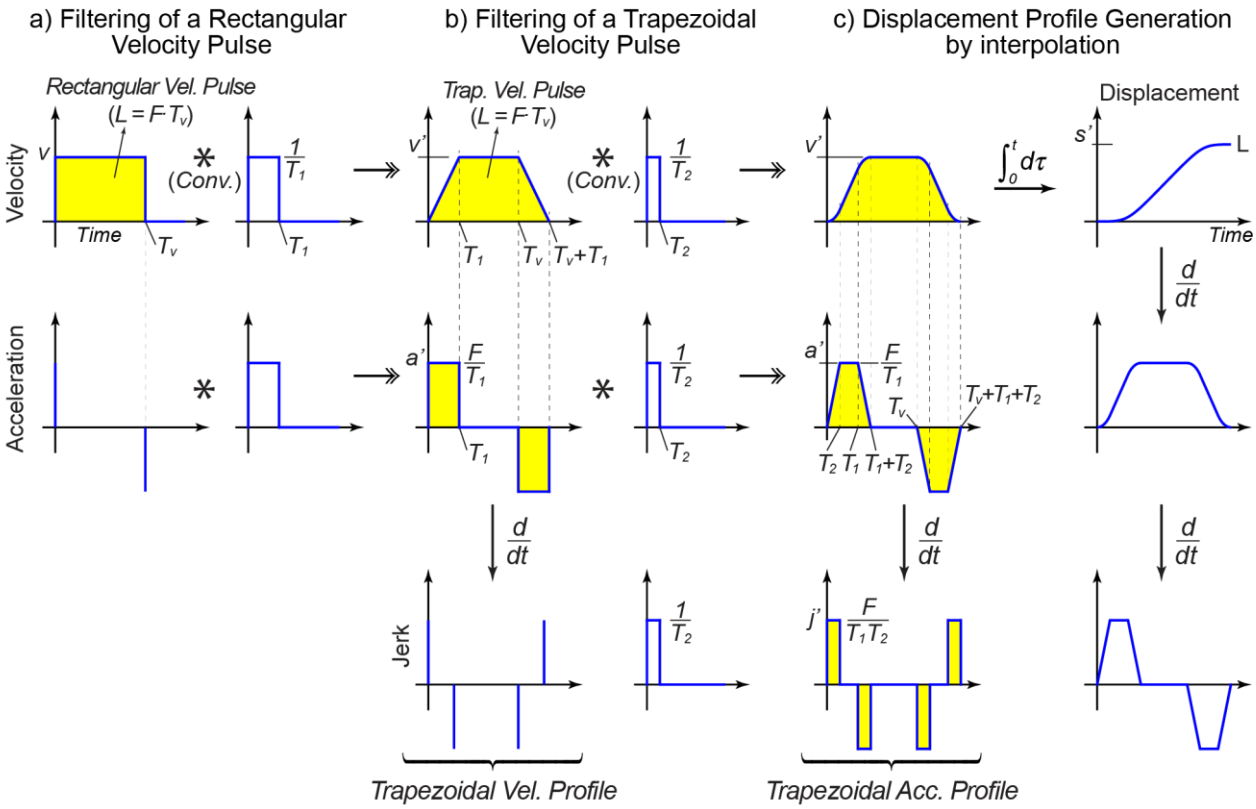

Figure 2: FIR filtering based smooth trajectory generation

Filtered kinematic profiles can be analyzed through analytical solution of convolution [26]. Consider a simple trapezoidal velocity profile. Convolution of a rectangular velocity pulse command *v(t)*, with the impulse response of the FIR filter from Eq. (2) is written as:

$$\begin{aligned} v'(t) &= v(t) * m(t) = \frac{1}{T_1}\int_0^t \Big(\big[v(\tau) - v(\tau - T_v) \big] \big[u(t-\tau) - u(t - T_1 - \tau) \big] \Big) d\tau \\ &= \frac{1}{T_1} \begin{bmatrix} \int_0^t v(\tau) u(t-\tau) d\tau - \int_0^t v(\tau) u(t - T_1 - \tau) d\tau \\ - \int_0^t v(\tau - T_v) u(t-\tau) d\tau + \int_0^t v(\tau - T_v) u(t - T_1 - \tau) d\tau \end{bmatrix} \end{aligned} \tag{3}$$

and the filtered velocity signal *v'(t)* is derived by evaluating above integrals,

$$v'(t) = \begin{cases} (F/T_1)t & , \ 0 \le t < T_1 \\ F & , \ T_1 \le t < T_v \\ F/T_1 \left(-t + T_v + T_1 \right) & , \ T_v \le t < T_v + T_1 \\ 0 & , \ T_v + T_1 \le t \end{cases} \tag{4}$$

where the resultant acceleration signal is obtained by differentiation as:

$$a'(t) = \begin{cases} F/T_1 & , \ 0 \le t < T_1 \\ 0 & , \ T_1 \le t < T_v \\ -F/T_1 & , \ T_v \le t < T_v + T_1 \end{cases} \tag{5}$$

Finally, smooth displacement profile *s'(t)* is generated by integration of the velocity profile.

The use of convolution enables analytical derivation of filtered profile kinematics. Above trapezoidal velocity profile is derived for the case of $T_v > T_1$ and also illustrated in Figure 3a. The peak acceleration depends on the filter's time constant and the commanded velocity, $A_{peak} = F/T_1$. Commanded velocity F is reached at filter's time constant, $t = T_1$, and the remaining cruise velocity duration becomes T_v-T_1. On the other hand, when the reference velocity pulse duration is equal or shorter than the filter's time delay $T_v \leq T_1$, motion kinematics alters. As depicted in Figure 3b, when $T_v = T_1$, no velocity cruise section occurs. As opposed, Figure 3c illustrates the case of $T_v < T_1$. In this case, commanded velocity cannot be reached and peak velocity becomes $V_{peak} = L/T_1$ with a cruise phase duration of T_1-T_v. Although omitted here, convolution can be used to analytically derive all these kinematic profiles. The overall motion duration is elongated by the amount of filter's time constant $T_v + T_1$.

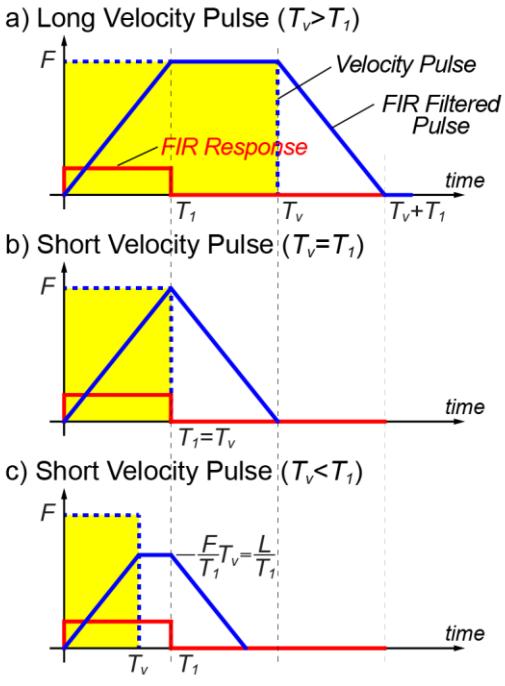

Figure 3: Trapezoidal velocity profile generated by single FIR filter

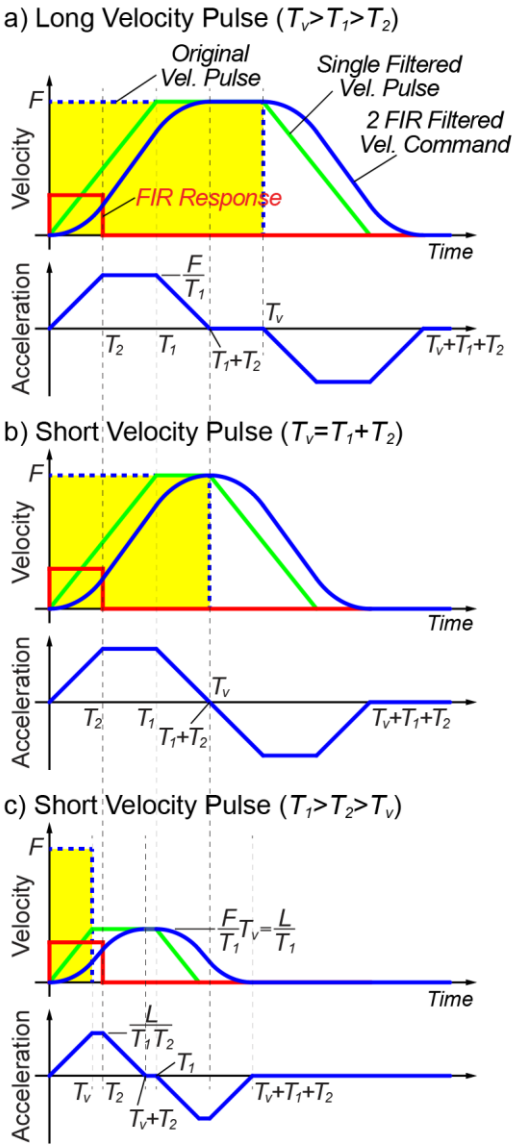

Figure 4: Trapezoidal acceleration profile generated by 2 FIR filters

In precision motion systems jerk [4] or even snap [8] limited velocity profiles are favored to generate smoother, more traceable reference motion profiles. As illustrated in Figure 2, utilizing 2 FIR filters with time constants T_1 and T_2 generates the well-known trapezoidal acceleration (jerk limited) feed profile. Figure 4 illustrates the motion profile generated by filtering a trapezoidal velocity pulse. The profile kinematics can be computed analytically by replacing rectangular velocity pulse command with the trapezoidal one in Eq.(3) , and for the case of $T_v>T_1>T_2$, the generated velocity profile is derived as:

$$v'(t)=\begin{cases}\frac{1}{2}\frac{F}{T_1T_2}t^2 & 0\le t<T_2\\ \frac{1}{2}\frac{FT_2}{T_1}+\frac{F}{T_1}\left(t-T_2\right) & T_2\le t<T_1\\ F-\frac{1}{2}\frac{F}{T_1T_2}\left(\left(T_1+T_2\right)-t\right)^2 & T_1\le t<T_1+T_2\\ F & T_1+T_2\le t<T_v\\ F-\frac{1}{2}\frac{F}{T_1T_2}\left(t-T_v\right)^2 & T_v\le t<T_v+T_2\\ F-\frac{1}{2}\frac{FT_2}{T_1}-\frac{F}{T_1}\left(t-\left(T_v+T_2\right)\right) & T_v+T_2\le t<T_v+T_1\\ \frac{1}{2}\frac{F}{T_1T_2}\left(\left(T_v+T_1+T_2\right)-t\right)^2 & T_v+T_1\le t<T_v+T_1+T_2\\ 0 & T_v+T_1+T_2\le t\end{cases}, \qquad (6)$$

and the corresponding filtered acceleration profile *a'(t)* becomes:

$$a'(t)=\begin{cases}\frac{F}{T_1T_2}t & 0\le t<T_2\\ \frac{F}{T_1} & T_2\le t<T_1\\ \frac{F}{T_1}-\frac{F}{T_1T_2}\left(t-T_1\right) & T_1\le t<T_1+T_2\\ 0 & T_1+T_2\le t<T_v\\ -\frac{F}{T_1T_2}\left(t-T_v\right) & T_v\le t<T_v+T_2\\ -\frac{F}{T_1} & T_v+T_2\le t<T_v+T_1\\ -\frac{F}{T_1}+\frac{F}{T_1T_2}\left(t-\left(T_v+T_1\right)\right) & T_v+T_1\le t<T_v+T_1+T_2\\ 0 & T_v+T_1+T_2\le t\end{cases} \qquad (7)$$

As observed from Eq.(6), filtered (interpolated) motion profile is determined by the filter delays, T_1 and T_2, and the length of the reference velocity pulse, T_v. There are in total 8 combinations. Owing to the linearity of filtering operation, the order of filters in the chain does not matter, and 3 main cases characterize the interpolated profiles; namely, $T_v>T_1+T_2$, $T_v=T_1+T_2$, or $T_v<T_1+T_2$. Figure 4a depicts the most common case when the velocity pulse is longer than total sum of filter time constants, $T_v>T_1+T_2$ and $T_1>T_2$. Note that in this case, a full 7-segmented jerk limited acceleration profile [4] with cruise phase can be generated (Eq.(6)). However, if a rapid (high speed) move on a short travel distance is commanded, T_v may become smaller. For

instance, cruise velocity phase may disappear completely if $T_v=T_1+T_2$ as shown in Figure 4b. Furthermore, if $T_v<T_1+T_2$, commanded path velocity cannot be reached. In this case, peak velocity is computed as $V_{peak}=L/T_1$ (See Figure 4c). The total motion duration is elongated by the filter delay as, $T_v+ T_d$, where $T_d = T_1 + T_2$.

2.2. Frequency shaping of interpolated trajectories

The FIR filter structure also provides effective means to control frequency spectrum of the generated trajectories. Filtered acceleration $a'(t)$ profile controls the torque/force delivered by the feed drive, which induces excitation to the overall motion system. If frequency spectrum of the reference acceleration profile contains components near the lightly damped structural modes of the machine structure, it initiates forced vibrations [11], [13].

For a rectangular pulse velocity input, the acceleration command consists of set of impulses separated by T_v (See Figure 2a). Consider only a single acceleration impulse with a magnitude h

$$a(t) = \begin{cases} h, & t = 0 \\ 0, & t \neq 0 \end{cases} \tag{8}$$

convolved with the chain of FIR filters; the frequency spectrum of resultant acceleration profile simply becomes frequency response of the FIR filters in the chain, evaluated as:

$$a'(j\omega) = hM_1(j\omega)M_2(j\omega)...M_n(j\omega) \tag{9}$$

and frequency (ω) response of a single FIR filter can be computed from Eq.(1) as:

$$M_i(j\omega) = \frac{1}{T_i}\frac{1-e^{-j\omega T_i}}{j\omega} \rightarrow \left|M_i(j\omega)\right| = \frac{\sin\left(\frac{\omega T_i}{2}\right)}{\frac{\omega T_i}{2}} \tag{10}$$

Consequently, frequency spectrum of the acceleration profile becomes multiplication of sinc [27] functions from Eq. (10) as:

$$\left|a'(j\omega)\right| = \prod_{i=1}^{N}\left|\frac{\sin\left(\omega T_i / 2\right)}{\omega T_i / 2}\right| = \prod_{i=1}^{N}\left|\frac{\sin\left(\pi\omega / \omega_i\right)}{\pi\omega / \omega_i}\right|, \quad \text{where} \quad \omega_i = \frac{2\pi}{T_i} \tag{11}$$

The above property can be exploited to choose time constant of the FIR filter to avoid exciting lightly damped structural frequencies of the machine tool. Every *sinc* function creates periodic notches (ripples), which can be matched with the resonant frequency of the motion system by setting,

$$\omega_i = \frac{\omega_r}{k} \leftrightarrow T_i = k\frac{2\pi}{\omega_r} \tag{12}$$

An example is presented in Figure 5. Simply setting time constant of the filters in the chain to the natural periods of the resonant modes $k = 1, T_i = 2\pi/\omega_r$ introduces shortest filter delay into the motion while avoiding excitation of resonances. It is also notable that there is close resemblance between Input Shaper [11][12] and the FIR filter. Input Shapers have the property to cancel any vibration at the half of the vibration period, π/ω_r.

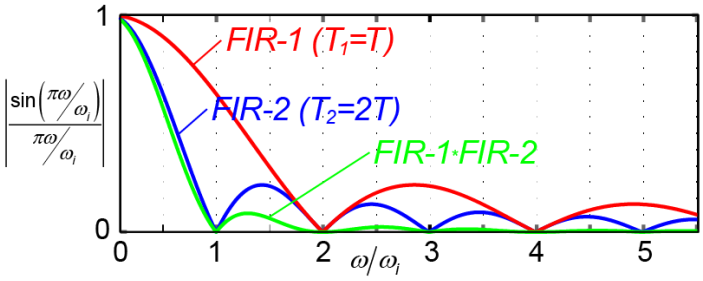

Figure 5: Frequency response of FIR filter

2.3. FIR based interpolation of linear and circular paths

2.3.1. Linear interpolation

Interpolation of single axis motion based on FIR filtering is presented in the previous sections. This technique can be extended to generate point-to-point (P2P) multi-axis linear motion. Figure 6 outlines the process to interpolate planar P2P linear motion between two points, $P_s = [x_s, y_s]^T$ and $P_e = [x_e, y_e]^T$. Firstly, the path length is computed from the Euclidean distance, $L = \|P_e - P_s\|$, and the tangential feed pulse F for a duration of $T_v = L/F$ is generated as shown in Figure 6b. The feed pulse is dissolved into its Cartesian velocity pulse components based on the path geometry,

$$\left.\begin{aligned} v_x(t) &= v(t)\frac{x_e - x_s}{L} = v(t)\cos(\alpha) \\ v_y(t) &= v(t)\frac{y_e - y_s}{L} = v(t)\sin(\alpha) \end{aligned}\right\} \text{ where } \alpha = \tan^{-1}\left(\frac{y_e - y_s}{x_e - x_s}\right) \qquad (13)$$

and smooth axis velocity commands are interpolated by applying FIR filtering. Note that, time constants of the filters are set identical so that the resultant motion is coordinated. Finally, the interpolated axis velocity commands are integrated to interpolate position commands.

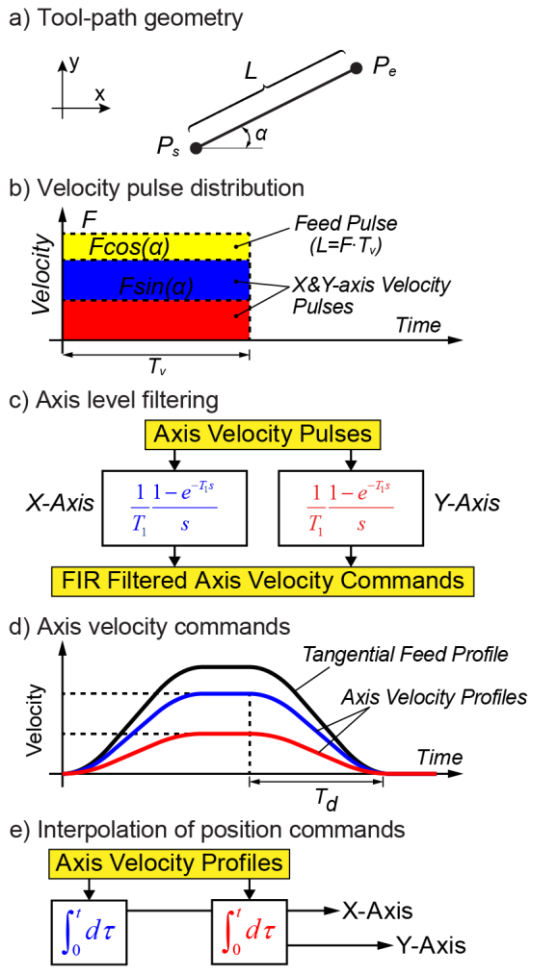

Figure 6: Multi-axis interpolation based on FIR filtering

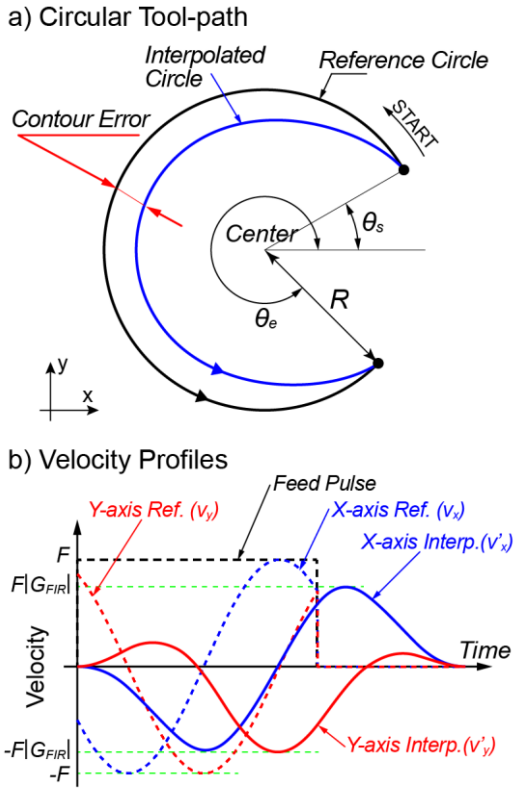

Figure 7: Circular interpolation and corresponding velocity profiles

2.3.2. Circular interpolation

Next, the approach for linear interpolation is adapted to interpolate circular paths. During circular interpolation, the total travel length becomes the arc length of the circular path $L = R\Delta\theta$ where R is the arc radius and $\Delta\theta = \theta_e - \theta_s$ is the difference between starting and ending angular positions (See Figure 7). Omitting the arc center, rectangular feed pulse F is dissolved into its axis components based on the circular geometry as:

$$\underbrace{\begin{Bmatrix} v_x(t) = -F\sin\left(\frac{F}{R}t+\theta_s\right) \\ v_y(t) = F\cos\left(\frac{F}{R}t+\theta_s\right) \end{Bmatrix}}_{\text{Velocity Commands}} \text{ and } \underbrace{\begin{Bmatrix} s_x(t) = R\cos\left(\frac{F}{R}t+\theta_s\right) \\ s_y(t) = R\sin\left(\frac{F}{R}t+\theta_s\right) \end{Bmatrix}}_{\text{Position Commands}} \tag{14}$$

Notice that as opposed to the linear motion, reference axis velocity commands during circular interpolation are not in rectangular pulse form but rather sinusoidal signals. The reference axis commands are then filtered through chain of FIR filters and integrated to interpolate the circular path. Figure 7 illustrates the reference and interpolated velocity commands during circular interpolation.

Notice that sinusoidal axis motion commands are generated at the rotational frequency of $\omega_c = F/R$, and they are modulated by the frequency response of FIR filters. At steady state, the filtered axis motion commands can be written from Eqs.(14) and (1) as:

$$\left.\begin{aligned} s'_x(t) &= R\left|G_{FIR}\right|_{\omega=\omega_c}\cos\left(\frac{F}{R}t+\theta_{init}\right)\\ s'_y(t) &= R\left|G_{FIR}\right|_{\omega=\omega_c}\sin\left(\frac{F}{R}t+\theta_{init}\right)\end{aligned}\right\} \tag{15}$$

where $G_{FIR}=\prod_{i=1}^{N} M_i(j\omega)$ is frequency response function of FIR filter. The discrepancy between reference and interpolated (filtered) circular motion commands result in an interpolation contour error ε as shown in Figure 7. This contour error is measured normal to the commanded circle, and its steady state value can be calculated from Eqs.(14) and (15) as:

$$\begin{aligned}\varepsilon &= \sqrt{\left(s_x - s'_x\right)^2+\left(s_y-s'_y\right)^2}\\ &= \sqrt{\left(R\left(1-G_{FIR}\right)\cos\left(\frac{F}{R}t+\theta_s\right)\right)^2+\left(R\left(1-G_{FIR}\right)\sin\left(\frac{F}{R}t+\theta_s\right)\right)^2}\\ &= R\left(1-G_{FIR}\right)\end{aligned} \tag{16}$$

The steady state value of the contour error is controlled by magnitude of the frequency response of the FIR filter at the fundamental frequency of the circular motion, ω_c. By lowering the feedrate *F*, excitation frequency can be altered, and ε can be confined by a user-specified tolerance value. Without losing generality, let us consider a trapezoidal acceleration profile generated by 2-FIR filters. The magnitude of the frequency response at ω_c can be evaluated from Eq. (11) as:

$$\left|G_{FIR}\right|_{w=w_c} = \left|\frac{\sin\left(\frac{T_1F}{2R}\right)}{\frac{T_1F}{2R}}\frac{\sin\left(\frac{T_2F}{2R}\right)}{\frac{T_2F}{2R}}\right| \tag{17}$$

4-term Taylor expansion can be applied to Eq.(17) to obtain a polynomial expression:

$$\left|G_{FIR}\right| \approx \left(1-\frac{\left(T_1w_c/2\right)^2}{3!}+\frac{\left(T_1w_c/2\right)^4}{5!}\right)\left(1-\frac{\left(T_2w_c/2\right)^2}{3!}+\frac{\left(T_2w_c/2\right)^4}{5!}\right) \tag{18}$$

and substituting Eq.(18) into Eq. (16) yields the relationship between the contouring error and the feedrate as:

$$\begin{aligned}&\varepsilon - R\left(1-\left|G_{FIR}\right|_{\omega=\omega_c}\right)=0\\ &\Rightarrow x^4-80\left(\frac{1}{T_1^2}+\frac{1}{T_2^2}\right)x^3+640\left(\frac{10}{T_1^2T_2^2}+\frac{3}{T_1^4}+\frac{3}{T_2^4}\right)x^2-153600\left(\frac{1}{T_1^4T_2^2}+\frac{1}{T_1^2T_2^4}\right)x+3686400\frac{\varepsilon}{T_1^4T_2^4R}=0\end{aligned} \tag{19}$$

where

$$x=\omega_c^2=\left(\frac{F}{R}\right)^2 \tag{20}$$

Eq.(19) is a *4th* order polynomial whose roots can be solved conveniently in real-time, and the maximum feedrate to bound the contour error by a tolerance value can be obtained from Eq.(20). It should be noted that the polynomial approximation used in Eq.(18) only approximates $\left|G_{FIR}\right|$ at low frequency; namely, below the

first notch (ripple) of FIR filters. The first notch is typically matched with one of the structural resonances to mitigate vibrations [1][12][24] and to minimize overall filter delay. Hence, the 4-term Taylor expansion is suitable for practice.

3. Multi-segmented tool-path interpolation strategy

Previous section discussed generation of basic linear and circular trajectories based on FIR filtering of axis velocity pulse commands. This section focuses on continuous and accurate interpolation of multi-segmented tool-paths.

Figure 8 presents the overall strategy for online FIR based interpolation of multi-segmented NC tool-paths. Each segment of the tool-path in a NC block/G-code is represented by a timed feed pulse. As introduced in the previous section, depending on the interpolation type, e.g. linear or circular, tangential feed pulse is dissolved into its axis velocity pulses, which are then filtered through chain of FIR filters to generate smooth axis reference motion commands. The NC part program can be interpolated based on two types of motion; namely, "point-to-point" (P2P), or non-stop "contouring". The motion type can be controlled by adjusting the "dwell" time between consecutive feed pulses (See Figure 8). For instance, in P2P mode, an instantaneous stop between the programmed segments is desired. This kind of motion strategy is typically employed in pick-and-place operations, ultra-precision machining and measurement. P2P motion can be achieved by simply accounting for the FIR filter delay and adding a *dwell* time between consecutive feed pulses that is equal to the filter delay T_d as outlined in Figure 8. Figure 9a shows an example P2P motion generated along 2 linear segments. Consecutive feed pulses are filtered with a dwell time of $T_d=T_1+T_2$ (in case of 2 FIR filters) to generate a full stop at P_1 and the corresponding velocity profile is shown in Figure 9b.

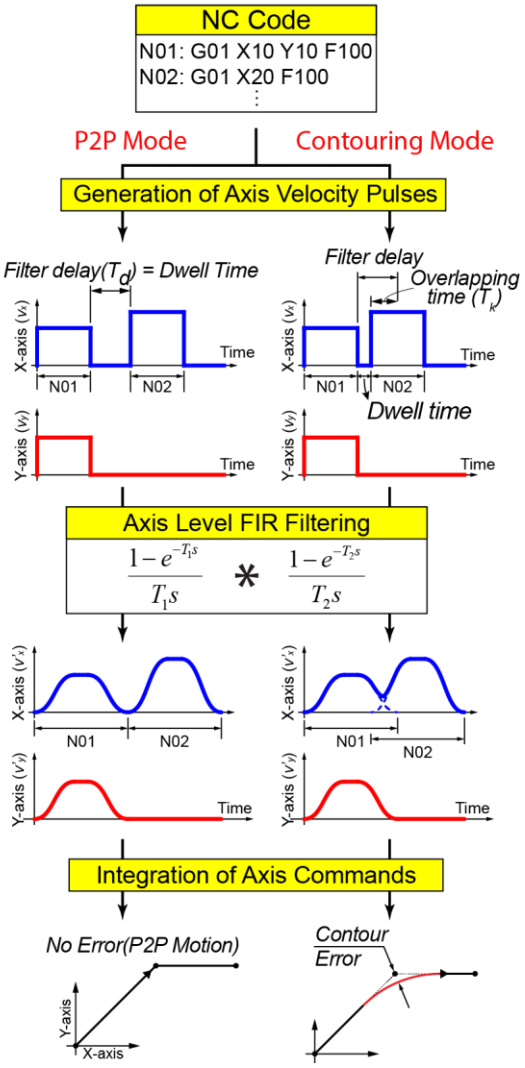

Figure 8: Overall path interpolation strategy

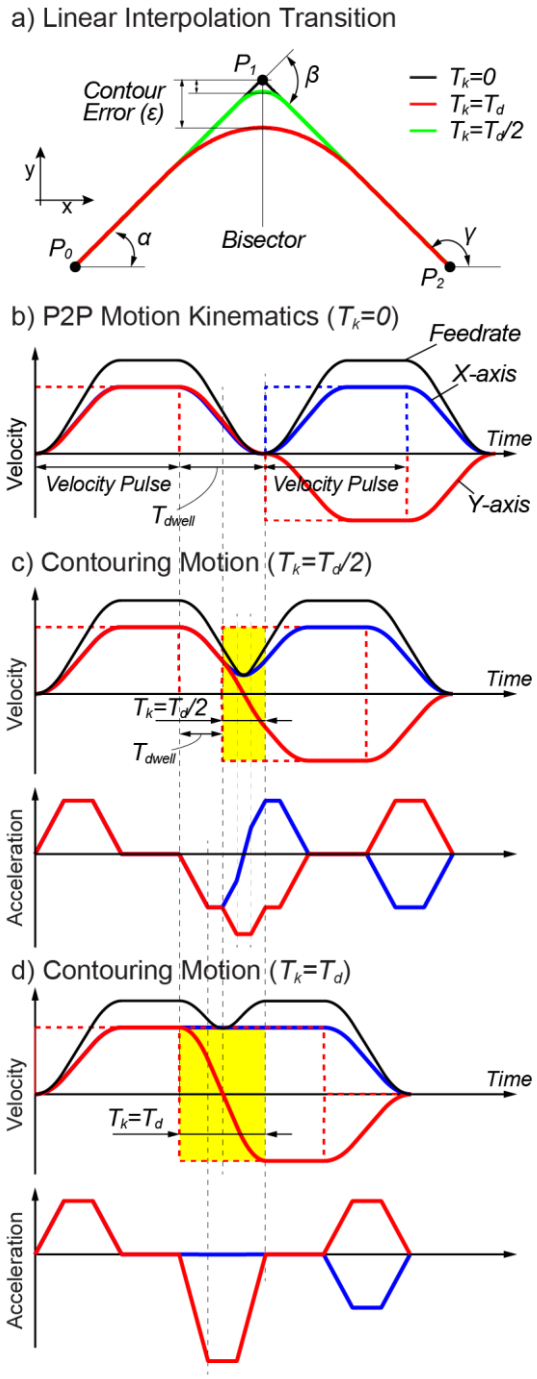

Figure 9: Kinematic profiles during contouring motion

3.1. Contour error control during non-stop linear interpolation

On the other hand, in high-speed machining, uninterrupted accurate contouring motion is desired. As presented in Figure 8, non-stop *"contouring"* motion can be generated by interpolating consecutive feed pulses without fully waiting for the filter delay to die out. This enforces convolution of consecutive feed pulse to begin with non-zero initial conditions, and through precise control of the *dwell* time, contouring errors [29][30] along segment transitions can be confined.

Firstly, let us define an "overlapping time", T_k, to control the *overlap* of convolution of the consecutive feed pulses:

$$0 \le T_k \le T_d \tag{21}$$

If $T_k=0$, the *dwell* time is equal to the total filter delay T_d, and as shown in Figure 9a and b, a P2P motion is generated. Figure 9c illustrates the case when $T_k=T_d/2$. In this case, consecutive segment interpolation is initiated before feed motion of the 1^{st} block comes to a full-stop. As a result, feed direction is altered continuously. Due to this gradual change in the feed direction an interpolation error, ε, occurs around the junction point of consecutive path segments (See Figure 9a). When the overlapping time T_k is increased to its upper limit $T_k=T_d$, *no dwell* time is inserted between consecutive feed pulses. Since change in the feed direction is also initiated earlier, larger interpolation contour error occurs as shown in Figure 9d.

The contour error around the segment junction due to non-stop change in the feed direction can be controlled analytically. Consider the generic path shown in Figure 9, the deceleration motion towards midpoint, P_1 starts at $t=T_v$. When a non-zero overlapping time is set $T_k>0$, feed direction towards the endpoint P_2 is initiated with the start of convolution of the consecutive segment at $t=T_v+T_d-T_k$. Note that convolution of the 1^{st} segment finishes at $t=T_v+T_d$, which marks the completion of feed direction change. If the feedrate F at consecutive segments is identical, total axis (x *and* y) velocity traverse, i.e. change in feed direction, is controlled directly by the angle between linear segments as:

$$\underbrace{\begin{Bmatrix} F_x^- = F\cos(\alpha) \\ F_y^- = F\sin(\alpha) \end{Bmatrix}}_{1^{st}\text{ Segment Velocities}}, \underbrace{\begin{Bmatrix} F_x^+ = F\cos(\gamma) \\ F_y^+ = F\sin(\gamma) \end{Bmatrix}}_{2^{nd}\text{ Segment Velocities}} \text{ where } \begin{Bmatrix} F_x^+ \\ F_y^+ \end{Bmatrix} = \underbrace{\begin{bmatrix} \cos(\beta) & -\sin(\beta) \\ \sin(\beta) & \cos(\beta) \end{bmatrix}}_{\text{Rotation Matrix}} \begin{Bmatrix} F_x^- \\ F_y^- \end{Bmatrix} \tag{22}$$

where $F_{x,y}^-$ represent axis velocities along the 1^{st} linear segment and $F_{x,y}^+$ represent velocities on the 2^{nd} (consecutive) segment, and β is the angle between the linear segments. Thus, when feed pulses with identical amplitude F are commanded, deceleration/acceleration kinematics around the bisector from the segment junction P_1 becomes mirror-imaged (See Figure 9a). As a result, tangential feedrate exhibits its minimum in the middle of the segment transition at $t=T_v+T_d-T_k/2$. Similarly, the interpolated trajectory also becomes symmetric where the maximum deviation from the junction point occurs along the bisector at $t=T_v+T_d-T_k/2$.

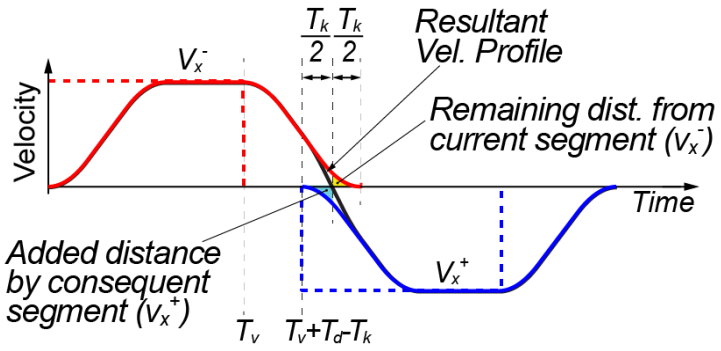

Figure 10: Axis kinematic profiles during segment transition

Kinematic profiles during segment transition depend on the overlapping time, T_k, and the filter delay, T_d. They can be computed by superimposing filtered velocity profiles of consecutive segments as shown in Figure 10. For a 7-segmented trapezoidal acceleration profile the x and y-axis velocities during segment transition can be calculated from Eq.(6) as:

$$v'_{x,y}(t)=\begin{cases} F^{-}_{x,y}-\frac{1}{2}\frac{F^{-}_{x,y}}{T_1T_2}\left(t-T_v\right)^2 & T_v\le t\le T_v+T_2 \\ F^{-}_{x,y}-\frac{1}{2}\frac{F^{-}_{x,y}T_2}{T_1}-\frac{F^{-}_{x,y}}{T_1}\left(t-\left(T_v+T_2\right)\right) & T_v+T_2\le t\le T_v+T_1 \\ \frac{1}{2}\frac{F^{-}_{x,y}}{T_1T_2}\left(\left(T_v+T_1+T_2\right)-t\right)^2 & T_v+T_1\le t\le T_v+T_1+T_2-T_k \\ \frac{1}{2}\frac{F^{-}_{x,y}}{T_1T_2}\left(\left(T_v+T_1+T_2\right)-t\right)^2+\frac{1}{2}\frac{F^{+}_{x,y}}{T_1T_2}\left(t-\left(T_v+T_1+T_2-T_k\right)\right)^2 & T_v+T_1+T_2-T_k\le t\le T_v+T_1+T_2 \\ \frac{1}{2}\frac{F^{+}_{x,y}}{T_1T_2}\left(t-\left(T_v+T_1+T_2-T_k\right)\right)^2 & T_v+T_1+T_2\le t\le T_v+T_1+2T_2-T_k \\ \frac{1}{2}\frac{F^{+}_{x,y}T_2}{T_1}+\frac{F^{+}_{x,y}}{T_1}\left(t-\left(T_v+T_1+2T_2-T_k\right)\right) & T_v+T_1+2T_2-T_k\le t\le T_v+2T_1+T_2-T_k \\ F^{+}_{x,y}-\frac{1}{2}\frac{F^{+}_{x,y}}{T_1T_2}\left(\left(T_v+2T_1+2T_2-T_k\right)-t\right)^2 & T_v+2T_1+T_2-T_k\le t\le T_v+2T_1+2T_2-T_k \end{cases} \tag{23}$$

for $T_k<2T_2$. Similarly, maximum contouring error along the bisector can be calculated by superimposing the remaining distance towards the midpoint P_1 during interpolation of the 1^{st} segment, and the distance traveled due to the convolution of the 2^{nd} segment. Based on Figure 10 Cartesian components of the maximum contour error can be written as:

$$\left.\begin{aligned} \varepsilon_x &= -\int_{T_v+T_d-T_k/2}^{T_v+T_d} v'^{-}_{x}d\tau+\int_{0}^{T_k/2} v'^{+}_{x}d\tau \\ \varepsilon_y &= -\int_{T_v+T_d-T_k/2}^{T_v+T_d} v'^{-}_{y}d\tau+\int_{0}^{T_k/2} v'^{+}_{y}d\tau \end{aligned}\right\} \tag{24}$$

and integrating axis velocity profiles $v'^{-}_{x,y}$ and $v'^{+}_{x,y}$ from Eq.(6) yields the maximum contour error during uninterrupted interpolation of consecutive linear segments from Eq.(24) as:

$$\varepsilon=\sqrt{\varepsilon_x^2+\varepsilon_y^2}=\begin{cases} \frac{T_k^{\,3}}{24T_1T_2}F\sin\left(\frac{\beta}{2}\right) & 0\le T_k\le 2T_2 \\ \frac{4T_2^{\,2}-6T_2T_k+3T_k^{\,2}}{12T_1}F\sin\left(\frac{\beta}{2}\right) & 2T_2<T_k\le T_1+T_2 \end{cases} \tag{25}$$

Finally, for a predetermined contour error tolerance the overlapping time can be solved from Eq.(25) as:

$$T_k=\begin{cases} \sqrt[3]{\frac{24T_1T_2e}{F\sin\left(b/2\right)}} & 0\le T_k\le 2T_2 \\ T_2+\sqrt{\frac{4T_1e}{F\sin\left(b/2\right)}-\frac{T_2^{\,2}}{3}} & 2T_2<T_k\le T_1+T_2 \end{cases} \tag{26}$$

As presented above, the *dwell* time between consecutive interpolation of feed pulses is controlled by the amount of T_k from Eq.(26), and interpolation contour error is confined by a predetermined value efficiently.

3.2. Control of contour errors during non-stop linear and circular interpolation

As shown in Figure 11, contour errors occur during non-stop interpolation through circular (G2/G3) and linear (G1) segments as well. The dwell time control method presented in the previous section can be adapted to confine these contour errors by approximating the change in the feed direction.

During circular interpolation, interpolated tool motion settles down on a circular path that has a smaller radius than the reference one due to the FIR filter dynamics (See Figure 7). As shown in Figure 11, the feed direction at the start of the circular interpolation is bounded between the tangent vector $\vec{\mathbf{t}}^{+}_{ref}$ of the reference path and the tangent vector $\vec{\mathbf{t}}^{+}_{filt}$ of the interpolated path shown. $\vec{\mathbf{t}}^{+}_{ref}$ is known from the reference path geometry, and $\vec{\mathbf{t}}^{+}_{filt}$ can be computed by the geometry through the following relationship:

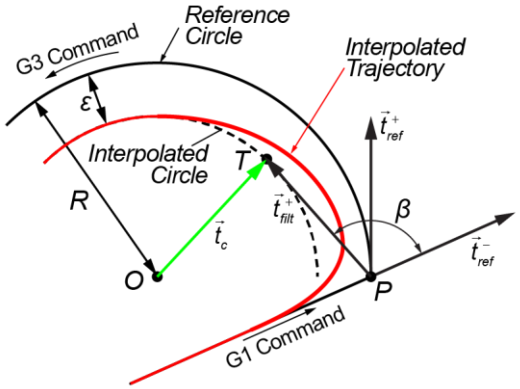

Figure 11: Linear to circular interpolation transition

$$\vec{t}_c \perp \vec{t}^{+}_{filt} \rightarrow \vec{t}_c \cdot \vec{t}^{+}_{filt} = \left|\vec{t}_c\right|\left|\vec{t}^{+}_{filt}\right|\cos\left(\frac{\pi}{2}\right) = 0 \tag{27}$$

where

$$\left.\begin{aligned} \left|\vec{t}_c\right| &= \sqrt{\left(T_x - O_x\right)^2 + \left(T_y - O_y\right)^2} \\ \left|\vec{t}_{filt}\right| &= \sqrt{\left(T_x - P_x\right)^2 + \left(T_y - P_y\right)^2} \end{aligned}\right\} \tag{28}$$

Note that norm of $\vec{\mathbf{t}}_c$ is known from Eq.(16),

$$\left|\vec{t}_c\right| = \sqrt{\left(T_x - O_x\right)^2 + \left(T_y - O_y\right)^2} = R - \varepsilon \tag{29}$$

and substituting Eqs.(29) and (28) in (27) yields:

$$\begin{aligned} \vec{t}_{filt} = &\left(\frac{(R-\varepsilon)^2\left(P_x - O_x\right) \pm (R-\varepsilon)\sqrt{R^2 - (R-\varepsilon)^2}\left(P_y - O_y\right)}{R^2} + O_x - P_x \right)\vec{i} + \\ &\left(\frac{(R-\varepsilon)^2\left(P_y - O_y\right) \mp (R-\varepsilon)\sqrt{R^2 - (R-\varepsilon)^2}\left(P_x - O_x\right)}{R^2} + O_y - P_y \right)\vec{j} \end{aligned} \tag{30}$$

where $\vec{\mathbf{i}}$ and $\vec{\mathbf{j}}$ are the unit directional vectors in *x* and *y* directions, respectively. Feed direction during linear to circular segment G01→G02/03 transition is then bounded between the $\vec{\mathbf{t}}^{+}_{ref}$ and $\vec{\mathbf{t}}^{+}_{filt}$, and hence the largest change can be approximated as shown in Figure 11 as:

$$\beta = \arccos\left(\min\left(\frac{\vec{t}^{-}_{ref} \cdot \vec{t}^{+}_{ref}}{\left|\vec{t}^{-}_{ref}\right|\left|\vec{t}^{+}_{ref}\right|}, \frac{\vec{t}^{-}_{ref} \cdot \vec{t}^{+}_{filt}}{\left|\vec{t}^{-}_{ref}\right|\left|\vec{t}^{+}_{filt}\right|} \right) \right) \tag{31}$$

where $\vec{t}^{-}_{ref}$ is the feed vector along the linear path. Eq.(31) is used to calculate the overlapping time T_k and control the maximum value of the contour error. In a similar fashion, transition from circular to linear segment G02/03→G01 is depicted in Figure 12a, and Eq. (31) can be adapted for this case as:

$$\beta = \arccos\left(\min\left(\frac{\vec{t}^{-}_{ref} \cdot \vec{t}^{+}_{ref}}{\left|\vec{t}^{-}_{ref}\right|\left|\vec{t}^{+}_{ref}\right|}, \frac{\vec{t}^{-}_{filt} \cdot \vec{t}^{+}_{ref}}{\left|\vec{t}^{-}_{filt}\right|\left|\vec{t}^{+}_{ref}\right|} \right) \right) \tag{32}$$

Finally, the transition between two consecutive circular segments G02/G03→G02/G03 is illustrated in Figure 12b. In this particular case, Eqs.(31)-(32) needs to be expanded to contain all possible combinations to bound the feed direction, and the largest angular change, i.e. worst case, is determined as:

$$\beta = \arccos\left(\min\left(\frac{\vec{t}^{-}_{ref} \cdot \vec{t}^{+}_{ref}}{\left|\vec{t}^{-}_{ref}\right|\left|\vec{t}^{+}_{ref}\right|}, \frac{\vec{t}^{-}_{ref} \cdot \vec{t}^{+}_{filt}}{\left|\vec{t}^{-}_{ref}\right|\left|\vec{t}^{+}_{filt}\right|}, \frac{\vec{t}^{-}_{filt} \cdot \vec{t}^{+}_{ref}}{\left|\vec{t}^{-}_{filt}\right|\left|\vec{t}^{+}_{ref}\right|}, \frac{\vec{t}^{-}_{filt} \cdot \vec{t}^{+}_{filt}}{\left|\vec{t}^{-}_{filt}\right|\left|\vec{t}^{+}_{filt}\right|} \right) \right) \tag{33}$$

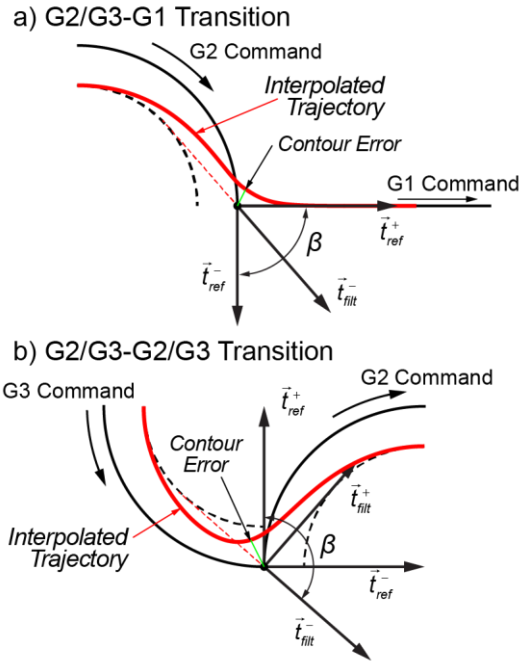

Figure 12: Feed direction during circular and linear transitions

4. Illustrative example

This section demonstrates application of the proposed FIR based block timing control technique to accurately interpolate machining tool-paths. The reference tool-path shown in Figure 13 is given in the Gcode/CL program defined by two G01 linear segments followed by circular move (G02). In order to interpolate it with a jerk limited trapezoidal acceleration profile, 2 FIR filters are used with time constants set to T_1=50 [msec] and T_2 =30 [msec]. The command feedrate is set to F=200 [mm/sec], and the maximum interpolation error tolerance is ε =100 [μm].

The path is interpolated based on the P2P and the proposed non-stop contouring type interpolation techniques (See Figure 8). Interpolation results are summarized in Figure 13. Figure 13a and b compare P2P and the contouring type interpolated tool trajectories, and resultant feedrate profiles. As shown, in case of P2P interpolation, the motion undergoes a full stop at each segment junction. A *dwell time* identical to the

total FIR filter delay $T_d=30+50=80$ [msec] is inserted between the blocks. The total cycle time for P2P motion results to $T_{total}=1.137$ [sec]. Proposed FIR filtering based contouring type interpolation technique can generate accurate non-stop feed motion. Contouring errors around transition of linear and circular segment junctions as well as the circular contour are precisely kept at and below the ε =100 [μm] tolerance value. The circular interpolation error is bounded by lowering the feedrate to 84.2 [mm/sec] from Eqs.(19) and (20). The contour error around segment transitions are controlled by calculating the overlapping time T_k based on the change in the feed direction from Eq.(26). The overall cycle time is reduced to $T_{total}=1.074$ [sec]. Next section presents interpolation of a more complex tool-path on an actual motion stage.

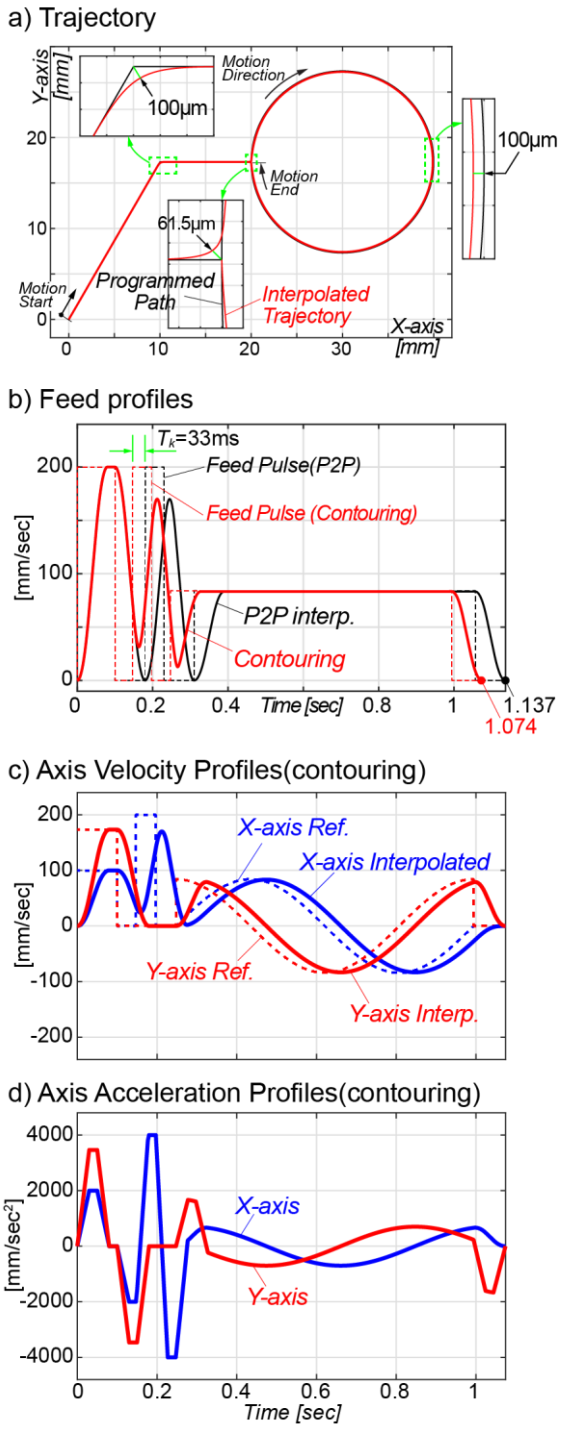

Figure 13: FIR based interpolation of multi-segmented path.

5. Experimental validation

5.1. Setup and implementation

Experimental validation and benchmark comparisons of the proposed technique are performed on the Cartesian *X*-*Y* motion system shown in Figure 14. The planar *X*-*Y* motion table is driven by 3 linear motors. The heavier *X*-axis is designed as gantry and carries the lighter *Y*-axis. In order to implement proposed algorithms, servo amplifiers are set to operate in torque (current) control mode. Closed loop control is implemented in the Dspace DS1103® real time control system by reading linear encoder feedback at a resolution of 0.1 [μm] and commanding torque signal to the servos at a closed loop sampling interval of T_s=0.1 [msec]. Both *X* and *Y* drives are controlled by P-PI cascade motion controllers with velocity feed-forward action. The position feedback control bandwidths of the axes are roughly matched at $w_n = 50\ [Hz]$ to ensure good motion synchronization and contouring [29].

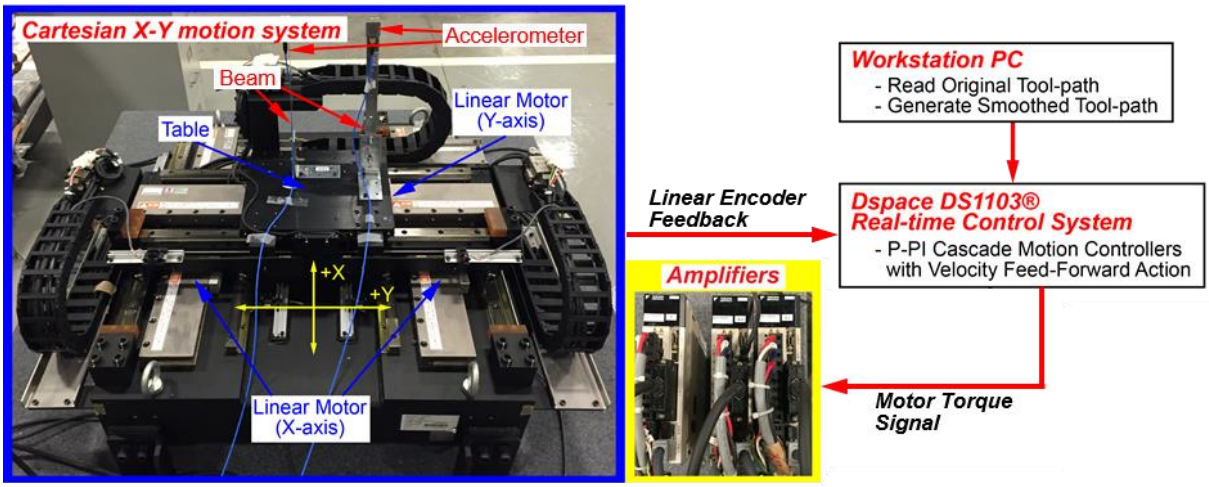

Figure 14: Experimental XY motion platform

Next, to generate motion commands at discrete time instants kT_s, FIR filter's transfer function from Eq.(1) needs to be discretized. A simple Euler's backward differentiation technique [26] is applied to derive the *z-domain* transfer function as:

$$M_i(z) = \frac{1}{N_i}\frac{1-z^{-N_i}}{1-z^{-1}} \tag{34}$$

where $N_i = T_i / T_s$ is the number of (delay) samples of the filter. The filtered velocity commands $v'_{x,y}$ are generated by implementing Eq.(34) through the following simple difference equation:

$$v'_{x,y}(k) = v'_{x,y}(k-1) + \frac{1}{N_i}\left(v_{x,y}(k) - v_{x,y}(k-N_i)\right) \tag{35}$$

where *k* is the sample counter. Note that generation of the filtered velocity commands from above difference equation requires only 2 additions and 1 multiplication for a single FIR filter.

5.2. Experimental Results

In the 1^{st} experiment, the clover shaped tool-path shown in Figure 15 is interpolated with the P2P and contouring interpolation techniques. As shown, this tool-path consists of 5 linear and 5 circular segments. The feedrate along the tool-path is set to *F = 200* [mm/sec], and the maximum interpolation contour tolerance is ε =100 [μm]. 2 FIR filters are used to interpolate the tool-path with time constants set to T_1=*30* and T_2=*25* [msec]. As shown in Figure 15, the tool-path is interpolated non-stop within desired contouring

tolerance. Maximum contouring errors both around linear and circular segment transitions as well as along the circular sections are respected. Kinematic profiles are shown in Figure 16. As shown, feedrate is lowered along circular sections and also at segment junctions to generate accurate transition. Figure 16b depicts the feed pulse timing and the resultant feedrate profiles. The cycle time during contouring interpolation is clearly shorter and axis motion profiles are acceleration and jerk limited (See Figure 16c and d). Contouring errors are measured experimentally [29] and presented in Figure 17 as well. As shown, the tool-path is interpolated within the given contour error tolerance. Experimentally measured contour errors show small discrepancy from the interpolated ones due to feedback tracking dynamics of the servo system. These errors are not accounted for in the proposed technique, and although small in this experiment (<15 [μm]), they can be further improved by well-known feed-forward control techniques [26].

A 2^{nd} experiment is performed to showcase vibration suppression capabilities of the proposed interpolation scheme. The starfish shaped tool-path shown in Figure 18 is interpolated using the proposed FIR filtering based technique and compared against another technique that fits small Bezier segments at segment junctions to realize a continuous motion transition and thereby non-stop contouring motion [28]. The tool-path consists of 125 short linear segments. The programmed feedrate is set to F=50[mm/sec], and the contouring tolerance is ε=30[μm]. 2 FIR filters are used to interpolate the path with trapezoidal acceleration profiles. The filter time constants are tuned to avoid unwanted vibrations. As shown in Figure 14, 2 flexible beams are placed on the X-Y table. The beams are flexible in orthogonal directions, e.g. X and Y, where their 1^{st} bending modes are identified with accelerometers (PCB-3711E1110G) mounted on the top of the beams as: $\omega_x \approx 7.4[Hz]$ and $\omega_y \approx 9.2\ [Hz]$. In order to avoid exciting the lightly damped beam resonances, FIR filter delays are set to T_1=136.1[msec] and T_2=110.9[msec], accordingly. Figure 19 shows feedrate profiles of the interpolation techniques. Note that, due to short linear segments, the programmed feedrate is never reached. As a result, interpolated feed profiles and the corresponding accelerations fluctuate aggressively. Nevertheless, the proposed FIR based filtering technique delivers the fastest cycle time while respecting desired contouring tolerance along the entire tool-path. Figure 20 compares frequency spectrum of the interpolated acceleration profiles. As shown, FIR based interpolation technique exhibits attenuated frequency spectrum especially around the resonances of flexible beam structures. Most of the excitation is kept in the lower frequencies. However, the Bezier based technique simply spreads the excitation at a much wider bandwidth. Figure 21 presents beam accelerations measured through the attached accelerometers. As shown, the level of acceleration of the beams are significantly less as the motion stage tracks the FIR based interpolated trajectory. The maximum acceleration is same as the interpolated one from the reference trajectory. As compared, the acceleration level is significantly higher on the Bezier based trajectory generation technique. Finally, Figure 22 shows the frequency spectrum of the measured beam accelerations. As shown, the Bezier based technique clearly excites the resonances and causes beams to vibration heavily. On the other hand, FIR based technique does not induce any unwanted vibrations and the beams move as a rigid body with the motion table. This experiment clearly demonstrates that the proposed technique can interpolate complex tool-paths accurately, deliver rapid non-stop contouring motion and at the same time mitigate unwanted residual vibrations.

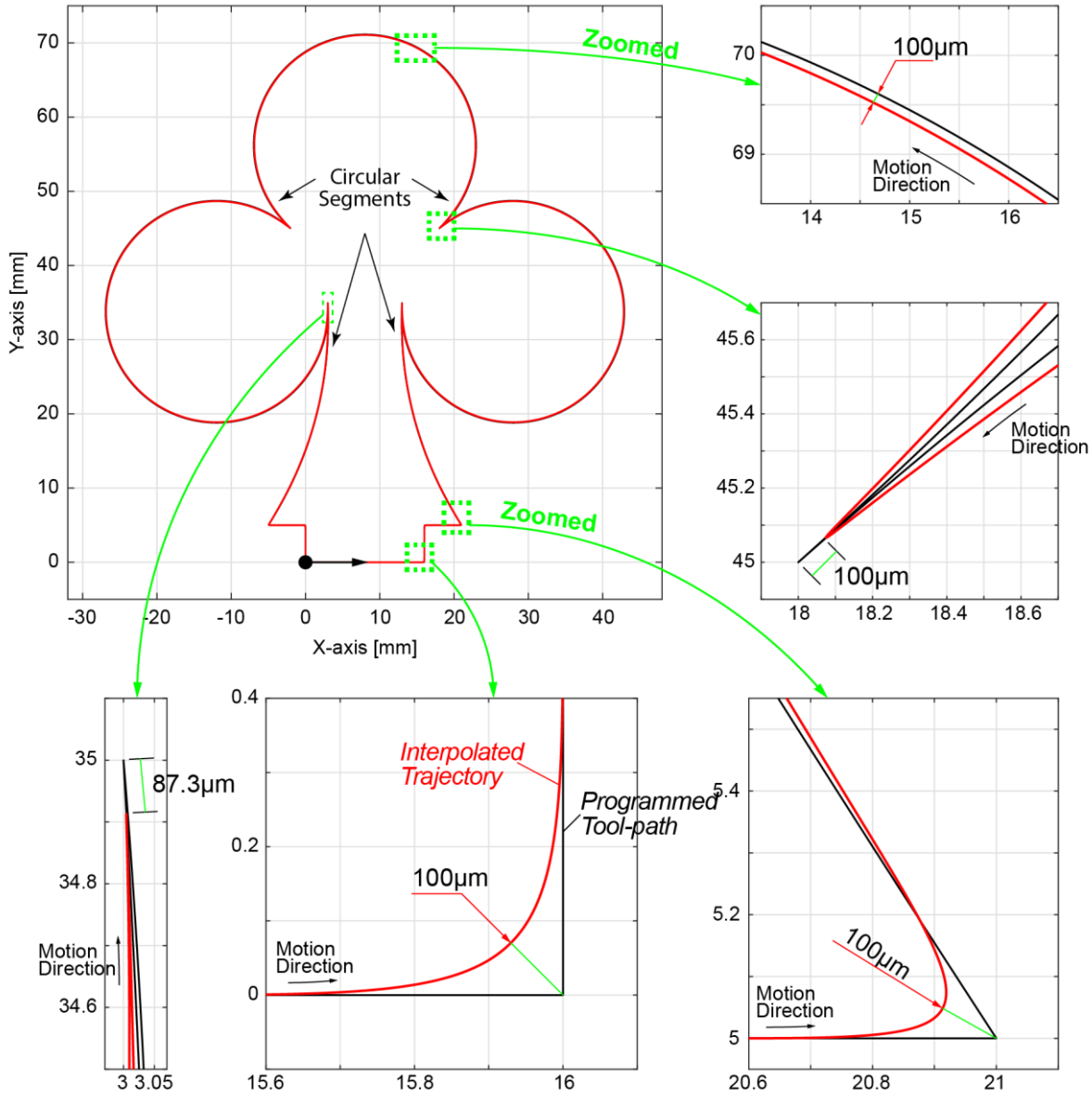

Figure 15: Clover shaped tool-path

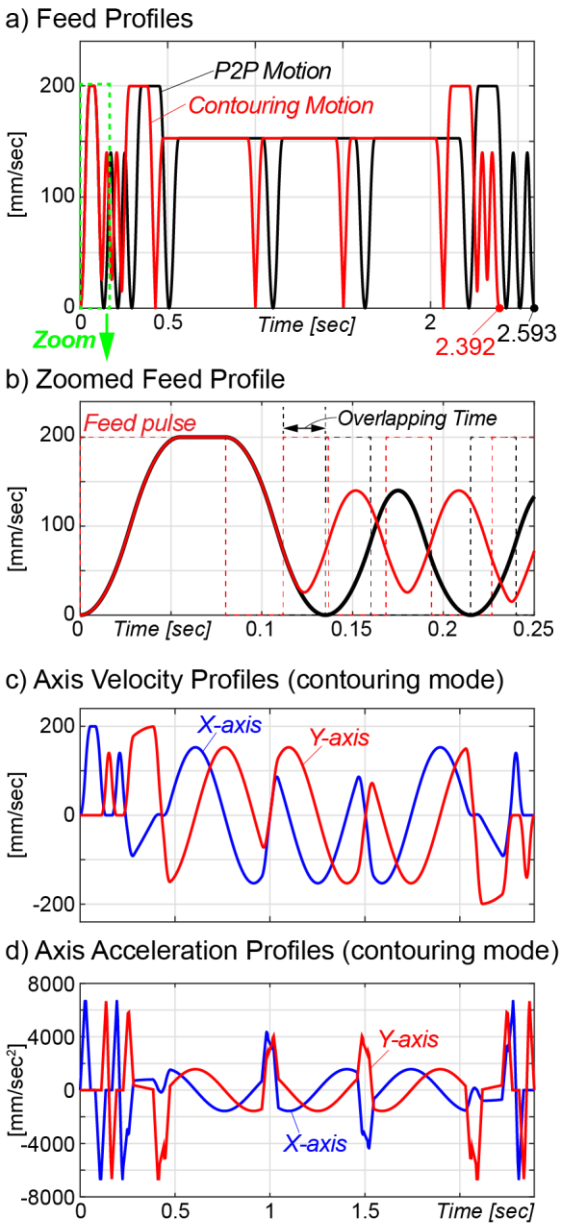

Figure 16: Interpolated kinematic profiles along clover shaped tool-path

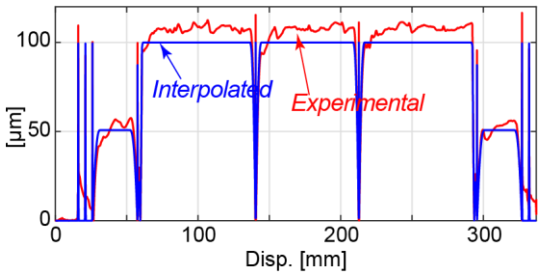

Figure 17: Contour errors during clover tool-path

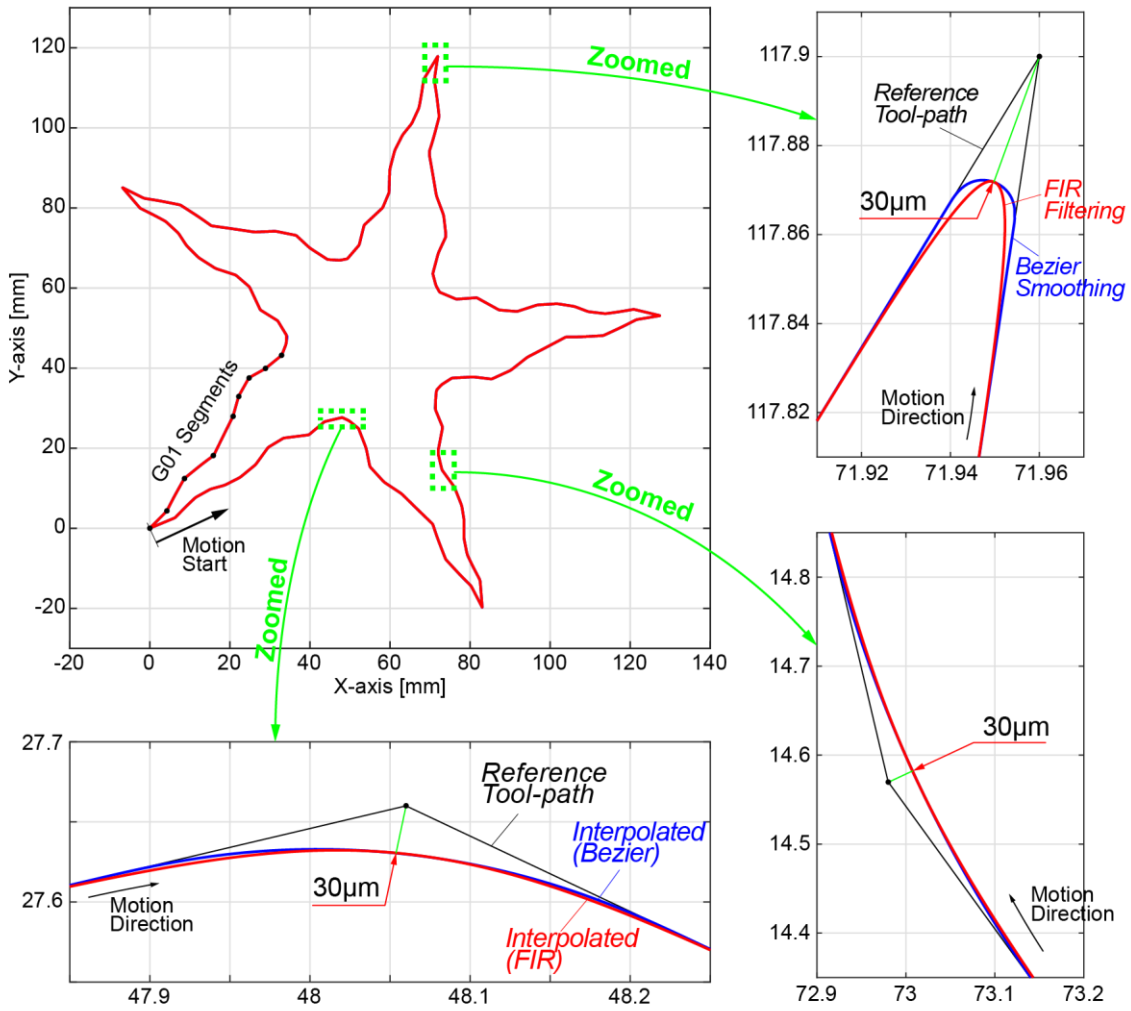

Figure 18: Starfish shaped tool-path

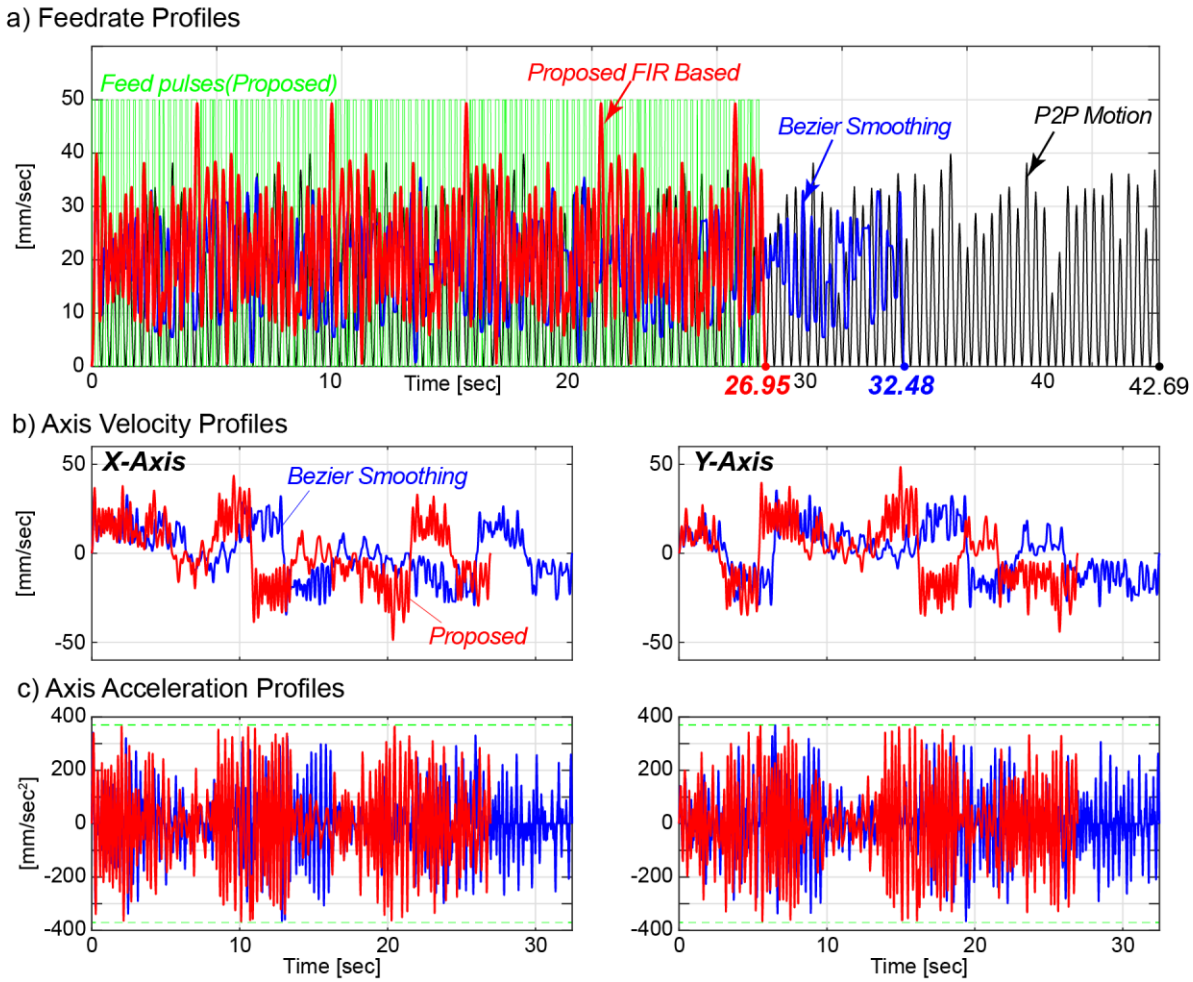

Figure 19: Interpolated kinematic profiles along starfish shaped tool-path

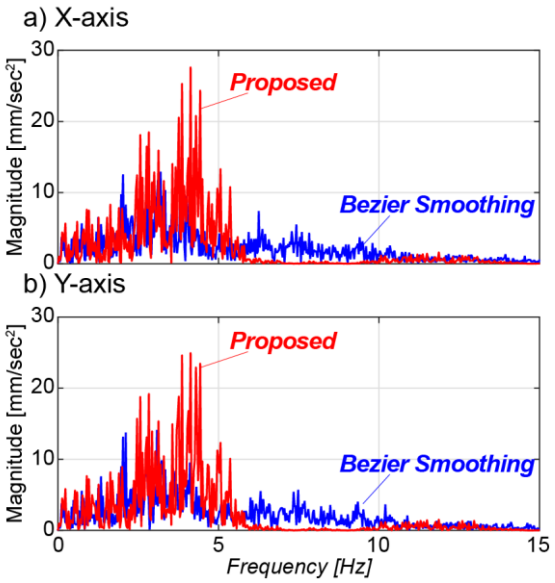

Figure 20: DFT of interpolated axis acceleration profiles

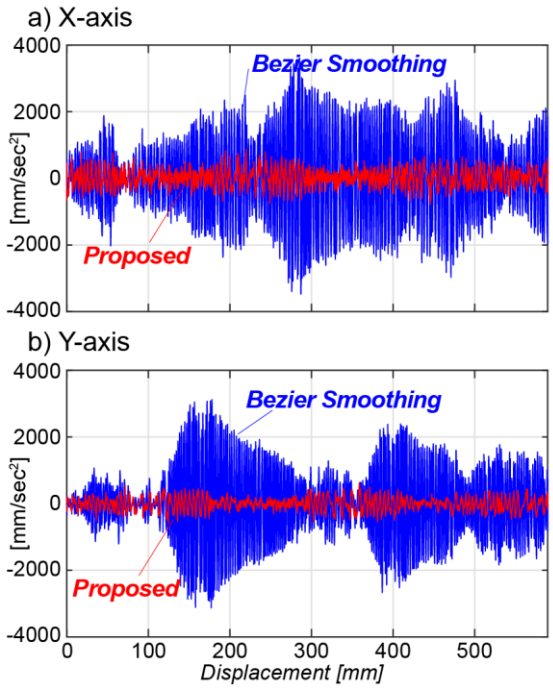

Figure 21: Experimentally measured beam accelerations

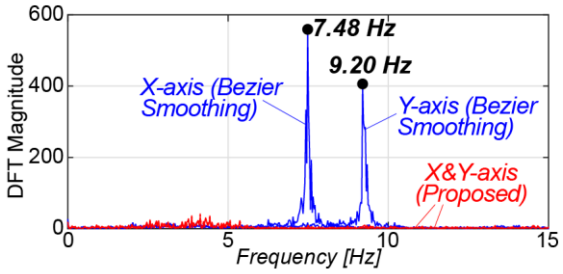

Figure 22: DFT of beam accelerations

6. Conclusions

A novel online trajectory generation scheme has been proposed for Cartesian machines and motion systems to generate high-speed and accurate feed motion. Owing to its simple filtering structure, proposed scheme can interpolate linear and circular paths with high kinematic continuity and minimum computational load making it suitable for real-time processors. The proposed block timing technique considers the change in the feed direction and the total delay in the filter chain to generate accurate non-stop rapid feed motion. For the first time, interpolation errors that occur during both linear and circular segment transitions as well as circular arcs are considered making the proposed scheme comprehensive for multi-segmented paths. Furthermore, by tuning the filter delays with respect to the dynamics of the motion system, frequency spectrum of the acceleration profile is shaped and unwanted residual vibrations are avoided. Experimental results validated that the proposed technique can interpolate multi-segmented tool-paths accurately. As compared to the state of the art technique, the proposed interpolation method can eliminate unwanted vibrations and reduce the cycle time up to ~20% while utilizing same level of acceleration proving it to be a practical and effective online interpolation technique form modern NC systems.

References

[1] Altintas, Y., 2000, Manufacturing Automation: Metal Cutting Mechanics, Machine Tool Vibrations, and CNC Design, Cambridge University Press.

[2] Choi, Y. K., & Banerjee, A. (2007). Tool path generation and tolerance analysis for free-form surfaces. International Journal of machine Tools and manufacture, 47(3), 689-696.

[3] Barre, P. J., Bearee, R., Borne, P., & Dumetz, E. (2005). Influence of a jerk controlled movement law on the vibratory behaviour of high-dynamics systems. Journal of Intelligent and Robotic Systems, 42(3), 275-293.

[4] Erkorkmaz, K., & Altintas, Y. (2001). High speed CNC system design. Part I: jerk limited trajectory generation and quintic spline interpolation. International Journal of machine tools and manufacture, 41(9), 1323-1345.

[5] Erkorkmaz, K., Alzaydi, A., Elfizy, A., & Engin, S. (2011). Time-optimal trajectory generation for 5-axis on-the-fly laser drilling. CIRP Annals-Manufacturing Technology, 60(1), 411-414.

[6] Flash, T., & Hogan, N. (1985). The coordination of arm movements: an experimentally confirmed mathematical model. The journal of Neuroscience, 5(7), 1688-1703.

[7] Sencer, B., Altintas, Y., & Croft, E. (2008). Feed optimization for five-axis CNC machine tools with drive constraints. International Journal of Machine Tools and Manufacture, 48(7), 733-745.

[8] Lambrechts motion, P., Boerlage, M., & Steinbuch, M. (2005). Trajectory planning and feedforward design for electromechanical systems. Control Engineering Practice, 13, 145–157.

[9] Rymansaib, Z., Iravani, P. and Sahinkaya, M.N., 2013, July. Exponential trajectory generation for point to point motions. In Advanced Intelligent Mechatronics (AIM), 2013 IEEE/ASME International Conference on (pp. 906-911). IEEE.

[10] Visioli, A., 2000. Trajectory planning of robot manipulators by using algebraic and trigonometric splines. Robotica, 18(6), pp.611-631.

[11] Singhose, W. E., Searing, W. P., & Singer, N. C. (1996). Improving repeatability of coordinate measuring machines with shaped command signals. Precision Engineering, 18(2), 138-146.

[12] Singer, N. C., & Seering, W. P. (1990). Preshaping command inputs to reduce system vibration. Journal of Dynamic Systems, Measurement, and Control, 112(1), 76-82.

[13] Altintas, Y. and Khoshdarregi, M.R., 2012. Contour error control of CNC machine tools with vibration avoidance. CIRP annals-manufacturing technology, 61(1), pp.335-338.

[14] Jeon JW, Kim Y-K. FPGA based acceleration and deceleration circuit for industrial robots and CNC machine tools. Mechatronics 2002;12:635–42.

[15] Jones S, Goodall R, Gooch M. Targeted processor architectures for high-performance controller implementation. Contr Eng Pract 1998;6:867–78.

[16] Osornio-Rios, R.A., de Jesus Romero-Troncoso, R., Herrera-Ruiz, G. and Castañeda-Miranda, R., 2007. Computationally efficient parametric analysis of discrete-time polynomial based acceleration–deceleration profile generation for industrial robotics and CNC machinery. Mechatronics, 17(9), pp.511-523.

[17] Kim, D.I., Jeon, J.W. and Kim, S., 1994. Software acceleration/deceleration methods for industrial robots and CNC machine tools. Mechatronics, 4(1), pp.37-53.

[18] Bonfe , M., & Secchi, C. (2010). Online smooth trajectory planning for mobile robots by means of nonlinear filters. In 2010 IEEE/RSJ international conference on intelligent robots and systems (IROS) (pp. 4299–4304).

[19] Zanasi, R., Bianco, C. G. L., & Tonielli, A. (2000). Nonlinear filter for the generation of smooth trajectories. Automatica, 36, 439–448.

[20] Zanasi, R., & Morselli, R. (2003). Discrete minimum time tracking problem for a chain of three integrators with bounded input. Automatica, 39(9), 1643–1649.

[21] Zheng, C., Su, Y., & Muller, P. (2009). Simple online smooth trajectory generations for industrial systems. Mechatronics, 19, 571–576.

[22] Chen, C.S. and Lee, A.C., 1998. Design of acceleration/deceleration profiles in motion control based on digital FIR filters. International Journal of Machine Tools and Manufacture, 38(7), pp.799-825. Vancouver

[23] Biagiotti, L. and Melchiorri, C., 2012. FIR filters for online trajectory planning with time-and frequency-domain specifications. *Control Engineering Practice*, *20*(12), pp.1385-1399.

[24] Sencer, B., Ishizaki, K., & Shamoto, E. (2015). High speed cornering strategy with confined contour error and vibration suppression for CNC machine tools. CIRP Annals-Manufacturing Technology, 64(1), 369-372.

[25] Tsai, M. S., & Huang, Y. C. (2016). A novel integrated dynamic acceleration/deceleration interpolation algorithm for a CNC controller. The International Journal of Advanced Manufacturing Technology, 1-14.

[26] Ogata, K. and Yang, Y., 1970. Modern control engineering.

[27] Wiener, N. (1988). The Fourier integral and certain of its applications. CUP Archive.

[28] Sencer, B., Ishizaki, K., & Shamoto, E. (2015). A curvature optimal sharp corner smoothing algorithm for high-speed feed motion generation of NC systems along linear tool paths. The International Journal of Advanced Manufacturing Technology, 76(9-12), 1977-1992.

[29] Erkorkmaz, K., Yeung, C. H., & Altintas, Y. (2006). Virtual CNC system. Part II. High speed contouring application. International Journal of Machine Tools and Manufacture, 46(10), 1124-1138.

[30] Goto, S., Nakamura, M. and Kyura, N., 1995, May. Trajectory generation of industrial mechatronic systems to achieve accurate contour control performance under torque saturation. In Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on (Vol. 3, pp. 2401-2406). IEEE.

[31] Yeh, S. S., & Hsu, P. L. (2004). Perfectly matched feedback control and its integrated design for multiaxis motion systems. Journal of Dynamic Systems, Measurement, and Control, 126(3), 547-557.