論文 / 著書情報
Article / Book Information

Title	KINEMATIC LOCAL CORNER SMOOTHING OF 5-AXIS LINEAR TOOL-PATHS
Authors	Shingo Tajima, Burak Sencer
Citation	Proceedings of the International Symposium on Flexible Automation, Vol. 2018, pp. 217-224
Pub. date	2018

Proceedings of 2018 ISFA
2018 International Symposium on Flexible Automation
Kanazawa, Japan, 15–19 July, 2018

ISFA2018-L020

KINEMATIC LOCAL CORNER SMOOTHING OF 5-AXIS LINEAR TOOL-PATHS

Shingo Tajima
School of Mechanical, Industrial and
Manufacturing Engineering
Oregon State University
219A Dearborn Hall
Corvallis, OR, 97331
tajimas@oregonstate.edu

Burak Sencer
School of Mechanical, Industrial and
Manufacturing Engineering
Oregon State University
219A Dearborn Hall
Corvallis, OR, 97331
burak.sencer@oregonstate.edu

ABSTRACT

5-axis machine tools are widely utilized in manufacturing complex sculptured components. Typically, tool-paths for 5-axis machine tools are discontinuous and they must be smoothened to generate continuous motion. This paper proposes a novel real-time interpolation algorithm for 5-axis machine tools and industrial robots to generate continuous and rapid feed motion along discrete point-to-point tool-paths by locally blending tool position and orientation, i.e. tool-pose, within user-specified tolerances. Proposed algorithm utilizes a filtering technique to smoothen pose trajectory of the tool and continuously interpolate 6DOF motion in real-time accurately. Tool-pose errors are controlled locally at junction points, i.e. corners, of the tool-path by timing consecutive feed commands precisely. Kinematic limits of the machine axes are considered in determining maximum feedrate along consecutive moves. Simulation studies show that proposed algorithm can interpolate 5-axis machining tool-paths accurately within kinematic limits in a computationally effective and real-time suitable scheme.

INTRODUCTION

Parts for aerospace, and die-and-mold industries contain complex sculptured geometries that are designed on CAD systems using smooth parametric curves such as splines and NURBS-curves [1]. In machining along those complex sculptured surfaces, the tool-tip and tool orientation needs to be controlled separately to generate the desired surface finish [2]. 5-axis machine tools are capable of controlling the 6 degrees of freedom (DOF) tool motion, and thus they are heavily utilized in today's advanced manufacturing industries.

In the current state of the art, CAM systems discretize original smooth workpiece surface with a series of short point-to-point (P2P) moves [3]. These discrete moves are only position continuous, and they don't facilitate continuous interpolation of the tool motion through the junction points, i.e. corners of consecutive moves. Therefore, tool motion has to undergo an instantaneous stop to change the feed direction. Or, those consecutive moves need to blended employing computationally expensive curve fitting, so called the blending, techniques [4]-[7].

In 3-axis machining, local blending of consecutive moves, i.e. corner smoothing or path blending approach is well-established and already programmed into the NC units of most Cartesian machine tools. For instance, high order splines such as quintic Bezier or a NURBS curves [4][5] are used to blend Cartesian tool-tip motion between consecutive P2P moves. Fitting of C2 continuous blending curves is necessary to generate an acceleration continuous motion transition. The path error introduced during blending must be controlled to ensure that the manufacturing accuracy is not violated. Along a fitted path, kinematic profiles are planned to attain a near time-optimal and smooth motion [6]. This overall process of accurate geometric blending curve fitting and motion planning motion is a computationally stringent trajectory generation scheme [7]. Recently, computationally efficient filtering approaches are also proposed. These approaches based on filtering the discrete velocity commands so that the interpolate trajectory is smooth and continuous. The control of interpolation errors due to filtering is critical for accurate machining operation [8].

For 5-axis CNC machines, on the other hand, accurate blending of discrete moves is more challenging [9]. The tool-path contains both translational motion of the tool center point (TCP), and also the rotational motion of the tool axis. Both the translational motion of the tool in Cartesian coordinates, and the tool orientation in spherical coordinates [10] must be blended continuously [11] in a synchronized matter [12]. This is known as the synchronized tool-pose interpolation [11][14]. Current machine tool literature tries to address this "tool-pose" blending

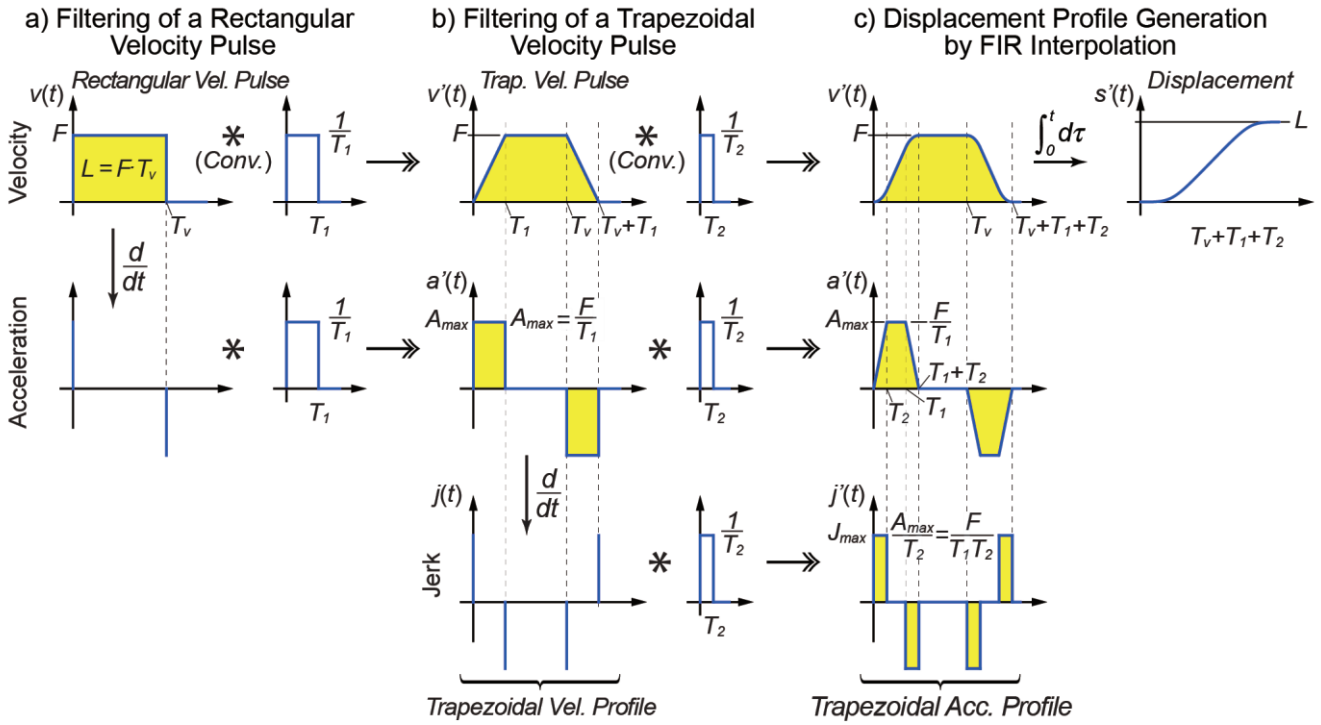

Figure 1: Trajectory generation using chain of FIR filters.

problem in 2-steps where TCP is blended using high order splines. Then, tool orientation is blended using spherical B-Splines [13] in spherical coordinates. The 6DOF motion of the tool is realized through synchronous interpolation of both curves. However, this 2-step approach is rather computationally expensive and interpolation contouring errors are difficult to control [12]. Furthermore, synchronization of linear and rotary tool motion is inefficient and may lead to feed fluctuations that in return generates poor surface finish [14]. Recent efforts are directed towards utilizing two parametric curves fitted in Cartesian coordinates to interpolate tool orientation and realize a more robust continuous interpolation [15].

A part from the machine tool literature, robotics literature addressed accurate and continuous tool-pose interpolation problem using quaternions and direct continuous interpolation of transformation matrices [16][17]. However, those mathematical frameworks are not suitable for accurate control of blending errors. Blending in joint coordinate systems is not a feasible approach for accurate control of blending errors due to the non-linear robot kinematics [17].

This paper proposes a new method to continuously interpolate TCP and tool orientation along series of P2P moves based on the Finite Impulse Response (FIR) filtering. FIR filters can be implemented as moving average filters and provide a computationally efficient framework for real-time implementation [18]. Proposed technique allows accurate control of TCP and angular blending errors. Analytical expressions are generated and simulation studies verify the effectiveness of the methods. To the authors knowledge, this is the first attempt to utilize a simple filtering approach for smooth and accurate interpolation of 5-axis tool-paths.

FIR FILTERING BASED TRAJECTORY GENERATION

FIR filters can generate a smooth trajectory in real-time by its characteristic as a moving average filter. This section introduces the FIR filtering techniques for generating high order reference motion commands and also continuous interpolation of P2P Cartesian tool motion [8].

Smooth trajectory generation based on FIR filtering

Firstly, FIR filters provide an effective means for generating high-order point-to-point (P2P) Cartesian trajectories [18]. Figure 1 briefly depicts how chain of FIR filters can be utilized to generate acceleration and jerk limited single axis P2P trajectories.

A 1st order FIR filter with a unity gain can be defined in Laplace domain (s) as:

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

where T_i is the time constant which could also be called the delay of the filter. The impulse response of the FIR filter is defined as:

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

and when a pulse velocity command $v(t)$ at a magnitude of F is convolved with FIR filter for a duration of $T_v > T_i$, filtered velocity $v'(t)$ is calculated as:

$$v'(t) = v(t) * m(t) = \frac{1}{T_i}\int_0^t \left(\begin{bmatrix} v(\tau) - v(\tau - T_v) \end{bmatrix} \begin{bmatrix} u(t-\tau) - u(t-T_i-\tau) \end{bmatrix} \right) d\tau$$

$$= \frac{1}{T_i} \begin{bmatrix} \int_0^t v(\tau)u(t-\tau)d\tau - \int_0^t v(\tau)u(t-T_i-\tau)d\tau \\ -\int_0^t v(t-T_v)u(t-\tau)d\tau \\ +\int_0^t v(t-T_v)u(t-T_i-\tau)d\tau \end{bmatrix} \quad (3)$$

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

As noticed, the FIR filter acts as moving average and integrates commanded velocity pulse. The resultant velocity profile $v'(t)$ becomes a trapezoidal waveform and acceleration $a'(t)$ becomes rectangular pulse (See Fig. 1). Peak acceleration is controlled by the filter's time constant, $A_{max} = F/T_i$. Due to the formal definition of convolution, the total cycle time is sum of the input velocity pulse duration and the filter's time constant, $T_{total} = T_v + T_i$. Duration of the velocity pulse, on the other hand, is computed based on the travel distance L as $T_v = L/F$. Since FIR filter has unity gain, the area underneath velocity pulse does not change and thus integration of $v'(t)$ satisfies target position command $s'(t)$.

If a second FIR filter is added to the chain, it convolves rectangular acceleration signal. The jerk profile becomes rectangular pulse (See Fig. 1), and its maximum value depends on the 2nd FIR filter's time constant, i.e. $J_{max} = A_{max}/T_2$. Total travel time is further stretched by the sum of all filter delays in the chain as:

$$T_d = T_1 + T_2 + \cdots + T_n \quad (5)$$

FIR filter based Cartesian corner smoothing

Previous section presented the framework for how exact P2P moves are generated by chain of FIR filters. As dictated by Eq. (5), the filter delays, T_i, elongate total motion duration. As a result, if consecutive P2P moves are planned, a "*dwell*" duration "T_{dwell}" which equals to the total filter delay T_d needs to be added in between the moves in order to wait for the filter delay. Therefore, the motion undergoes a full-stop, and so the tool can linearly travel along consecutive waypoints. This scheme can be implemented by FIR filtering of velocity pulses in axis level.

In most motion systems, such as machine tools and robotic manipulators, continuous motion is required to minimize cycle time. In this case, linear moves must be blended within a tolerance band. In the proposed approach, this can be achieved by commanding consecutive axis velocity pulses without fully waiting for the FIR filter delay to die out.

Figure 2 shows two linear P2P motion commands interpolated for a 2D Cartesian case. If consecutive moves are interpolated with a dwell time of the filter delay computed from (5), a sharp cornered geometry with zero blending error can be realized (See Fig. 2a-b). On the other hand, if consecutive commands are continuously interpolated through FIR filter with no dwell time, a non-stop motion can be achieved while leaving

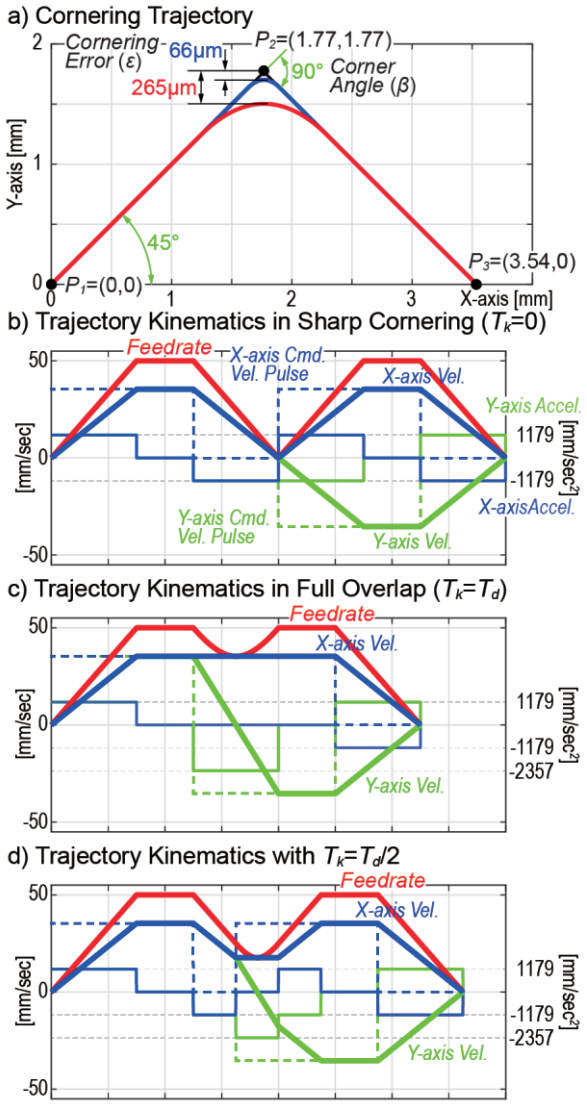

Figure 2: Blending error control for 2D Cartesian case.

a large error around the junction point (corner) of P2P moves (See Fig. 2c).

In the proposed technique, corner blending error (ε) is controlled by adjusting the dwell time between consecutive P2P motion commands. As shown in Fig. 2d, we time the consecutive move's velocity pulse and call it the "overlapping time" (OT). The OT, T_k controls the blending errors at the junction (corner) point. The cornering error ε occurs in the middle of the cornering transition if consecutive velocity pulse magnitudes are identical. Thus, the maximum cornering error can be expressed analytically as a function of filter delay T_i, corner angle β, and the overlapping time T_k from Eq. (3)-(4). For instance, if a single FIR with delay T_1 is used to filter X and Y axis velocity pulses with the magnitude of F, a trapezoidal velocity profile is generated and blending error is controlled analytically as a function of OT as:

$$\varepsilon = \sqrt{\frac{F^2 T_k^4 (1-\cos(\beta))}{32T_1^2}} \rightarrow T_k = \sqrt[4]{\frac{32T_1^2\varepsilon^2}{F^2(1-\cos(\beta))}} \quad (6)$$

and for 2 FIR filters with delays T1 and T2, acceleration profile becomes jerk limited and OT, T_k, is calculated as:

$$T_k = \begin{cases} \sqrt[3]{\dfrac{24T_1T_2\varepsilon}{F\sin(\beta/2)}} & , 0 \le T_k \le 2T_2 \\ T_2 + \sqrt{\dfrac{4T_1\varepsilon}{F\sin(\beta/2)} - \dfrac{T_2^2}{3}} & , 2T_2 < T_k \le T_1 + T_2 \end{cases} \quad \text{for } T_1 > T_2 \tag{7}$$

As given in Eq. (6) and (7), for a desired corner blending error, OT can be determined analytically by the proposed technique. Following sections present the extension to control blending errors along 5-axis machining tool-paths.

FIR FILTER BASED 6DOF INTERPOLATION

This section presents the FIR filtering techniques for generating smooth trajectory along a linear 6DoF motion command and the local corner smoothing technique to generate a continuous motion profile along a series of linear 6DoF segments by using the FIR filters.

FIR filter based 6DoF interpolation

A typical 5-axis machine or industrial robot motion command in workpiece coordinate system is given by a sequence of discrete P2P moves [12]. Each tool position is defined by three Cartesian coordinates of its tool center point (TCP), $\boldsymbol{P} = [P_x, P_y, P_z]$, and angular orientation vector of the cutter axis, $\boldsymbol{O} = [O_i, O_j, O_k]$. The technique presented in the previous section can be used to continuously interpolate translational tool motion in 3D Cartesian coordinates, but the tool orientation cannot be interpolated. To overcome this, 6DoF motion of the tool is generated by interpolating $\boldsymbol{P}$ and a second set of Cartesian position commands $\boldsymbol{Q} = [Q_x, Q_y, Q_z]$ along the tool axis. This approach is presented in Fig. 3. As shown, $\boldsymbol{Q}$ is generated by offsetting P along orientation vectors:

$$\vec{\mathbf{O}} = \frac{\mathbf{Q}-\mathbf{P}}{\|\mathbf{Q}-\mathbf{P}\|} = \frac{\mathbf{Q}-\mathbf{P}}{H} \tag{8}$$

where H is the offset distance. The offset distance can be determined based on the length of the tool. As shown in Fig. 3, synchronized interpolation of translational and rotation tool motion is achieved by simultaneously interpolating velocity pulses along the P and Q-paths, F_p and F_q, through chain of FIR filters. Duration of the velocity pulses for P-path is calculated by the user set feedrate F and length of P2P moves. F_q is either selected as $F_q = F_p$, or it is modulated so that tool rotational motion is synchronized with the translation. Accurate P2P motion is then generated by introducing a dwell between the moves, i.e. by setting overlapping time (OT), $T_k = T_d - T_{dwell}$. Figure 4 illustrates 6 DoF motion planning scheme with proposed FIR filtering technique.

FIR filter based 6DoF continuous interpolation

To generate an accurate continuous motion, blending errors for the tool center point (TCP) and tool axis orientation must be controlled. Figure 5 shows definition of TCP and orientation blending errors. As shown, TCP blending errors (ε_p) occur due to blending of the P-path. ε_p can be controlled through the technique presented in the section by modifying Eq. (7) for the 3D Cartesian case and hence the overlapping time T_k with the blending position error ε_p is computed as:

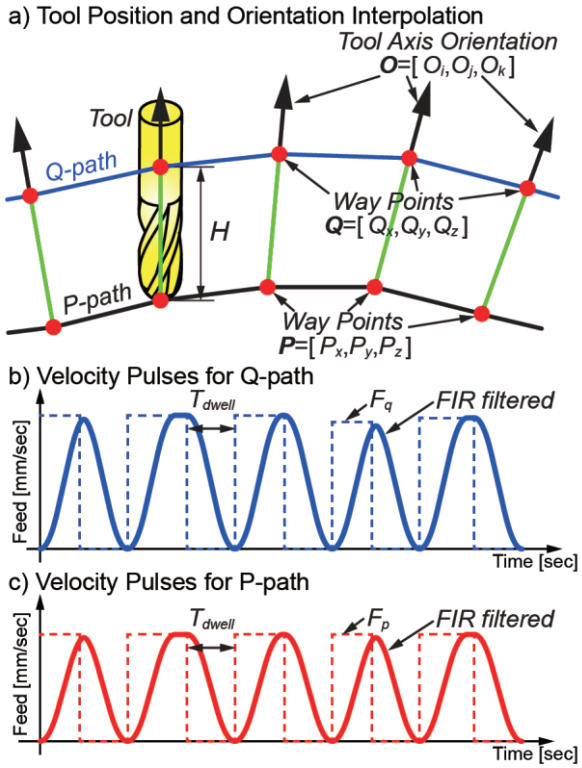

Figure 3: 6DoF interpolation of 5-axis paths.

$$T_{k,\varepsilon_p} = \begin{cases} \sqrt[3]{\dfrac{48T_1T_2\varepsilon_p}{\left\|F_{p2}\mathbf{t}_{\mathbf{p2}} - F_{p1}\mathbf{t}_{\mathbf{p1}}\right\|}} & , 0 \le T_k \le 2T_2 \\ T_2 + \sqrt{\dfrac{8T_1\varepsilon_p}{\left\|F_{p2}\mathbf{t}_{\mathbf{p2}} - F_{p1}\mathbf{t}_{\mathbf{p1}}\right\|} - \dfrac{T_2^2}{3}} & , 2T_2 < T_k \le T_1 + T_2 \end{cases} \tag{9}$$

where, F_{p1} and F_{p2} are velocity pulses along consecutive P2P moves. $\boldsymbol{t_{p1}}$ and $\boldsymbol{t_{p2}}$ are unit vectors representing motion directions between the moves. Cartesian blending errors for the Q-path, ε_q, can be computed in a similar fashion from Eq. (9) by replacing F_p with F_q and $\boldsymbol{t_p}$ with $\boldsymbol{t_q}$. The orientation blending error ε_θ is defined from Fig. 5 as:

$$\varepsilon_\theta = \arccos\left(\frac{\mathbf{O}_2 \cdot \mathbf{O}_m}{\|\mathbf{O}_2\|\|\mathbf{O}_m\|}\right) \tag{10}$$

where $\boldsymbol{O_2}$ is the original tool axis orientation vector at the corner, and $\boldsymbol{O_m}$ is the tool axis orientation vector at the midpoint of the corner blend computed in terms of the blending errors of the P and Q-paths as:

$$\mathbf{O}_m = \boldsymbol{\varepsilon}_q + \mathbf{O}_2 - \boldsymbol{\varepsilon}_p \tag{11}$$

The overlapping time T_k to satisfy orientation errors ε_θ is then calculated using Eq. (9)-(11) as:

$$aT_{k,\varepsilon_\theta}{}^2 + bT_{k,\varepsilon_\theta} + c = 0, \quad \text{if } 0 \le T_k \le 2T_2$$

$$\begin{cases} a = |\mathbf{M}|^2\cos^2(\varepsilon_\theta) - (\mathbf{O}_2 \cdot \mathbf{M})^2 \\ b = 2\left(\cos^2(\varepsilon_\theta) - 1\right)(\mathbf{O}_2 \cdot \mathbf{M}) \\ c = \cos^2(\varepsilon_\theta) - 1 \end{cases} \tag{12}$$

$$\text{where} \quad \mathbf{M} = \frac{\left(F_{q2}\mathbf{t}_{\mathbf{q2}} - F_{q1}\mathbf{t}_{\mathbf{q1}}\right) - \left(F_{p2}\mathbf{t}_{\mathbf{p2}} - F_{p1}\mathbf{t}_{\mathbf{p1}}\right)}{48T_1T_2}$$

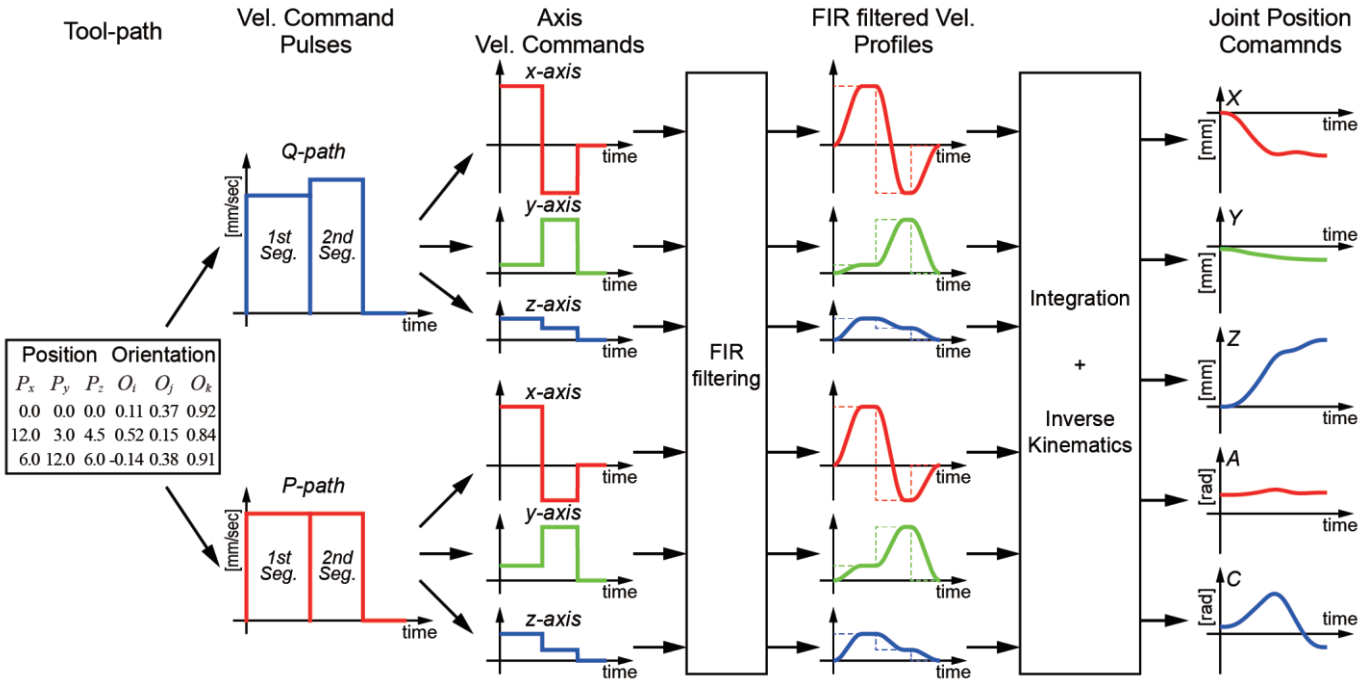

Figure 4: 6-DoF motion planning scheme with proposed FIR filtering technique.

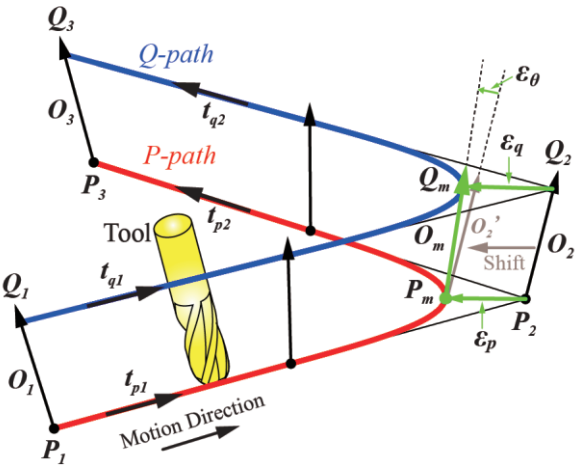

Figure 5: TCP and tool orientation blending errors.

and

$$aT_{k,\varepsilon_\theta}{}^4 + bT_{k,\varepsilon_\theta}{}^3 + cT_{k,\varepsilon_\theta}{}^2 + dT_{k,\varepsilon_\theta} + e = 0\,,$$

$$\text{if } 2T_2 \le T_k \le T_1 + T_2$$

$$\begin{cases} a = 9f\,, \quad b = -36T_2 f\,, \quad c = 60T_2^2 f + 6g \\ d = -48T_2^3 f - 12T_2 g\,, \quad e = 16T_2^4 f + 8T_2^2 g + h \end{cases} \tag{13}$$

$$\text{where} \quad \begin{cases} f = |\mathbf{M}|^2 \cos^2(\varepsilon_\theta) - (\mathbf{O_2}\cdot\mathbf{M})^2 \\ g = \left(\cos^2(\varepsilon_\theta) - 1\right)(\mathbf{O_2}\cdot\mathbf{M}) \\ h = \cos^2(\varepsilon_\theta) - 1 \end{cases}$$

$$\text{and} \quad \mathbf{M} = \frac{\left(F_{q2}\mathbf{t_{q2}} - F_{q1}\mathbf{t_{q1}}\right) - \left(F_{p2}\mathbf{t_{p2}} - F_{p1}\mathbf{t_{p1}}\right)}{24T_1}$$

The overlapping time $T_{k,\varepsilon\theta}$ which satisfies the orientation error tolerance is obtained from Eqs. (12) and (13). Finally, to satisfy both the TCP and orientation blending errors, the smaller of the overlapping time (OT) is selected from Eq. (9) and Eqs. (12)-(13). Notice that smaller OT introduces larger dwell time between consecutive velocity pulses to satisfy the worst case of the errors, which can be either the TCP, i.e. position, or the orientation component.

Incorporation of axis kinematic limits

Each axis, drive, of the machine has its own kinematic limits, such as the velocity, acceleration and jerk limitations. The amplitude of command velocity pulse must be selected so that when the tool motion is interpolated, it does not violate axis kinematic limits and so the drives are not saturated. This adjustment is rather computationally expensive and requires kinematic model of the machine and iterations between the command velocity pulses in workpiece coordinate system (WCS) and each axis kinematics in joint coordinate system (JCS) by using the forward and inverse kinematics [19]. To avoid this iterative calculation technique, amplitudes of command velocity pulses are adjusted based on the characteristic of FIR filters as follows.

$$\alpha = \max\left\{1, \max\left\{\frac{|v_i(t)|}{V_{\max,i}}\right\}, \sqrt{\max\left\{\frac{|a_i(t)|}{A_{\max,i}}\right\}}, \sqrt[3]{\max\left\{\frac{|j_i(t)|}{J_{\max,i}}\right\}}\right\}, \tag{14}$$

$$0 \le t \le T_v + T_d$$

where α is a scaling parameter which adjusts the command pulse width, T_v by scaling the velocity pulses, F_p and F_q, by (1/α). i depicts axes/joints of 5-axis CNC machine. v_i, a_i and j_i are the velocity, acceleration and jerk of joint i, respectability. $V_{max,i}$, $A_{max,i}$ and $J_{max,i}$ are axis/joint velocity, acceleration and jerk limits.

The α in Eq. (14) adjusts the commanded velocity pulse during the cruise period based on the relationship between each drive's limits and each kinematic profile. However, as shown in Fig. 1, acceleration and jerk profiles may have larger amplitudes

during acceleration and deceleration sections. Therefore, in addition to α of Eq. (14), an extra scaling parameter α^* is used to consider the acceleration and jerk limits during the acceleration and deceleration period.

$$\alpha^* = \max\left\{1, \max\left\{\frac{|a_i(\tau)|}{A_{\max,i}}\right\}, \max\left\{\frac{|j_i(\tau)|}{J_{\max,i}}\right\}\right\}, \quad \begin{cases} 0 \le \tau \le T_d \\ T_v \le \tau \le T_v + T_d \end{cases} \tag{15}$$

Then, the larger value of α of Eq. (14) or α^* of Eq. (15) adjusts the command velocity pulse width and amplitude. By optimizing α and α^* during acceleration and cruise sections of the tool-path joint kinematic limits can be incorporated into the interpolation scheme. Note that, as α and α^* become larger than 1 the velocity pulses become smaller and overall motion is elongated. Thus, those scaling parameters are stretching the overall cycle time.

SIMULATION RESULTS

This section presents an illustrative example of the proposed dual FIR filtering based 6DoF motion blending technique applied on 5-axis tool-paths. Typically, 5-axis machine tools have 3-Cartesian axes complemented with 2 rotary axes to control the tool orientation. In this study, kinematics of a 5-axis machine tool is considered with rotary and tilting axis placed on the workpiece side. Corresponding kinematic transformations can be obtained conveniently from existing literature [14].

Figure 6 illustrates a simple test tool-path composed of two consecutive P2P moves. Commanded position and orientation vectors are listed in Table 1. Chain of 2 FIR filters are used to interpolate the P and Q-paths to generate the jerk limited acceleration profile in workpiece coordinates with the tool offset distance H = 1 [mm]. Filter time constants (delays) are set to T_1 = 100 [msec] and T_2 = 60 [msec]. User programmed feedrate for the tool-path is set to F_p = 30 [mm/sec]. Velocity, acceleration and jerk limits of the Cartesian axes (X, Y and Z) are set to V_{Tmax} = 30 [mm/sec], A_{Tmax} = 500 [mm/sec^2] and J_{Tmax} = 10000 [mm/sec^3] respectively. Kinematic limits of the rotational axes (A and C) are set to V_{Rmax} = 5 [rad/sec], A_{Rmax} = 50 [rad/sec^2] and J_{Rmax} = 1000 [rad/sec^3] respectively. To generate continuous motion along the original tool-path, the user specified maximum position and orientation blending tolerances are set to ε_p = 300 [µm] and ε_θ = 0.03 [rad].

Figure 6 shows the interpolated smooth path. As shown, TCP blending error is limited to exactly 300 [µm] with the cornering overlapping time of T_k = 125.9 [msec]. Maximum orientation blending error, on the other hand, is simulated to be 15.4 [mrad], which is less than the set maximum tolerance value. This is due to the fact that the proposed technique cannot satisfy both tolerances exactly. Only the worst-case scenario is considered from Eq. (9) and (12)-(13). Figure 7 shows interpolated kinematic velocity profiles of P and Q-paths in the workpiece coordinate system (WCS). As observed, although desired feedrate of P-path was originally set to F_p = 30 [mm/sec], it is reduced to 28.2 [mm/sec] and 26.2 [mm/sec] on the consecutive segments using Eqs. (14)-(15). Similarly, the velocity pulse of Q-path, F_q, is also reduced from its set value of 30 [mm/sec] down to 29.4 [mm/sec] and 27.8 [mm/sec] on the consecutive segments. Therefore, joint velocity and acceleration limits are respected, and the rotational motion is synchronized with the translation. The dwell time, T_{dwell}, between velocity pulses delivered to the FIR filters can be observed clearly Fig. 7. Finally, joint kinematic profiles are generated and shown in Fig. 8. As shown, the acceleration continuous non-stop motion is

TABLE I. COMMANDED WAY-POINTS OF POSITION AND ORIENTATION

Way-Point	Tool tip Position [mm] (*P*-path)			Tool axis Orientation Vector		
	P_x	P_y	P_z	O_i	O_j	O_k
1	0.0	0.0	0.0	-0.108	-0.365	0.925
2	12.0	3.0	4.5	0.518	-0.155	0.841
3	6.0	12.0	6.0	0.142	0.381	0.914

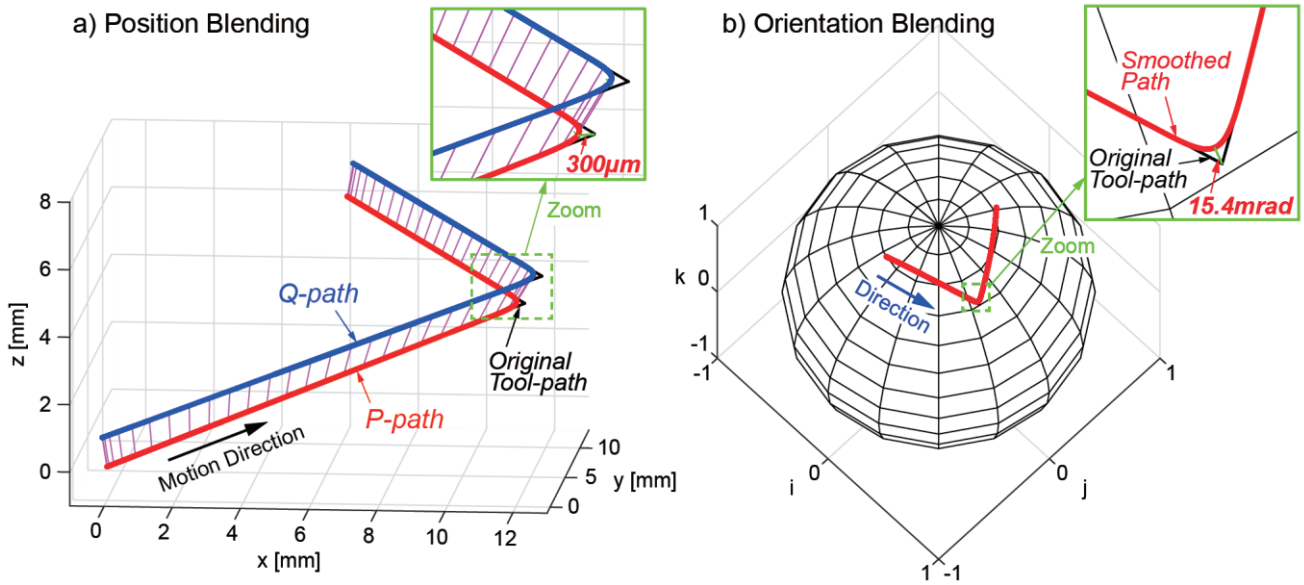

Figure 6: Tool-path position and orientation interpolation.

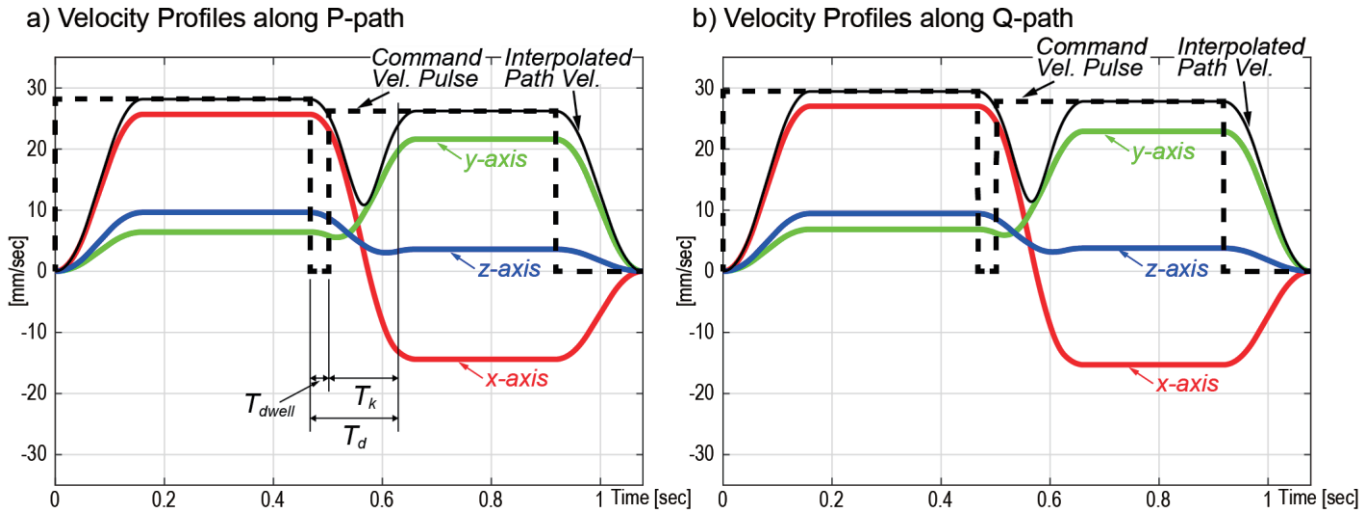

Figure 7: Kinematic profiles in work coordinate system.

interpolated within the kinematic limits of the drives and user defined position and orientation tolerances.

CONCLUSIONS

This paper proposed a novel corner smoothing technique for 5-axis machine tools based on the dual FIR filter interpolation. The proposed technique smoothly interpolates TCP and tool axis orientation simultaneously. The simulation result verify that the proposed technique can blend the sharp corner with respect to predefined position and orientation tolerances within the kinematic limits of the drives. Experimental implementation and validations are planned for future work.

ACKNOWLEDGMENTS

We thank Brother Industries, LTD. for funding and supporting this research project.

REFERENCES

[1] L. Piegl, W. Tiller, 2003, "The NURBS Book," (2nd Edition) Springer-Verlag, Berlin Heidelberg.

[2] A. Lasemi, D.Y. Xue, P.H. Gu, 2010, "Recent development in CNC machining of freeform surfaces: A state-of-the-art review," Computer-Aided Design, 48, 227-241.

[3] 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.

[4] 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.

[5] Duan, M., & Okwudire, C., 2015, "Minimum-time cornering for CNC machines using an optimal control method with NURBS parameterization," The International Journal of Advanced Manufacturing Technology, 1-14.

[6] L. Zhang, Y. You, J. He, X. Yang, 2011, "The transition algorithm based on parametric spline curve for high-speed machining of continuous short line segments," The International Journal of Advanced Manufacturing Technology, 52, pp. 245–254

[7] M. Heng, K. Erkorkmaz, 2010, "Design of a NURBS interpolator with minimal feed fluctuation and continuous feed modulation capability," International Journal of Machine Tools and Manufacture 50, 281–293.

[8] Tajima, S., Sencer, B., & Shamoto, E., 2017, "Accurate Interpolation of Machining Tool-paths Based on FIR Filtering," Precision Engineering (Under press).

[9] Y. R. Hwang, C. S. Liang, 1998, "Cutting error analysis for spindle-tilting type five-axis NC machines," The International Journal of Advanced Manufacturing Technology, 14, 399–405.

[10] E. B. Dam, M. Koch and M. Lillholm, 1998, "Quaternions, Interpolation and Animation," Department of Computer Science, University of Copenhagen, Tech. Rep. DIKU-TR-98/5.

[11] R.V. Fleisig, A.D. Spence, 2001, "A constant feed and reduced angular acceleration interpolation algorithm for multi-axis machining," Computer Aided Design 33 (1), 1–15.

[12] X. Beudaert, S. Lavernhe, C. Tournier, 2013, "5-axis local corner rounding of linear tool path discontinuities," International Journal of Machine Tools & Manufacture, 73, 9-16.

[13] W. Ge, Z. Huang, G. Wang, 2007, "Interpolating solid orientations with a C2-Continuous B-Spline quaternion curve," Technologies for E-Learning and Digital Entertainment 4469, 606–615

[14] S. Tulsysn, Y. Altintas, 2015, "Local toolpath smoothing for five-axis machine tools," International Journal or Machine Tools & Manufacture, 96, 15-26.

[15] J. Shi, Q. Bi, L.M. Zhu, et al, 2015, "Corner rounding of linear five-axis tool path by dual PH curves blending," International Journal of Machine Tools & Manufacture, 88, 223–236.

[16] K. Shoemake, 1985, "Animating rotation with quaternion curves," SIGGRAPH Computer Graphics 19 (3), 245–254.

[17] M.-C. Ho, Y.-R. Hwang, C.-H. Hu, 2003, "Five-axis tool

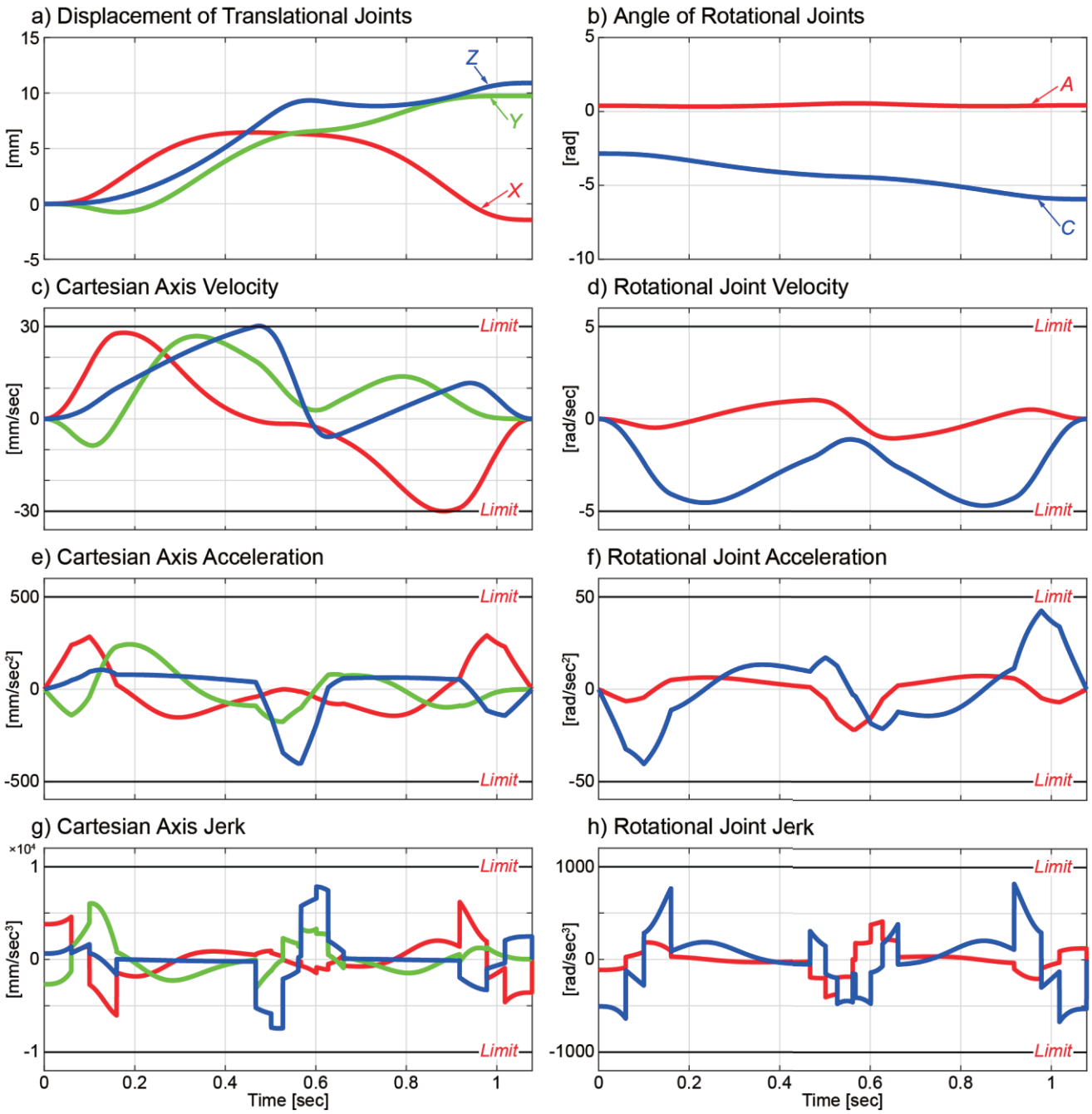

Figure 8: Kinematic profiles in machine (joint) coordinate system.

orientation smoothing using quaternion interpolation algorithm," International Journal of Machine Tools and Manufacture 43 (12), 1259–1267.

[18] 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.

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