Proceedings Paper:

Ward, R. orcid.org/0000-0002-6201-0285, Sencer, B., Panoutsos, G. orcid.org/0000-0002-7395-8418 et al. (1 more author) (2022) On-The-Fly CNC interpolation using frequency-domain FFT-based filtering. In: Carpanzano, E., Boer, C. and Valente, A., (eds.) Procedia CIRP. 55th CIRP Conference on Manufacturing Systems - CMS2022, 29 Jun - 01 Jul 2022, Lugano, Switzerland. Elsevier , pp. 1571-1576.

https://doi.org/10.1016/j.procir.2022.05.193

Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 107 (2022) 1571–1576

www.elsevier.com/locate/procedia

55th CIRP Conference on Manufacturing Systems

On-The-Fly CNC interpolation using frequency-domain FFT-based filtering

Rob Ward[a,c], Burak Sencer[b], George Panoutsos[a], Erdem Ozturk[c]

[a]*Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, S1 3JD, UK*
[b]*Mechanical Engineering Department, Oregon State University, Corvallis, OR, USA*
[c]*Advanced Manufacturing Research Centre, University of Sheffield, Rotherham, S60 5TZ, UK*

* Corresponding author. *E-mail address:* r.ward@amrc.co.uk

Abstract

Finite Impulse Response filtering is increasingly becoming the interpolation method of choice in modern computer numerically controlled (CNC) machining centres. Recently, On-The-Fly (OTF) interpolation was introduced using direct convolution to adaptively change the FIR filter time constant throughout the toolpath. In this paper, it is shown that computational advantages can be gained by using frequency domain methods instead. This paper introduces a novel on-the-fly CNC interpolation method using Fast Fourier Transforms (FFTs). The presented OTF FFT method demonstrates up to 10 times increase in computational speed than the direct convolution OTF method. The effectiveness of the proposed method is validated in simulation based case studies.

Keywords: CNC Interpolation ; FFT Convolution; Motion Control;

1. Introduction

Interpolation of CNC toolpaths using Finite Impulse Response (FIR) filtering has been proven a computationally efficient method of trajectory generation. Existing FIR based methods use direct convolution in the discrete domain to filter reference commands and generate smooth feed drive reference trajectories. Within these methods, the FIR filter time constant, which determines the kinematic properties of the reference signal, is calculated from the maximum commanded feedrate set by the part program. Therefore, for toolpaths with varying feedrate commands (such as feedrate scheduled toolpaths), the FIR filter is sub-optimal for all but the maximum feedrate commands leading to slower drive responses and longer overall machining cycle times. One method to overcome this limitation is to use On-The-Fly (OTF) interpolation and change the kinematic properties of the reference signal online. OTF interpolation optimises the reference signal for each change in feedrate along the toolpath and reduces the overall machining cycle time. Optimising the kinematics throughout the toolpath requires changing the FIR filter time constant online. However, this imposes significant computational and practical implementation challenges. To date, all FIR-based filtering methods use the same FIR filter time constants throughout the toolpath and interpolate using direct convolution of a single input signal with the FIR filter [10, 4, 3, 7]. Using methods from real-time audio processing [1] we address the implementation challenges twofold by 1) presenting a method of adapting the FIR filter time constants online and 2) significantly improving the computational speed of the NC interpolation.

By segmenting each cutter location (CL)-line of the G-code, the input signals can be split and filtered separately by an optimised FIR filter followed by signal reconstruction to form the full smoothed reference signals. The segmented filtering can be achieved either by direct convolution or via frequency domain methods. The convolution theorem states that multiplication in the frequency domain is equivalent to convolution in the time domain [6], and as such the use of Fast Fourier Transforms (FFTs) can offer greater performance computationally for real-time Numerical Control (NC) interpolation. The advantages of fast FFT convolution are seen when the original signals are long and subsequently segmented into smaller blocks. This is the case for circular convolution using the block transform method. The segmented filtered output signals are concatenated with an overlapped section where the values are added, this is termed Overlap-Add (OLA) signal reconstruction [6][1]. The result is the linear interpolation of the original input signal.

10.1016/j.procir.2022.05.193

Unlike previous research [12][9], no single continuous pulse train signals are generated and subsequently filtered. The CL-blocks are individually filtered using frequency domain methods and converted back into the time domain. The OLA method of signal reconstruction is used to control the cornering feedrates which in turn are used to control the tool centre point (TCP) blending tolerances. The FIR filter is designed separately for each individual CL command and this provides on-the-fly filtering to optimise acceleration and jerk, resulting in a reduced machining cycle time compared to standard FIR based filtering. In this paper, we propose FFT based convolution as a computationally efficient method interpolating NC toolpaths whilst simultaneously handling kinematic and tolerance constraints.

2. Application to CNC Interpolation

This research implements the circular convolution (block transform) method to linearly interpolate each individual velocity pulse separately and reconstruct the final input signal from the convolved parts. The block transform method applied to CNC interpolation is presented in the following sections for point-to-point (P2P) and non-stop continuous TCP motion.

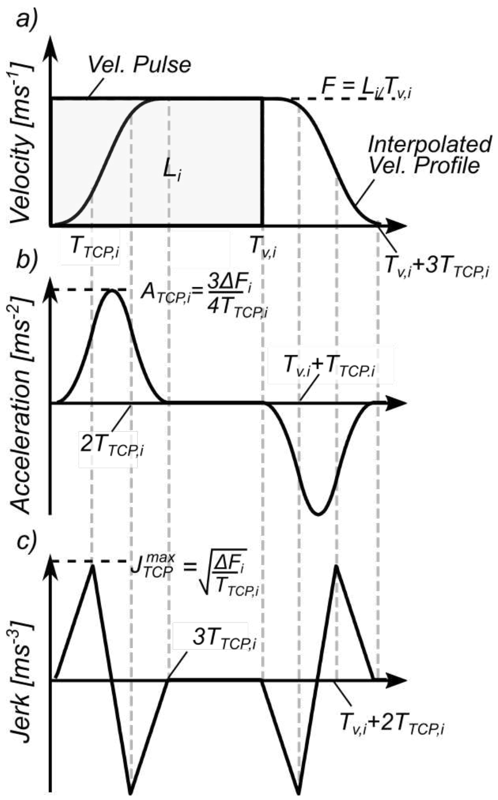

Fig. 1. Kinematic profiles generated by FIR filtering

A P2P linear G01 move commands the tool motion from the starting TCP position $\mathbf{P}_s = \left[P_{s,x}, P_{s,y}, P_{s,z}\right]^T$ to the final TCP position defined by $\mathbf{P}_e$. In 3-axis machining the machining feedrate F is defined as the linear speed of the TCP. For each G01 command denoted by subscript i, where $i = [1, ..., N_{G01}]$ and N_{G01} is the number of G01 commands, the segmented input signals $u_i[n]$ are represented by a vector of discrete feedrate values F_i:

$$u_i[n] = [F_i, F_i, F_i, \ldots, F_i]\,, \tag{1}$$

where $u_i[n] \in \mathbb{R}^{1\mathrm{x}N_{v,i}}$, $N_{v,i} = T_{v,i}/T_s$, $T_{v,i} = L_i/F_i$ and $L_i = \left\|\mathbf{P}_{e,i} - \mathbf{P}_{s,i}\right\|_2$.

By interpolating feedrate signals using FIR-based interpolation, it has been shown that filtering the input signal 3 times using matching FIR filter time constants the filter response approaches that from a Gaussian filter [12] as shown in Fig.1. The FIR filter time constant is derived analytically from either feed drive acceleration or jerk limit. The FIR filter signal $h_i[n]$ is constructed as:

$$h_i[n] = \frac{1}{N_{f,i}}[1, 1, 1, \ldots, 1]\,, \tag{2}$$

where $h_i[n] \in \mathbb{R}^{1\mathrm{x}N_{f,i}}$, $N_{f,i} = T_{TCP,i}/T_s$, T_s is the interpolation sampling time and

$$T_{TCP,i} = \max\left\{\frac{3\Delta F_i}{4A_{TCP}^{\max}}, \sqrt{\frac{\Delta F_i}{J_{TCP}^{\max}}}\right\}. \tag{3}$$

It is therefore $h_i(n)$ which controls the kinematic properties of the final interpolated signal based on $T_{TCP,i}$. Which, unlike in previous research where a single T_{TCP} was applied globally to the whole convolution operation based on a worst case feedrate, $T_{TCP,i}$ is now selected independently for each individual G01 command based on the segment feedrate F_i. Therefore each individual G01 response is optimised to ensure the maximum kinematic performance. This allows "On-The-Fly" (OTF) filtering to change the machine tool kinematic response throughout the toolpath in real-time.

Next, in order to conduct FFT operations the segmented input signals $u_i(n)$ and FIR filter response vectors $h_i(n)$ are each padded with zeros to length M_i to prevent aliasing [5] when using overlap-add convolution. As a minimum, $M_i = N_{v,i} + 3N_{f,i} - 1$, which covers the entire G01 travel length. However for computational efficiency, the length M_i should equal a power of 2 ($M_i = 2^x$, where $x = \text{ceiling}(\log_2(N_{v,i} + 3N_{f,i}))$, and account for the FIR filter to be applied 3 times to the input signal to generate a smooth C^3 continuous velocity signal.

$$u_{i,zp}[n] = [[F_i, F_i, F_i, \ldots, F_i], [0, 0, 0, \ldots, 0]]\,, \tag{4}$$

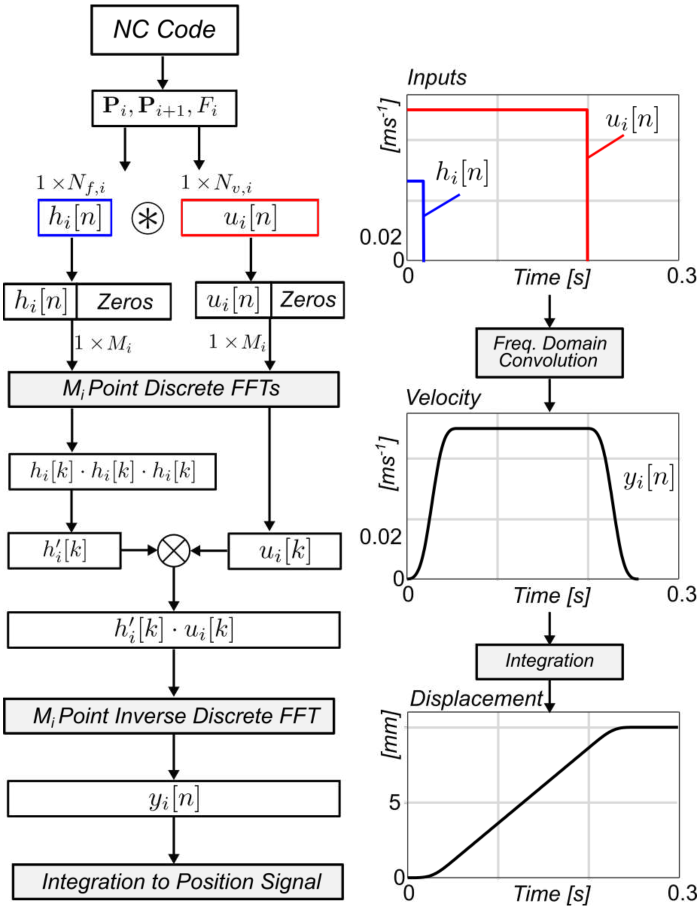

Fig. 2. FFT-based interpolation method applied to a single CL-line

$$h_{i,zp}[n] = \frac{1}{N_{f,i}} \left[[1, 1, 1, \ldots, 1], [0, 0, 0, \ldots, 0]\right], \tag{5}$$

where subscript zp denotes a zero padded vector.

The zero padded input signal $u_{i,zp}[n]$ and impulse response signal $h_{i,zp}[n]$ are transformed using M_i-point Discrete FFTs:

$$u_i[k] = FFT_{M_i}\left(u_{i,zp}[n]\right), \tag{6}$$

$$h_i[k] = FFT_{M_i}\left(h_{i,zp}[n]\right). \tag{7}$$

First, the impulse response $h_i[k]$ is multiplied 3 times in the frequency domain to achieve the equivalent 3 filter convolution:

$$h_i'[k] = h_i[k] \cdot h_i[k] \cdot h_i[k], \tag{8}$$

where $h_i'[k]$ represents the triple FIR filter. Next $h_i'[k]$ and $u_i[k]$ are multiplied in the frequency domain to achieve convolution:

$$y_i[k] = h_i'[k] \cdot u_i[k]. \tag{9}$$

Finally, an M_i-point inverse FFT is applied to $y_i[k]$ to generate the convolved segmented signal in the discrete time domain:

$$y_i[n] = IFFT_{M_i}\left(y_i[k]\right), \tag{10}$$

where $y_i[n] \in \mathbb{R}^{1\mathrm{x}M_i}$. The resulting output signal $y_i(n)$ has been extended to M_i when compared to the segmented input signal $u_i[n]$ of length $N_{v,i}$. Only the first $N_{v,i} + 3N_{f,i}$ values of the segmented signals are utilised in the signal reconstruction, therefore the segmented output signals are truncated and the remaining values (all zeros) are discarded. Finally, smoothed segmented velocity signal (10) is integrated to generate the interpolated position command. For P2P motion the individual position signals are concatenated end to end to generate the final reconstructed output signal. The end to end reconstruction generates an instantaneous stop at the start and end of each CL-line resulting in P2P motion.

Continuous TCP motion is achieved when the individual output signals are concatenated with an overlap to yield the final reconstructed output signal [11]. In traditional OLA methods of signal reconstruction the final $3N_{f,i} - 1$ values of the output signal $y_i[n]$ would be added to the first $3N_{f,i} - 1$ values of the next output signal $y_{i+1}[n]$, hence the term "overlap-add". The overlap can be used as a method of constraining TCP contouring errors by controlling the cornering feedrates between two successive CL-lines. The overlap is analytically determined based on consecutive feedrates, FIR filter time constants (set to satisfy acceleration and jerk limits) and TCP blending tolerance (and tool orientation blending tolerance in 5-axis machining). The generalised reconstructed reference signals are denoted by:

$$y'(n) = y_i'(n) \overset{T_{o,i}}{\bigcap} y_{i+1}'(n) \overset{T_{o,i+1}}{\bigcap} y_{i+2}'(n) \ldots y_{N_{G01}}'(n), \tag{11}$$

where $\overset{T_{o,i}}{\bigcap}$ represents the overlap of $T_{o,i}$ between adjacent segments during signal reconstruction.

To demonstrate the proposed method a simple corner motion commanded by the G-code given in Fig.3 is generated by OTF FIR interpolation and compared with standard FIR interpolation. The individual velocity vectors are defined by (1), the interpolated velocity profiles are generated using (10) and reconstructed using (11). The overlap time T_o is calculated analytically to satisfy the user defined TCP tolerance constraint ε_{TCP}. The overlap time is a function of commanded feedrates F_i, drive kinematic constraints (acceleration and/or jerk limits), cornering angle θ_{TCP} and tolerance ε_{TCP} [11]. Note, the toolpath is commanded at 2 different feedrates, therefore, for the OTF method each motion is kinematically optimised (velocity and acceleration profiles shown in Figs.3c and d respectively). The adaptive FIR filter time constant (3) is shown in Fig.3d. The overall cycle time for the standard method is 0.603s compared with 0.589 for the OTF method, which is a 2.32% reduction in cycle time. Further cycle time reductions are witnessed in full toolpaths. Finally, integration of the smoothed axis velocity profiles results in the interpolated TCP trajectory which satisfies TCP tolerance as presented in Fig.3a.

In practical implementation, the interpolated position reference signals are generated and then sent to the motion control system, therefore any locally applied feed overrides would behave as standard and change the TCP feedrate as commanded by the operator.

3. Illustrative Case Studies

This section presents two case studies to demonstrate the computational efficacy of the FFT based OTF interpolation method. Each of the 2 case study toolpaths were interpolated 10^3 times to determine the average computation speeds.

3.1. 5-Axis Spiral Semi-Finishing Toolpath

The first case study is based on a highly discritised 5-axis spiral toolpath as shown in Fig.4. The toolpath is commanded at 2000mm/min and composed of 6919 G01 commands where the average CL-line is 1.28 mm in length. The toolpath is interpolated on-the-fly using 3 different methods. The first uses segmented direct convolution implemented through difference equations, the second uses FFT based convolution, and the final method once again uses FFT convolution but with optimised input lengths. The computations were conducted on a 64-bit Intel(R) Xenon(R) 3.00GHz CPU. Table 1 shows the results from case study 1 and Figure 5 shows the resulting average computation times throughout the toolpath for each CL.

Inspecting Table 1, the mean computation time per output for the direct convolution method implemented via difference equations is 3.5×10^{-6} seconds. The two FFT methods are 3.7×10^{-7} and 3.4×10^{-8} seconds, corresponding to a 9.3 and 10.2 times improvement for the FFT and optimised FFT methods respectively.

3.2. 3-Axis Roughing Toolpath

The second case study is based on a coarsely discritised 3-axis roughing toolpath to show the impact of CL discritisation on the interpolation process. The roughing toolpath is shown in Fig.6. The toolpath is commanded at 3000mm/min and is made up of 167 G01 commands averaging 112.3mm in length.

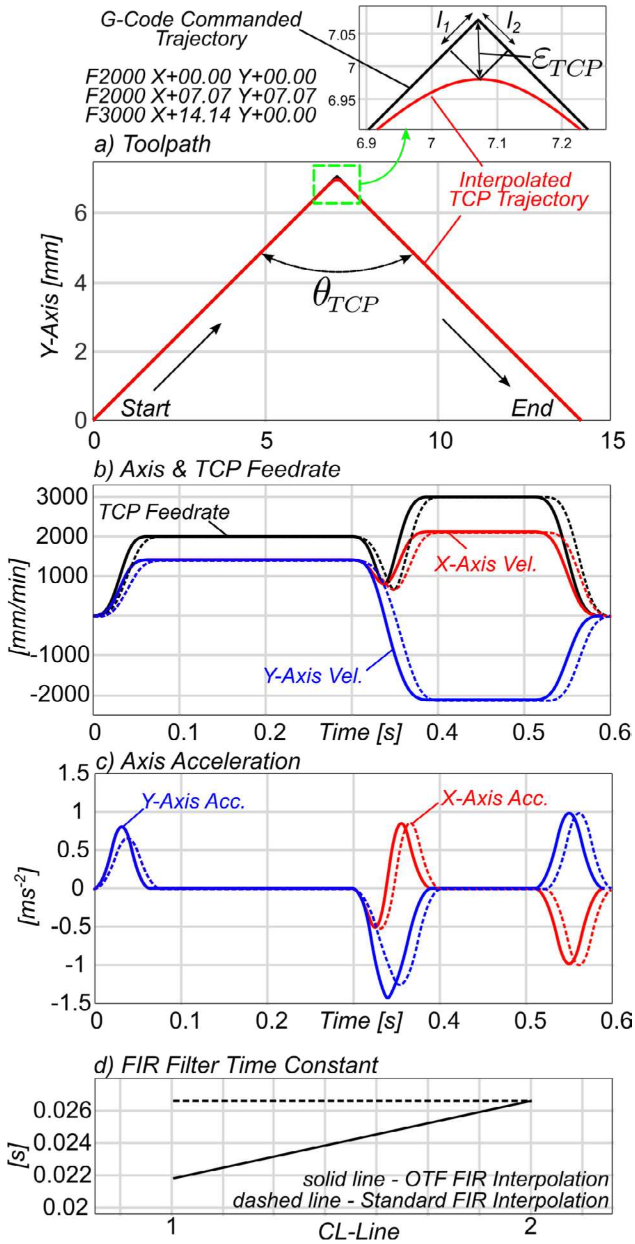

Fig. 3. Interpolated and commanded TCP Position, kinematic profiles and FIR filter time constants for standard and OTF FIR interpolation

Table 1. Comparison of Computational Cost of Convolution Method for 5-Axis Finishing Spiral Toolpath.

Convolution Method	Mean Time Per Output (s)	Improvement (x times)
DC - Difference Equation	3.5×10^{-6}	
FFT, $M_i = N_{v,i} + 3N_{f,i} - 1$	3.7×10^{-7}	9.3
FFT, $M_i = 2^x$	3.4×10^{-7}	10.2

From Table 2, the mean computation time per output for the direct convolution method implemented via difference equations is 5.0×10^{-6} seconds. The two FFT methods are 8.2×10^{-7} and 2.4×10^{-7} seconds, corresponding to a 6.1 and 10.6 times

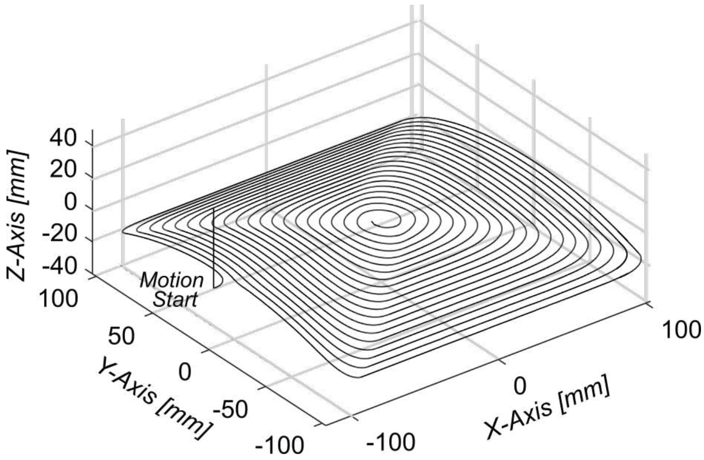

Fig. 4. Illustrative Case Study: Spiral Toolpath TCP Trajectory

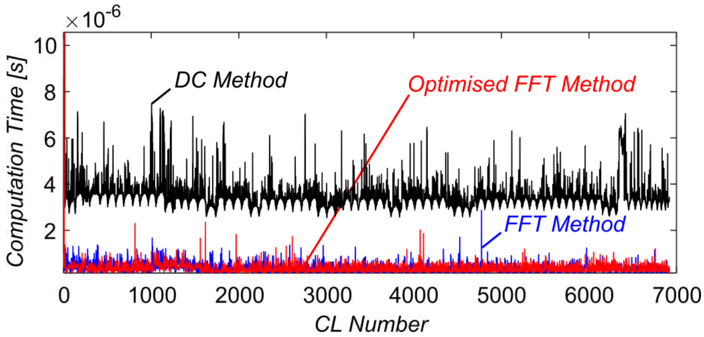

Fig. 5. Spiral toolpath - computation times

improvement for the FFT and optimised FFT methods respectively.

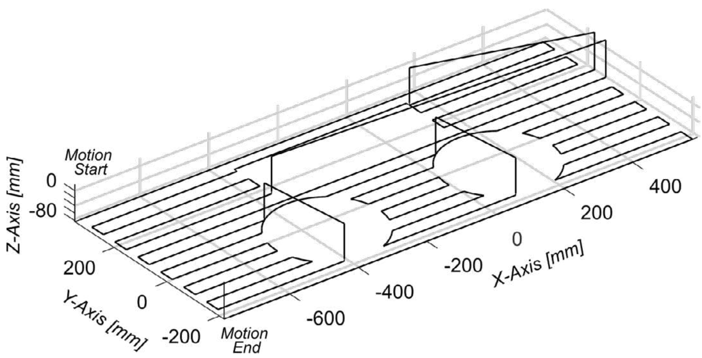

Fig. 6. Illustrative Case Study: Roughing Toolpath TCP Trajectory

Table 2. Comparison of Computational Cost of Convolution Method for 3-Axis Roughing Toolpath.

Convolution Method	Mean Time Per Output (s)	Improvement (x times)
DC - Difference Equation	5.0×10^{-6}	-
FFT, $M_i = N_{v,i} + 3N_{f,i} - 1$	8.2×10^{-7}	6.1
FFT, $M_i = 2^x$	2.4×10^{-7}	10.6.

4. Discussion

As demonstrated in section 3 there are significant advantages to using frequency domain methods for NC interpolation when compared to the difference equation implementation. Further advantages are gained when optimising the input lengths to base 2 integers. For short signals where $M_i < 64$ direct convolution offers faster implementation than FFT methods. However, for $M_i > 64$, FFT based convolution offers a significant computational advantage due to the efficiency of the Cooley-Tukey FFT algorithm [2] when calculating DFTs resulting in fewer calculations [8]. In fact, the computational cost of direct convolution using difference equations scales to $O\left(M_i^2\right)$ compared to the FFT method when M_i is a power of 2 scales to $O\left(M_i \log_2 M_i\right)$.

As interpolator clock cycles become faster it is important to note that the lengths of input and filter signals will increase accordingly. Therefore implementing computational efficient methods of NC interpolation is vital.

5. Conclusions

Until recently, published research in FIR based NC interpolation has not considered adapting the FIR filter time constant throughout the toolpath. Newly introduced On-The-Fly methods can optimise the kinematic properties of the tool motion for the whole toolpath rather than based on a worst case feedrate scenario. This paper has proposed a novel method of OTF interpolation which also addresses the real-time computational cost of the algorithm. The main findings of the work are:

- On-The-Fly interpolation of NC toolpaths can be conducted by segmenting input signals and individually linearly interpolating each CL-line using convolution in the frequency domain.
- FFT based methods of convolution demonstrate significantly faster computational times per interpolated output than direct convolution methods.
- FFT based methods can be further optimised to reduce time of computation by selecting input and filter lengths to match exponential to the base 2 whole numbers.

Further work will investigate the frequency domain approach to locally control the frequency spectrum of the interpolated position reference commands.

Acknowledgements

This research was supported by the Advanced Manufacturing Research Centre members and EPSRC (grant EP/L016257/1).

References

[1] Borgerding, M., 2006. Turning overlap-save into a multiband mixing, downsampling filter bank. IEEE Signal Process. Mag. 23, 158–161. doi:10.1109/MSP.2006.1598092.

[2] Cooley, J.W., Tukey, J.W., 1965. An algorithm for the machine calculation of complex Fourier series. Math. Comput. 19, 297–301.

[3] Jiang, Y., Han, J., Xia, L., Lu, L., Tian, X., Liu, H., 2020. A decoupled five-axis local smoothing interpolation method to achieve continuous acceleration of tool axis. Int. J. Adv. Manuf. Technol. 111, 449–470. doi:10.1007/s00170-020-05936-0.

[4] Liu, Y., Wan, M., Qin, X.B., Xiao, Q.B., Zhang, W.H., 2020. FIR filter-based continuous interpolation of G01 commands with bounded axial and tangential kinematics in industrial five-axis machine tools. Int. J. Mech. Sci. 169, 105325. URL: https://doi.org/10.1016/j.ijmecsci.2019.105325, doi:10.1016/j.ijmecsci.2019.105325.

[5] Oshana, R., 2006. DSP Software Development Techniques for Embedded and Real-Time Systems. Elsevier Science & Technology. doi:10.1016/B978-0-7506-7759-2.X5000-5.

[6] Smith, S.W., 2003. Moving Average Filters. Newnes, Boston. URL: http://www.sciencedirect.com/science/article/pii/B9780750674447500522, doi:10.1016/b978-0-7506-7444-7/50052-2.

[7] Song, D.N., Ma, J.W., Zhong, Y.G., Xiao, D., Yao, J.J., Zhou, C., 2021. A fully real-time spline interpolation algorithm with axial jerk constraint based on FIR filtering. Int. J. Adv. Manuf. Technol. 113, 1873–1886. doi:10.1007/s00170-021-06738-8.

[8] Strum, R.D., Kirk, D.E., 1988. First Principles of Discrete Systems and Digital Signal Processing. Addison-Wesley.

[9] Tajima, S., Sencer, B., 2019. Accurate real-time interpolation of 5-axis tool-paths with local corner smoothing. Int. J. Mach. Tools Manuf. 142, 1–15. URL: https://doi.org/10.1016/j.ijmachtools.2019.04.005, doi:10.1016/j.ijmachtools.2019.04.005.

[10] Tajima, S., Sencer, B., Shamoto, E., 2018. Accurate interpolation of machining tool-paths based on FIR filtering. Precis. Eng. 52, 332–344. URL: http://dx.doi.org/10.1016/j.precisioneng.2018.01.016, doi:10.1016/j.precisioneng.2018.01.016.

[11] Ward, R., 2022. Smooth Trajectory Generation for 5-Axis CNC Machine Tools. Ph.D. thesis. University of Sheffield.

[12] Ward, R., Sencer, B., Jones, B., Ozturk, E., 2021. Accurate prediction of machining feedrate and cycle times considering interpolator dynamics. Int. J. Adv. Manuf. Technol. 116, 417–438. doi:10.1007/s00170-021-07211-2.