CIRP Annals - Manufacturing Technology 72 (2023) 317–320

Contents lists available at ScienceDirect

CIRP Annals - Manufacturing Technology

journal homepage: https://www.editorialmanager.com/CIRP/default.aspx

Real-time trajectory generation for dual-stage feed drive systems

Shingo Tajima[a], Burak Sencer[b,*]

[a] Tokyo Institute of Technology, Yokohama, Japan
[b] Oregon State University, Corvallis, USA

Submitted by Mikel Zatarain, IDEKO, Spain

ARTICLE INFO

Article history:
Available online 26 April 2023

Keyword:
Computer numerical control (CNC)
Feed
Trajectory generation

ABSTRACT

Machine tools are designed using dual-stage feed drives with slow and fast stages when speed and accuracy must be delivered over large work-volume. The slow stage provides large-stroke positioning whereas the fast stage executes short-stroke accurate tool-tip placement. This paper presents a real-time trajectory generator (governor) that distributes reference motion commands so that coordinated high-speed dual-stage contouring motion can be realized. A novel dynamic command scaling (DCS) approach is developed to generate trajectories in real-time that fully utilize the stroke (range), velocity, acceleration and dynamic limits of the dual-stages drives. Demonstration in 2D laser patterning shows >2 times increase in productivity.

1. Introduction

To increase accuracy and productivity over large work volume, machine tool feed drives are designed with redundant actuation stages providing different bandwidths and operating ranges. Diamond turning machines equipped with Fast Tool Servos (FTS) are the most well-known applications of redundant dual (two)-stage feed drives [1,2]. Dual-stage feed drives are built typically on the axis-on-axis architecture (See Fig. 1a) where a large stroke slow stage carries a high-bandwidth, high-resolution short stroke fast stage. The fast stage provides high-bandwidth actuation within a short stoke, and it is typically used to compensate for the static and dynamic errors of the slow stage. Such functionality can be realized in practice by feeding the slow stage's errors as reference position commands to the fast stage [3].

More recently, dual-stage feed drives are used in multi-axis contouring, e.g. laser processing (cutting, patterning, sintering) and micro-machining [4] applications where fast stage is driven by ultra-high-speed galvo mirrors or magnetically levitated actuators reaching 3 to 5 g accelerations [5]. The overreaching goal in these applications is to use the fast stage not only to improve dynamic accuracy, but primarily to minimize the processing (cycle) times. For instance, small features should be processed rapidly by only actuating the fast stage, whereas large features are accessed by engaging the slow drive into motion, which continuously carries the fast drive to the vicinity of the target profile as shown in Fig. 1b and c. Real-time generation of reference trajectories that utilize slow and fast stages in such a coordinated manner while respecting their kinematic and dynamic limits is a challenge.

Dual-stage feed drives can be modeled as multi-input single-output systems subject to kinematic and dynamic constraints. Slow stage provides the large stroke motion with limited velocity, accelerations, and positioning bandwidth whereas the fast stage provides up to 10 times wider bandwidth and peak acceleration but only within a short stroke. Model predictive control (MPC) framework [6,7] can be used to incorporate these contradicting kinematic constraints and solve for optimal command distribution. However, command sampling frequencies in these applications reach 10 to 100 kHz, and thus MPC problems cannot be solved in real-time. Instead, complementary filtering [7] method has been used in practice. Reference trajectories are low-pass filtered and send to the slow stage, whereas fast stage receives the high-frequency signal. Since smooth trajectories contain limited energy in high frequency, fast stage moves only in short range. However, there is no guarantee that fast stage stroke limits will be respected. Although practical, the frequency separation based approach underutilizes actuator capability and may violate kinematic limits. Time-domain constraints must be imposed for safe operation.

This paper presents a novel real-time trajectory generation approach for dual-stage feed drives that jointly incorporate

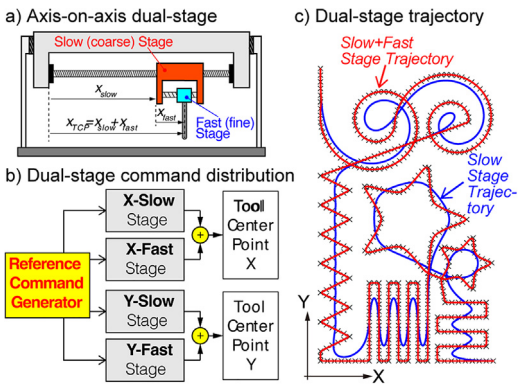

Fig. 1. Dual-stage feed drive and contouring motion generation.

* Corresponding author.
E-mail address: burak.sencer@oregonstate.edu (B. Sencer).

https://doi.org/10.1016/j.cirp.2023.04.070
0007-8506/

frequency and time-domain constraints to ensure both stages are operated within their dynamic and kinematic limits. Finite impulse response (FIR) filtering strategy is adapted to design a trajectory governor that can be 1) implemented at high sampling rates, 2) frequency and time domain signal separation is achieved, and 3) unwanted structural vibrations can be avoided. A novel dynamic command scaling (DCS) strategy is developed, which scales motion commands in real-time so that all axis kinematic limits (range, velocity, acceleration) are respected. Overall, the proposed real-time trajectory generator utilizes the slow and fast stages in a coordinated fashion within their dynamic and kinematic limits for maximizing productivity in high-speed large-workspace contouring.

2. Command distribution between slow and fast stage

Fig. 2 illustrates the proposed dual-stage interpolation strategy. It uses low-pass filtering [8] based interpolation and distributes non-stop contouring motion commands between slow and fast stages. CL-data points $\boldsymbol{P}_k = [P_{x,k}, P_{y,k}, P_{z,k}]^T, k = 1 \dots M$ of the toolpath are first linearly interpolated in cartesian (x,y,z) directions. This is achieved by describing linear (G01) moves between CL-points by series of tangential feed pulse commands F_k, each with a pulse-width of $T_{v,k} = \frac{\| \boldsymbol{P}_{k+1} - \boldsymbol{P}_k \|}{F_k} = \frac{L_k}{F_k}$ (See Fig. 2b). Tangential feed pulses $\langle F_k, T_{v,k} \rangle$ are then projected onto their cartesian directional components:

$$\boldsymbol{v}_k = \left[v_{x,k}, v_{y,k}, v_{z,k} \right]^T = F_k \vec{\boldsymbol{t}}_k \text{ where } \vec{\boldsymbol{t}}_k = (\boldsymbol{P}_k - \boldsymbol{P}_{k-1})/L_k \quad (1)$$

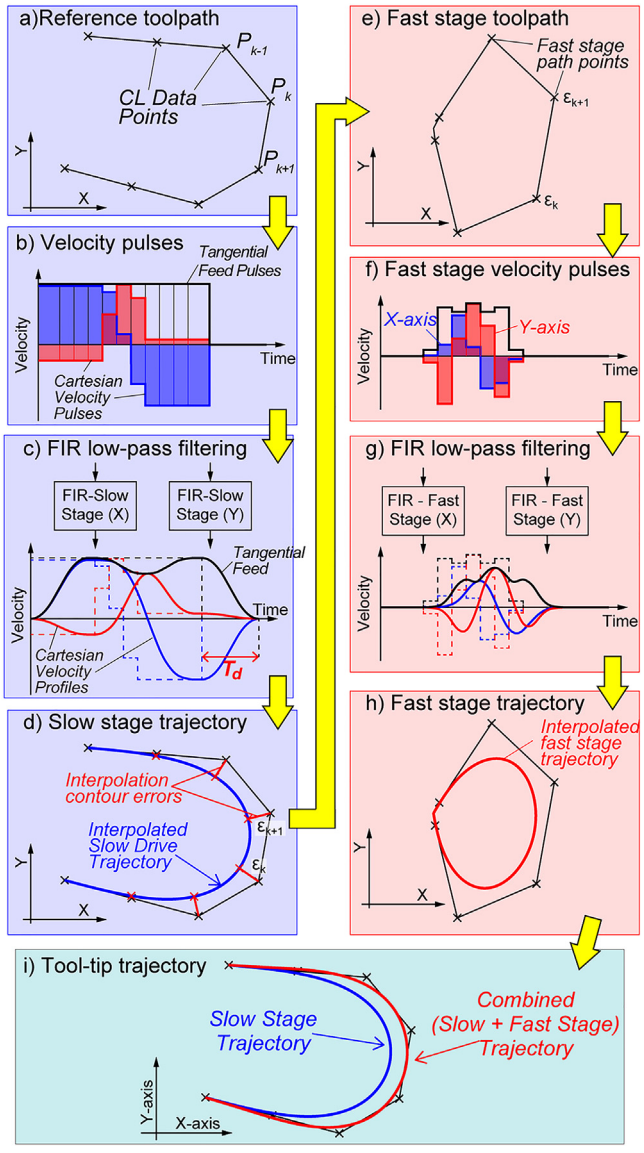

Fig. 2. Proposed real-time trajectory generation strategy.

and as shown in Fig. 2c they are filtered (smoothened) by 2nd order unity-gain FIR low-pass filters (G_{LFP}):

$$G_{LPF} = \frac{1}{T_1 T_2} \frac{(1 - e^{-sT_1})(1 - e^{-sT_2})}{s^2} \quad (2)$$

where T_1, T_2 are the filter time constants. Note that the filter (Eq. 2) contains 2 free integrators $(1/s^2)$, and thus filtering a pulse velocity signal generates the well-known jerk-limited trajectory profile (See Fig. 2c) [8,9]. Filter time constants control peak (max) velocity, acceleration and jerk values as well their frequency spectrum. The filter in Eq. (2) also acts as an input shaper [10], which means that matching T_1, T_2 with the natural periods of structural resonances eliminates triggering inertial (residual) vibrations during rapid feed motion [8–10]. As shown in Fig. 2c, the filter time constants also determine the delay $T_d = T_1 + T_2$ added to the motion.

As shown in Fig. 2d, low-pass filtering interpolates a smooth cartesian tool trajectory that blends linear CL-lines while at the same time deviating from the original toolpath geometry. This deviation is due to the filter's (low-pass) transient response and called as the interpolation contour error. Although introducing interpolation errors in the reference trajectory seem to be a major drawback of this interpolation strategy, it is used as an advantage in the proposed approach as described in the following.

The idea is to use the interpolated trajectory with contour errors as reference to the slow stage, and then utilize the fast stage to compensate for these interpolation errors. To realize that slow stage's interpolation error profile is predicted and commanded as feed pulses to the fast stage (See Fig. 2e-f). Fast stage's pulses are then smoothened by a new set of FIR filters with smaller time constants so that its wider dynamic bandwidth is utilized. Note that fast stage's trajectory will also show interpolation errors (See Fig. 2h), but they are smaller and can be kept within process tolerances by adjusting its motion, which is explained in Section 3. Owing to its filtering based framework, the proposed approach is real-time suitable and provides the following advantages:

- Firstly, as long as slow-stage's interpolation contour errors are geometrically confined within the stroke limit of the fast-stage, slow stage simply carries the fast stage close to the path geometry, and the fast stage executes final tool positioning.
- Synchronized motion is achieved conveniently where the slow stage becomes the "master". When its motion slows down, the overall motion becomes slower, which allows efficient control over interpolation errors and kinematic limits.
- Using separate set of filters for slow and fast stages allows individual control over command frequency spectrum for better utilizing stage bandwidths and avoiding vibrations.

2 key bottlenecks must be addressed to benefit from the above advantages: 1) interpolation errors must be confined, and 2) drive kinematic limits must be imposed.

3. Interpolation contour error and kinematic limit control

Slow stage's interpolation errors are predicted and confined within the fast stage stroke limits to ensure safe operation. As shown in Fig. 3a, interpolation errors occur due to the transient response of the FIR filter [9]. Each feed pulse $\langle F_k, T_{v,k} \rangle$ interpolates the motion along the feed direction $(\vec{\boldsymbol{t}}_k)$ for a duration of $T_{v,k}+T_d$. Overlap of responses (See Fig. 3b) cause interpolation errors because motion along the next feed direction (red trapezoids/triangles in Fig. 3b) are commanded before the previous (blue) ones are fully completed. The contour error ϵ_k from the k^{th} CL-data point $(\boldsymbol{P}_k)$ can be calculated from the filter's pulse response and feed direction $\vec{\boldsymbol{t}}_k$ as:

$$\epsilon_k = \| \sum_{i=0}^{m-1} \boldsymbol{t}_{k-i} l_{k-i} - \sum_{i=1}^{n} \boldsymbol{t}_{k+i} l_{k+i} \| \quad (3)$$

where l_{k-i} and l_{k+i} denote distances to be traveled due to m previous and n next velocity pulse responses that overlap around the target,

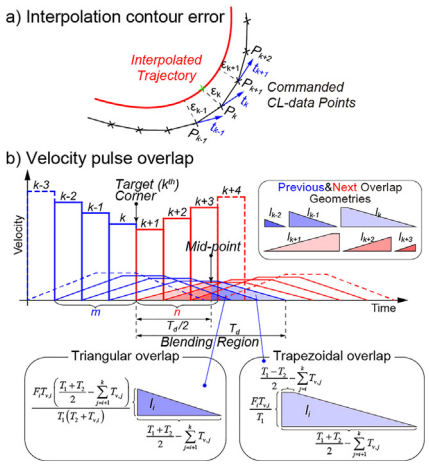

Fig. 3. Interpolation contour error prediction. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$\boldsymbol{P}_k$. They are calculated by approximating the filter response as trapezoids and then evaluating the area underneath:

$$l_{k-i} = \begin{cases} \dfrac{F_{k-i}T_{v,k-i}}{2T_1}\left(T_1 - T_{v,k-i} - 2\sum_{j=0}^{i-1} T_{v,k-j}\right) & \text{for} \sum_{j=0}^{i} T_{v,k-j} \le \dfrac{T_1 - T_2}{2} \\ \dfrac{F_{k-i}T_{v,k-i}}{2T_1(T_2 + T_{v,k-i})}\left(\dfrac{T_1+T_2}{2} - \sum_{j=0}^{i-1} T_{v,k-j}\right) & \text{for} \sum_{j=0}^{i} T_{v,k-j} > \dfrac{T_1 - T_2}{2} \end{cases} \quad (4)$$

The trapezoidal response approximation (Fig. 3b) used here is for computational efficiency. It slightly over-estimates the travel distance and the contour errors, but such over-estimation is safe.

3.1. Interpolation contour error control

Once interpolation contour errors are predicted from Eq. (3), they are controlled by 2 novel Dynamic Command Scaling (DCS) strategies. Notice from Fig. 3b that the base and height of trapezoidal velocity responses are controlled by the filter time constants (T_1,T_2) and the feed pulse $\langle F_k, T_{v,k}\rangle$. DCS strategies are used to either modulate the feed pulse width/height, or the filter time constant to control the pulse response and thus limit the interpolation contour errors.

The first strategy is to define a feed override factor, $\alpha_k \le 1$, which scales (stretches) the feed pulses to slow down the motion:

$$F_k^* = \alpha_k F_k \text{ and } T_v^* = T_v/a_k \quad (5)$$

Eqs. (3) and 4 can be re-written with scaled pulse $\langle F_k^*, T_v^*\rangle$, and feed override α_k can be solved from Eq. (3) to satisfy a tolerance:

$$\epsilon_k^* = \epsilon_{tol} \Rightarrow (a_1 - \epsilon_{tol}^2)\alpha_k^4 + a_2\alpha_k^3 + a_3\alpha_k^2 + a_4\alpha_k + a_5 = 0 \quad (6)$$

where $a_1 \ldots a_5$ are the polynomial coefficients that can be calculated analytically and omitted here. When scaling the slow stage's feed pulses, ϵ_{tol} can be set as the stroke limit of the fast stage. It should also be noted that multiple $(m + n)$ feed pulses affect k^{th} contour error, and they are overridden together.

Another strategy is to scale the filter's transient response by a delay override factor, $\beta_k \le 1$, which shrinks the filter time constants by $T_{1,2}^* = \beta_k T_{1,2}$. Eqs. (3) and 4 can be re-written by substituting $T_{1,2} \leftarrow T_{1,2}^*$, and $\beta_k$can be solved from Eq. (3) similarly:

$$\epsilon_k^* = \epsilon_{tol} \Rightarrow b_1\beta^4 + b_2\beta^3 + (b_3 - \epsilon_{tol}^2)\beta^2 + b_4\beta + b_5 = 0 \quad (7)$$

where $b_1..b_5$ can be computed analytically.

Some remarks about the proposed DCS strategies must be given. 1) Both DCS approaches can be applied sequentially to confine the errors. 2) Overriding the feedrate by α slows down the motion by lowering the feed and stretching the pulse width, which ensures that the interpolated motion is closer to the commanded geometry (analogous to the feed-override knob found on an NC). 3) Scaling the filter time constant by β shortens the motion duration while at the same time minimizes the errors because the transient response becomes shorter. However, it distorts filter's frequency response and deteriorates vibration mitigation functionality. 4) α and β are found efficiently in real-time from the roots of Eqs. (6) and 7 via few numerical, e.g. Newton's, iterations. 5) Slow stage is the master. Once DCS is applied on it, it affects fast stage's motion. For instance, once feed override is applied, both the slow and the fast stage motion slow down. The slow stage travels closer to the toolpath, which keeps the fast stage within its actuation range. At the same time fast stage's errors decrease and the total tool motion becomes more accurate. Section 3.3 summarizes DCS application strategy.

3.2. Kinematic limit control

In addition to interpolation tolerance control, DCS can be used to impose kinematic limits (velocity and acceleration). Let us assume that the feed pulse widths are short, $T_{v,k} < T_2 < T_1$. This assumption is generally acceptable since high-speed contouring paths contain short linear moves and high feeds [10]. For a single short CL-block, the achievable peak velocity v_p and acceleration a_p can be estimated from the filter response [11] as (See Fig. 4a):

$$v_p = \frac{FT_v}{T_1} = \frac{L}{T_1} \text{ and } a_p = \frac{FT_v}{T_1T_2} = \frac{L}{T_1T_2} \quad (8)$$

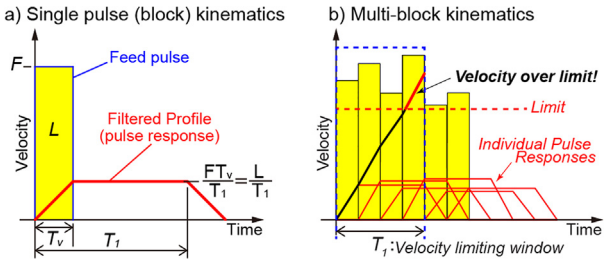

Fig. 4. Kinematic limit control.

Notice from Eq. (8) that the peak velocity is controlled by the block length L and the filter time constant T_1. This implies that the travel distance of a CL-block within T_1 motion command window must not exceed $L_{vmax} < T_1 v_{max}$ to constrain the peak velocity, $v_p < v_{max}$. Similarly, peak acceleration is controlled jointly by T_1 and T_2, and this indicates that the commanded travel distance must be kept $L_{amax} < T_1T_2a_{max}$ to satisfy the acceleration limit, $a_p < a_{max}$ within T_1T_2 motion window.

Now, consider p (successive) blocks in a look-ahead-window fashion in Fig. 4b. Peak velocity and accelerations simply become superposition of multiple filter pulse responses as:

$$v_p = \frac{1}{T_1}\sum_{k=1..p} L_k \text{ and } a_p = \frac{1}{T_1T_2}\sum_{k=1..q} L_k \quad (9)$$

and the feed override factor (α) can be used to stretch multiple pulse widths so that the total commanded travel distance $\sum_{k=1..p} L_k$ commanded within T_1 and T_1T_2 motion look-a-head windows are constrained to bound v_p and a_p as:

$$\alpha_{v_p} = \frac{L_{vmax}}{\sum_{k=1..p} L_k} \text{ and } \alpha_{a_p} = \frac{L_{amax}}{\sum_{k=1..q} L_k} \quad (10)$$

$\min\{\alpha_{vp}, \alpha_{ap}\}$ must be selected to satisfy both limits. Also note that Eq. (10) is applied in drive level to impose axis kinematic limits. Since feed override also controls the interpolation errors (Eqs. (6) and (7)), smallest of all the feed override factors must be applied.

Next, the delay override factor (β) can be used to scale the filter time constants and impose kinematic limits as well. Please note from Eq. (8) that delay override linearly scales the peak velocity $v_p = L/(\beta T_1)$ whereas peak acceleration is scaled by β^2, $a_p = L/(\beta^2 T_1 T_2)$. Due to space limitations, it is omitted here but the delay override can be similarly applied over multiple blocks that fall within the filter delay T_1or T_2, and suitable override factor can be selected to jointly constrain the peak kinematic values.

3.3. Real-time implementation and DCS application strategy

DCS strategies are applied sequentially within a look-a-head window where multiple feed pulses are modified simultaneously to control both stroke and kinematic limits as follows:

1. Feed override (α) is applied first (Eqs. (6) and (10)), to stretch slow stage feed pulses for satisfying both slow stage kinematic (velocity, acceleration) and fast stage's stroke limits.
2. Next, fast stage feed pulses are generated based on the slow stage interpolation errors (See Fig. 2e).
3. Delay override β is applied on the fast stage pulses using Eq. 7 to confine tool-tip positioning errors within a tolerance.
4. Lastly, feed override (α) is applied again on slow and fast stages using Eq. (10) to satisfy fast stage kinematic (velocity, acceleration) limits.

4. Experimental results

Experiments are conducted on a dual-stage laser engraving system shown in Fig. 5a. Slow stage is a linear motor driven gantry platform that carries the fixture and workpiece within stroke limit of the fast stage, which positions the laser beam. Fast stage is driven by voice-coil motors and has a total stroke of +/−3 mm. Both stages are controlled by P-PI control with position bandwidths of roughly 50 Hz (slow) and 120 Hz (fast). Table 1 summarizes kinematic limits and key trajectory parameters.

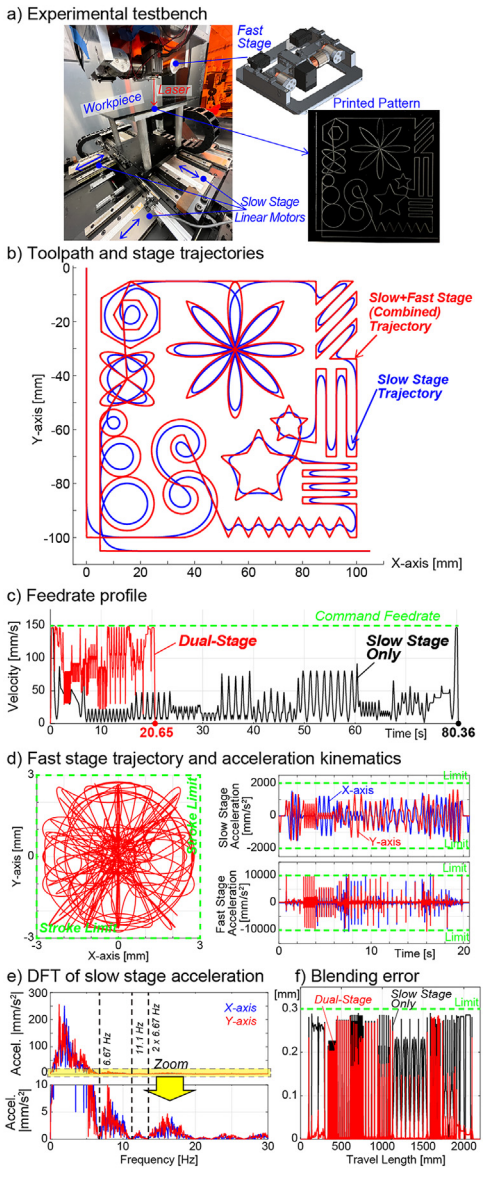

Fig. 5. Experimental results.

Table 1
Trajectory parameters.

Stage	Filter time const.	Stroke	Velocity Lim.	Acc. Lim.
Slow	T_1=150, T_2=90 ms	–	150 mm/s	2000 mm/s^2
Fast	T_1=20, T_2=10 ms	+/- 3 mm	150 mm/s	10,000 mm/s^2

The laser engraved(printed) pattern is shown in Fig. 5a. The toolpath consists of 3801 CL-data points. Fig. 5b shows the slow and fast stage motion. As shown, proposed trajectory governor commands slow stage to carry the workpiece within fine stage's motion range, and fine stage accurately follows the sharp contours of the workpiece within set tolerance (300 µm). Total motion duration is 20.7 s, and the algorithm required only 10 s on a typical PC to interpolate entire trajectory, which validates its real-time suitability. Fig. 5c also shows that as compared to executing the same trajectory only using a slow stage, the proposed algorithm effectively uses of dual-stage kinematic limits and improves productivity by >2 times. Stage acceleration limits and tolerances are respected, and frequency spectrum of the motion is controlled. DFT of slow stage acceleration exhibits notches around 6.67 and 11.1 Hz. These frequencies correspond to the filter time constants (Table 1) and can be used to avoid residual vibrations.

5. Conclusions

This paper presented a novel reference trajectory governor for machine tool feed drives equipped with dual-stage servo systems. Without solving complex optimization algorithms, proposed approach can distribute coordinated motion commands between slow and fast stages to efficiently utilize their dynamic and kinematic limits. It voids range saturation of the fast stage and provides frequency shaping of reference commands for vibration mitigation. As opposed to spline-based trajectory generation techniques, proposed approach is highly efficient and real-time suitable. It can be used to improve productivity of various laser processing (cutting, drilling, printing, patterning) equipment, and high-speed machine tools with dual-stage feed drives.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Hong GS, San Wong Y (2012) Profile error compensation in fast tool servo diamond turning of micro-structured surfaces. *IJMTM* 52(1):13–23.
[2] Altintas Y, Woronko A (2002) A piezo tool actuator for precision turning of hardened shafts. *CIRP Annals* 51(1):303–306.
[3] Staroselsky S, Stelson KA (1988) Two-stage actuation for improved accuracy of contouring. *ACC* : 127–132.
[4] Yuen A, Altintas Y (2016) Trajectory generation and control of a 9 axis CNC micromachining center. *CIRP Annals* 65(1):349–352.
[5] Lu XD, Trumper DL (2005) Ultrafast tool servos for diamond turning. *CIRP annals* 54(1):383–388.
[6] Di Cairano S (2018) Cascaded reference Governor–MPC for motion control of two-stage manufacturing machines. *IEEE SCT* 27(5):2030–2044.
[7] Antonello R, Oboe (2015) *Feasible Trajectory Generation For a Dual Stage Positioning System Using a Simplified Model Predictive Control Approach* March. IEEE (ICM), 608–613.
[8] Sencer B, Ishizaki K, Shamoto E (2015) High speed cornering strategy with confined contour error and vibration suppression for CNC machine tools. *CIRP Annals* 64(1):369–372.
[9] Tajima S, Sencer B (2020) Real-time trajectory generation for 5-axis machine tools with singularity avoidance. *CIRP Annals* 69(1):349–352.
[10] Altintas Y, Khoshdarregi MR (2012) Contour error control of CNC machine tools with vibration avoidance. *CIRP annals* 61(1):335–338.
[11] Tajima S, Sencer B (2022) Online interpolation of 5-axis machining toolpaths with global blending. *IJMTM* 175:103862.