

Simultaneous servo error pre-compensation and feedrate optimization with tolerance constraints using linear programming

Heejin Kim · Chinedum E. Okwudire

Abstract Servo error pre-compensation and feedrate optimization are often performed independently to improve the accuracy and speed of manufacturing machines. However, this independent approach leads to unnecessary trade-offs between productivity and quality in manufacturing. This paper proposes a novel linear programming approach for combined servo error pre-compensation and feedrate optimization, subject to contour error (tolerance) and kinematic constraints. The incorporation of servo error pre-compensation into feedrate optimization allows for faster motions without violating tolerance constraints. Experiments carried out on a 3D printer and precision motion stage are respectively used to demonstrate up to 43% and 47% reduction in cycle time without compromising part quality using the proposed compared to the independent approach.

Keywords Feedrate optimization · Pre-compensation · Filtered B-splines · Contour error · CNC · Linear programming

1 Introduction

A wide range of manufacturing machines use feed drives powered by computer numerical control (CNC) to generate motion commands. Two critical requirements of manufacturing are productivity and quality, which often involve a trade-off between speed and accuracy of feed drives [1]. This trade-off is typically handled, in practice, by maximizing speed so long as an accuracy (tolerance) level is not violated. Servo errors are a major

source of inaccuracy in feed drives. They can be caused by commanded motion (i.e., motion-induced servo errors) or disturbance forces like friction and manufacturing process forces. Motion-induced servo errors are very important in determining the trade-off between speed and accuracy because servo controllers always have limited bandwidth. This means that faster motion commands lead to larger servo errors. One way of reducing motion-induced servo errors is through pre-compensation (i.e., feedforward compensation). Knowledge of the machine's servo dynamics is used to modify the motion commands offline or online in the CNC interpolator to reduce servo errors. Examples of servo error pre-compensation (SEP) include zero phase error tracking controller [2], iterative method [3], path-modification via inverse dynamics [4], input shaper [5], analytical prediction and compensation of contour error [6], model predictive control framework [7], trajectory pre-filter [8], cross-coupled pre-compensation [9], mirror compensation with Taylor's expansion [10], adaptive cross-coupled prediction compensation [11], cross-coupled dynamic friction control [12], and filtered B splines [13, 14]. However, available SEP approaches focus on reducing or minimizing servo error without trying to maximize feedrate subject to tolerance constraints.

On the other hand, there are numerous works on feedrate optimization (FO) subject to tolerance constraints. Traditionally, tolerance constraints are introduced implicitly into FO by imposing velocity, acceleration and jerk limits [15–18]. Some works have explicitly added tracking or chordal accuracy constraints to FO [19–25]. In practice, SEP and FO are combined by first performing FO and then applying the optimized results to SEP for error minimization, as shown by the one-way arrow between FO and SEP in Fig. 1(a). However, this independent (or sequential) approach could lead to sub

Heejin Kim · Chinedum E. Okwudire (✉)
Department of Mechanical Engineering, University of Michigan,
2350 Hayward, Ann Arbor, MI, USA
E-mail: okwudire@umich.edu

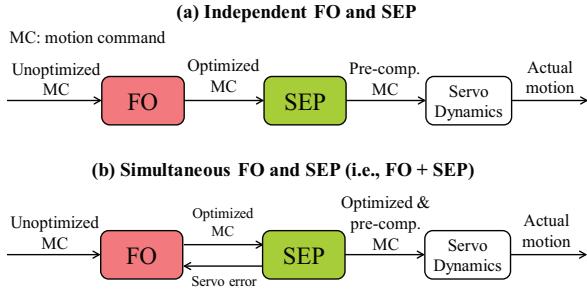


Fig. 1 Comparison of (a), independent FO and SEP – standard practice – and (b), proposed concept of simultaneous FO and SEP (i.e., FO+SEP)

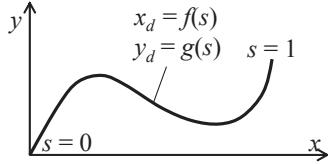


Fig. 2 Parametric planar curve as function of path variable, s

optimality, as FO does not benefit from the reduction of error provided by SEP in maximizing feedrate. A better approach is to incorporate information from SEP into FO, as shown by the two-way arrow between FO and SEP in Fig. 1(b). To our best knowledge, such a combined technique for FO and SEP has not been explored in the open literature.

This paper proposes, for the first time, a linear programming (LP) approach for simultaneous FO and SEP. The systematic incorporation of SEP into FO expands the feasible region for FO, thus allowing for faster motions without violating tolerance constraints. In addition, the use of LP makes the proposed method computationally efficient and mathematically elegant. The outline of the paper is as follows: Section 2 contrasts time-based LP and available path-based LP [15, 16] approaches for FO. It shows that time-based LP is similar to path-based LP in terms of computational efficiency but is superior to path-based LP in handling jerk constraints and incorporating servo dynamics into FO. Section 3 presents the proposed approach for simultaneous FO and SEP (i.e., FO+SEP) using time-based LP. Section 4 validates the effectiveness of the proposed FO+SEP approach relative to FO in simulations and experiments carried out on a 3D printer and planar motion stage. Conclusions and future work are presented in Section 5.

2 Time-based vs Path-based LP for Feedrate Optimization

Fig. 2 illustrates an arbitrary, curved path in the x - y plane with path parameter $s \in [0, 1]$. Note that s is a function of time t (i.e., $s = s(t)$). Let $x_d = f(s)$ and $y_d = g(s)$ denote a pair of parametric equations in s , representing the x and y components of desired position, respectively.

An increasingly popular approach is to perform FO via path-based LP, i.e., using s as the independent variable [15, 16]. In path-based LP, it is assumed that s represents arc length, i.e., distance travelled along the curve, which is normalized by total travel length L . Let x_d and y_d be already known in the defined domain of s . Then, the kinematic limits F_{max} , A_{max} , and J_{max} on feedrate, axis acceleration and axis jerk respectively, can be imposed as

$$\begin{aligned} L |\dot{s}| &\leq F_{max} \\ \left| \frac{d^2 x_d(s)}{dt^2} \right| &= |x_d''(s)\dot{s}^2 + x_d'(s)\ddot{s}| \leq A_{max} \\ \left| \frac{d^3 x_d(s)}{dt^3} \right| &= |x_d'''(s)\dot{s}^3 + 3x_d''(s)\dot{s}\ddot{s} + x_d'(s)\dddot{s}| \leq J_{max} \end{aligned} \quad (1)$$

for $\forall s$, where $x_d'(s)$, $x_d''(s)$, and $x_d'''(s)$ denote geometric derivatives of $x_d(s)$ with respect to s ; \dot{s} , \ddot{s} , and \dddot{s} are tangential velocity, acceleration, and jerk, respectively. The y -axis acceleration and jerk limits are imposed in the same manner. Note that, instead of imposing feedrate limits as in Eq. (1), axis velocity limits could be imposed in addition to axis acceleration and jerk limits [15, 16].

To facilitate path-based LP, a new parameter $q = \dot{s}^2$ is introduced to remove the nonlinearity in Eq. (1). With q , the following substitutions hold:

$$\begin{aligned} \dot{s} &= \sqrt{q}, & \dot{s}^2 &= q, & \dot{s}^3 &= q\sqrt{q} \\ \ddot{s} &= \frac{1}{2}q', & \ddot{s} &= \frac{1}{2}q''\sqrt{q}, & \ddot{s}\ddot{s} &= \frac{1}{2}q'\sqrt{q} \end{aligned} \quad \left. \right\} \quad (2)$$

With Eq. (2), the feedrate optimization with the same kinematic constraints is formulated as Eq. (3) for $\forall s$:

$$\begin{aligned} \min_q \quad & \int_0^1 q(s)ds \\ \text{s.t.} \quad & L\sqrt{q(s)} \leq F_{max} \\ & \left| x_d''(s)q(s) + \frac{1}{2}x_d'(s)q'(s) \right| \leq A_{max} \\ & \left| x_d'''(s)q(s) + \frac{3}{2}x_d''(s)q'(s) + \frac{1}{2}x_d'(s)q''(s) \right| \sqrt{q(s)} \\ & \leq J_{max} \end{aligned}$$

(3)

Here, the feedrate constraint can be linearized by squaring both sides, and the $q'(s)$ and $q''(s)$ terms in acceleration can be linearized by B-spline parametrization of $q(s)$ with respect to s [15, 16]. However, because the $\sqrt{q(s)}$ term in the jerk limit is still nonlinear, $q(s)$ is replaced by a precomputed upper bound $q^*(s)$. One candidate for $q^*(s)$ is the solution obtained with only velocity and acceleration constraints [15, 16] in Eq. (3). Then, the jerk constraint in Eq. (3) is reformulated using pseudo jerk $\bar{j}(s)$ as:

$$\left| x_d'''(s)q(s) + \frac{3}{2}x_d''(s)q'(s) + \frac{1}{2}x_d'(s)q''(s) \right| \sqrt{q^*(s)} = |\bar{j}(s)| \leq J_{max} \quad (4)$$

Although path-based LP is capable of imposing linear feedrate and axis acceleration constraints on FO, it cannot impose linear axis jerk constraints without the use of pseudo jerk, at the cost of optimality [15, 16]. Moreover, because path-based LP uses q as the independent variable, it is limited in its ability to accommodate servo dynamics, which uses time t as the independent variable.

Therefore, in this work, we formulate a time-based LP approach for FO, adapted from the model predictive contour control framework proposed by Lam et al. [26]. Let $s(t)$ be discretized with fixed sampling interval T_s and expressed as a vector $\mathbf{s} = \{s(0), s(1), \dots, s(N-1)\}^T$. Then, FO can be formulated as:

$$\begin{aligned} \min_{\mathbf{s}} & - \sum_{k=0}^{N-1} s(k) \\ \text{s.t.} & \quad s(k-1) \leq s(k) \leq 1, \quad \forall k = 1, 2, \dots, N-1 \end{aligned} \quad (5)$$

The idea of Eq. (5) is that to minimize total time, the sum of $s(k)$ over N time steps must be maximized – i.e., the path from $s(0) = 0$ to $s(N-1) = 1$ should be traversed as fast as possible, while satisfying the monotonicity and endpoint constraints on \mathbf{s} in Eq. (5). In addition, it should satisfy kinematic constraints F_{max} , A_{max} , and J_{max} on feedrate, axis acceleration and axis jerk, respectively:

$$\begin{aligned} L \frac{D[\mathbf{s}]}{T_s} & \leq \mathbf{F}_{max} \\ \left| \frac{D^2[\hat{\mathbf{x}}_d]}{T_s^2} \right|, \left| \frac{D^2[\hat{\mathbf{y}}_d]}{T_s^2} \right| & \leq \mathbf{A}_{max} \\ \left| \frac{D^3[\hat{\mathbf{x}}_d]}{T_s^3} \right|, \left| \frac{D^3[\hat{\mathbf{y}}_d]}{T_s^3} \right| & \leq \mathbf{J}_{max} \end{aligned} \quad (6)$$

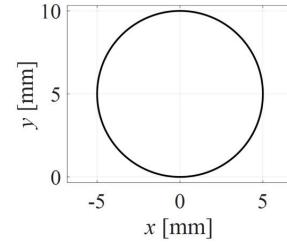


Fig. 3 Desired toolpath

Here, D denotes finite difference operator, while \mathbf{F}_{max} , \mathbf{A}_{max} , and \mathbf{J}_{max} are vectorized representations of the corresponding kinematic limits, and $\hat{\mathbf{x}}_d$ and $\hat{\mathbf{y}}_d$ are vectorized versions of $\hat{x}_d(k)$ and $\hat{y}_d(k)$, respectively, similar to \mathbf{s} . This notation is maintained hereinafter. The terms $x_d = f(s)$ and $y_d = g(s)$ are generally nonlinear in s . Thus, at each time step k , they are linearized with linearization points $s^e(k)$ estimated from an initial unoptimized trajectory as

$$\hat{x}_d(k) = \frac{\partial f(s)}{\partial s} \Big|_{s=s^e(k)} \cdot (s(k) - s^e(k)) + f(s^e(k)) \quad (7)$$

and $\hat{y}_d(k)$ is obtained by linearizing $g(s)$ in the same manner.

To compare time-based and path-based LP, a circular toolpath with radius, $R = 5$ mm is employed, as illustrated in Fig. 3. The kinematic constraints are $F_{max} = 30$ mm/s, $A_{max} = 0.5$ m/s², and $J_{max} = 5$ m/s³. For both path-based and time-based LP, \mathbf{s} is discretized and represented using B-splines [16, 20] as

$$\mathbf{s} = \mathbf{N}_s \mathbf{p}_s \quad (8)$$

where \mathbf{p}_s is the control point vector of length n_p ; \mathbf{N}_s is the basis function matrix. By using \mathbf{p}_s as the optimization variable in place of \mathbf{s} , the problem size is substantially reduced because $n_p \ll N$. Here, a 5th degree B-spline with uniform knot vector and $n_p = 40$ control points are used.

Time-based LP is initialized using an unoptimized trajectory generated using trapezoidal acceleration profile (TAP) [8] with the just-given kinematic limits. First, only the feedrate and axis acceleration limits are imposed on path-based and time-based LP. In this case, they yield almost the same feedrate profile, as shown in Fig. 4. As a result, their cycle time and computation time (using MATLAB® R2019a on a Windows PC with Intel Core i7-8750H CPU and 16 GB RAM) is similar, as summarized in Table 1. This shows that both methods have similar computational efficiency, under similar conditions. Next, the constraint on axis jerk and pseudo-jerk of $J_{max} = 5$ m/s³ is introduced. Fig. 4 and Table 1 shows that the cycle time becomes 1.42 s for

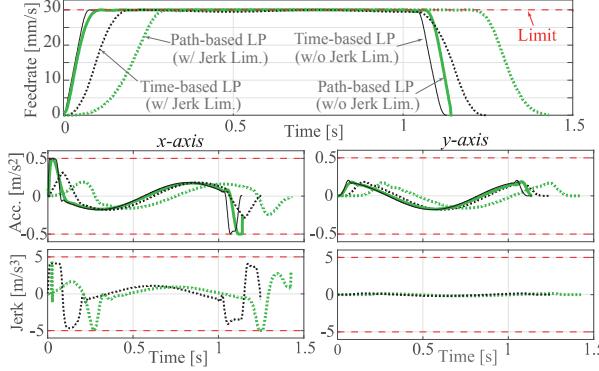


Fig. 4 Feedrate, axis acceleration and axis jerk profiles of path- and time-based LP with and without jerk constraints imposed

Table 1 Comparison of cycle and computation time of path- and time-based LP with and without jerk constraints imposed

FO algorithm	Cycle time [s]	Computation time [s]
w/o jerk constraints	Time-based LP	1.13
w/o jerk constraints	Path-based LP	0.75
w/ jerk constraints	Time-based LP	1.14
w/ jerk constraints	Path-based LP	1.25
		1.89
		1.42
		1.35

path-based LP and 1.25 s for time-based LP. This discrepancy shows the sub-optimality introduced by the pseudo-jerk relaxation. The computation time is also summarized in Table 1.

Time-based LP is also superior to path-based LP because it can incorporate any linear servo dynamics into FO. Conversely, path-based LP can only accommodate servo dynamics that are linear with regard to velocity and acceleration [19, 20], without need for approximation. Given these advantages, the time-based LP formulated in this section is selected for the simultaneous SEP and FO approach proposed in the next section.

3 Simultaneous FO and SEP using Time-based LP

3.1 Framework of Simultaneous FO and SEP using Time-based LP

Fig. 5 shows a block diagram of the proposed simultaneous SEP and FO. The idea is to impose contour error (tolerance) constraints on FO taking SEP into account. Contour error, denoted as CE hereinafter, has been selected as the accuracy index in FO because it directly impacts the ability of part quality to meet tolerance specifications in manufacturing [3–7, 27]. However, because the proposed approach uses LP, CE must be estimated using linear dynamics. To do this, linearized

desired x -axis position, \hat{x}_d , is used to generate modified position command \hat{x}_{dm} using a SEP process represented by C_x . A linear model, \hat{G}_x , of the actual servo dynamics, G_x , is used to estimate the x -axis position as \hat{x} and tracking error as $\hat{e}_x = \hat{x}_d - \hat{x}$. A similar process is followed to obtain $\hat{e}_y = \hat{y}_d - \hat{y}$, using C_y and \hat{G}_y .

CE is defined as the orthogonal distance between an actual trajectory point at time k and the reference toolpath [9], denoted as $\epsilon_c(k)$ in Fig. 6. Approximate CE, denoted as $\hat{\epsilon}_c(k)$, can be computed from the axis tracking errors using a linear estimation [27] as

$$\begin{aligned}\hat{\epsilon}_c(k) &= -\sin \theta(k) \hat{e}_x(k) + \cos \theta(k) \hat{e}_y(k) \\ &= -\sin \theta(k) (1 - \hat{G}_x C_x) \hat{x}_d(k) \\ &\quad + \cos \theta(k) (1 - \hat{G}_y C_y) \hat{y}_d(k)\end{aligned}\quad (9)$$

where $\theta(k)$ is the angle of incline of the curve (\hat{x}_d, \hat{y}_d) at time step k . The linear approximation of CE $\hat{\epsilon}_c$ (including the effects of SEP using C_x and C_y) is imposed as an additional constraint on the time-based LP formulation of Section 2 as:

$$\begin{aligned}|\hat{\epsilon}_c| &= |-\sin \theta(\mathbf{I} - \hat{G}_x C_x) \hat{x}_d \\ &\quad + \cos \theta(\mathbf{I} - \hat{G}_y C_y) \hat{y}_d| \leq \hat{E}_{max}\end{aligned}\quad (10)$$

where \hat{E}_{max} is the vectorized form of the maximum allowable approximate (i.e., linearized) CE, \hat{E}_{max} ; C_x , \hat{G}_x , C_y and \hat{G}_y are matrix (lifted) versions of the corresponding system dynamics [28]; and \mathbf{I} is the identity matrix. The implication is that a model of SEP is incorporated into FO, yielding FO+SEP. The optimized x_d and y_d from FO+SEP are then applied to the actual servo dynamics, G_x and G_y , after being pre-compensated using C_x and C_y , respectively. Note that if $C_x = C_y = 1$, then it means that no SEP is considered in FO.

3.2 Realization of SEP in FO+SEP using filtered B splines

It is worth pointing out that C_x and C_y can be any linear SEP (feedforward tracking control) method, e.g., [2, 4–6, 8, 9, 11, 13, 14]. However, among the available linear SEP methods, the filtered B spline (FBS) approach [13, 14] stands out because of its effectiveness and versatility in handling any type of linear system dynamics [14]. Therefore, it is selected for SEP in this paper.

The FBS approach parameterizes modified command \mathbf{x}_{dm} (see Fig. 5) using B splines as $\mathbf{x}_{dm} = \mathbf{N}_x \mathbf{p}_x$, where \mathbf{N}_x is the basis function matrix of degree m and

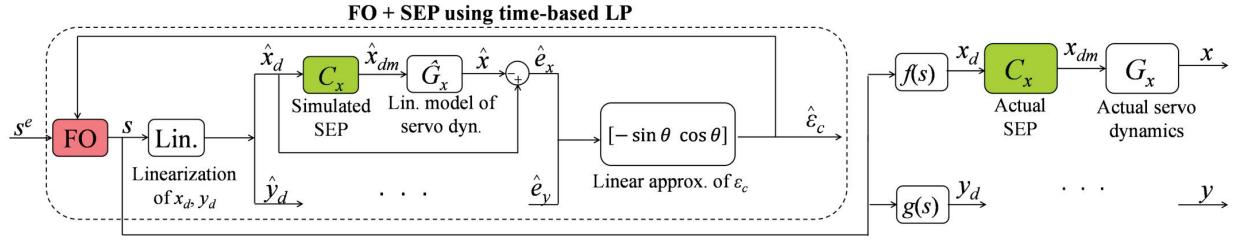


Fig. 5 Block diagram of the proposed simultaneous SEP and FO method (y -component of SEP and servo dynamics omitted for simplicity)

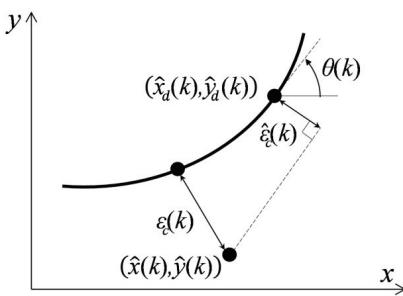


Fig. 6 Contouring error ϵ_c and its linear approximation $\hat{\epsilon}_c$

\mathbf{p}_x is a vector of n control points as

$$\begin{aligned} & \begin{bmatrix} x_{dm}(0) \\ x_{dm}(1) \\ \vdots \\ x_{dm}(N-1) \end{bmatrix} \\ &= \underbrace{\begin{bmatrix} N_{0,m}(\xi_0) & N_{1,m}(\xi_0) & \dots & N_{n-1,m}(\xi_0) \\ N_{0,m}(\xi_1) & N_{1,m}(\xi_1) & \dots & N_{n-1,m}(\xi_1) \\ \vdots & \vdots & \ddots & \vdots \\ N_{0,m}(\xi_{N-1}) & N_{1,m}(\xi_{N-1}) & \dots & N_{n-1,m}(\xi_{N-1}) \end{bmatrix}}_{=\mathbf{N}_x} \underbrace{\begin{bmatrix} p_x(0) \\ p_x(1) \\ \vdots \\ p_x(n-1) \end{bmatrix}}_{=\mathbf{p}_x} \quad (11) \end{aligned}$$

where $\xi \in [0, 1]$ is the spline parameter, representing normalized time; it is uniformly discretized into $\xi_0, \xi_1, \dots, \xi_{N-1}$. Each basis function $N_{j,m}(\xi)$ is defined as

$$\begin{aligned} N_{j,m}(\xi) &= \frac{\xi - \bar{g}_j}{\bar{g}_{j+m} - \bar{g}_j} N_{j,m-1}(\xi) \\ &+ \frac{\bar{g}_{j+m+1} - \xi}{\bar{g}_{j+m+1} - \bar{g}_{j+1}} N_{j+1,m-1}(\xi) \quad (12) \end{aligned}$$

$$N_{j,0}(\xi) = \begin{cases} 1 & \bar{g}_j \leq \xi \leq \bar{g}_{j+1} \\ 0 & \text{otherwise} \end{cases}$$

where $j = 0, 1, \dots, n-1$, and $\bar{g} = [\bar{g}_0, \bar{g}_1, \dots, \bar{g}_{m+n}]$ is a normalized uniformly-spaced knot vector defined in $[0, 1]$. Accordingly, the system output is expressed as $\mathbf{x} \approx \tilde{\mathbf{G}}_x \mathbf{x}_{dm} = \tilde{\mathbf{N}}_x \mathbf{p}_x$, where $\tilde{\mathbf{N}}_x = \tilde{\mathbf{G}}_x \mathbf{N}_x$ (i.e., \mathbf{N}_x

filtered by $\tilde{\mathbf{G}}_x$). The tracking error is modeled as Eq. (13):

$$\mathbf{e}_x = \mathbf{x}_d - \mathbf{x} \approx \mathbf{x}_d - \tilde{\mathbf{N}}_x \mathbf{p}_x \quad (13)$$

Then, the least-squares solution for minimizing $\mathbf{e}_x^T \mathbf{e}_x$ yields optimal coefficients \mathbf{p}_x^* as:

$$\mathbf{p}_x^* = (\tilde{\mathbf{N}}_x^T \tilde{\mathbf{N}}_x)^{-1} \tilde{\mathbf{N}}_x \mathbf{x}_d = \tilde{\mathbf{N}}_x^\dagger \mathbf{x}_d \quad (14)$$

where \dagger represents pseudoinverse. Therefore, $\mathbf{x}_{dm} = \mathbf{N}_x \mathbf{p}_x^* = \mathbf{N}_x \tilde{\mathbf{N}}_x^\dagger \mathbf{x}_d$, which leads to $\mathbf{C}_x = \mathbf{N}_x \tilde{\mathbf{N}}_x^\dagger$. The same process is applied to \mathbf{y}_d . Accordingly, $\mathbf{C}_x = \mathbf{N}_x \tilde{\mathbf{N}}_x^\dagger$ and $\mathbf{C}_y = \mathbf{N}_y \tilde{\mathbf{N}}_y^\dagger$ are substituted into Eq. (10) to realize FO+SEP using the FBS approach.

4 Experimental Validation

For validation of the proposed FO+SEP approach, two experimental setups are used. The first set of experiments, described in Section 4.1, is carried out on a desktop 3D printer commonly used for rapid prototyping. The second set of experiments, described in Section 4.2, is carried out on a linear motor driven planar motion stage typically used in industry for precision positioning. Demonstration of the proposed method on two experimental setups helps to show its versatility.

4.1 Desktop 3D Printer

4.1.1 Experimental Setup

A Lulzbot Taz 6 3D printer is used, as shown in Fig. 7. The optimization algorithms are implemented on dSPACE DS1202 real-time control board running at 1 kHz sampling rate, connected to DRV8825 stepper motor drivers for x , y , z , and e - (extruder) axes stepper motors. ADXL335 accelerometers are attached on the build plate and extruder to measure x , y -axes acceleration.

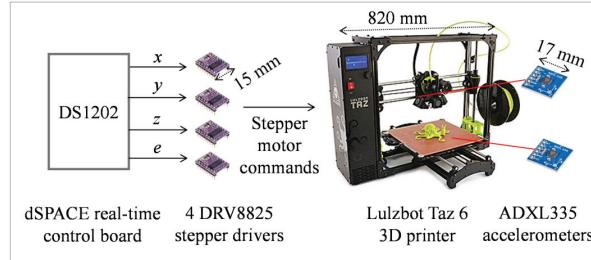


Fig. 7 Experimental set up

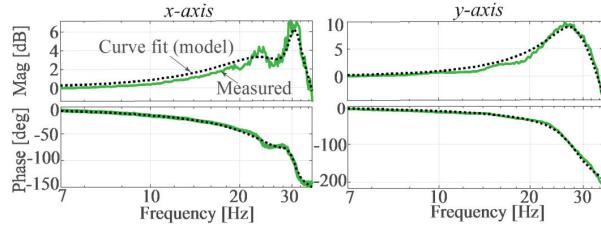


Fig. 8 Measured and curve fitted FRFs of x and y axes of 3D printer

To execute FO and FO+SEP with error constraints, the x and y axis servo dynamics of the printer must be measured in the form of frequency response functions (FRFs) and modeled, via curve fitting, as \hat{G}_x and \hat{G}_y . Fig. 8 shows the measured and modeled FRFs of the x - and y -axes of the printer. The input of each FRF are swept sine acceleration commands to the stepper motor, and the output is relative acceleration between the build plate and nozzle measured using the two ADXL335 accelerometers. The discrete-time transfer function representation of \hat{G}_x and \hat{G}_y are shown in Eq. (15), where the open-loop bandwidth is located around 30 Hz for both axes.

$$\begin{aligned} \hat{G}_x &= \frac{0.021z^5 - 0.061z^4 + 0.044z^3 + 0.033z^2 - 0.056z + 0.012}{z^6 - 5.627z^5 + 13.38z^4 - 17.2z^3 + 12.6z^2 - 4.994z + 0.836} \\ \hat{G}_y &= \frac{0.018z^5 - 0.053z^4 + 0.038z^3 + 0.027z^2 - 0.048z + 0.017}{z^6 - 5.648z^5 + 13.48z^4 - 17.4z^3 + 12.8z^2 - 5.093z + 0.856} \end{aligned} \quad (15)$$

Moreover, in recovery of x , y axes displacement from acceleration measurements, a Luenberger state observer [29] is used. Observer gains are chosen such that the dynamics of the observer error (i.e., difference between estimated position using the linear system model in Eq. (15) and observed position) obtains global asymptotic convergence with observer frequency $f = 10$ Hz.

4.1.2 Benchmarking to Determine Approximate CE Limit

Unoptimized position commands generated using trapezoidal acceleration profile (TAP) [8] are used for benchmarking to determine suitable approximate CE

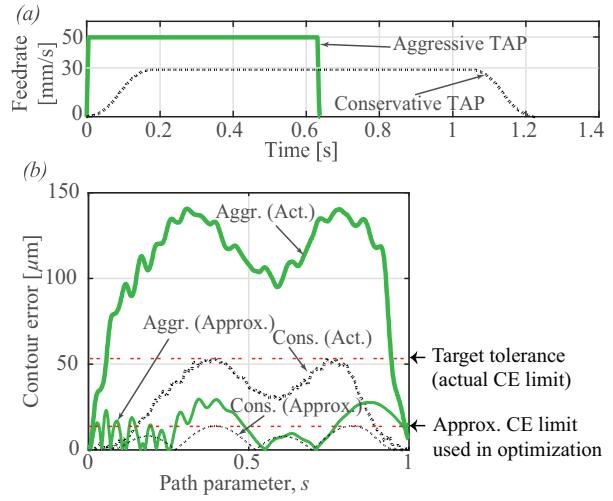


Fig. 9 (a). Commanded feedrate and (b) simulated (approximated) and actual (measured) CE profiles of conservative and aggressive TAP motion commands

limit to traverse a circle of 5 mm radius. Two sets of kinematic limits are used. They are:

- Conservative: $F_{max} = 30$ mm/s, $A_{max} = 0.5$ m/s², $J_{max} = 5$ m/s³;
- Aggressive: $F_{max} = 50$ mm/s, $A_{max} = 10$ m/s², $J_{max} = 5000$ m/s³

Fig. 9(a) shows the TAP feedrate profile generated using the conservative and aggressive kinematic limits; the acceleration and jerk profiles are omitted for the sake of brevity. Fig. 9(b) shows simulated (approximated) and actual (measured) CE profiles of the conservative and aggressive TAP position commands applied to the 3D printer. The simulations are performed using the curve fit linear dynamic model in Eq. (15). Conservative TAP yields maximum simulated and actual CEs of 14 μ m and 54 μ m, respectively. Conversely, Aggressive TAP yields maximum simulated and actual CEs of 30 μ m and 141 μ m, respectively. The reason for the discrepancy between the simulated and actual CEs is due to dynamics like friction and geometric errors not included in the linear model, as well as errors due to linear approximation of the CE in Eq. (9) and constraint equations in Eq. (6) and (10). As a result of these discrepancies between the linear dynamics/approximations and the actual dynamics, it is very important to determine the approximate CE limits (\hat{E}_{max}), used in the proposed FO+SEP, that correspond to acceptable tolerance (i.e., actual CE). Prior work [30] has shown that the aggressive TAP results in poor print quality, while the conservative TAP (with maximum actual CE of 54 μ m) yields acceptable print quality on the Taz 6 printer. Therefore, $\hat{E}_{max} = 14$ μ m is selected as the approximate

CE for LP-based optimization to help keep actual CEs close to the target 54 μm in reality.

4.1.3 Optimization Results using FO and FO+SEP

We compare FO and the proposed FO+SEP with a goal to achieve similar accuracy as Conservative TAP in Fig. 9(b) with the shortest cycle time. To do this, the aggressive kinematic limits in Section 4.1.2 are imposed on both FO and FO+SEP, together with an approximate CE limit of $\hat{E}_{\max} = 14 \mu\text{m}$ using Eq. (10). For FO, $\mathbf{C}_x = \mathbf{C}_y = \mathbf{I}$ (i.e., tolerance constraints are imposed without SEP). However, for FO+SEP, \mathbf{C}_x and \mathbf{C}_y are generated via the FBS approach described in Section 3.2, using a 5th degree B-spline with uniform knot vector and $n = 40$ control points. Another 5th degree B-spline with uniform knot vector and $n_p = 40$ control points is used to parametrize \mathbf{s} to reduce the problem size, as explained in Section 2. Both the FO and FO+SEP cases are initialized using unoptimized TAP trajectories.

Fig. 10 shows the commanded feedrate, acceleration, and jerk profiles of FO and FO+SEP. Fig. 11 shows the simulated (approximated) and actual (measured) CE profiles. Both FO and FO+SEP enforce the approximate CE limit in the LP optimization, leading to the system staying close to the target in experiments. However, FO has to slow down because it hits the approximate CE limit while FO+SEP is able to stay very close to the maximum speed throughout the motion. As a result, FO+SEP completes the motion in 0.64 s, which is 43% faster than FO at 1.13 s, as summarized in Table 2. Note that implementing SEP after FO (i.e., independent approach) would not lower the cycle time of FO; it would only reduce the CE, which has little or no practical value if the desired tolerance has already been met. The computation time for FO+SEP is 1.8 s; FO's is much higher at 37.5 s because it is operating very close to the imposed error constraint.

To further validate our findings, a cylinder of height 8.3 mm consisting of three concentric circular toolpath of radii 4.39 mm, 4.69 mm and 5 mm are printed using the same 3D printer, as shown in Fig. 12. Conservative and Aggressive TAP as well as FO and FO+SEP, as discussed above, are applied to each circular toolpath at each layer of the print. Fig. 13 shows the side and top view of the printed cylinders for the four cases. FO and FO+SEP save 10.9% and 50.5% in cycle time, respectively, compared to Conservative TAP, while maintaining similar surface quality. However, Aggressive TAP results in poor surface quality, though it takes a similar length of time as FO+SEP to print.

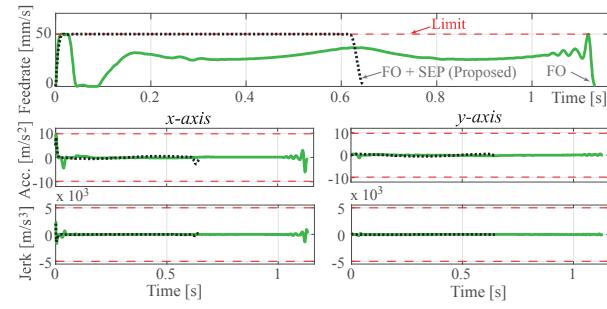


Fig. 10 Feedrate, acceleration and jerk profiles of trajectories generated by FO and FO+SEP using aggressive kinematic limits and $\hat{E}_{\max} = 14 \mu\text{m}$

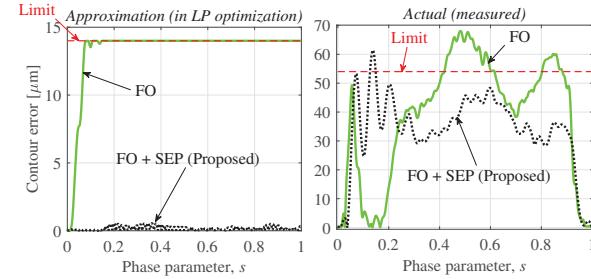


Fig. 11 Simulated (approximated) and actual (measured) contour error (CE) profiles of FO and FO+SEP

Table 2 Comparison of cycle and computation time of FO and FO + SEP

	Cycle time [s]	Computation time [s]
FO	1.13	37.5
FO + SEP (Proposed)	0.64	1.8

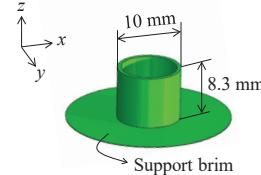


Fig. 12 CAD model of cylinder

4.2 Precision Motion Stage

The performance of FO+SEP is also validated without the use of state observer, by testing on a precision motion stage with direct encoder feedback. This is to show that FO+SEP is applicable to various types of servo systems that suffer from limited bandwidth.

4.2.1 Experimental Setup

A biaxial linear-motor-driven motion stage (Aerotech ALS 25010) is used, as shown in Fig. 14. The opti-

Algorithm	Conservative TAP	Aggressive TAP	FO	FO + SEP (Proposed)
Side view				
Top view				
Time (min)	11:06	5:24	9:53	5:30

Fig. 13 Side and top view of printed cylinders using Conservative TAP, Aggressive TAP, FO and FO+SEP on a support brim

mization algorithms of FO and FO+SEP are implemented on dSPACE DS1103 real-time control board running at 1 kHz sampling rate, connected to Soloist CP controller/drive. Each axis is controlled by pre-tuned closed-loop P-PI controller and velocity feedforward. The planar motion stage is equipped with optical linear encoders with resolution $0.1 \mu\text{m}$ to provide position feedback on each axis.

As with the 3D printer, the servo dynamics of the x and y axis are measured in the form of FRFs and fitted as transfer functions. Fig 15 shows the measured and modeled FRFs of each axis of the planar motion stage. The input of each FRF are position commands constructed by swept sine acceleration to the servomotor, and output is the position measured by encoders on each axis.

4.2.2 Benchmarking to Determine Approximate CE Limit

Unoptimized position commands generated using a Conservative TAP [8] are used for benchmarking to determine suitable approximate CE error in traversing a circle of 5 mm radius. Fig. 16(a) shows the TAP feedrate profile generated using conservative kinematic limits of $F_{max} = 40 \text{ mm/s}$, $A_{max} = 0.4 \text{ m/s}^2$, $J_{max} = 4 \text{ m/s}^3$; the acceleration and jerk profiles are omitted for the sake of brevity.

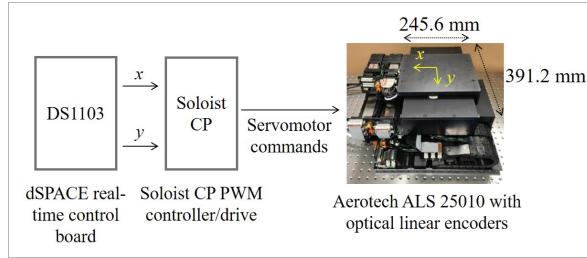


Fig. 14 Experimental setup

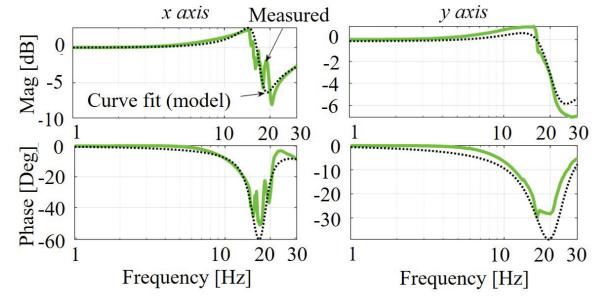


Fig. 15 Measured and curve fitted FRFs of x and y axes of planar motion stage

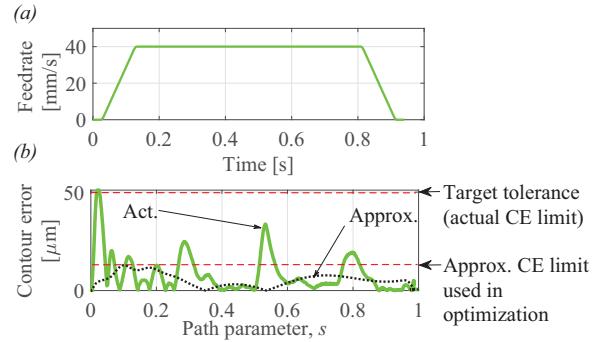


Fig. 16 (a) Commanded feedrate and (b) simulated (approximated) and actual (measured) CE profile of TAP motion commands

Fig. 16(b) shows simulated (approximated) and actual (measured) CE profiles of the TAP position commands applied to the planar motion stage. The simulations are performed using the curve fit linear dynamic model of Fig. 15. The conservative TAP yields maximum simulated and actual CE of $13 \mu\text{m}$ and $50 \mu\text{m}$, respectively. Because the actual contouring accuracy of the TAP is considered to be satisfactory, $\hat{E}_{max} = 13 \mu\text{m}$ is selected as the approximate CE limit for LP-based optimization, to help keep CE close to the target $50 \mu\text{m}$ in reality.

4.2.3 Optimization Results using FO and FO+SEP

We compare FO and the proposed FO+SEP with a goal to achieve similar accuracy as Conservative TAP in Fig. 16(b) with the shortest cycle time. To do this, aggressive kinematic limits are imposed on both FO and FO+SEP as: $F_{max} = 80 \text{ mm/s}$, $A_{max} = 8 \text{ m/s}^2$, $J_{max} = 8,000 \text{ m/s}^3$. In addition, approximate CE limit of $\hat{E}_{max} = 13 \mu\text{m}$ is imposed. In FO, $C_x = C_y = \mathbf{I}$, and in FO+SEP, C_x and C_y are generated via the FBS approach described in Section 3.2 using a 5th degree B-spline with uniform knot vector and $n = 30$ control points. To reduce the problem size, another 5th degree

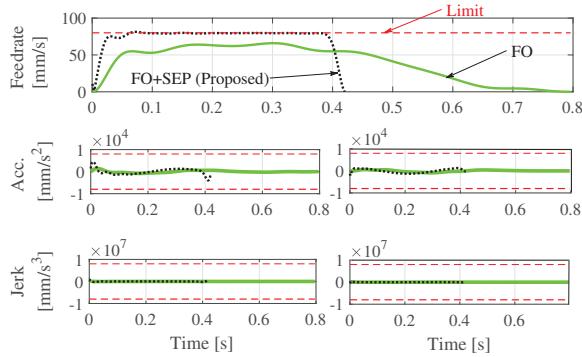


Fig. 17 Feedrate, acceleration and jerk profiles of trajectories generated by FO and FO+SEP using aggressive kinematic limits and $\hat{E}_{max} = 13 \mu\text{m}$

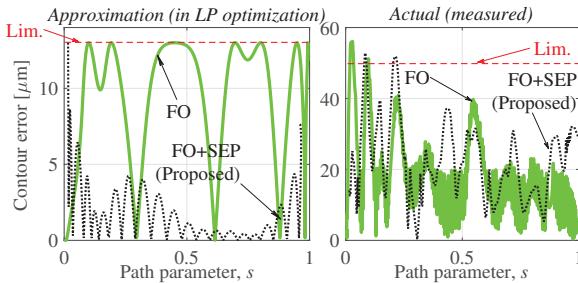


Fig. 18 Simulated (approximated) and actual (measured) CE profiles of FO and FO+SEP

B-spline with uniform knot vector and $n_p = 30$ control points is used to parametrize s . Both the FO and FO+SEP are initialized using unoptimized TAP trajectories.

Fig. 17 shows the commanded feedrate, acceleration, and jerk profiles of FO and FO+SEP. Fig. 18 shows the simulated (approximated) and actual (measured) CE profiles. Both FO and FO+SEP enforce the tolerance in simulations, which enforces the experimental error close to the target. However, as was in the experiment with the 3D printer, FO has to slow down because it hits the CE limit while FO+SEP is able to stay very close to the maximum speed throughout the motion. Consequently, FO+SEP completes the motion in 0.42 s, which is 47% faster than FO at 0.79 s, as summarized in Table 3. Note that the cycle time of FO is only 0.15 s (i.e., 16.0%) faster than the conservative TAP. The computation time for FO+SEP is 0.04 s; FO's is higher at 0.12 s because it is operating very close to the approximate error limit imposed.

Table 3 Comparison of cycle and computation time of FO and FO + SEP

	Cycle time [s]	Computation time [s]
FO	0.79	0.12
FO + SEP (Proposed)	0.42	0.04

5 Conclusions and Future Work

This paper has introduced a new concept of simultaneous FO and SEP (i.e., FO+SEP), and proposed a novel approach for realizing FO+SEP using time-based LP.

A time-based LP approach, which uses time as the independent variable, is formulated and compared with commonly-used path-based LP. Time-based LP is preferable to path-based LP in two aspects: (1) axis jerk constraints can be imposed without the use pseudo-jerk approximation, and (2) any general linear dynamics constraints can be incorporated. It is shown in the simulations that time-based LP provides an elegant and computationally efficient approach for FO+SEP.

Compared to the standard practice of performing FO and SEP independently, FO+SEP relaxes the error tolerance constraints imposed on FO, allowing shorter cycle times without violating tolerance constraints. Experiments carried out on a 3D printer and precision motion stage yielded up to 43% and 47% reduction, respectively, in cycle time using FO+SEP compared to FO, subject to the same tolerance and kinematic constraints.

Future work will explore limited-preview (windowing) [14,16] into FO+SEP to enable its application to longer trajectories. Implementation of sequential linear programming approach will also be studied to reduce the linearization error [31]. Lastly, incorporation of actuator (e.g., torque) limits [20] into FO+SEP will also be explored to help expand its performance benefits over FO.

Acknowledgements This work is funded by the National Science Foundation's award 1825133: Boosting the Speed and Accuracy of Vibration-Prone Manufacturing Machines at Low Cost through Software.

References

1. Altintas Y, Verl A, Brecher C, Uriarte L, Pritschow G (2011) Machine Tool Feed Drives. CIRP Annals 60(2):779-796
2. Tomizuka M (1987) Zero Error Tracking Algorithm for Digital Control. ASME J Dyn Sys Meas Control 109(1):65-68
3. Zhang D, Chen Y, and Chen Y (2016) Iterative Pre-Compensation Scheme of Tracking Error for Contouring Error Reduction. Int J Adv Manuf Technol 87(9-12):3279-3288

4. Ernesto CA, Farouki RT (2010) Solution of Inverse Dynamics Problems for Contour Error Minimization in CNC Machines. *Int J Adv Manuf Technol* 49(5):589-604
5. Altintas Y, Khoshdarregi MR (2012) Contour Error Control of CNC Machine Tools with Vibration Avoidance. *CIRP Annals* 61(1):335-338
6. Zhang K, Yuen A, Altintas Y (2013) Pre-Compensation of Contour Errors in Five-Axis CNC Machine Tools. *Int J Mach Tool Manu* 74:1-11
7. Yang S, Ghasemi AH, Lu X, Okwudire CE (2015) Pre-Compensation of Servo Contour Errors Using a Model Predictive Control Framework. *Int J Mach Tool Manu* 98:50-60
8. Gordon DJ, Erkorkmaz K (2013) Accurate Control of Ball Screw Drives Using Pole-Placement Vibration Damping and a Novel Trajectory Prefilter. *Precision Engineering* 37(2):308-322
9. Chin JH, Lin TC (1997) Cross-Coupled Precompensation Method for the Contouring Accuracy of Computer Numerically Controlled Machine Tools. *Int J Mach Tool Manu* 37(7):947-967
10. Song D, Ma J, Jia Z, Gao Y (2017) Estimation and Compensation for Continuous-Path Running Trajectory Error in High-Feed-Speed Machining. *Int J Adv Manuf Technol* 89(5-8):1495-1508
11. Liu W, Sun Y, Yuan X, Chen M (2016) A New Approach to the Pre-Compensation of Contour Errors For Three-Axis Machine Tools Using an Adaptive Cross-Coupled Controller. *Int J Adv Manuf Technol* 90(9-12):3711-3725
12. Liu W, Ren F, Sun Y, Jiang S (2018) Contour Error Pre-Compensation for Three-Axis Machine Tools by Using Cross-Coupled Dynamic Friction Control. *Int J Adv Manuf Technol* 98(1):551-563
13. Okwudire C, Ramani K, Duan M (2016) A Trajectory Optimization Method for Improved Tracking of Motion Commands Using CNC Machines That Experience Unwanted Vibration. *CIRP Annals* 65(1):373-376
14. Duan M, Yoon D, Okwudire CE (2018) A Limited-Preview Filtered B-Spline Approach to Tracking Control-With Application to Vibration-Induced Error Compensation of a 3D Printer. *Mechatronics* 56:287-296
15. Fan W, Gao XS, Lee CH, Zhang K, Zhang Q (2013) Time-Optimal Interpolation for Five-Axis CNC Machining along Parametric Tool Path Based on Linear Programming. *Int J Adv Manuf Technol* 69(5-8):1373-1388
16. Erkorkmaz K, Chen QGC, Zhao MY, Beudaert X, and Gao XS (2017) Linear Programming and Windowing Based Feedrate Optimization for Spline Toolpaths. *CIRP Annals* 66(1):393-396
17. Bharathi A, Dong J (2016) Feedrate Optimization for Smooth Minimum-Time Trajectory Generation with Higher Order Constraints. *Int J Adv Manuf Technol* 82(5):1029-1040
18. Zhang Q, Li SR (2013) Efficient Computation of Smooth Minimum Time Trajectory for CNC Machining. *Int J Adv Manuf Technol* 68:683-692
19. Guo J, Zhang Q, Gao XS, Li H (2015) Time Optimal Feedrate Generation with Confined Tracking Error based on Linear Programming. *Journal of Systems Science and Complexity* 28(1):90-95
20. Zhang Y, Ye P, Zhao M, Zhang H (2019) Dynamic Feedrate Optimization for Parametric Toolpath with Data-Based Tracking Error Prediction. *Mechanical Systems and Signal Processing* 120:221-223
21. Zhang K, Yuan CM, Gao XS (2013) Efficient Algorithm for Time-Optimal Feedrate Planning and Smoothing with Confined Chord Error and Acceleration. *Int J Adv Manuf Technol* 66(9-12):1685-1697
22. Sun Y, Zhao Y, Bao Y, Guo D (2015) A Smooth Curve Evolution Approach to the Feedrate Planning on Five-Axis Toolpath with Geometric and Kinematic Constraints. *Int J Mach Tool Manu* 97:86-97
23. Liang F, Zhao J, Ji S (2017) An Iterative Feed Rate Scheduling Method with Confined High Order Constraints in Parametric Interpolation. *Int J Adv Manuf Technol* 92(5):2001-2015
24. Lu L, Zhang L, Ji S, Han Y, Zhao J (2016) An Offline Predictive Feedrate Scheduling Method for Parametric Interpolation Considering the Constraints in Trajectory and Drive Systems. *Int J Adv Manuf Technol* 83(9-12):2143-2157
25. Lai HY, Lin KY, Tseng SJ, Ueng WD (2008) On the Development of a Parametric Interpolator with Confined Chord Error, Feedrate, Acceleration and Jerk. *Int J Adv Manuf Technol* 37(1):104-121
26. Lam D, Manzie C, Good MC (2012) Model Predictive Contouring Control for Biaxial Systems. *IEEE Transactions on Control Systems Technology* 21(2):552-559
27. Koren Y, Lo CC (1991) Variable-Gain Cross-Coupling Controller for Contouring. *CIRP Annals* 40(1):371-374
28. Ramani KS, Duan M, Okwudire CE, Ulsoy AG (2017) Tracking Control of Linear Time-invariant Nonminimum Phase Systems using Filtered Basis Functions. *Journal of Dynamic Systems, Measurement, and Control* 139(1):011001-1 to 011001-11
29. Luenberger DG (1971) An Introduction to Observers. *IEEE Trans. On Automatic Control* 16(6):596-602
30. Okwudire C, Huggi S, Supe S, Huang C, Zeng B (2018) Low-Level Control of 3D Printers from the Cloud: A Step toward 3D Printer Control as a Service. *Inventions* 3(3):56
31. Nocedal J, Wright SJ (2006) Numerical Optimization. Berlin, New York