Software-based Compensation of Vibration-induced Errors of a Commercial Desktop 3D Printer

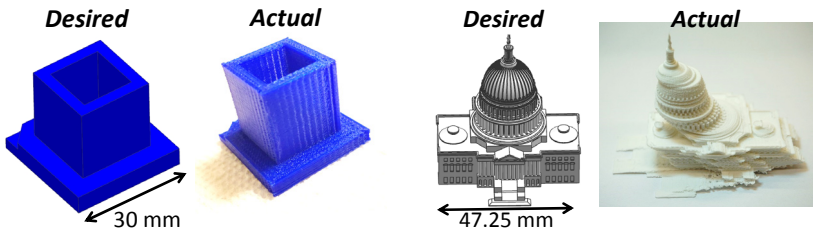
D. Yoon, M. Duan, C.E. Okwudire
2350 Hayward Street, Ann Arbor, MI 48109
University of Michigan
okwudire@umich.edu

Abstract
Commercial desktop 3D printers are designed with light, but flexible, structures and driven with stepper motors in order to reduce their cost, size and weight. As a result, parts manufactured on such 3D printers suffer from surface waviness and registration errors (caused by stepper motors skipping counts) due to excessive vibration triggered by the motion of the print head or build platform. Such vibration-induced errors not only mar the aesthetics of 3D printed parts, but often lead to highly distorted and hence scrapped parts. The most common solution to this problem, not involving hardware modifications, is to reduce motion speed at the expense of productivity. This paper demonstrates the use of a software-based method called filtered B-splines (FBS) for mitigating vibration-induced errors of 3D printers without sacrificing productivity. In the FBS method, the motion command sent to the machine is parameterized using B-splines, whose basis functions are filtered using a model of the machine's dynamics. The control points associated with the filtered basis functions are then selected to minimize tracking errors. The effectiveness of the FBS method in mitigating surface waviness and registration errors is demonstrated on parts printed on a commercial desktop 3D printer.

Keywords
3D printer, Vibration, Control, Motion Command Generation, B-splines

1. Introduction

3D printing (aka additive manufacturing) is becoming a popular means for prototyping and manufacturing of complex part geometries. It is currently a $7.3 billion industry and, by 2020, the market for 3D printers and the accompanying services is expected to eclipse $21 billion [1]. Due to their relatively low cost, the commercial use of desktop 3D printers (i.e., printers priced below $5,000) is expanding in industries where customization and personalized products are important for consumers, e.g., prosthetics, jewelry, and accessories [2]. The cost of desktop 3D printers has been lowered, in large part, by constructing them using fewer and lighter structural members, and by controlling them using stepper motors driven by open-source micro-controllers. However, the resulting structural flexibilities cause the printers to vibrate excessively as the print head and/or build platform move rapidly during printing, thus causing errors that compromise printing performance. One common error caused by excessive vibration is surface waviness which mars the aesthetics of the 3D printed part, as shown in Figure 1 (a). Moreover, as the machine vibrates, its stepper motors (which have no feedback sensors) are often unable to provide enough torque to hold position, hence they skip counts (i.e., lose track of their position) resulting in so-called registration errors. As shown in Figure 1 (b), registration errors often distort the printed part severely, causing it to be scrapped.

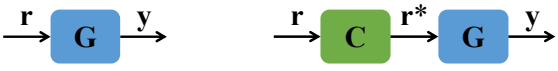


(a) (b)

Figure 1. Excessive vibration of desktop 3D printers causes (a) aesthetically unpleasing surface waviness, and (b) scrapped part due to registration errors.

Vibration-induced errors in CNC manufacturing machines can be dealt with via hardware modifications, e.g., by stiffening the machine or adding damping [3]. However, such solutions often involve additional material costs and weight, which are not desirable for desktop 3D printers. The errors could also be mitigated via active vibration control techniques [3]. However, such control techniques require feedback sensors and high-sample-rate real-time control hardware, which are not commercially viable for low-cost desktop 3D printers. An alternative approach, that does not require extra hardware or active control, is to generate the motion commands sent to the machine such that unwanted vibration is avoided or reduced (i.e., feedforward control) [3]. The most popular approach for achieving this goal, used on most desktop 3D printers, is to generate motion commands that are smooth (i.e., commands that have little or no high-frequency content). However, the attenuated high-frequency content of smooth motion commands often implies loss of motion speed (productivity), and results in suboptimal vibration reduction, because knowledge of the machine's dynamics is not exploited [3, 4]. Better vibration reduction can be achieved by using filters (e.g., FIR filters) or input shapers to generate motion commands with low frequency content around the resonance modes of the machine (assuming that its dynamics is known). However, filters and input shapers introduce time delays, which sacrifice productivity, and could severely degrade accuracy [4]. Okwudire et al. [4] presented a filtered B-spline (FBS) motion command optimization method for minimizing vibration-induced tracking errors subject to kinematic limits of a machine, without introducing time delays. The effectiveness of the FBS method was demonstrated in laboratory experiments by measuring the vibration of a metal block attached to a motion stage. However, its effectiveness in improving the quality of a manufactured part via online implementation was not studied. In this paper, the FBS method is briefly reviewed, and its online implementation on a commercial desktop 3D printer with structural flexibilities is presented. Parts are manufactured on the 3D printer, with and without the FBS method, and significant reductions in vibration-induced surface waviness and registration errors, without sacrificing productivity, are demonstrated.

2. Overview of motion command optimization using filtered B-splines [4]



(a) (b)

Figure 2. Servo dynamics of CNC machine (a) without, and (b) with motion command optimization.

Figure 2 (a) shows the servo dynamics, **G**, of a CNC machine to which motion command, **r**, is inputted and actual motion, **y**, is outputted. Due to the dynamics (e.g., structural flexibilities) present in **G**, tracking error, **e** = **r** – **y**, is created. In order to minimize **e**, a motion command optimizer, **C**, is introduced, as shown in Figure 2 (b), to modify **r** to an optimal form, **r***, based

on knowledge of the machine dynamics. To achieve this goal, **r*** is assumed to be parameterized (temporally, not spatially) using a B-spline curve of degree m and $n + 1$ coefficients (control points), such that [4]

$$\underbrace{\begin{bmatrix} r^*(0) \\ r^*(1) \\ \vdots \\ r^*(E) \end{bmatrix}}_{\mathbf{r}^*} = \underbrace{\begin{bmatrix} N_{0,m}(\xi_0) & N_{1,m}(\xi_0) & \cdots & N_{n,m}(\xi_0) \\ N_{0,m}(\xi_1) & N_{1,m}(\xi_1) & \cdots & N_{n,m}(\xi_1) \\ \vdots & \vdots & \ddots & \vdots \\ N_{0,m}(\xi_E) & N_{1,m}(\xi_E) & \cdots & N_{n,m}(\xi_E) \end{bmatrix}}_{\mathbf{N}} \underbrace{\begin{bmatrix} p^*(0) \\ p^*(1) \\ \vdots \\ p^*(n) \end{bmatrix}}_{\mathbf{p}^*}; \; n \le E \tag{1}$$

where **N** and **p** are the basis function matrix and coefficient vector, respectively, while 0, 1, …, E are discrete time steps, and $\xi \in [0, 1]$ is the spline parameter, representing normalized time. A uniform knot vector $\mathbf{g} = \begin{bmatrix} g_0 & g_1 & \cdots & g_{m+n+1} \end{bmatrix}^{\mathrm{T}}$ and matrix **N** are generated as

$$g_k = \begin{cases} 0 & 0 \le k \le m \\ \dfrac{k-m}{n-m+1} & m+1 \le k \le n \\ 1 & n+1 \le k \le m+n+1 \end{cases} \tag{2}$$

$$N_{j,m}(\xi) = \frac{\xi - g_j}{g_{j+m} - g_j} N_{j,m-1}(\xi) + \frac{g_{j+m+1} - \xi}{g_{j+m+1} - g_{j+1}} N_{j+1,m-1}(\xi); \quad N_{j,0}(\xi) = \begin{cases} 1 & g_j \le \xi < g_{j+1} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Applying the modified (i.e., optimized) motion command, **r***, to the machine yields

$$\mathbf{y} = \mathbf{G}\mathbf{r}^* = \mathbf{G}\mathbf{N}\mathbf{p}^* = \tilde{\mathbf{N}}\mathbf{p}^* \tag{4}$$

where $\tilde{\mathbf{N}}$ is the filtered basis function matrix (i.e., **N** filtered by **G**). The optimal coefficients, **p***, can be calculated by minimizing the 2-norm of **e**, resulting in the well-known least-squares solution:

$$\mathbf{p}^* = \left(\left(\tilde{\mathbf{N}}^{\mathrm{T}} \tilde{\mathbf{N}} \right)^{-1} \tilde{\mathbf{N}}^{\mathrm{T}} \right) \mathbf{r} \tag{5}$$

The motion command optimizer, **C**, therefore becomes

$$\mathbf{C} = \mathbf{N} \left(\left(\tilde{\mathbf{N}}^{\mathrm{T}} \tilde{\mathbf{N}} \right)^{-1} \tilde{\mathbf{N}}^{\mathrm{T}} \right) \tag{6}$$

B-splines exhibit a local property, meaning that a change in the lth coefficient, $p^*(l)$, influences only a small subset of **r***. Using this local property, **p*** (hence **r***) can be calculated in small batches, as in look-ahead functionalities of modern CNCs, without requiring the controller to read the entire G-code in advance [5]. This feature drastically reduces the computational cost of the FBS method [5], thus facilitating recursive online implementation of the motion command optimizer, **C**, on low-cost desktop 3D printers, as demonstrated below.

3. Application of FBS method to commercial desktop 3D printer

3.1. Experimental set-up: Commercial desktop 3D printer

Figure 3 shows a commercially available desktop 3D printer (HICTOP Prusa i3) used in this paper to demonstrate the compensation of vibration-induced errors based on the FBS method. The motion of the printer's build platform is along the x-axis, while its print head moves along the y- and z-axes. All three axes of the printer are controlled by stepper motors, but the focus of this study is on controlling its x- and y-axis motions which generate significant vibration, due to

the printer's flexible structure, as its print head and build platform move. Figure 4 shows the measured and curve fit *x*- and *y*-axis frequency response functions (FRFs) of the 3D printer. The FRFs are measured by applying swept sine acceleration signals to the printer's stepper motors (each having 12.5 μm stepping resolution) and measuring the relative acceleration of the build platform and print head using accelerometers (PCB Piezotronics 393B05). The curve fit model is generated using MATLAB®'s *invfreqs* function. To identify the FRFs and implement the FBS method, the printer's proprietary motion controller is bypassed. Instead, axis-level motion (i.e., step and direction) commands are sent to the printer's stepper motors at 1 kHz sampling rate using a real-time controller (dSPACE DS1007 and DS5203) via stepper motor drives (Pololu DRV8825). A script written in MATLAB® reads a G-code file (generated using Cura™ software package) and parses the G-code information into axis-level motion commands, emulating motion command generators of 3D printers. The real-time controller optimizes the motion commands using the FBS method following the procedure outlined in the preceding section.

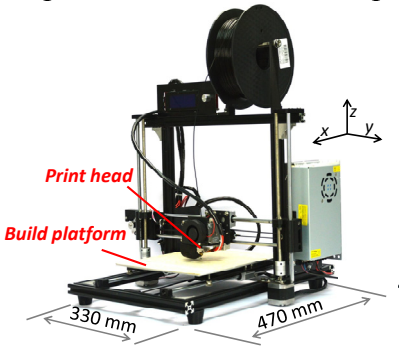


Figure 3. Commercial desktop 3D printer (HICTOP Prusa i3) used in this study.

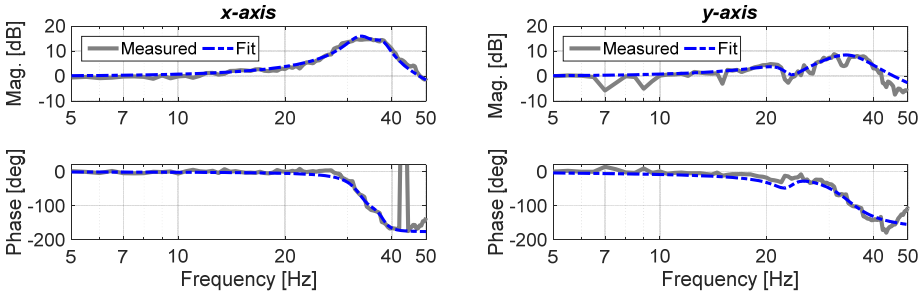


Figure 4. Frequency response functions of x- and y-axes of the 3D printer. Least-squares curve fitting is used to identify the axis level dynamics.

3.2. Case study I: Mitigation of surface waviness of 3D printed part

The square block model shown in Figure 1 (a) is printed using the 3D printer of Figure 3 with different acceleration limits imposed on the motion commands (while maintaining the feedrate at 60 mm/s for all cases). For each case, the part is printed using the unmodified motion commands (as the baseline approach), as well as the motion commands optimized via the FBS method. Figure 5 (a) and (b) show the blocks printed using the two methods (both using acceleration limit of 7 m/s^2), and the 3D surface profile, *h*, of the highlighted surfaces, measured using a laser displacement sensor (Keyence LK G-10). Notice that the vibration-induced surface waviness of the baseline case is significantly reduced by the FBS method. Figure 6 compares the surface roughness (RMS *h* values) of blocks printed using different acceleration limits. Observe that the surface quality of the baseline approach deteriorates significantly at higher acceleration limits compared to that of the FBS method, which stays relatively consistent for all acceleration limits investigated.

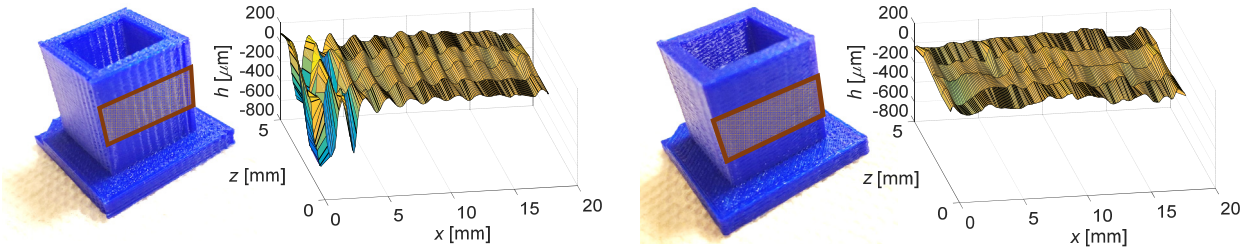


Figure 5. Comparison of photographs and measured surface profiles (h) of the highlighted surfaces of blocks printed using (a) baseline approach and (b) FBS method (both with acceleration limit of 7 m/s^2). Vibration-induced surface waviness of baseline case is greatly attenuated using FBS.

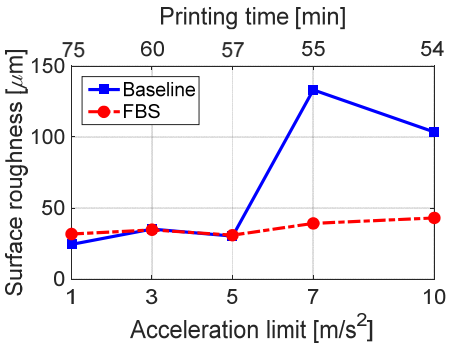
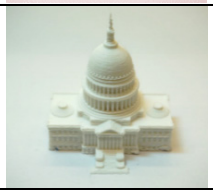
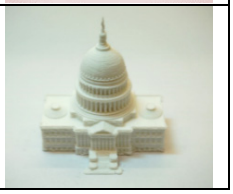


Figure 6. Surface roughness of 3D printed blocks comparing x- & y-axes commands generated using baseline (unmodified) and FBS (optimized) methods for different acceleration limits (total printing time).

3.3. Case study II: Mitigation of registration errors in 3D printed part

A scale model of the US Capitol, shown in Figure 1 (b), is printed to demonstrate the common problem of vibration-induced registration errors in 3D printed parts, and the ability of the FBS method to mitigate them. The part is printed using the baseline approach and the FBS method, with different acceleration limits, while keeping the feedrate at 60 mm/s. As shown in Table 1, registration errors occur with the baseline case as the acceleration limit is increased above 3 m/s^2, causing severe distortion of the printed parts. However, the parts printed using the FBS method do not exhibit such registration errors. As a result, the FBS method enables quality prints at much shorter printing times compared to the baseline case.

Table 1. 3D printed models of US Capitol using different acceleration limits (which influences total printing time). The FBS method eliminates the registration errors observed in the baseline case at higher acceleration limits; the feedrate is 60 mm/s for all cases.

Acc. limit	1 m/s^2	3 m/s^2	5 m/s^2	7 m/s^2	10 m/s^2
Printing time	3:59 h	2:42 h	2:22 h	2:12 h	2:06 h
Baseline					
FBS					

To explain the reason for the registration errors, Figure 7 compares the measured *x*-axis acceleration between print head and build platform for the baseline approach and FBS method (both with 7 m/s^2 acceleration limit). Notice that even though the acceleration of the motion command is limited to 7 m/s^2 (excepts for a few spikes here and there due to short distance travels), the actual acceleration of the baseline case turns out to be more than double this limit, because the build platform's motion heavily excites the resonance modes of the machine. The excessive vibration (i.e., acceleration) causes inertial loads that exceed the holding torque of the motors, hence they skip counts and lose track of their position. On the other hand, the FBS method maintains actual acceleration levels that are closer to the desired limit, hence avoiding the registration errors.

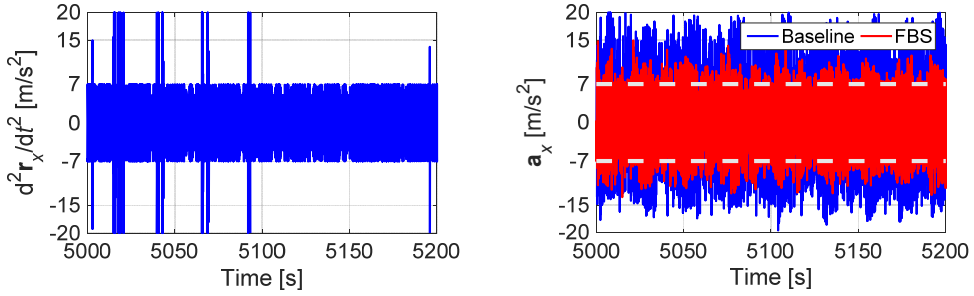


Figure 7. (a) Acceleration profile of x-axis reference command (unmodified). (b) On-machine measurement of x-axis acceleration. Dashed lines indicate the acceleration limit (7m/s^2) imposed.

4. Conclusion and future work

The filtered B-spline (FBS) method [4] is implemented on a commercial desktop 3D printer to mitigate its vibration-induced errors. Two commonly observed classes of vibration-induced errors in 3D printed parts (i.e., surface waviness and registration errors) are investigated. Compared to the baseline approach, the FBS method is shown to significantly reduce surface waviness and eliminate registration errors for the cases studied, thus allowing the production of high quality printed parts at higher speed (i.e., shorter manufacturing time). Future work will implement the FBS method on low-cost open-source micro-controllers like Arduino and Raspberry Pi, rather than dSPACE, to further demonstrate its low computational cost and suitability for commercial desktop 3D printers.

References

[1] Wohlers T, 3D Printing and additive manufacturing state of the industry annual worldwide progress report, Wohlers Report, 2015.

[2] Zaleski A, Here's why 2016 could be 3D printing's breakout Year, Fortune, http://fortune.com/2015/12/30/2016-consumer-3d-printing/, Accessed on Feb. 14th, 2017.

[3] Altintas Y, Verl A, Brecher C, Uriarte L, Pritschow G, Machine tool feed drives, CIRP Annals-Manufacturing Technology, 60(2), 779-796, 2011.

[4] Okwudire CE, Ramani KS, Duan M, A trajectory optimization method for improved tracking of motion commands using CNC machines that experience unwanted vibration, CIRP Annals-Manufacturing Technology, 65(1), 373-376, 2016.

[5] Duan M, Yoon D, Okwudire CE, A limited-preview filtered B-spline approach to tracking control – with application to vibration-induced error compensation of a commercial 3D printer, Under preparation.