

Received 20 December 2022, accepted 4 February 2023, date of publication 13 February 2023, date of current version 16 February 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3244194

RESEARCH ARTICLE

A Physics-Guided Data-Driven Feedforward Tracking Controller for Systems With Unmodeled Dynamics—Applied to 3D Printing

CHENG-HAO CHOU¹, MOLONG DUAN^{2,3}, (Member, IEEE),
AND CHINEDUM E. OKWUDIRE¹, (Senior Member, IEEE)

¹Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI 48109, USA

²Department of Mechanical and Aerospace Engineering, Hong Kong University of Science and Technology, Hong Kong, SAR, China

³HKUST Shenzhen-Hong Kong Collaborative Innovation Research Institute, Shenzhen, China

Corresponding author: Chinedum E. Okwudire (okwudire@umich.edu)

This work was supported in part by the National Science Foundation under Grant 1931950, in part by CISCO Systems Inc., and in part by the Hong Kong University of Science and Technology (HKUST)-Guangzhou University (GZU) Joint Research Collaboration Fund.

ABSTRACT A hybrid (i.e., physics-guided data-driven) feedforward tracking controller is proposed for systems with unmodeled linear or nonlinear dynamics. The proposed controller is based on the filtered basis functions (FBF) approach, and hence called a hybrid FBF controller. It formulates the feedforward control input to a system as a linear combination of a set of basis functions whose coefficients are selected to minimize tracking errors. To predict the system response and thereby the tracking errors, the basis functions are filtered using a combination of two linear models. The first model is physics-based and remains unaltered during the execution of the controller, while the second is data-driven and is continuously updated during the execution of the controller. To ensure its practicality and safe learning, the proposed hybrid FBF controller is equipped with the abilities to handle delays in data acquisition and to detect impending instability due to its inherent data-driven feedback loop. The effectiveness of the hybrid FBF controller is demonstrated via application to vibration compensation of a 3D printer with unmodeled linear and nonlinear dynamics. Thanks to the proposed hybrid FBF controller, the tracking accuracy of the 3D printer and the print quality are both significantly improved in experiments involving high-speed printing, compared to standard FBF controller that does not incorporate a data-driven model. Furthermore, the ability of the hybrid FBF controller to detect, and hence to potentially avoid, impending instability is demonstrated offline using data collected online from experiments.

INDEX TERMS Computer numerical control, data-driven modeling, motion control, nonlinear dynamical systems, predictive control, smart manufacturing, three-dimensional printing, trajectory tracking, vibration control.

I. INTRODUCTION

Tracking control is important in a wide range of applications, such as manufacturing, robotics, and aeronautics. The goal of the tracking control is to force the output of a dynamic system to follow a desired trajectory by minimizing the tracking error, i.e., the difference between the output and the desired trajectories. In particular, feedforward (FF) techniques play an important role in tracking control. Compared

to feedback (FB) control, FF control can preemptively compensate tracking error, which is impossible using FB control. In some applications, such as 3D printing, where open-loop-controlled stepper motors are typically used for motion generation, FF is the only resource for control. Moreover, even in applications where sensors are available, they may be unsuitable for real-time FB control due to practical constraints, like sensor and data acquisition delays, excessive noise, observability or stability concerns. Moreover, FF control can always be combined with FB control to improve overall tracking accuracy.

The associate editor coordinating the review of this manuscript and approving it for publication was Zheng Chen¹.

Theoretically, perfect FF tracking control can be achieved by exact model inversion. In linear systems, the model inversion can be simplified to pole-zero cancellation. However, due to the common occurrences of uncancellable zeros and uncertainty in system dynamics, exact model inversion may lead to unbounded control signals or poor tracking performance in practice. To solve this issue, various FF controllers based on approximate model inversion have been proposed in the literature, as summarized in review papers like [1], [2], and [3]. Examples include the zero phase error tracking controller (ZPETC) [4], zero magnitude error tracking controller (ZMETC) [5], extended bandwidth ZPETC [6], and model matching controller [7], to name a few. A relatively recent addition to the available tracking control methods is the filtered basis functions (FBF) approach [8], [9], [10], [11], [12], [13]. The origin of the FBF approach can be traced back to the work of Frueh and Phan [14] on inverse linear quadratic iterative learning control (ILC). It formulates the FF control input to a system as a linear combination of a set of basis functions that are filtered using the system's model to predict the system's output. Then, the unknown coefficients of the basis functions are then selected to minimize tracking error. A major advantage of the FBF approach over several other FF tracking control techniques is its versatility; it is applicable to any linear system and has been shown to yield accurate tracking regardless of the location of zeros of the system in the complex plane [8]. One application that has benefited from the versatility of the FBF approach is tracking control of vibration-prone 3D printers, which have a variety of underlying dynamics [9], [12], [13], [15].

However, a major deficiency of standard FF tracking control techniques is their inability to handle unmodeled dynamics prevalent in practice. For example, the vibration dynamics of 3D printers often have unmodeled nonlinear dynamics, e.g., due to friction, backlash and nonlinear belt stiffness [9], [12]. These nonlinear dynamics are left out of the identified linear models used for FF tracking control. Moreover, the vibration dynamics of 3D printers vary over time due to printer wear or end-user modifications. These changes to the printer dynamics are unknown, and hence not captured in the models used for FF tracking control. A variety of strategies that utilize data to correct the controller or the model have been proposed in the literature to enable FF tracking controllers to handle unmodeled dynamics. The existing methods can generally be broken down into two main categories: (1) iterative learning control (ILC) and (2) adaptive FF control. ILC updates feedforward control input signals based on data gathered from past iterations [16]. Hence, it can learn and adapt to unmodeled dynamic behaviors. A major advantage of ILC is that it does not require a parametric model to operate, and hence it can learn unmodeled dynamics with an unknown structure. Besides, ILC can also be directly incorporated with other controllers to adjust the reference without changing the controller settings [17]. However,

ILC's requirement of repeating trajectories or disturbances, or the assumptions that are made in order to apply ILC to nonrepeating trajectories, do not apply to many practical scenarios, such as tracking control in 3D printing where motions are typically nonrepetitive and complex due to customization. Moreover, when the stability or convergence is being analyzed, the broad knowledge of plant dynamics is typically required [16]. Adaptive FF controllers, on the other hand, tune the parameters of the controllers to account for unmodeled or changing dynamics [18], [19], [20], [21], [22], [23]. While there are some adaptive FF controllers that must be tuned iteratively using repeating trajectories or trials, e.g., [18], [20], and [21], literature has been exploring the approach to improve the extrapolation performance of the iterative tuning process [18], and besides not all adaptive FF controllers need this requirement. However, a major shortcoming of adaptive FF controllers is that they typically require a fixed structure with associated parameters to be tuned online [18], [19], [20], [21], [22], [23], despite the fact that the stability properties can be obtained more easily due to fixed structure. This diminishes their usefulness in situations where the exact structure of the unmodeled dynamics is a priori unknown or changing, such as the unknown aftermarket modification of 3D printers.

To handle both nonrepeating trajectories and the unmodeled dynamics with unknown or complex structure, neural networks (NN) have also been incorporated into FF or other controllers, due to their loose structure and ability to approximate any mathematical function [24], [25], [26], [27], [28], [29], [30]. However, their high dimensional and nonlinear nature typically make them difficult to be trained online or their stability to be rigorously analyzed to ensure safe learning and control. As a result, most existing works on NN-based controllers either ignore the stability of the NN portion [24], [25], [26], [27], [28] or analyze its stability under very strict conditions that are not validated experimentally [29], [30].

As an alternative approach, the present authors have proposed a hybrid (i.e., physics-guided data-driven) model for servo systems entirely based on linear models [31]. In the hybrid model, the output of a structured linear physics-based model is fed into a loosely structured linear data-driven model that is continuously updated online. It was shown through simulations and experiments that, thanks to the loose-structured data-driven component, the hybrid model was able to learn unknown nonlinear dynamics and improve overall model prediction compared to its purely physics-based counterpart. A preliminary version [32] of the present paper has incorporated the linear hybrid model into the FBF approach, where it was shown in simulations that the resulting hybrid FBF controller could significantly improve the tracking performance of a system with unmodeled nonlinear dynamics. However, the preliminary investigation in [32] failed to address two issues that are key to the practicality of the hybrid FBF controller, namely: (1) how

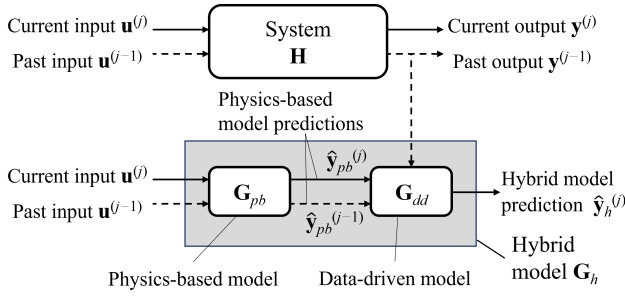


FIGURE 1. Overall framework of the linear hybrid model G_h [31].

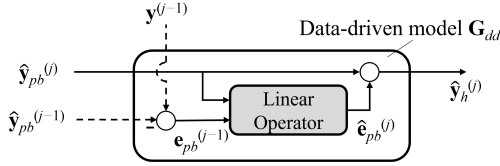


FIGURE 2. Detailed mechanism of the data-driven portion of G_h [31].

to handle the delays in the data acquisition for the inputs to the data-driven model, and (2) how to assess the stability of the hybrid FBF controller due to the inherent feedback loop introduced by its data-driven component. Moreover, the preliminary work did not validate the effectiveness of the hybrid FBF controller in experiments. To address these shortcomings, together with its preliminary version [32], which is reviewed in Section II, the original contributions of this paper are:

- 1) In Section II, proposing a physics-guided data-driven FF tracking controller based on the FBF approach. The proposed method, called the hybrid FBF controller, is able to provide accurate tracking for systems with unmodeled dynamics. Moreover, it is formulated to account for measurement delays which can occur in practice.
- 2) In Section III, proposing a rigorous stability analysis approach that can detect impending instability of the hybrid FBF controller due to training in order to avert instability.
- 3) In Section IV, demonstrating the effectiveness of the proposed hybrid FBF controller and stability analysis methods in online and offline experiments on a vibration-prone 3D printer with unmodeled linear and nonlinear dynamics.

These contributions are followed by the conclusions and the future work in Section V.

II. FORMULATION OF THE HYBRID FBF CONTROLLER

A. REVIEW OF THE LINEAR HYBRID MODEL

Consider a stable and causal SISO dynamic system H , which is sampled at a constant interval T_s . The input and output of H are respectively denoted by $u(k)$ and $y(k)$, where $k = 0, 1, 2, \dots$, is the time step sampled at the fixed interval T_s . In addition, suppose that the inputs and outputs are generated

and measured in small batches. The signal in the j -th batch is denoted by the superscript (j) and is defined, taking the measured output y as an example, as the vector

$$\mathbf{y}^{(j)} \triangleq [y(jN), y(jN + 1), \dots, y((j + 1)N - 1)]^T \quad (1)$$

where N is the length of the small batches and $j = 0, 1, 2, \dots$ is the batch index.

For traditional FF tracking controllers, a physics-based model G_{pb} , such as represented by a transfer function or a lifted representation, is used to predict the response of H . The controller then drives its prediction, denoted as $\hat{\mathbf{y}}_{pb}$, to be as close as possible to the desired trajectory $\mathbf{y}_d$. However, when there exists unmodeled dynamics that are not captured by G_{pb} , the inaccurate prediction of $\hat{\mathbf{y}}_{pb}$ can degrade the performance of the controller.

To enhance the prediction accuracy of servo systems, the authors have proposed a linear hybrid model [31]. As shown in Fig. 1, the linear hybrid model G_h cascades a linear physics-based model G_{pb} and a linear data-driven model G_{dd} to output a more accurate prediction $\hat{\mathbf{y}}_h$. More specifically, G_{dd} aims to complement G_{pb} using past measured data. To predict the output in the j -th batch, it takes the past and current physics-based predictions, $\hat{\mathbf{y}}_{pb}^{(j-1)}$ and $\hat{\mathbf{y}}_{pb}^{(j)}$, as well as the past measured output $\mathbf{y}^{(j-1)}$ as inputs, i.e.,

$$\begin{aligned} \hat{\mathbf{y}}_h^{(j)} &= \mathbf{G}_{dd} \left(\hat{\mathbf{y}}_{pb}^{(j-1)}, \hat{\mathbf{y}}_{pb}^{(j)}, \mathbf{y}^{(j-1)} \right) \\ &= \mathbf{G}_{dd} \left(\mathbf{G}_{pb} \mathbf{u}^{(j-1)}, \mathbf{G}_{pb} \mathbf{u}^{(j)}, \mathbf{y}^{(j-1)} \right) \\ &= \mathbf{G}_h \left(\mathbf{u}^{(j-1)}, \mathbf{u}^{(j)}, \mathbf{y}^{(j-1)} \right) \end{aligned} \quad (2)$$

In more detail, $\mathbf{G}_{dd}$ is constructed based on a linear regression model, and its detailed structure is shown in Fig. 2. The linear regression model estimates the physics-based model prediction error $\mathbf{e}_{pb} \triangleq \mathbf{y} - \hat{\mathbf{y}}_{pb}$ recursively for each time step, which is then added to $\hat{\mathbf{y}}_{pb}$ to obtain the final hybrid model prediction $\hat{\mathbf{y}}_h$. Therefore, using $\mathbf{G}_h$, $\hat{\mathbf{y}}_h^{(j)}$ is computed by recursively applying

$$\hat{\mathbf{y}}_h(k) = \hat{\mathbf{y}}_{pb}(k) + \hat{\mathbf{e}}_{pb}(k) = \hat{\mathbf{y}}_{pb}(k) + \hat{\mathbf{w}}^{(j)\top} \boldsymbol{\phi}(k) \quad (3)$$

for each time step k belongs to the j -th batch, i.e., $k \in \{jN, jN + 1, \dots, (j + 1)N - 1\}$, where $\boldsymbol{\phi}(k)$ is the feature vector for the prediction of $e_{pb}(k)$ and $\hat{\mathbf{w}}^{(j)}$ is the model weights trained with the data prior to the j -th batch. Furthermore, the feature vector $\boldsymbol{\phi}(k)$ is designed as

$$\boldsymbol{\phi}(k) = [1, \hat{\mathbf{y}}_{pb}(k - q + 1), \dots, \hat{\mathbf{y}}_{pb}(k), e_{pb}(k - p), \dots, e_{pb}(k - 1)]^T \quad (4)$$

where q and p are design parameters representing the number of $\hat{\mathbf{y}}_{pb}$ and e_{pb} terms included in $\boldsymbol{\phi}$, which can be decided based on the system dynamics and the prediction horizon [31], [32]. Besides, for all unavailable e_{pb} in Eq. (4), i.e., $e_{pb}(k)$ for $k \geq jN$, e_{pb} is replaced with the estimated values $\hat{e}_{pb}$ calculated from Eq. (3). On the other hand, $\hat{\mathbf{w}}^{(j)}$ is trained by minimizing the difference between e_{pb} and $\hat{e}_{pb}$ for

all time steps prior to the j -th batch, through the following regularized least squares optimization problem

$$\hat{\mathbf{w}}^{(j)} = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{k=0}^{jN-1} [e_{pb}(k) - \mathbf{w}^\top \boldsymbol{\phi}(k)]^2 + \lambda \|\mathbf{w}\|_2^2 \quad (5)$$

where $\lambda > 0$ is a tunable regularization factor that prevents overfitting. Besides, in practice, recursive least squares are adopted to solve Eq. (5) to prevent increasing computational expense as more data are collected [31], [32].

Remark 1: Even though the hybrid model $\mathbf{G}_h$ is linear, it is able to predict unmodeled nonlinear behavior by approximating it using the past measured data in a piecewise linear fashion [31]. Moreover, though $\mathbf{G}_h$ is loosely parameterized with q and p , it regresses on past prediction errors caused by unmodeled dynamics, thus making it versatile in its representation of unmodeled dynamics using linear combinations of past errors [31].

Remark 2: The recursive nature of $\mathbf{G}_h$ described in Eq. (3) allows it to predict a system's output with a prediction horizon of arbitrary length or a prediction with multiple batches ahead, despite the fact that the prediction accuracy drops as the horizon expands in practice. For example, the hybrid model prediction of $(j+1)$ -th batch can be formulated as

$$\begin{aligned} \hat{\mathbf{y}}_h^{(j+1)} &= \mathbf{G}_{dd} \left(\hat{\mathbf{y}}_{pb}^{(j)}, \hat{\mathbf{y}}_{pb}^{(j+1)}, \hat{\mathbf{y}}_h^{(j)} \right) \\ &= \mathbf{G}_{dd} \left(\hat{\mathbf{y}}_{pb}^{(j)}, \hat{\mathbf{y}}_{pb}^{(j+1)}, \mathbf{G}_{dd}(\hat{\mathbf{y}}_{pb}^{(j-1)}, \hat{\mathbf{y}}_{pb}^{(j)}, \mathbf{y}^{(j-1)}) \right) \\ &= \mathbf{G}_{dd,2} \left(\begin{bmatrix} \hat{\mathbf{y}}_{pb}^{(j-1)} \\ \hat{\mathbf{y}}_{pb}^{(j)} \end{bmatrix}, \hat{\mathbf{y}}_{pb}^{(j+1)}, \mathbf{y}^{(j-1)} \right) \\ &= \mathbf{G}_{h,2} \left(\begin{bmatrix} \mathbf{u}^{(j-1)} \\ \mathbf{u}^{(j)} \end{bmatrix}, \mathbf{u}^{(j+1)}, \mathbf{y}^{(j-1)} \right) \end{aligned} \quad (6)$$

where $\mathbf{G}_{dd,2}$ and $\mathbf{G}_{h,2}$ respectively define the data-driven and hybrid prediction models for two batches ahead.

B. DERIVATION OF HYBRID FBF CONTROLLER

As described in Sections I and II-A, the task of a FF tracking controller is to minimize the error between the predicted system response $\hat{\mathbf{y}}$ and the desired trajectory $\mathbf{y}_d$. Since the performance of the FF controller is closely linked to the accuracy of its prediction model, the linear hybrid model $\mathbf{G}_h$ is more favorable than the physics-based model $\mathbf{G}_{pb}$. Thus, $\mathbf{G}_h$ is applied to the receding horizon version of the standard FBF approach [9] to enhance its tracking performance.

Since $\mathbf{G}_h$ is linear, according to Eq. (2) and Eq. (3), the hybrid model prediction $\hat{\mathbf{y}}^{(j)}$ can be expressed as linear operation involving $\hat{\mathbf{y}}_{pb}^{(j-1)}$, $\hat{\mathbf{y}}_{pb}^{(j)}$, and $\mathbf{e}_{pb}^{(j-1)}$ as

$$\hat{\mathbf{y}}_h^{(j)} = \mathbf{G}_{dd} \left(\hat{\mathbf{y}}_{pb}^{(j-1)}, \hat{\mathbf{y}}_{pb}^{(j)}, \mathbf{y}^{(j-1)} \right) = \mathbf{L}_{dd}^{(j)} \cdot \begin{bmatrix} 1 \\ \hat{\mathbf{y}}_{pb}^{(j-1)} \\ \hat{\mathbf{y}}_{pb}^{(j)} \\ \mathbf{e}_{pb}^{(j-1)} \end{bmatrix} \quad (7)$$

where the $\mathbf{L}_{dd}^{(j)}$ matrix is constructed based on $\hat{\mathbf{w}}^{(j)}$ [32] and Eq. (7) can be further rewritten as follows

$$\hat{\mathbf{y}}_h^{(j)} = \mathbf{L}_{dd,a}^{(j)} \hat{\mathbf{y}}_{pb}^{(j)} + \mathbf{L}_{dd,uy}^{(j)} \hat{\mathbf{y}}_{pb}^{(j-1)} + \mathbf{L}_{dd,ue}^{(j)} \mathbf{e}_{pb}^{(j-1)} + \mathbf{L}_{dd,u1}^{(j)} \quad (8)$$

where the matrices $\mathbf{L}_{dd,a}^{(j)}$, $\mathbf{L}_{dd,uy}^{(j)}$, $\mathbf{L}_{dd,ue}^{(j)}$, and $\mathbf{L}_{dd,u1}^{(j)}$ are the submatrices of $\mathbf{L}_{dd}^{(j)}$.

Although the prediction result in Eq. (8) may generate the control input for the j -th batch, in practice, there are bound to be delays between the end of batch $j-1$ and the availability of the data captured at batch $j-1$. Therefore, $\mathbf{y}^{(j-1)}$ (and $\mathbf{e}_{pb}^{(j-1)}$) are not immediately available at the end of batch $j-1$, making the use of Eq. (8) impractical. To address this problem, without loss of generality, suppose the delay in acquiring $\mathbf{y}^{(j-1)}$ is one batch length, i.e., the latest available measurement is $\mathbf{y}^{(j-2)}$ when optimizing the inputs for the j -th batch. As described in Eq. (6), $\mathbf{y}^{(j-1)}$ can be substituted with $\hat{\mathbf{y}}_h^{(j-1)}$. In other words, the hybrid prediction $\hat{\mathbf{y}}_h^{(j)}$ can be rewritten as

$$\hat{\mathbf{y}}_h^{(j)} = \mathbf{G}_{dd,2} \left(\begin{bmatrix} \hat{\mathbf{y}}_{pb}^{(j-2)} \\ \hat{\mathbf{y}}_{pb}^{(j-1)} \end{bmatrix}, \hat{\mathbf{y}}_{pb}^{(j)}, \mathbf{y}^{(j-2)} \right) \quad (9)$$

Moreover, as discussed in [9] and in the preliminary work [32], to ensure the continuity of the optimized input, an overlapping window, denoted by the superscript $[j]$, is required; it is defined as

$$\mathbf{y}^{[j]} \triangleq [\mathbf{y}(jN), \mathbf{y}(jN+1), \dots, \mathbf{y}(jN+N_w-1)]^\top \quad (10)$$

where N_w is the window length. Without loss of generality, N_w in this paper is defined to be twice of the batch length N , i.e., $N_w = 2N$. Therefore, $\hat{\mathbf{y}}_h^{(j+1)}$ also needs to be predicted for control purposes and it can be expressed by

$$\hat{\mathbf{y}}_h^{(j+1)} = \mathbf{G}_{dd,3} \left(\begin{bmatrix} \hat{\mathbf{y}}_{pb}^{(j-2)} \\ \hat{\mathbf{y}}_{pb}^{(j-1)} \\ \hat{\mathbf{y}}_{pb}^{(j)} \end{bmatrix}, \hat{\mathbf{y}}_{pb}^{(j+1)}, \mathbf{y}^{(j-2)} \right) \quad (11)$$

Combining Eqs. (9) and (11), one can write the hybrid model prediction in the j -th window, denoted by $\hat{\mathbf{y}}_h^{[j]}$, as

$$\hat{\mathbf{y}}_h^{[j]} = \begin{bmatrix} \hat{\mathbf{y}}_h^{(j)} \\ \hat{\mathbf{y}}_h^{(j+1)} \end{bmatrix} = \mathbf{G}_{dd}^{[j]} \left(\begin{bmatrix} \hat{\mathbf{y}}_{pb}^{(j-2)} \\ \hat{\mathbf{y}}_{pb}^{(j-1)} \end{bmatrix}, \begin{bmatrix} \hat{\mathbf{y}}_{pb}^{(j)} \\ \hat{\mathbf{y}}_{pb}^{(j+1)} \end{bmatrix}, \mathbf{y}^{(j-2)} \right) \quad (12)$$

where $\mathbf{G}_{dd}^{[j]}$ denotes the trained data-driven model used to compute $\hat{\mathbf{y}}_h^{[j]}$. Eq. (12) can be further expressed as a linear operation, similar to Eq. (8), as

$$\hat{\mathbf{y}}_h^{[j]} = \mathbf{L}_{dd,a}^{[j]} \hat{\mathbf{y}}_{pb}^{[j]} + \mathbf{L}_{dd,uy}^{[j]} \hat{\mathbf{y}}_{pb}^{[j-1]} + \mathbf{L}_{dd,ue}^{[j]} \mathbf{e}_{pb}^{[j-1]} + \mathbf{L}_{dd,u1}^{[j]} \quad (13)$$

where $\hat{\mathbf{y}}_{pb}^{[j]}$, $\hat{\mathbf{y}}_{pb}^{[j-1]}$, and $\mathbf{e}_{pb}^{[j-1]}$ respectively represent

$$\hat{\mathbf{y}}_{pb}^{[j]} = \begin{bmatrix} \hat{\mathbf{y}}_{pb}^{(j)} \\ \hat{\mathbf{y}}_{pb}^{(j+1)} \end{bmatrix}, \hat{\mathbf{y}}_{pb}^{[j-1]} = \begin{bmatrix} \hat{\mathbf{y}}_{pb}^{(j-2)} \\ \hat{\mathbf{y}}_{pb}^{(j-1)} \end{bmatrix}, \mathbf{e}_{pb}^{[j-1]} = \mathbf{e}_{pb}^{(j-2)} \quad (14)$$

while $\mathbf{L}_{dd,a}^{[j]}$, $\mathbf{L}_{dd,uy}^{[j]}$, $\mathbf{L}_{dd,ue}^{[j]}$, and $\mathbf{L}_{dd,u1}^{[j]}$ here are constructed based on $\hat{\mathbf{w}}^{(j-1)}$ (due to the unavailability of $\mathbf{y}^{(j-1)}$) and can be

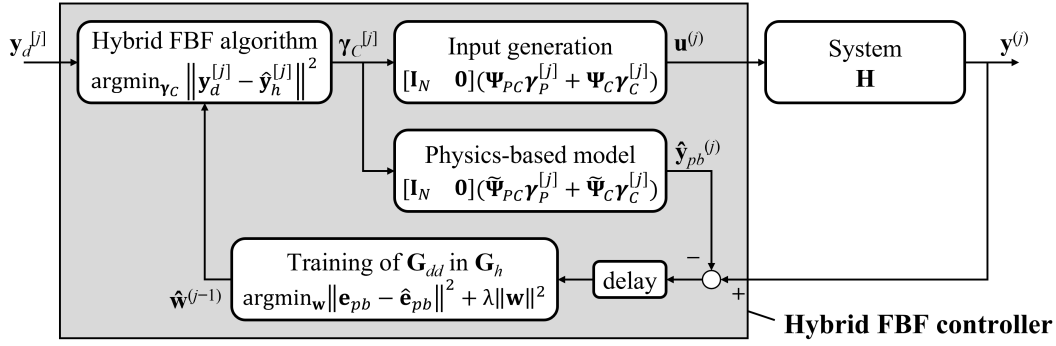


FIGURE 3. Simplified diagram for the overall framework of the hybrid FBF approach.

computed by replacing $\mathbf{e}_{pb}^{(j-1)}$ with $(\hat{\mathbf{y}}_h^{(j-1)} - \hat{\mathbf{y}}_{pb}^{(j-1)})$ and then comparing the coefficients of the linear operators.

To apply the prediction of Eq. (13) to the FBF approach, $\hat{\mathbf{y}}_h^{[j]}$ needs to be expressed as a function of the control inputs. Recall that the FBF approach formulates the control inputs $\mathbf{u}$ as the linear combination of a set of basis functions, which is then concatenated into a matrix Ψ , with unknown coefficients, denoted as $\boldsymbol{\gamma}$. That is, $\mathbf{u} = \Psi\boldsymbol{\gamma}$. Accordingly, the term in Eq. (13) that is affected by the optimized inputs, i.e., $\hat{\mathbf{y}}_{pb}^{[j]}$, is represented as a function of $\boldsymbol{\gamma}$. Furthermore, when implemented in receding horizon, as in [9], $\hat{\mathbf{y}}_{pb}$ can be decomposed into past, current, and future portions with respect to window j , i.e.,

$$\begin{aligned}\hat{\mathbf{y}}_{pb,P}^{[j]} &\triangleq [y_{pb}(0), y_{pb}(1), \dots, y_{pb}(jN-1)]^\top \\ \hat{\mathbf{y}}_{pb,C}^{[j]} &\triangleq \hat{\mathbf{y}}_{pb}^{[j]} \\ \hat{\mathbf{y}}_{pb,F}^{[j]} &\triangleq [y_{pb}(jN+N_w), y_{pb}(jN+N_w+1), \dots]^\top\end{aligned}\quad (15)$$

where P , C , and F respectively represent the past, current and future. Accordingly, $\hat{\mathbf{y}}_{pb}$ is formulated as

$$\begin{bmatrix} \hat{\mathbf{y}}_{pb,P}^{[j]} \\ \hat{\mathbf{y}}_{pb,C}^{[j]} \\ \hat{\mathbf{y}}_{pb,F}^{[j]} \end{bmatrix} = \begin{bmatrix} \tilde{\Psi}_P & \mathbf{0} & \mathbf{0} \\ \tilde{\Psi}_{PC} & \tilde{\Psi}_C & \mathbf{0} \\ \tilde{\Psi}_{PF} & \tilde{\Psi}_{CF} & \tilde{\Psi}_F \end{bmatrix} \begin{bmatrix} \boldsymbol{\gamma}_P^{[j]} \\ \boldsymbol{\gamma}_C^{[j]} \\ \boldsymbol{\gamma}_F^{[j]} \end{bmatrix}\quad (16)$$

where $\boldsymbol{\gamma}_P^{[j]}$, $\boldsymbol{\gamma}_C^{[j]}$, $\boldsymbol{\gamma}_F^{[j]}$ represent the vectors of coefficients corresponding to past, current, and future windows, $\tilde{\Psi}$ matrices represent the concatenation of the basis functions, Ψ , filtered by $\mathbf{G}_{pb}$, with the subscripts P , C , F of $\tilde{\Psi}$ denoting the input-output effects in the past, current and future sections, and the subscripts PC denoting the effects of the past inputs on the current outputs, similarly for PF and CF . Also, note that $\tilde{\Psi}$ is time-invariant when the batch length N is a multiple of the number of time steps represented by one coefficient γ .

From Eqs. (15) and (16), $\hat{\mathbf{y}}_{pb}^{[j]}$ is represented as a function of $\boldsymbol{\gamma}$ by

$$\hat{\mathbf{y}}_{pb}^{[j]} = \tilde{\Psi}_C \boldsymbol{\gamma}_C^{[j]} + \tilde{\Psi}_{PC} \boldsymbol{\gamma}_P^{[j]}\quad (17)$$

which is then inserted to Eq. (13). Replacing the unalterable terms in Eq. (13), i.e., the terms from the past ($\hat{\mathbf{y}}_{pb}^{[j-1]}$ and

$\mathbf{e}_{pb}^{[j-1]}$) and the bias term, with $\Phi_u^{[j]}$ and concatenating their linear operators as $\mathbf{L}_{dd,u}^{[j]}$, one can formulate $\hat{\mathbf{y}}_h^{[j]}$ as

$$\hat{\mathbf{y}}_h^{[j]} = \mathbf{L}_{dd,a}^{[j]} \tilde{\Psi}_C \boldsymbol{\gamma}_C^{[j]} + \mathbf{L}_{dd,a}^{[j]} \tilde{\Psi}_{PC} \boldsymbol{\gamma}_P^{[j]} + \mathbf{L}_{dd,u}^{[j]} \Phi_u^{[j]}\quad (18)$$

Finally, the tracking error is minimized window by window [9]. Given a desired trajectory for the j -th window $\mathbf{y}_d^{[j]}$, since $\boldsymbol{\gamma}_C^{[j]}$ is the only optimization variable, the objective function is given by

$$\min_{\boldsymbol{\gamma}_C^{[j]}} \|\mathbf{e}^{[j]}\|_2^2 \min_{\boldsymbol{\gamma}_C^{[j]}} \|\mathbf{y}_d^{[j]} - \hat{\mathbf{y}}_h^{[j]}\|_2^2\quad (19)$$

Therefore, the optimal $\boldsymbol{\gamma}_C^{[j]}$ for the hybrid FBF controller is calculated as

$$\boldsymbol{\gamma}_C^{[j]} = \left(\mathbf{L}_{dd,a}^{[j]} \tilde{\Psi}_C \right)^\dagger \left(\mathbf{y}_d^{[j]} - \mathbf{L}_{dd,a}^{[j]} \tilde{\Psi}_{PC} \boldsymbol{\gamma}_P^{[j]} - \mathbf{L}_{dd,u}^{[j]} \Phi_u^{[j]} \right)\quad (20)$$

Then, as in [9], the optimized control input for the j -th batch $\mathbf{u}^{(j)}$ is reconstructed by the linear combination of the basis functions, with only the first N time steps extracted, i.e.,

$$\mathbf{u}^{(j)} = [\mathbf{I}_N \quad \mathbf{0}] \left(\Psi_C \boldsymbol{\gamma}_C^{[j]} + \Psi_{PC} \boldsymbol{\gamma}_P^{[j]} \right)\quad (21)$$

After the optimized $\mathbf{u}^{(j)}$ is fed to the system, and the corresponding output $\mathbf{y}^{(j)}$ is measured, the linear hybrid model $\mathbf{G}_h$ is further updated by Eq. (5) using recursive least squares algorithm, with additional $\mathbf{e}_{pb}(k)$ terms coming from

$$\begin{aligned}\mathbf{e}_{pb}^{(j)} &= \mathbf{y}^{(j)} - \hat{\mathbf{y}}_{pb}^{(j)} \\ &= \mathbf{y}^{(j)} - [\mathbf{I}_N \quad \mathbf{0}] \left(\tilde{\Psi}_C \boldsymbol{\gamma}_C^{[j]} + \tilde{\Psi}_{PC} \boldsymbol{\gamma}_P^{[j]} \right)\end{aligned}\quad (22)$$

A flowchart for the hybrid FBF controller is shown in Fig. 3.

Remark 3: When the model weight $\hat{\mathbf{w}}$ of the linear hybrid model $\mathbf{G}_h$ is set equal to zero, $\mathbf{L}_{dd,a}^{[j]}$ and $\mathbf{L}_{dd,u}^{[j]}$ in Eq. (18) will respectively become an identity matrix and a zero matrix, thereby reducing the hybrid FBF approach to the standard FBF method of [9]. Under this scenario, it can also be inferred from Eq. (3) that $\hat{\mathbf{y}}_h = \hat{\mathbf{y}}_{pb}$.

III. STABILITY ANALYSIS FOR HYBRID FBF CONTROLLER

Although the linear hybrid model can give more accurate prediction [31] and accordingly lead to more accurate tracking using the hybrid FBF controller, the hybrid FBF controller may potentially suffer from instability due to the feedback loop introduced by the regression on past data. More specifically, according to Eq. (20), the feedback of the $\Phi_u^{[j]}$ term, which includes the past outputs, may potentially cause the overall closed-loop dynamics to become unstable. Furthermore, the varying values of the $\mathbf{L}_{dd}$ matrices in Eq. (20) due to training can also affect the stability of the system.

Therefore, to ensure the safety of the controller, we propose an analytical approach to check the stability of the system when using the hybrid FBF controller in this section. In particular, since the stability of the hybrid FBF controller largely depends on the measured data from the past, which is unknown in advance, the stability analysis proposed in this section aims to check if there will be any risk of instability for the next batch, given the current trained results of $\hat{\mathbf{w}}$. In practice, once impending instability is detected, a variety of mitigating actions can be taken to prevent it. For example, learning could be turned off, allowing the system to revert to the standard FBF controller, to use the model weights before the instability happens, or to run in a safe mode using conservative speed and acceleration.

To perform the stability check, we assume that the linear hybrid model has converged to accurate predictions, and thus the rate of weight changes due to updating is sufficiently low. This assumption is reasonable because, in practice, the hybrid model should be adequately trained for a few cycles to ensure that it is sufficiently accurate before using it for control. Given this assumption, the general idea of the stability check method is to write the linear hybrid model in the state-space format as the approximation of the actual system, i.e., approximating $\mathbf{y} = \hat{\mathbf{y}}_h$, and then formulating the input to the system as a control feedback law utilizing the hybrid FBF controller. That is, the linear hybrid model, described in Eq. (13), and the hybrid FBF controller, formulated as Eq. (20), are rewritten into the form of

$$\begin{aligned}\mathbf{x}^{[j+1]} &= \mathbf{A}^{[j]}\mathbf{x}^{[j]} + \mathbf{B}^{[j]}\mathbf{u}^{[j]} \\ \mathbf{u}^{[j]} &= \mathbf{K}^{[j]}\mathbf{x}^{[j]} + \mathbf{M}^{[j]}\mathbf{r}^{[j]}\end{aligned}\quad (23)$$

where the input vector $\mathbf{u}^{[j]}$ and the reference vector $\mathbf{r}^{[j]}$ are represented by the FBF coefficients $\boldsymbol{\gamma}_C^{[j]}$ and the desired trajectory $\mathbf{y}_d^{[j]}$, respectively. For the design of the state vector, $\mathbf{x}^{[j]}$, based on Eqs. (13), (14) and (17), $\hat{\mathbf{y}}_{pb}$ and $\hat{\mathbf{y}}_h$ from $(j-2)$ -th to j -th batch, as well as $\boldsymbol{\gamma}_P^{[j]}$ and the bias term must be included. Accordingly, $\mathbf{x}^{[j]}$, $\mathbf{u}^{[j]}$, and $\mathbf{r}^{[j]}$ are designed as

$$\begin{aligned}\mathbf{x}^{[j]} &= [\hat{\mathbf{y}}_h^{(j-2)\top}, \hat{\mathbf{y}}_h^{(j-1)\top}, \hat{\mathbf{y}}_h^{(j)\top}, \\ &\quad \hat{\mathbf{y}}_{pb}^{(j-2)\top}, \hat{\mathbf{y}}_{pb}^{(j-1)\top}, \hat{\mathbf{y}}_{pb}^{(j)\top}, \boldsymbol{\gamma}_P^{[j]\top}, 1]^\top\end{aligned}\quad (24a)$$

$$\mathbf{u}^{[j]} = \boldsymbol{\gamma}_C^{[j]\top} = [\boldsymbol{\gamma}^{(j)\top}, \boldsymbol{\gamma}^{(j+1)\top}]^\top\quad (24b)$$

$$\mathbf{r}^{[j]} = \mathbf{y}_d^{[j]} = [\mathbf{y}_d^{(j)\top}, \mathbf{y}_d^{(j+1)\top}]^\top\quad (24c)$$

Furthermore, by comparison of the linear operators of Eq. (23) with that of Eqs. (13), (17), (18), and (20), the matrices $\mathbf{A}^{[j]}$, $\mathbf{B}^{[j]}$, $\mathbf{K}^{[j]}$, and $\mathbf{M}^{[j]}$ are given by

$$\mathbf{A}^{[j]} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_N & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{21} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{24} & \mathbf{A}_{25} & \mathbf{0} & \mathbf{A}_{27} & \mathbf{A}_{28} \\ \mathbf{A}_{31} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{34} & \mathbf{A}_{35} & \mathbf{0} & \mathbf{A}_{37} & \mathbf{A}_{38} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_N & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{57} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{67} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{77} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}\quad (25)$$

$$\mathbf{B}^{[j]} = [\mathbf{0} \quad \mathbf{B}_2^\top \quad \mathbf{B}_3^\top \quad \mathbf{0} \quad \mathbf{B}_5^\top \quad \mathbf{B}_6^\top \quad \mathbf{B}_7^\top \quad \mathbf{0}]^\top\quad (26)$$

$$\mathbf{K}^{[j]} = [\mathbf{K}_1 \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{K}_4 \quad \mathbf{K}_5 \quad \mathbf{0} \quad \mathbf{K}_7 \quad \mathbf{K}_8]\quad (27)$$

$$\mathbf{M}^{[j]} = (\mathbf{L}_{dd,a}^{[j]} \tilde{\Psi}_C)^\dagger\quad (28)$$

where $\mathbf{I}_N$ are N -by- N identity matrices, $\mathbf{0}$ are zero matrices with appropriate size, and the submatrices $\mathbf{A}_{ij}$, $\mathbf{B}_{ij}$, and $\mathbf{K}_{ij}$ are defined as follows:

$$\begin{aligned}\begin{bmatrix} \mathbf{A}_{21} \\ \mathbf{A}_{31} \end{bmatrix} &= \mathbf{L}_{dd,ue}^{[j]}, \quad \begin{bmatrix} \mathbf{A}_{28} \\ \mathbf{A}_{38} \end{bmatrix} = \mathbf{L}_{dd,u1}^{[j]}, \\ \begin{bmatrix} \mathbf{A}_{24} \\ \mathbf{A}_{34} \end{bmatrix} &= \mathbf{L}_{dd,uy}^{[j]} \begin{bmatrix} \mathbf{I}_N \\ \mathbf{0} \end{bmatrix} - \mathbf{L}_{dd,ue}^{[j]}, \\ \begin{bmatrix} \mathbf{A}_{25} \\ \mathbf{A}_{35} \end{bmatrix} &= \mathbf{L}_{dd,uy}^{[j]} \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_N \end{bmatrix}, \\ \begin{bmatrix} \mathbf{A}_{27} \\ \mathbf{A}_{37} \end{bmatrix} &= \mathbf{L}_{dd,a}^{[j]} \tilde{\Psi}_{PC}, \quad \begin{bmatrix} \mathbf{B}_2 \\ \mathbf{B}_3 \end{bmatrix} = \mathbf{L}_{dd,a}^{[j]} \tilde{\Psi}_C, \\ \begin{bmatrix} \mathbf{A}_{57} \\ \mathbf{A}_{67} \end{bmatrix} &= \tilde{\Psi}_{PC}, \quad \begin{bmatrix} \mathbf{B}_5 \\ \mathbf{B}_6 \end{bmatrix} = \tilde{\Psi}_C, \\ \mathbf{A}_{77} &= \begin{bmatrix} \mathbf{0} & \mathbf{I}_{n_p-n_c} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad \mathbf{B}_7 = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{I}_{n_c} & \mathbf{0} \end{bmatrix}, \\ \mathbf{K}_i &= -(\mathbf{L}_{dd,a}^{[j]} \tilde{\Psi}_C)^\dagger \begin{bmatrix} \mathbf{A}_{2i} \\ \mathbf{A}_{3i} \end{bmatrix} \quad (\text{for } i = 1, 4, 5, 7, 8)\end{aligned}\quad (29)$$

where n_p and n_c are the sizes of $\boldsymbol{\gamma}_P^{[j]}$ and $\boldsymbol{\gamma}_C^{[j]}$, respectively.

Finally, despite the fact that the matrices, $\mathbf{A}^{[j]}$, $\mathbf{B}^{[j]}$, $\mathbf{K}^{[j]}$ and $\mathbf{M}^{[j]}$, can vary with respect to time due to the training, based on the assumption that the linear hybrid model is converging, the overall dynamics described in Eq. (23) can be assumed to be a slow linear time-varying system before instability happens due to the low rate of weight changes (validated in Section IV-E), and thus the closed-loop dynamics can be approximated as a linear time-invariant (LTI) system. Accordingly, based on LTI stability analysis, it can be determined that the hybrid FBF controller is unstable when the closed-loop state matrix $(\mathbf{A}^{[j]} + \mathbf{B}^{[j]}\mathbf{K}^{[j]})$ has eigenvalues outside the unit disk. Hence, instability is predicted to occur in the next batches when the spectral radius (i.e., largest absolute eigenvalue) exceeds 1.

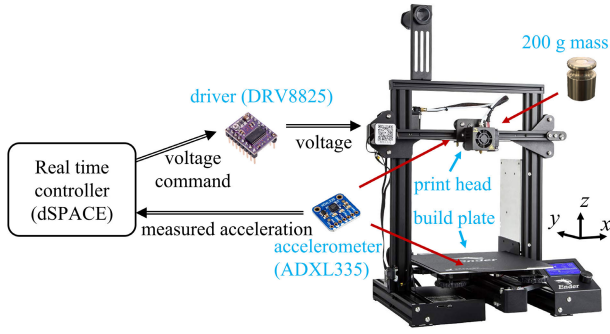


FIGURE 4. The printer and controller setup used for the experimental case studies (the 200 g mass is added for case study 2 and 3 only).

IV. APPLICATION TO 3D PRINTING

In this section, the proposed hybrid FBF controller and stability analysis approach are validated experimentally on a desktop 3D printer.

A. BACKGROUND

Desktop 3D printers are typically made of lightweight material and are driven by stepper motors through belts. As a result, they experience unwanted vibration during printing, leading to a loss of printed part quality. A standard FBF controller, with no data-driven component, has been successfully used to compensate the unwanted vibration, enabling high-speed printing while maintaining high print quality [9], [15]. However, the physics-based model used in the standard FBF controller does not capture the nonlinear vibration behavior of desktop 3D printers, e.g., due to nonlinear friction, backlash, and belt stiffness. Moreover, 3D printers wear over time, and it is not unusual for end-users to perform aftermarket modifications to their desktop printers, e.g., adding a new nozzle or installing a webcam, thus altering the printers' dynamics in unknown ways. The standard FBF approach is unable to handle these unmodeled dynamics. However, next-generation desktop 3D printers are being equipped with low-cost accelerometers that can measure vibration. In the following subsections, we explore the effectiveness of the hybrid FBF controller to leverage data gathered from on-board accelerometers to compensate unmodeled vibration dynamics of a desktop 3D printer.

B. EXPERIMENTAL SETUP

Figure 4 shows the experimental set up. It consists of a Creality Ender 3 Pro desktop 3D printer. A 200 g mass can be attached to the printer's x -axis to simulate an aftermarket modification to the printer (e.g., the installation of a heavier printhead or a webcam). Two low-cost accelerometers (Adafruit ADXL335) are attached to the printer's x - and y -axis to collect acceleration signals, which are converted to position signals using a Luenberger observer with a bandwidth of 5 Hz, where the low frequency components rely more on the identified model and the high frequency components on the measured accelerations. All control

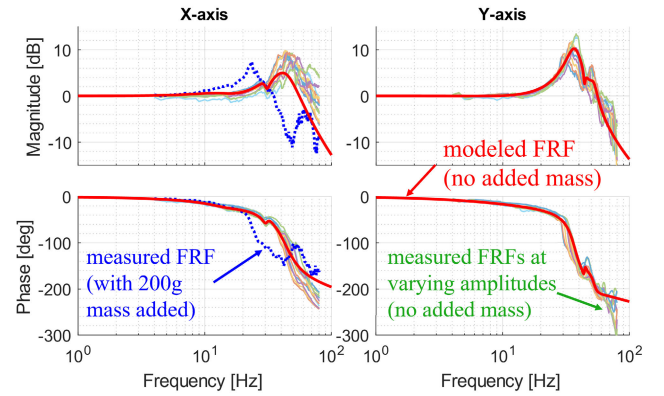


FIGURE 5. Comparison of (1) solid lines: the FRF of the identified model, (2) solid lines with faded colors: the measured FRFs from the unmodified printer (no added mass) using different amplitudes of sine sweeps, and (3) dotted lines: for x -axis only, the measured FRF from the retrofitted printer (with 200 g mass added).

algorithms tested on the printer are implemented on dSPACE DS1107 and dSPACE DS5203 FPGA board at fixed sampling rates of 1 kHz (for control) and 40kHz (for data collection and stepper motor command generation). The stepper motor commands are delivered to the printer's motors using DRV8825 stepper motor drivers.

As shown in Fig. 4, the x -axis of the printer is the extruder while the y -axis is the print bed, so the two axes can be seen as independent dynamics. Figure 5 shows the printer's x - and y -axis frequency response functions (FRFs). The lines with faded colors show the FRFs measured at varying input acceleration amplitudes without the 200 g mass added to the printer. Due to the nonlinear dynamics of the printer, the FRFs vary significantly with the input amplitude. Also shown by the solid lines is a linear transfer function model fit using the measured FRFs for each axis. The identified transfer functions for the x - and y -axes are given by

$$\begin{aligned} G_{pb,x} &= \frac{b_5 s^5 + b_4 s^4 + b_3 s^3 + b_2 s^2 + b_1 s + b_0}{s^6 + a_5 s^5 + a_4 s^4 + a_3 s^3 + a_2 s^2 + a_1 s + a_0} \\ G_{pb,y} &= \frac{d_6 s^6 + d_5 s^5 + d_4 s^4 + d_3 s^3 + d_2 s^2 + d_1 s + d_0}{s^7 + c_6 s^6 + c_5 s^5 + c_4 s^4 + c_3 s^3 + c_2 s^2 + c_1 s + c_0} \end{aligned} \quad (30)$$

where

$$\begin{aligned} b_5 &= -62.48, & a_5 &= 242.6, \\ b_4 &= 5.91 \times 10^4, & a_4 &= 1.36 \times 10^5, \\ b_3 &= 3.82 \times 10^6, & a_3 &= 1.73 \times 10^7, \\ b_2 &= 2.96 \times 10^9, & a_2 &= 4.22 \times 10^9, \\ b_1 &= 1.96 \times 10^{11}, & a_1 &= 2.75 \times 10^{11}, \\ b_0 &= 2.29 \times 10^{13}, & a_0 &= 2.29 \times 10^{13}, \\ d_6 &= -84.79, & c_6 &= 211.2, \\ d_5 &= 2.87 \times 10^4, & c_5 &= 2.56 \times 10^5, \\ d_4 &= -8.03 \times 10^6, & c_4 &= 4.11 \times 10^7, \\ d_3 &= 6.45 \times 10^9, & c_3 &= 2.07 \times 10^{10}, \end{aligned}$$

TABLE 1. The controller parameters of the standard and the hybrid FBF used for the experimental case studies.

Sampling rate	1 kHz (for control) 40 kHz (for data transmission)
Observer bandwidth	5 Hz
Accelerometers low-pass cutoff freq.	200 Hz
B-spline degree	5
B-spline knot vector spacing	10 time steps (= 0.01 s)
N (batch length)	70 time steps (= 0.07 s)
N_w (window length)	140 time steps (= $2N$)
q (number of y_{pb} terms in Φ)	4
p (number of e_{pb} terms in Φ)	50
λ (regularization factor)	0.01

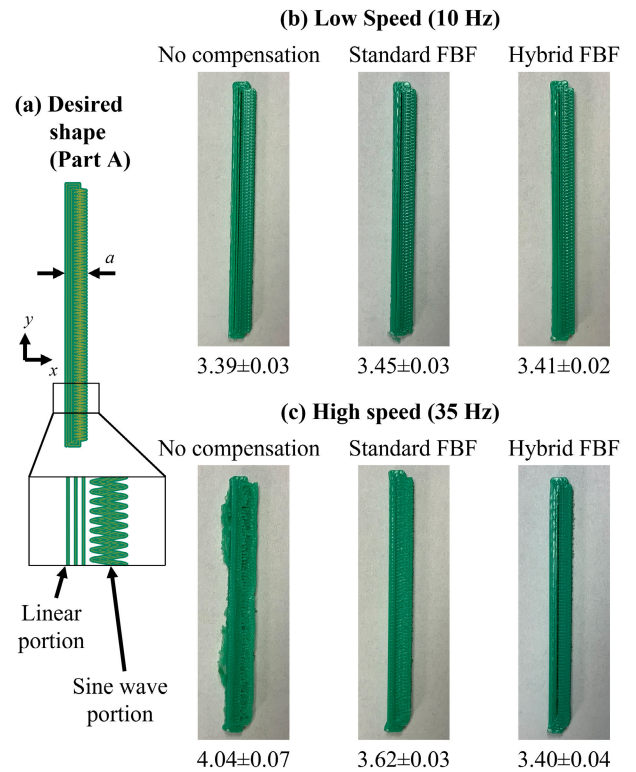
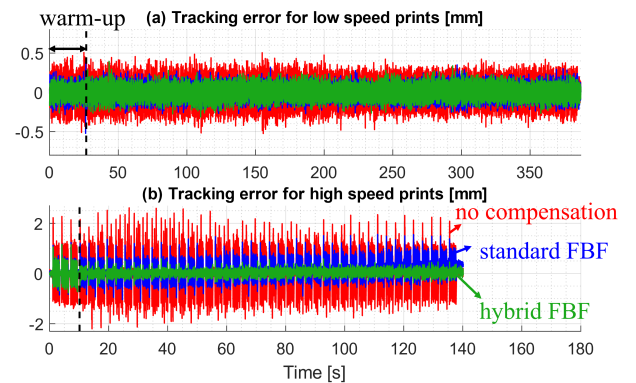
$$\begin{aligned}
 d_2 &= 3.86 \times 10^{11}, & c_2 &= 2.39 \times 10^{12}, \\
 d_1 &= 3.29 \times 10^{14}, & c_1 &= 5.28 \times 10^{14}, \\
 d_0 &= 3.74 \times 10^{16}, & c_0 &= 3.74 \times 10^{16}.
 \end{aligned}$$

Notice that the fit FRF, which is used to generate the physics-based model $\mathbf{G}_{pb}$ used in the standard and hybrid FBF controllers, does not model the nonlinear amplitude-dependent behavior of the FRFs. Instead, it only models the average of all measured FRFs, with larger weighting on the lower-frequency portion, which is more common for most trajectories and is the main signal source of the identified model for the observer. As a result, in addition to the amplitude-dependent behavior, the fit FRF and the measured FRFs may also have larger deviation at higher frequency trajectories. Also shown by the dotted lines on the x -axis plot of Fig. 5 is an FRF measured when the 200 g mass is added to x -axis. Notice that this modification causes the printer's FRF to further deviate from the modeled FRF. Two case studies that evaluate the effectiveness of the proposed hybrid FBF controller in handling these unmodeled dynamics are described below. A third case study evaluates the stability of the hybrid FBF controller. The basis functions used in these three case studies are B-spline basis functions, and the detailed controller parameters are summarized in Table 1.

C. CASE STUDY 1: EVALUATION OF HYBRID FBF CONTROLLER ON UNMODIFIED PRINTER

This case study aims to show the quality improvement of 3D printed parts using the hybrid FBF controller on the printer of Fig. 4 without the 200 g mass added. The unmodeled dynamics are therefore due to the nonlinear amplitude-dependent behavior shown in Fig. 5. Part A, shown in Fig. 6(a), is printed without compensation and with compensation using the standard and hybrid FBF controllers. The part consists of linear and sine-wave motions. The sine wave motions have a wavelength of 0.5 mm. Therefore, when traversing the sine wave, the x -axis vibrates at a frequency proportional to the feed rate of the printer, and the frequency of the sine wave can be obtained by: frequency = (y-axis speed)/wavelength.

Figure 6(b) and (c) show the results of Part A with the sine wave printed at low speed (i.e., 10 Hz) and high speed

**FIGURE 6.** The (a) desired shape for case study 1 (i.e., Part A) and the comparison of the quality of parts printed at (b) low speed and (c) high speed using different controllers. (The numbers show the mean width a and its standard deviation of the parts from 10 measured values, in mm.).**FIGURE 7.** Comparison of the x -axis tracking error using different controllers for (a) low-speed and (b) high-speed printing for case study 1.

(i.e., 35 Hz) without compensation, and with compensation using the standard and hybrid FBF controllers. The width a shown in Fig. 6(a) is used as a figure of merit. It is obtained by measuring 10 different locations of the printed parts using a vernier caliper and computing the mean and the standard deviation as summarized in Fig. 6(b) and (c). At low speed, all three prints have similar quality, as shown by their similar mean values of a . This is because, as seen from Fig. 5, at 10 Hz, the resonance of the machine is not excited, and there is hardly any vibration to compensate. Moreover,

the nonlinear behavior of the machine is not prevalent. However, at 35 Hz, the qualities of the parts are markedly different. The mean width of the parts printed without compensation and with compensation using the standard FBF controller are respectively 19.17% and 4.93% larger than their low-speed counterparts due to uncompensated vibration. The standard FBF controller does a poor job because it is unable to compensate the unmodeled nonlinear vibration around 35 Hz. However, the hybrid FBF controller does a much better job at compensation, leading to only 0.29% error of mean value of a at high speed compared to at low speed.

The performance of using the hybrid FBF controller can also be observed from the time history of the tracking error. Here, the estimation of the observer is treated as the true position of the printer. Figure 7 compares the tracking error of the parts printed without compensation and with compensation using the standard and hybrid FBF controllers. At low speed, as shown in Fig. 7(a), the standard and hybrid FBF controllers have similar levels of tracking error, while that of the uncompensated is larger. At high speed, Fig. 7(b) shows that the uncompensated and the standard FBF control cases both have significantly larger tracking errors than the hybrid FBF controller beyond the warm-up period, where the standard FBF controller are used in place of the hybrid FBF controller to allow G_{dd} to be fully trained with sufficient data. The overall root mean square (RMS) tracking error between the uncompensated, standard FBF, and hybrid FBF are respectively 143.3, 74.7, and 73.2 μm for low speed printing and 702.6, 358.6, and 129.8 μm for high-speed printing. Again, this confirms the superior performance of the hybrid FBF controller in compensating unmodeled nonlinear vibration.

D. CASE STUDY 2: EVALUATION OF HYBRID FBF CONTROLLER ON RETROFITTED PRINTER

In this case study, Part B shown in Fig. 8(a) is printed on the 3D printer in Fig. 4 with the 200 g mass added. As shown in Fig. 5, the added mass leads to significant mismatch between the modeled and actual x -axis dynamics. Part B is printed without compensation, and with compensation using the standard and hybrid FBF controllers at low speed (i.e., 25 mm/s wall speed) and high speed (i.e., 100 mm/s wall speed). The feed rate (other than the wall speed), acceleration, and jerk limits for both speed settings are respectively:

$$v_{\text{lim}} = 100 \text{ mm/s}, a_{\text{lim}} = 10 \text{ m/s}^2, j_{\text{lim}} = 5000 \text{ m/s}^3 \quad (31)$$

As shown in Fig. 8(b), at low speed, the quality of the part is similar for all three cases. However, at high speed, as shown in Fig. 8(c), there is a loss of quality on the “Y” face of the cube due to unmodeled x -axis vibration. The loss of quality is severe without compensation. Its severity is slightly reduced using the standard FBF controller. However, the hybrid FBF controller significantly reduces the loss of quality thanks to its ability to leverage data gathered from the on-board

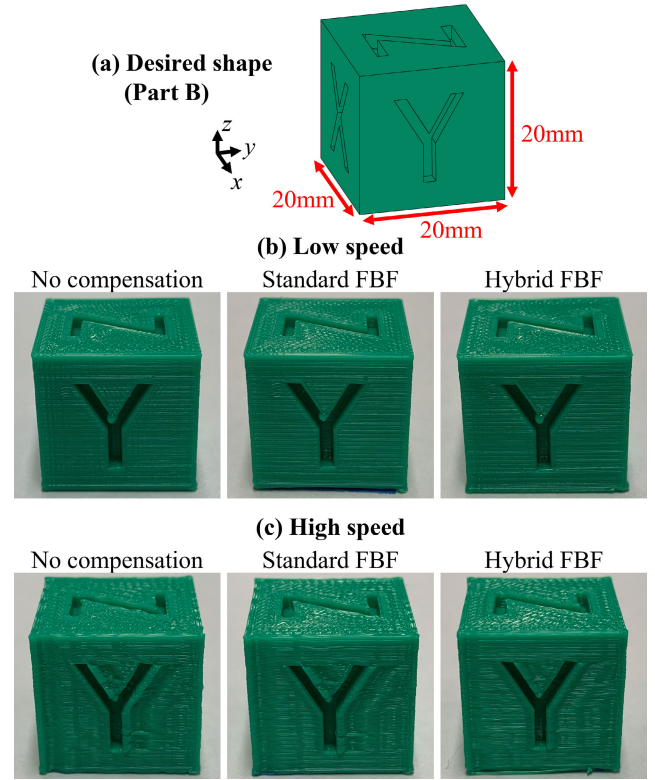


FIGURE 8. The (a) desired shape for case study 2 (i.e., Part B) and the comparison of the print quality using different controllers printed at (b) low speed and (c) high speed.

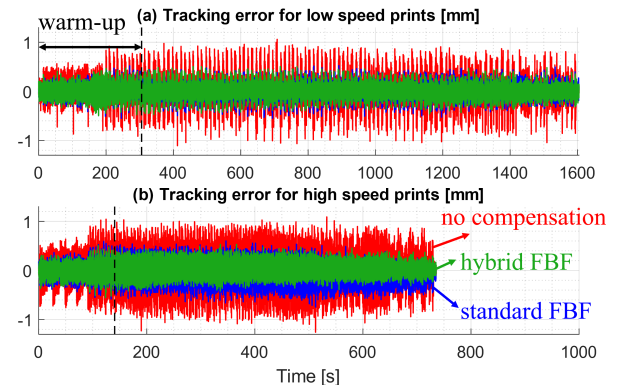


FIGURE 9. Comparison of the x -axis tracking error using different controllers for (a) low-speed and (b) high-speed printing for case study 2.

accelerometers to learn the unmodeled dynamics. Note that the loss of quality in the high-speed case is not perfectly compensated by the hybrid FBF controller. One reason is that the compensation using the hybrid model is based on the position signals observed from acceleration signals using the modeled FRF in Fig. 5, which can be highly inaccurate after retrofitting the printer. This limitation, however, could be alleviated if the position signals are gathered directly from a position sensor. Another reason is that the training of the key feature, i.e., the “Y” letter notch, is diluted by other parts of

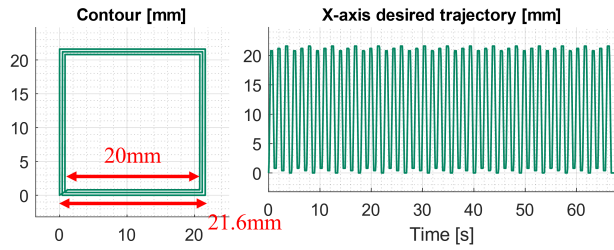


FIGURE 10. The desired shape and the x-axis trajectory (the axis containing instability) for Case Study 3.

the shape that do not have the same geometric pattern. This problem can be addressed by performing more intelligent training updates of G_{dd} using features with similar geometric patterns.

Similar to Case Study 1, the quantitative tracking performance of the hybrid FBF is shown in the time history plots of the tracking error in Fig. 9. As shown in Fig. 9(a), at low speed, the uncompensated case has some spikes in the tracking error, which are compensated by both the standard and the hybrid FBF. These spikes of tracking error occur mostly in the high-speed infill (interior) printing, and thus are not reflected on the surface quality of the printed part shown in Fig. 8(b). However, for high-speed printing, Fig. 9(b) shows that the uncompensated cases have large tracking error at almost every instant. The standard FBF can slightly compensate it, while the hybrid FBF can further improve the compensation of the standard FBF after the warm-up section. The overall RMS tracking error between the uncompensated, standard FBF, and hybrid FBF are respectively 146.6, 77.5, and 80.1 μm for low speed printing and 265.5, 140.3, and 99.2 μm for high-speed printing.

E. CASE STUDY 3: VALIDATION OF THE STABILITY ANALYSIS METHOD

No stability issues were encountered in executing the two case studies reported above. Therefore, in this section, a special case study is created to trigger instability and evaluate the ability of the proposed stability analysis approach to detect the impending instability in advance. Figure 10 shows the motion trajectory used in this case study. It is a hollow square shape that is 3-layer-thick and 15-layer-high. The feed rate, acceleration, and jerk limits used are respectively:

$$v_{\text{lim}} = 60 \text{ mm/s}, a_{\text{lim}} = 3 \text{ m/s}^2, j_{\text{lim}} = 6000 \text{ m/s}^3 \quad (32)$$

Figure 11 shows the control trajectories, measured acceleration, the training results (i.e., the model weights for G_h), and the stability analysis results for this case study. The hybrid FBF controller is warmed up for 5.5 s at the beginning of the motion as it begins to be applied to vibration compensation thereafter. However, at time $t \approx 11.5$ s, the controller becomes unstable, which outputs diverging control trajectory (due to skipping steps, the actual control command stops) and leads to a sudden spike in acceleration and in some

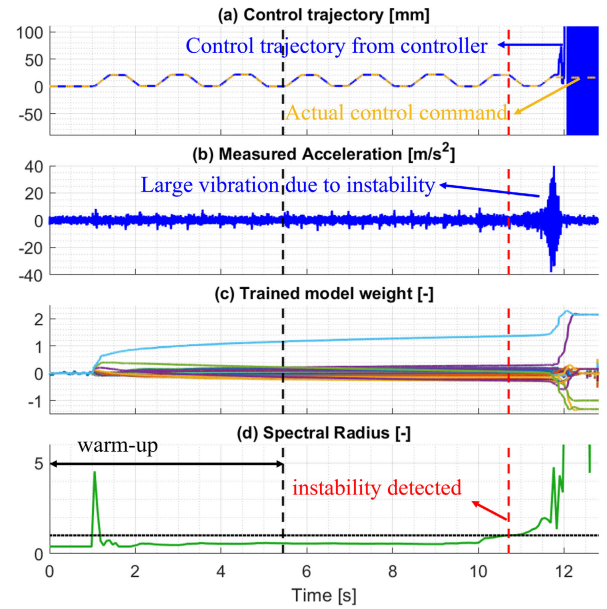


FIGURE 11. The x-axis (a) control trajectory, (b) measured acceleration, (c) training results, and (d) stability analysis results for Case Study 3.

of the trained model weights. Using the stability analysis approach proposed in Section III, which in this case study is performed offline using data gathered online from the x-axis accelerometer, this instability can be detected in advance by the spectral radius approaching unity at $t = 10.7$ s, just before the instability sets in. This gives the controller a 0.8 s lead time to take mitigating measures to avert instability.

From this case study, it is notable that, the calculation of the eigenvalues can be computationally expensive due to the large size of $(\mathbf{A}^{[l]} + \mathbf{B}^{[l]}\mathbf{K}^{[l]})$. Hence, in practice, if the computational resources are limited but no stability issue is tolerable, the model could also be updated task by task. For example, in 3D printing, the same weights could be used throughout the whole printing part. After the current printing task is finished, they are updated offline, checked for stability, and used for the next printing task. Otherwise, with sufficient computational resources, such as utilizing parallel computing, the stability assessment can also be implemented online to detect the impending instability and take mitigating actions, e.g., by switching off the hybrid controller and switching to a safe mode known to be stable (such as the standard FBF controller with a conservative printing speed to ensure the quality).

V. CONCLUSION AND FUTURE WORK

This paper has proposed a physics-guided data-driven controller, called the hybrid filtered basis functions (hybrid FBF) controller, for feedforward tracking control of systems with unmodeled dynamics. In contrast with the standard FBF controller that uses only a fixed physics-based model, the hybrid FBF controller consists of a fixed physics-based

model and a varying data-driven model that is continuously updated during the execution of the controller to account for unmodeled dynamics. The effect of delays due to data acquisition are incorporated into the formulation of the data-driven portion of the controller to enhance its practicality. Moreover, given that the hybrid FBF controller has an inherent feedback loop that could lead to instability and unsafe learning during its operation, a rigorous stability analysis method is proposed to enable the detection and aversion of impending instability.

The performance of the hybrid FBF is demonstrated experimentally using two case studies involving the compensation of the tracking errors of a vibration-prone desktop 3D printer with unmodeled linear and nonlinear dynamics. In both cases, the hybrid FBF controller is shown to significantly outperform the standard FBF controller, particularly in high-speed printing where the effects of unmodeled dynamics are prominent. Moreover, in the third case study, the proposed stability analysis approach is evaluated offline using accelerometer data collected online from the desktop 3D printer. The proposed approach is shown to be able to detect impending instability in advance, allowing time for mitigating measures to be taken to avert instability. This bodes well for the practicality of the hybrid FBF controller, as it facilitates the safe implementation of the controller.

There are deficiencies of the hybrid FBF controller that will be addressed in future work. For example, in the second case study involving a retrofitted 3D printer, the hybrid FBF controller had some unresolved quality issues on the printed objects partly due to inconsistent learning from portions of the printed part with different geometries and vibration behaviors. This deficiency could be solved by creating multiple data-driven models for different geometric features and operating conditions. Another opportunity for future work is to explore learning from one system to another, e.g., across a network of 3D printers. This would likely involve the use of privacy-preserving federated learning [33].

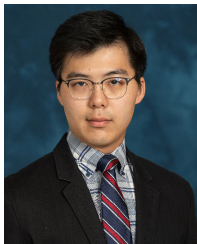
ACKNOWLEDGMENT

A company founded by Chinedum E. Okwudire holds a commercial license for the filtered B spline (FBS) algorithm, which is the version of the standard FBF controller used in this article.

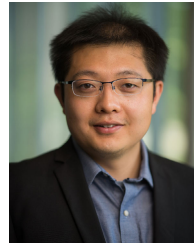
REFERENCES

- [1] G. M. Clayton, S. Tien, K. K. Leang, Q. Zou, and S. Devasia, "A review of feedforward control approaches in nanopositioning for high-speed SPM," *J. Dyn. Syst., Meas., Control*, vol. 131, no. 6, pp. 1–19, Nov. 2009.
- [2] B. P. Rigney, L. Y. Pao, and D. A. Lawrence, "Nonminimum phase dynamic inversion for settle time applications," *IEEE Trans. Control Syst. Technol.*, vol. 17, no. 5, pp. 989–1005, Sep. 2009.
- [3] J. V. Zundert and T. Oomen, "On inversion-based approaches for feedforward and ILC," *IFAC Mechatronics*, vol. 50, pp. 282–291, Apr. 2018.
- [4] M. Tomizuka, "Zero phase error tracking algorithm for digital control," *ASME Trans. J. Dyn. Syst. Meas. Control*, vol. 109, no. 1, pp. 65–68, Mar. 1987.
- [5] J. A. Butterworth, L. Y. Pao, and D. Y. Abramovitch, "Analysis and comparison of three discrete-time feedforward model-inverse control techniques for nonminimum-phase systems," *Mechatronics*, vol. 22, no. 5, pp. 577–587, 2012.
- [6] D. Torfs, J. De Schutter, and J. Swevers, "Extended bandwidth zero phase error tracking control of nonminimal phase systems," *ASME Trans. J. Dyn. Syst. Meas. Control*, vol. 114, pp. 347–351, Sep. 1992.
- [7] J. T. Wen and B. Potsaid, "An experimental study of a high performance motion control system," in *Proc. Amer. Control Conf.*, vol. 6, 2004, pp. 5158–5163.
- [8] K. S. Ramani, M. Duan, C. E. Okwudire, and A. G. Ulsoy, "Tracking control of linear time-invariant nonminimum phase systems using filtered basis functions," *J. Dyn. Syst., Meas., Control*, vol. 139, no. 1, Jan. 2017, Art. no. 011001.
- [9] M. Duan, D. Yoon, and C. E. Okwudire, "A limited-preview filtered B-spline approach to tracking control—With application to vibration-induced error compensation of a 3D printer," *Mechatronics*, vol. 56, pp. 287–296, Dec. 2018.
- [10] K. S. Ramani, M. Duan, C. E. Okwudire, and A. G. Ulsoy, "Optimal selection of basis functions for minimum-effort tracking control of nonminimum phase systems using filtered basis functions," *J. Dyn. Syst., Meas., Control*, vol. 141, no. 11, Nov. 2019, Art. no. 111009.
- [11] R. Romagnoli and E. Garone, "A general framework for approximated model stable inversion," *Automatica*, vol. 101, pp. 182–189, Mar. 2019.
- [12] K. S. Ramani and C. E. Okwudire, "Optimal selection of basis functions for robust tracking control of uncertain linear systems—With application to three-dimensional printing," *J. Dyn. Syst., Meas., Control*, vol. 143, no. 10, pp. 1–10, Oct. 2021.
- [13] K. S. Ramani, N. Edoimioya, and C. E. Okwudire, "A robust filtered basis functions approach for feedforward tracking control—With application to a vibration-prone 3-D printer," *IEEE/ASME Trans. Mechatronics*, vol. 25, no. 5, pp. 2556–2564, Oct. 2020.
- [14] J. A. Frueh and M. Q. Phan, "Linear quadratic optimal learning control (LQL)," *Int. J. Control*, vol. 73, no. 10, pp. 832–839, Jan. 2000.
- [15] N. Edoimioya, K. S. Ramani, and C. E. Okwudire, "Software compensation of undesirable racking motion of H-frame 3D printers using filtered B-splines," *Additive Manuf.*, vol. 47, Nov. 2021, Art. no. 102290.
- [16] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control," *IEEE Control Syst. Mag.*, vol. 26, no. 3, pp. 96–114, Jun. 2006.
- [17] C. Hu, Z. Hu, Y. Zhu, Z. Wang, and S. He, "Model-data driven learning adaptive robust control of precision mechatronic motion systems with comparative experiments," *IEEE Access*, vol. 6, pp. 78286–78296, 2018.
- [18] M. Cao, Y. Bo, and H. Gao, "Combined feedforward control and disturbance rejection control design for a wafer stage: A data-driven approach based on iterative parameter tuning," *IEEE Access*, vol. 8, pp. 181224–181232, 2020.
- [19] T. Sakai, S. Yahagi, and I. Kajiwar, "Two-degree-of-freedom controller design based on a data-driven estimation approach," *IEEE Access*, vol. 10, pp. 120475–120491, 2022.
- [20] F. Boeren, T. Oomen, and M. Steinbuch, "Iterative motion feedforward tuning: A data-driven approach based on instrumental variable identification," *Control Eng. Pract.*, vol. 37, pp. 11–19, Jan. 2015.
- [21] A. Dumanli and B. Sencer, "Data-driven iterative trajectory shaping for precision control of flexible feed drives," *IEEE/ASME Trans. Mechatronics*, vol. 26, no. 5, pp. 2735–2746, Oct. 2021.
- [22] T.-C. Tsao and M. Tomizuka, "Adaptive zero phase error tracking algorithm for digital control," *J. Dyn. Syst., Meas., Control*, vol. 109, no. 4, pp. 349–354, Dec. 1987.
- [23] W. Huang, K. Yang, Y. Zhu, and S. Lu, "Data-driven parameter tuning for rational feedforward controller: Achieving optimal estimation via instrumental variable," *IET Control Theory Appl.*, vol. 15, no. 7, pp. 937–948, Apr. 2021.
- [24] B. Wang, C. Liu, S. Chen, S. Dong, and J. Hu, "Data-driven digital direct position servo control by neural network with implicit optimal control law learned from discrete optimal position tracking data," *IEEE Access*, vol. 7, pp. 126962–126972, 2019.
- [25] N. Hassan and A. Saleem, "Neural network-based adaptive controller for trajectory tracking of wheeled mobile robots," *IEEE Access*, vol. 10, pp. 13582–13597, 2022.
- [26] J. Kon, D. Bruijnen, J. van de Wijdeven, M. Heertjes, and T. Oomen, "Physics-guided neural networks for feedforward control: An orthogonal projection-based approach," 2022, *arXiv:2201.03308*.

- [27] G. Otten, T. J. A. de Vries, J. van Amerongen, A. M. Rankers, and E. W. Gaal, "Linear motor motion control using a learning feedforward controller," *IEEE/ASME Trans. Mechatronics*, vol. 2, no. 3, pp. 179–187, Sep. 1997.
- [28] M. Bolderman, M. Lazar, and H. Butler, "On feedforward control using physics-guided neural networks: Training cost regularization and optimized initialization," 2022, *arXiv:2201.12088*.
- [29] A. K. Ishihara, S. Ben-Menahem, and N. Nguyen, "Protection ellipsoids for stability analysis of feedforward neural-net controllers," in *Proc. Int. Joint Conf. Neural Netw.*, Jun. 2009, pp. 1349–1356.
- [30] Y. Pan and H. Yu, "Biomimetic hybrid feedback feedforward neural-network learning control," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 6, pp. 1481–1487, Jun. 2017.
- [31] C.-H. Chou, M. Duan, and C. E. Okwudire, "A linear hybrid model for enhanced servo error pre-compensation of feed drives with unmodeled nonlinear dynamics," *CIRP Ann.*, vol. 70, no. 1, pp. 301–304, 2021.
- [32] C.-H. Chou, M. Duan, and C. E. Okwudire, "A hybrid filtered basis functions approach for tracking control of linear systems with unmodeled nonlinear dynamics," in *Proc. IEEE 17th Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2021, pp. 1176–1181.
- [33] R. Kontar, N. Shi, X. Yue, S. Chung, E. Byon, M. Chowdhury, J. Jin, W. Kontar, N. Masoud, M. Nouiehed, C. E. Okwudire, G. Raskutti, R. Saigal, K. Singh, and Z.-S. Ye, "The Internet of Federated Things (IoFT)," *IEEE Access*, vol. 9, pp. 156071–156113, 2021.



CHENG-HAO CHOU received the B.S.E. degree in mechanical engineering from the National Taiwan University and the M.S.E. degree in mechanical engineering from the University of Michigan, where he is currently pursuing the Ph.D. degree with the Department of Mechanical Engineering. His current research interest includes data-driven methods for control of manufacturing machines.



MOLONG DUAN (Member, IEEE) received the B.S. degree from Peking University, in 2012, and the M.S. and Ph.D. degrees from the Mechanical Engineering Department, University of Michigan, in 2013 and 2018, respectively. He is currently an Assistant Professor with the Department of Mechanical and Aerospace Engineering, The Hong Kong University of Science and Technology. His research interests include the modeling and control of advanced manufacturing systems, flexible structures, and robots.



CHINEDUM E. OKWUDIRE (Senior Member, IEEE) received the Ph.D. degree in mechanical engineering from The University of British Columbia, in 2009. He joined the Mechanical Engineering Faculty, University of Michigan, in 2011. Prior to joining the University of Michigan, he was the Mechatronic Systems Optimization Team Leader at DMG MORI USA, Davis, CA, USA. His research interests include exploiting knowledge at the intersection of machine design, control, and computing to boost the performance of manufacturing automation systems at low cost. He has received a number of awards, including the CAREER Award from the National Science Foundation, the Young Investigator Award from the International Symposium on Flexible Automation, the Outstanding Young Manufacturing Engineer Award from the Society of Manufacturing Engineers, the Ralph Teetor Educational Award from SAE International, and the Russell Severance Springer Visiting Professorship from UC Berkeley. He has coauthored a number of best paper award winning papers in the areas of control and mechatronics.

...