CIRP Annals - Manufacturing Technology 70 (2021) 301–304

Contents lists available at ScienceDirect

CIRP Annals - Manufacturing Technology

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

A linear hybrid model for enhanced servo error pre-compensation of feed drives with unmodeled nonlinear dynamics

Cheng-Hao Chou[a], Molong Duan[b], Chinedum E. Okwudire (2)[a,*]

[a] Smart and Sustainable Automation Research Laboratory, University of Michigan, Ann Arbor, USA
[b] Department of Aerospace Engineering, University of Michigan, Ann Arbor, USA

ARTICLE INFO

Article history:
Available online 12 May 2021

Keywords:
Feed drive
Machine learning
Hybrid modeling

ABSTRACT

Servo error pre-compensation (SEP) is commonly used to improve the accuracy of feed drives. Existing SEP approaches often involve the use of physics-based linear models (e.g., transfer functions) to predict servo errors, but suffer from inaccuracies due to unmodeled nonlinear dynamics in feed drives. This paper proposes a linear hybrid model for SEP that combines physics-based and data-driven linear models. The proposed model is shown to approximate nonlinearities unmodeled in physics-based linear models. In experiments on a precision feed drive, the proposed hybrid model improves the accuracy of servo error prediction by up to 38% compared to a physics-based model.

1. Introduction

A wide range of manufacturing machines use feed drives powered by computer numerical control (CNC) to generate motion commands [1]. The accuracy of feed drives significantly affects the quality of the parts produced by manufacturing machines. Servo errors are a major source of inaccuracy in feed drives. When caused by motion commands, servo errors can often be reduced by pre-compensation (i.e., feedforward compensation). A model of the machine's servo dynamics is used to modify motion commands offline or online in the CNC interpolator to reduce servo errors.

Examples of servo error pre-compensation (SEP) methods include zero phase error tracking controller [2], inverse compensation filter [3], input shaper [4], model reference feedforward control [5], hierarchical predictive control [6], trajectory pre-filter [7] with friction compensator [8], and filtered B-splines [9,10]. These methods often involve the use of transfer functions of feed drive dynamics to predict and compensate servo errors. Sometimes, the parameters of the transfer functions are tuned online using data collected from the feed drives [7,8].

A major shortcoming of using transfer function models for SEP is that they cannot incorporate nonlinearities. A common practice to alleviate this issue is to pre-emptively cancel out nonlinearities from system dynamics using models of nonlinear dynamics (e.g., friction) [5,7,8,11,12]. However, in several practical cases, the nonlinearities are unknown or do not have reliable models, hence they cannot be canceled out completely. For example, the nonlinear stiffness of a feed drive's cable carrier may be unknown, or the nonlinear friction in a feed drive's bearings may be poorly modeled. Another issue with nonlinear feed drive models, when available, is that they are not amenable to many SEP approaches which depend on linear models, e.g., [2–10].

The idea of hybrid modeling, where physics-based models are combined with data-driven (machine learning) models is gaining a lot of attention [13,14]. One benefit of hybrid modeling is that data, which is becoming abundant, can be used to complement physics-based models. This paper proposes a hybrid model to enable more accurate SEP of feed drives with unmodeled nonlinear (and linear) dynamics. In the proposed approach, the predictions of a linear physics-based model (e.g., derived from transfer function) are fed into a linear data-driven model to achieve a linear hybrid model. It is shown analytically and numerically that, though linear, the proposed hybrid model is able to approximate nonlinearities that are unmodeled by the physics-based model. As a result, it is able to more accurately predict servo errors, which is a pre-requisite for accurate SEP using linear models. Experiments carried out on a vibration-prone precision feed drive with unmodeled nonlinear dynamics demonstrate up to 38% improvement in servo error prediction accuracy using the proposed hybrid modeling approach compared to a standard physics-based model.

The outline of the paper is as follows: In Section 2, the proposed hybrid model is presented. In Section 3, analytical and numerical case studies are used to demonstrate how the proposed linear hybrid model approximates unmodeled nonlinearity in feed drives. A precision feed drive with a vibrating fixture and unmodeled nonlinear dynamics is presented in Section 4 and used to experimentally validate the benefits of the proposed hybrid model in accurately predicting servo errors. This is followed by conclusions and a discussion of future work in Section 5.

2. Proposed hybrid model for SEP

Assume that the stable SISO servo dynamics, $\mathbf{H}$, of a feed drive, is broken down into a linear portion ($\mathbf{H}_L$) which is accurately modeled and a nonlinear portion ($\mathbf{H}_{NL}$) which is unmodeled, as shown in Fig. 1. Let $x_d(k)$ and $x(k)$ represent discrete values of the desired and actual positions of the feed drive, where $k = 0, 1, 2, \ldots$ are the discrete time

* Corresponding author.
E-mail address: okwudire@umich.edu (C.E. Okwudire).

https://doi.org/10.1016/j.cirp.2021.04.070
0007-8506/

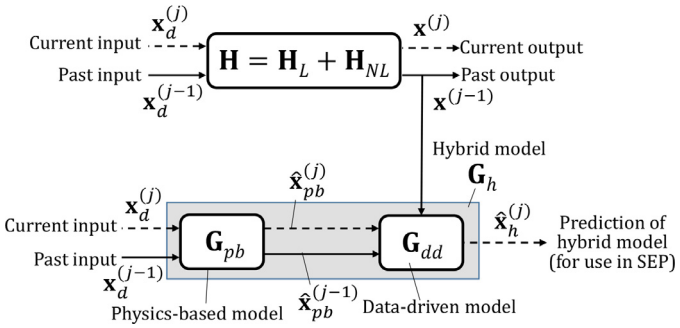

Fig. 1. General framework of the proposed hybrid model.

steps with a sampling interval T_s. Assume that the CNC interpolator of the feed drive has look-ahead capabilities such that the desired trajectory, x_d, is fed to the feed drive in small batches, defined as the vector

$$\mathbf{x}_d^{(j)} \triangleq \{x_d(jN_p), x_d(jN_p+1), \ldots, x_d((j+1)N_p-1)\}^T, \tag{1}$$

where N_p is the length of the look-ahead window, $j = 0, 1, 2, \ldots$ is the batch index. Let $\mathbf{x}^{(j)}$ denote the output position corresponding to $\mathbf{x}_d{}^{(j)}$. In standard SEP, a physics-based linear model, $\mathbf{G}_{pb}$, is typically used to predict $\mathbf{x}^{(j)}$ as

$$\begin{aligned}\hat{\mathbf{x}}_{pb}^{(j)} &\triangleq \{\hat{x}_{pb}(jN_p), \hat{x}_{pb}(jN_p+1), \ldots, \hat{x}_{pb}((j+1)N_p-1)\}^T \\ &= \mathbf{G}_{pb}\,\mathbf{x}_d^{(j)},\end{aligned} \tag{2}$$

where the subscript *pb* indicates that the prediction $\hat{\mathbf{x}}_{pb}{}^{(j)}$ is obtained using $\mathbf{G}_{pb}$, a matrix (lifted) representation of a linear system. However, since $\mathbf{G}_{pb}$ does not capture $\mathbf{H}_{NL}$, using $\mathbf{G}_{pb}$ may result in inaccurate SEP. To address this deficiency, $\mathbf{H}$ is modeled using a linear hybrid model (as shown in Fig. 1), $\mathbf{G}_h$, which combines the physics-based linear model, $\mathbf{G}_{pb}$, and a linear data-driven model, $\mathbf{G}_{dd}$. The goal of $\mathbf{G}_h$ is to return accurate predictions $\hat{\mathbf{x}}_h{}^{(j)}$ of $\mathbf{x}^{(j)}$ based on current and past inputs, $\mathbf{x}_d{}^{(j)}$ and $\mathbf{x}_d{}^{(j-1)}$, as well as past measured output $\mathbf{x}^{(j-1)}$. This is achieved by defining

$$\begin{aligned}\hat{\mathbf{x}}_h^{(j)} &\triangleq \mathbf{G}_{dd}\left(\hat{\mathbf{x}}_{pb}^{(j)},\ \hat{\mathbf{x}}_{pb}^{(j-1)},\ \mathbf{x}^{(j-1)}\right) = \mathbf{G}_{dd}\left(\mathbf{G}_{pb}\,\mathbf{x}_d^{(j)},\ \mathbf{G}_{pb}\,\mathbf{x}_d^{(j-1)},\ \mathbf{x}^{(j-1)}\right) \\ &= \mathbf{G}_h\left(\mathbf{x}_d^{(j)},\ \mathbf{x}_d^{(j-1)},\ \mathbf{x}^{(j-1)}\right),\end{aligned} \tag{3}$$

where $\mathbf{G}_{dd}$ is a data-driven model which learns the residual error of $\mathbf{G}_{pb}$ and then refines its prediction by using as its inputs the outputs of the physics-based model combined with the past output data measured from the feed drive. The accurate predictions $\hat{\mathbf{x}}_h{}^{(j)}$ of the proposed hybrid model can be used online in a windowed (limited-preview) approach [10] for SEP.

In order to keep the overall model linear to facilitate SEP, $\mathbf{G}_{dd}$ is built on a linear regression model. Fig. 2 shows the internal structure of $\mathbf{G}_{dd}$. Let $\mathbf{e}_{pb}{}^{(j)}$ be defined as the prediction error of $\mathbf{G}_{pb}$ (i.e., $\mathbf{e}_{pb}{}^{(j)} \triangleq \mathbf{x}^{(j)} - \hat{\mathbf{x}}_{pb}{}^{(j)}$). Accordingly, linear regression is used to determine $\hat{\mathbf{e}}_{pb}{}^{(j)}$ (the prediction of $\mathbf{e}_{pb}{}^{(j)}$) as a function of $\mathbf{e}_{pb}{}^{(j-1)}$, $\hat{\mathbf{x}}_{pb}{}^{(j)}$, and $\hat{\mathbf{x}}_{pb}{}^{(j-1)}$ by recursively applying the following formula

$$\hat{x}_h(k) = \hat{x}_{pb}(k) + \hat{e}_{pb}(k) = \hat{x}_{pb}(k) + \hat{\mathbf{w}}^{(j)T}\boldsymbol{\phi}(k), \tag{4}$$

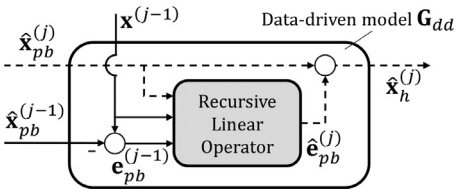

Fig. 2. Internal structure of the data-driven model ($\mathbf{G}_{dd}$).

where $k \in \{jN_p, \ldots, (j+1)N_p - 1\}$, i.e., the time steps defined within batch j; $\hat{\mathbf{w}}^{(j)}$ is the weight of the regression model for batch j, while $\boldsymbol{\phi}(k)$ is the pre-defined feature vector for the regression model. For each time step k, it is defined as

$$\boldsymbol{\phi}(k) = \left[1,\ \hat{x}_{pb}(k), \ldots, \hat{x}_{pb}(k-q), e_{pb}(k-1), \ldots,\ e_{pb}(k-p)\right]^T. \tag{5}$$

Here, $q < N_p$ and $p \le N_p$ are design parameters, where $q+1$ and p represent the number of time steps in $\hat{x}_{pb}$ and e_{pb} in the regressor. Notice from Fig. 2 that only the past values of e_{pb} (i.e., $\mathbf{e}_{pb}{}^{(j-1)}$) are available, meaning that in Eq. (5), $e_{pb}(k)$ is unavailable for $k \ge jN_p$. Therefore, for $k \ge jN_p$, $e_{pb}(k)$ is replaced with $\hat{e}_{pb}(k) \triangleq \hat{x}_h(k) - \hat{x}_{pb}(k)$. In Eq. (4), $\hat{\mathbf{w}}$ is updated recursively for each batch using recursive least squares algorithm to minimize the sum of square errors at all time steps with the following loss function:

$$\hat{\mathbf{w}}^{(j)} = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{k=0}^{jN_p-1} \left[e_{pb}(k) - \mathbf{w}^T\boldsymbol{\phi}(k)\right]^2 + \lambda \parallel \mathbf{w} \parallel_2^2, \tag{6}$$

where $\lambda > 0$ is a constant regularization factor added to prevent overfitting.

3. Analytical justification and numerical validation

3.1. Analytical justification for hybrid model

The hybrid model's capability to approximate unmodeled nonlinear servo dynamics is analytically explained in this section. Consider a feed drive with servo dynamics ***H*** given by

$$\begin{aligned}\mathbf{z}(k+1) &= \mathbf{A}\mathbf{z}(k) + \mathbf{B}x_d(k) + \boldsymbol{\Gamma}x_d^2(k), \\ x(k) &= \mathbf{C}\mathbf{z}(k),\end{aligned} \tag{7}$$

where, x_d and x are the desired and actual positions of the feed drive, respectively, while $\mathbf{z}$ is the vector of internal states; **A, B, C** and $\boldsymbol{\Gamma}$ are system matrices. Notice that, as a simple example but without loss of generality, $\mathbf{H}$ contains an unmodeled, input-dependent nonlinear term, $\boldsymbol{\Gamma}x_d^2(k)$. Assume that the system is stable, and the linear portion of the servo dynamics (i.e., matrices **A, B, C**) is accurately modeled. Then, given zero initial conditions, the system output $x(k)$ is written as

$$\begin{aligned}x(k) = {} & \mathbf{C}\mathbf{A}^{k-1}\mathbf{B}x_d(0) + \mathbf{C}\mathbf{A}^{k-1}\boldsymbol{\Gamma}x_d^2(0) + \cdots + \mathbf{C}\mathbf{B}\mathrm{x_d(k-1)} \\ & + \mathbf{C}\boldsymbol{\Gamma}\mathrm{x_d^2(k-1)}.\end{aligned} \tag{8}$$

Eq. (8) can be concatenated into matrix form as

$$\begin{bmatrix} x(1) \\ x(2) \\ x(3) \\ \vdots \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{CB} & 0 & 0 & \cdots \\ \mathbf{CAB} & \mathbf{CB} & 0 & \cdots \\ \mathbf{CA^2B} & \mathbf{CAB} & \mathbf{CB} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}\begin{bmatrix} x_d(0) \\ x_d(1) \\ x_d(2) \\ \vdots \end{bmatrix}}_{\mathbf{x}_{pb}} + \underbrace{\begin{bmatrix} \mathbf{C}\boldsymbol{\Gamma} & 0 & 0 & \cdots \\ \mathbf{CA}\boldsymbol{\Gamma} & \mathbf{C}\boldsymbol{\Gamma} & 0 & \cdots \\ \mathbf{CA^2}\boldsymbol{\Gamma} & \mathbf{CA}\boldsymbol{\Gamma} & \mathbf{C}\boldsymbol{\Gamma} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}\begin{bmatrix} x_d^2(0) \\ x_d^2(1) \\ x_d^2(2) \\ \vdots \end{bmatrix}}_{\mathbf{e}_{pb}}. \tag{9}$$

Since the linear part of the system is modeled by the physics-based model, the first matrix product represents $\hat{\mathbf{x}}_{pb}$. Therefore, the second matrix product, $\mathbf{e}_{pb}$, is re-written as

$$\begin{bmatrix} e_{pb}(1) \\ e_{pb}(2) \\ e_{pb}(3) \\ \vdots \end{bmatrix} = \begin{bmatrix} \mathbf{C}\boldsymbol{\Gamma} & 0 & 0 & \cdots \\ \mathbf{CA}\boldsymbol{\Gamma} & \mathbf{C}\boldsymbol{\Gamma} & 0 & \cdots \\ \mathbf{CA^2}\boldsymbol{\Gamma} & \mathbf{CA}\boldsymbol{\Gamma} & \mathbf{C}\boldsymbol{\Gamma} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}\begin{bmatrix} x_d^2(0) \\ x_d^2(1) \\ x_d^2(2) \\ \vdots \end{bmatrix}. \tag{10}$$

From Eq. (10), $e_{pb}(k)$ can be seen as a linear combination of the terms $\mathbf{C}\boldsymbol{\Gamma}$, $\mathbf{CA}\boldsymbol{\Gamma}$, …, $\mathbf{CA}^{k-1}\boldsymbol{\Gamma}$, while the nonlinear inputs x_d^2 at each time step are the corresponding weights. Moreover, since the system is stable, we assume that $\mathbf{CA}^n\boldsymbol{\Gamma} \approx 0$ for n greater than or equal some value p. Put differently, $e_{pb}(k)$ is the linear combination of $\mathbf{C}\boldsymbol{\Gamma}$, $\mathbf{CA}\boldsymbol{\Gamma}$, …, $\mathbf{CA}^{p-1}\boldsymbol{\Gamma}$, while its p previous inputs are the weights. Accordingly, at any time step k, if the sequences of the nonlinear inputs, i.e., $\{x_d^2(k-1), x_d^2(k-2), \ldots\ x_d^2(k-p)\}^T$, are linearly related to the past

sequences $\{x_d^2(k-2),\ x_d^2(k-3),\ \ldots\ x_d^2(k-p-1)\}^T$, $\{x_d^2(k-3),\ x_d^2(k-4),\ \ldots\ x_d^2(k-p-2)\}^T$, …, then future values of e_{pb} can be approximated by linear combinations of past values of e_{pb}. Therefore, by including past values of e_{pb} in the linear regressor of Eq. (5), future values of the prediction error caused by similar unmodeled nonlinearity can be approximated.

Remark: Even though the unmodeled dynamics is assumed to be square inputs above, the analysis can also be generalized to other system- or input-dependent, linear or nonlinear terms of time-invariant systems, as long as responses can be approximated by series expansion and then reformatted as in Eq. (10).

3.2. Numerical validation of hybrid model

To further support our analysis in Section 3.1, consider a feed drive whose linear servo dynamics H_L is represented by the spring-mass-damper model shown in Fig. 3, combined with an unmodeled nonlinear dynamic force H_{NL}. The equation of motion of the feed drive is expressed as

$$m\ddot{x} + c\dot{x} + kx + H_{NL} = kx_d + c\dot{x}_d, \tag{11}$$

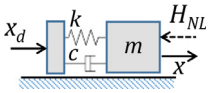

Fig. 3. Model of feed drive servo dynamics used for numerical study.

where $m = 1$ kg, $c = 15.7$ kg/s, and $k = 24674$ N/m. The linear system has a resonance mode at 25 Hz with 5% damping ratio. Assume that H_{NL} (in Newtons) is defined as

$$H_{NL} = 0.1\ \mathrm{sgn}(\dot{x})\ \dot{x}^2 + 1\ \mathrm{sgn}(x)\ x^2. \tag{12}$$

The nonlinear force could arise from uncancelled nonlinear damping from guideways or nonlinear stiffness from a cable carrier attached to the feed drive's table. As a result of H_{NL}, the dynamics of the feed drive becomes nonlinear as

$$m\ddot{x} + (c + 0.1|\dot{x}|)\dot{x} + (k + |x|)x = kx_d + c\dot{x}_d. \tag{13}$$

Simulations are conducted with x_d tracking a staircase position trajectory shown in Fig. 4(a) where the feed drive travels from 0 to 30 mm and then goes back to 0 mm in increments of 10 mm. The velocity, acceleration and jerk limits of the desired position trajectory are respectively set as follows:

$$F_{max} = 80\ \mathrm{mm/s}, A_{max} = 8\ \mathrm{m/s^2},\ \text{and}\ J_{max} = 1{,}000\ \mathrm{m/s^3}. \tag{14}$$

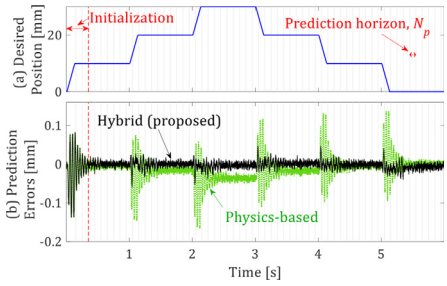

Fig. 4. (a) Desired staircase trajectory; and (b) comparison of servo position prediction errors of $\mathbf{G}_{pb}$ (green line) and $\mathbf{G}_h$ (black line). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The physics-based model, G_{pb}, is obtained from Eq. (11) with $H_{NL} = 0$, i.e., not capturing the nonlinearity. It is used to determine $\hat{x}_{pb}$. To obtain the hybrid model ($\mathbf{G}_h$), the actual position x of Eq. (13) is used, with some Gaussian noise $\varepsilon \sim N(0\ \mathrm{mm},\ 0.005^2\ \mathrm{mm}^2)$ added. The parameter q of $\mathbf{G}_h$ is set as 3 so that it implicitly includes up to the jerk information of the desired trajectory. The parameter p is set to 50, as half of the look ahead window size $N_p = 100$. The sampling interval is $T_s = 1$ ms. For each batch j of $\mathbf{x}_d^{(j)}$, the hybrid model is used to determine $\hat{\mathbf{x}}_h^{(j)}$, the N_p time steps ahead prediction of $\mathbf{x}^{(j)}$.

Fig. 4(b) compares the prediction errors of the physics-based and the hybrid models. Note that during the initialization period marked in the figure, which is arbitrarily set as 5% of the whole trajectory, $\hat{x}_h = \hat{x}_{pb}$, to give the hybrid model some time to get trained before being used for prediction. Otherwise, the performance of $\mathbf{G}_h$ in early batches may be worse than that of $\mathbf{G}_{pb}$. Beyond the initialization period, the proposed hybrid model is more accurate than the physics-based model because of its ability to approximate H_{NL} from data. The root mean squares (RMS) of the servo position prediction error of the hybrid model is 14.7 μm, which is 56.4% smaller than that of the physics-based model (33.7 μm).

4. Experimental validation on precision feed drive

Experimental results are presented in this section to compare the accuracy of the proposed hybrid model to a physics-based model in predicting servo errors of a feed drive.

4.1. Experimental setup and physics-based modeling

The biaxial linear motor-driven stage (Aerotech ALS 25010) shown in Fig. 5 is used for the experiments. The same stage was used in [9] to demonstrate SEP using the filtered basis functions (FBS) method. The stage is controlled using a traditional P/PI feedback controller, augmented with velocity and acceleration feedforward and feedforward friction compensation. The controller is implemented on a dSPACE 1103 real-time control board with 1 kHz sampling frequency (i.e., $T_s = 1$ ms). As shown in Fig. 5, the stage is equipped with a fixture consisting of a block mounted on a rod. The block is assumed to represent an apparatus (e.g., a workpiece or measurement device) whose X and Y positions are expected to track their respective desired trajectories, x_d and y_d accurately, despite inherent structural flexibilities. The acceleration of the fixture is measured using two unidirectional accelerometers (PCB Piezotronics 393B05) shown in Fig. 5.

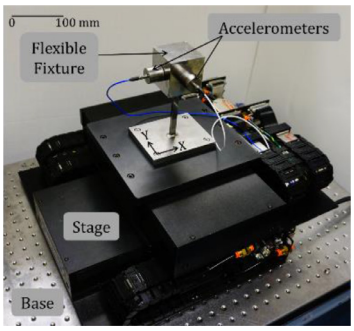

Fig. 5. Experimental setup-biaxial motion stage with a flexible fixture [9].

Fig. 6 shows the frequency response function (FRF) of the servo dynamics of each axis of the stage, generated by applying swept sine acceleration commands to the stage and measuring the accelerations of the fixture using the accelerometers. As discussed in [9], the linear servo dynamics of the stage with fixture for each axis is represented by the following physics-based model of a vibrating system

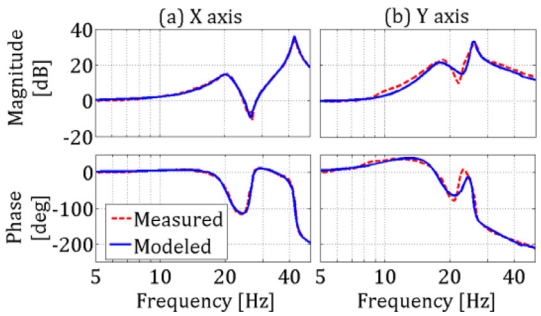

Fig. 6. Measured and modeled frequency response functions of X and Y axes of the experimental setup [9].

$$G_{pb}(s) = \sum_{i=1}^{M} \frac{\alpha_i + \beta_i s}{s^2 + 2\zeta_i \omega_{n,i} s + \omega_{n,i}^2}, \quad (15)$$

where $M = 4$ represents the number of vibration modes and the other system parameters, identified by curve fitting the FRFs, are presented in Table 1 for each axis. More details are available in [9]. In addition to the identified linear dynamics, the stage has unmodeled nonlinear dynamics, like the stiffness of the stage's cable carriers and accelerometer cables, and residual nonlinear friction that was uncancelled by the friction compensation.

Table 1
Model parameters of $G_{pb}(s)$ for X and Y axes of stage [9].

Axis-Mode#	$\omega_{n,i}$ [Hz]	ζ_i	α_i	β_i
X-1	20.52	0.092	15,797.5	54.3
X-2	34.94	0.540	−135,160.6	−587.7
X-3	42.53	0.029	189,225.5	−60.5
X-4	42.60	0.007	14,633.4	−67.9
Y-1	17.86	0.120	6,709.0	310.4
Y-2	25.70	0.021	42,872.2	169.4
Y-3	30.66	0.440	−43,178.2	−1260.2
Y-4	43.10	0.036	−966.3	7.5

4.2. Experimental results

The stage is commanded to track the spiral path shown in Fig. 7 by following jerk-limited position commands x_d and y_d having velocity, acceleration and jerk limits given in Eq. (14). The identified $G_{pb}(s)$ is used to determine the predicted position $\hat{x}_{pb}$ and $\hat{y}_{pb}$ for the X and Y axes, respectively, of the physics-based model. To obtain the data-driven model ($\mathbf{G}_{dd}$) used in the hybrid model ($\mathbf{G}_h$), the actual position of the flexible fixture is estimated from measured acceleration signals using an observer. For $\mathbf{G}_h$, a look-ahead window length of $N_p = 100$, together with $q = 3$ and $p = 50$, are used, as also employed and described in Section 3.

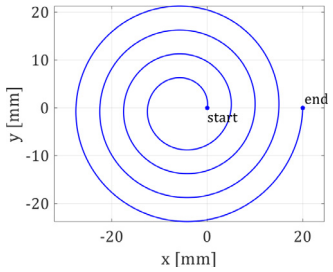

Fig. 7. Desired path: Circular spiral.

Fig. 8 shows the X and Y axes servo position prediction errors using the physics-based model and the proposed hybrid model for the spiral path shown in Fig. 7. As discussed in Section 3, during the initialization period, $\hat{x}_h = \hat{x}_{pb}$ is used to give the hybrid model some time to warm up before being used for prediction. The RMS of the X and Y axes prediction errors of the proposed hybrid model are 13.7 μm and 94.5 μm, which are respectively 38% and 27% more accurate than those of the physics-based model (i.e., 22 μm and 130.2 μm). However, Fig. 8 also shows that the hybrid model does not perform as well at the end of the motion when the stage is decelerating. This is because the hybrid model has not seen nor been trained by similar motion earlier. Also,

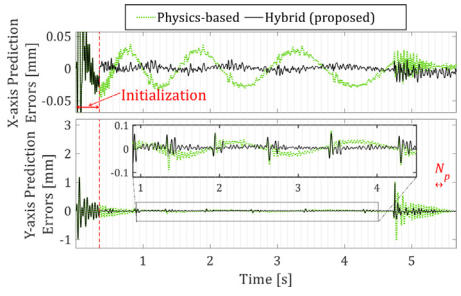

Fig. 8. Prediction errors of $\mathbf{G}_{pb}$ (green line) and $\mathbf{G}_h$ (black line) for the circular spiral (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.).

for the Y axis, the hybrid model does not capture the error spikes very well. This is because the spikes occur sparsely throughout the trajectory, meaning that the weights of the spikes in the training objective are relatively low. Nonetheless, the hybrid model is never overall less accurate than the physics-based model throughout the trajectory.

5. Conclusion and future work

This paper has proposed a linear hybrid model that combines physics-based and data-driven models to capture unmodeled nonlinearity in feed drive servo dynamics. The input to the proposed hybrid model is first filtered through the physics-based model to give a rough prediction, which is then refined by a linear data-driven model using past data acquired from the feed drive. Accordingly, the proposed hybrid model significantly improves the predictions of the physics-based model when the trajectory shows similar nonlinearity, as demonstrated analytically, numerically, and experimentally in this paper.

The accuracy of the hybrid model can be further improved by having weights that vary for different regions of a trajectory, e.g., different weights at locations of spikes in Fig. 8 than the rest of the trajectory. This would require an intelligent approach for segmenting the trajectory into portions with similar features during the training process. Such an approach will be explored in future work. Moreover, future work will also include a determination of the uncertainty of $\mathbf{G}_h$ and optimization of the initialization period as a function of the uncertainty. Lastly, the proposed linear hybrid model will be integrated into online SEP methods in future work.

Declaration of Competing Interest

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

Acknowledgments

The authors would like to thank Dr. Keval Ramani and Ms. Heejin Kim for their assistance with the experiments reported in this paper. This work is funded by the National Science Foundation's awards #1825133 and #1931950.

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 Phase Error Tracking Algorithm for Digital Control. *Journal Dynamic Systems Measurment Control* 109(1):65–68.
[3] Weck M, Ye G (1990) Sharp Corner Tracking Using the IKF Control Strategy. *CIRP Annals* 39(1):437–441.
[4] Altintas Y, Khoshdarregi MR (2012) Contour Error Control of CNC Machine Tools with Vibration Avoidance. *CIRP Annals* 61(1):335–338.
[5] Matsubara A, Nagaoka K, Fujita T (2011) Model-Reference Feedforward Controller Design for High-Accuracy Contouring Control of Machine Tool Axes. *CIRP Annals* 60(1):415–418.
[6] Susanu M, Dumur D (2006) Hierarchical Predictive Control within an Open Architecture Virtual Machine Tool. *CIRP Annals* 55(1):389–392.
[7] D.J. Gordon, K. Erkorkmaz (2013) Accurate Control of Ball Screw Drives Using Pole-Placement Vibration Damping and a Novel Trajectory Prefilter. Precision Engineering 37(2):308–322.
[8] Dumanli A, Sencer B (2019) Pre-Compensation of Servo Tracking Errors Through Data-Based Reference Trajectory Modification. *CIRP Annals* 68(1):397–400.
[9] 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.
[10] 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.
[11] Dong X, Liu X, Yoon D, Okwudire CE (2017) Simple and Robust Feedforward Compensation of Quadrant Glitches using a Compliant Joint. *CIRP Annals* 66(1):353–356.
[12] Jamaludin Z, Van Brussel H, Pipeleers G, Swevers J (2008) Accurate Motion Control of XY High-speed Linear Drives Using Friction Model Feedforward and Cutting Forces Estimation. *CIRP Annals* 57(1):403–406.
[13] Karpatne A, Atluri G, Faghmous JH, Steinbach M, Banerjee A, Ganguly A, Shekhar S, Samatova N, Kumar V (2017) Theory-Guided Data Science: a New Paradigm for Scientific Discovery from Data. *IEEE Transactions Knowledge Data Engineering* 29 (10):2318–2331.
[14] J. Willard, X. Jia, S. Xu, M. Steinbach, V. Kumar (2020) Integrating Physics-Based Modeling With Machine Learning: a Survey. arXiv Preprint arXiv:2003.04919.