Intelligent Feedrate Optimization Using an Uncertainty-Aware Digital Twin within a Model Predictive Control Framework

Heejin Kim [a], Raed A. Kontar [b] and Chinedum E. Okwudire [a,*]

[a]   Department of Mechanical Engineering, University of Michigan, Ann Arbor, 48105 MI, USA
[b]   Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, 48105 MI, USA
[*]   Correspondence: okwudire@umich.edu

Abstract: The future of intelligent manufacturing machines involves autonomous selection of process parameters to maximize productivity while maintaining quality within specified constraints. To effectively optimize process parameters, these machines need to adapt to existing uncertainties in the physical system. This paper proposes a novel framework and methodology for feedrate optimization that is based on a physics-informed data-driven digital twin with quantified uncertainty. The servo dynamics are modeled using a digital twin, which incorporates the known uncertainty in the physics-based models and predicts the distribution of contour error using a data-driven model that learns the unknown uncertainty on-the-fly by sensor measurements. Using the quantified uncertainty, the proposed feedrate optimization maximizes productivity while maintaining quality under desired servo error constraints and stringency (i.e., the tolerance for constraint violation under uncertainty) using a model predictive control framework. Experimental results obtained using a 3-axis desktop CNC machine tool and a desktop 3D printer demonstrate significant cycle time reductions of up to 38% and 17% respectively, while staying close to the error tolerances compared to the existing methods.

Keywords: smart manufacturing; feedrate optimization; digital twin; CNC milling; 3D printing

---

1. Introduction

Quality and productivity are two important and frequently competing factors in manufacturing. As a result, manufacturers strive to maximize productivity while adhering to quality constraints. In practice, achieving this goal has involved a trial-and-error approach. However, there is a growing demand for self-optimizing intelligent manufacturing machines that have the capability to autonomously optimize their speed while ensuring desired quality levels, eliminating the need for extensive trial-and-error [14].

Motion-induced servo error, which is one of the major sources of quality degradation in manufacturing machines, can result from the limited bandwidth of feedback controllers, flexible structures, nonlinear friction, and backlash. Another source of servo error is process force, such as cutting force. Given that motion- and process-force-induced servo errors tend to increase with higher speeds, there is a keen interest in maximizing the speed of motion while respecting the tolerances on motion- and/or process-induced servo errors.

Extensive research has been conducted in the field of feedrate optimization with the objective of maximizing the feedrate while respecting servo error constraints. The majority of feedrate optimization or feedrate scheduling techniques primarily focus on maximizing the feedrate while considering kinematic limits such as speed, acceleration, and jerk [1,25,31,34,35,39]. However, the existing studies in [1,25,31,34,35,39] do not incorporate dynamic constraints such as servo error and cutting force, resulting in the need for a cautious selection of kinematic limits to indirectly meet the requirements on dynamic constraints. This indirect approach is necessary due to the complex relationship between kinematics and dynamic constraints, which often leads to sub-optimal feedrates.

In order to directly enforce dynamic constraints, certain feedrate optimization methods incorporate limits on the servo error, in addition to kinematics, by using steady-state [28] or static [20,40] approximations of servo models associated with motion velocity and acceleration. However, their limited ability to directly incorporate dynamic aspects of servo error, such as dynamic servo error pre-compensation, hinders their accuracy and effectiveness in optimizing feedrate.

To directly incorporate dynamic components via physics-based models, numerous feedrate scheduling methods for CNC machines maximize feedrate in each NC block while keeping cutting force under desired levels via mechanistic force models [9,10,15,16,27,29,38,42]. Some feedrate scheduling techniques maximize feedrate while regulating machining error due to tool deflection [2,4,18,24,30,37,43] or force-induced servo error [21,41] under desired tolerance in CNC machine tools. A few works in feedrate optimization [12,13] constrain motion-induced error via linear physics-based models of servo dynamics. However, the works in [2,4,9,10,12,13,15,16,18,21,24,27,29, 30,37,38,41–43] are unable to effectively constrain actual cutting force or servo error in situations where uncertainties arise from nonlinear dynamics or disturbances that are not incorporated in the physics-based models. As a result, their capability to maximize feedrate while adhering to dynamic constraints is severely restricted.

There is increasing interest in the utilization of digital twins in manufacturing. A digital twin is a virtual representation, parallel to a physical system, built on a bi-directional link between simulation and actual data collection [14]. To effectively optimize feedrate with existence of uncertainties, digital twins can be used to provide more-accurate predictions of process or servo dynamics for feedrate optimization using data-driven models updated via sensor measurements. Model predictive control (MPC) is a framework often used for optimizing feedrate using digital twins. In MPC [23], predictions made using physics-based and data acquired from sensors are used to optimize an objective in a receding horizon fashion. In the context of feedrate optimization, a linear hybrid model was augmented with a periodic internal model in a MPC framework [11] to effectively predict and constrain servo errors due to motion and cutting forces. Luenberger state observers [44] were used in feedrate optimization to correct the initial system states of servo dynamic models in real-time for accurate contour error constraint enforcement, where the objective function was based on a tunable index of how far away an unattainable target position is from the current position. Similarly, a two-stage MPC approach was proposed in [33], where the first stage performed feedrate optimization with contour error constraints using a linear data-driven model, and the second stage further pre-compensated the contour error using artificial neural networks (ANNs). However, the data-driven models in [33] were trained offline through numerous experiments, which is time-consuming and may not be effective in predicting contour error in the presence of in-situ uncertainties that are not included in the training data. Moreover, the two-stage optimizations in [33] and the mixed objective functions used to balance quality and productivity in [33,44] require trial-and-error to tune the objective function weights to determine an acceptable quality level. Therefore, it is better to enforce quality requirements as contour error constraints that must be met, as is often the case in practice. Furthermore, the existing MPC methods in [11,33,44] do not quantify or exploit the uncertainty of the prediction in their feedrate optimization. Hence, they may not effectively adhere to constraints in the presence of high uncertainty due to a lack of training data, system variability and sudden changes in operating conditions.

To quantify uncertainties and impose robustness, studies exist on maximizing feedrate while regulating spindle power, where the spindle power is modeled using Gaussian process regression (GPR) [26]. The spindle power constraint is derived from a stochastic constraint with a fixed confidence level to safely optimize feedrate in uncertain environments. However, GPR in [26] is updated cycle-by-cycle, where the first cycle is initialized with a conservative feedrate profile followed by numerous sequential cycles for convergence. This requires optimizing a highly non-linear GPR objective [32] to estimate hyperparameters, which renders the process less adaptable to real-time control. Moreover, it does not constrain the servo error nor account for the propagation of model uncertainties to the servo error.

This oversight is critical in achieving desired part accuracy in feedrate optimization, especially for toolpaths with high curvatures that can create significant structural vibration.

To address the shortcomings of the existing works, this paper proposes an intelligent feedrate optimization with contour error constraints using an uncertainty-aware digital twin, by making the following original contributions:

1. It uses a novel physics-informed data-driven digital twin to predict the contour error and its uncertainty, where the digital twin incorporates the known uncertainty in the physics-based models and learns the unknown uncertainty by correcting the data-driven model on-the-fly using sensor feedback.
2. It formulates an intelligent feedrate optimization framework capable of maximizing feedrate while accurately constraining contour error under desired tolerance and stringency, based on the uncertainty estimation from the digital twin using a model predictive control framework.
3. It demonstrates the effectiveness of the proposed method by validating its performance in simulations and experiments using a desktop CNC machine tool and 3D printer.

The outline of the paper is as follows: Section 2 introduces our framework for intelligent feedrate optimization using an uncertainty-aware digital twin. Section 3 describes the methodology of the proposed digital twin for predicting the contour error distribution. Section 4 provides a formulation of the feedrate optimization with contour error and stringency constraints (i.e., the tolerance for constraint violation under uncertainty). Section 5 numerically validates the proposed method via a desktop CNC machine tol. Section 6 experimentally validates the proposed method via a desktop CNC machine tool and 3D printer. Finally, section 7 concludes the paper and discusses future work.

2. Framework for Feedrate Optimization with Uncertainty-Aware Digital Twin

The framework for the proposed intelligent feedrate optimization with an uncertainty-aware digital twin is depicted in Figure 1. First, a manufacturer submits a part together with the desired dimensions and contour error tolerance to an intelligent manufacturing machine. Then, the goal of the machine is to autonomously produce the part as quickly as possible while respecting the given error tolerance. The machine is equipped with a digital twin that predicts the contour error, which the machine can exploit for feedrate optimization with contour error constraints. Hence, the proposed framework is based on model predictive control.
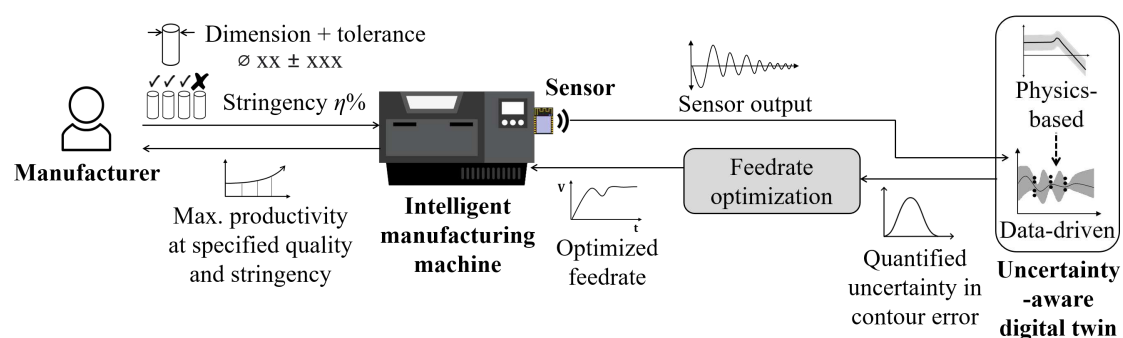
Figure 1. Diagram of intelligent feedrate optimization using an uncertainty-aware digital twin. A manufacturer provides a part tolerance and stringency (i.e., the tolerance for quality constraint violation under uncertainty). The intelligent machine leverages an uncertainty-aware digital twin to optimize feedrate while satisfying the tolerance and stringency requirements.

However, several uncertainties exist in the physical system. Some portions are known from available data or expert knowledge, while others, such as nonlinear dynamics, may be unknown. If not considered, the known and unknown uncertainties cause a violation in the contour error tolerance, hence the part quality, as illustrated in Figure 2(a).

However, given that uncertainty exists in enforcing tolerance constraints, manufacturers have different levels of stringency in enforcing constraints. For example, a manufacturer may want at least 99% of the produced parts to satisfy the constraints. Therefore, stringency reflects a manufacturer's tolerance for quality constraint violations under uncertainty. In this paper, we propose that, instead of relying on trial-and-error, manufacturers can impose a desired stringency $\eta\%$ on the given tolerance by incorporating the uncertainty of the contour error prediction as shown in Figure 2(b). Imposing the stringency represents constraining the worst case out of $\eta\%$ of the entire variation of contour error, so that $\eta\%$ of the manufactured parts adhere to the imposed kinematic and tolerance constraints under the given uncertainty.

To do so, the digital twin uses the known uncertainty from the physics-based models and trains a data-driven model using the machine's sensor measurements to learn the unknown uncertainty. The digital twin then predicts and quantifies the uncertainty of the contour error, which is used in feedrate optimization with desired tolerance and stringency on the contour error. Together with the uncertainty-aware digital twin, the feedrate optimization determines the fastest feedrate to run the machine while respecting the limits for the contour errors (and the kinematic limits of the machine) in a robust way. The measured sensor output is compared with the predicted output and used to adjust the digital twin and optimization algorithm in the next iteration of feedrate optimization.
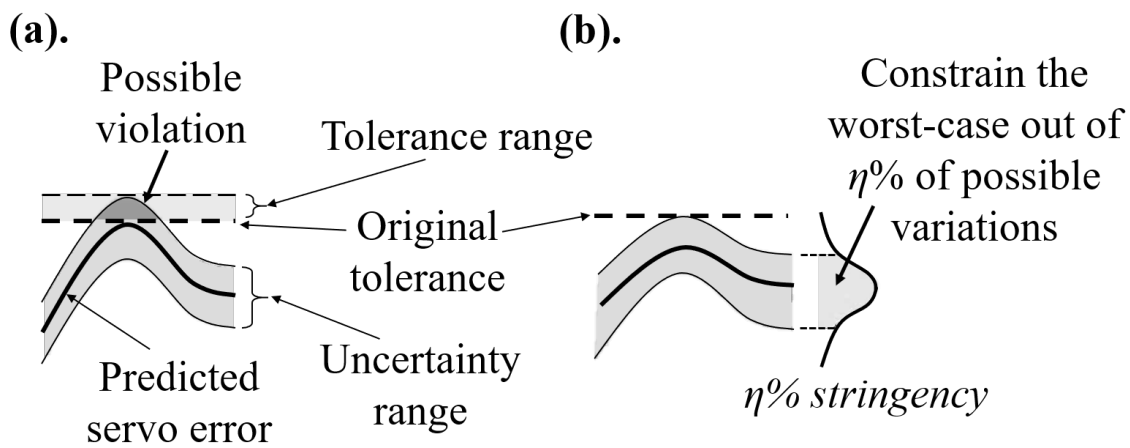
Figure 2. (a) Need for a tolerance range due to violation of error tolerance in the presence of uncertainties, (b) Proposed method of feedrate optimization with desired tolerance stringency using quantified uncertainty.

3. Contour Error Prediction with a Real-Time Uncertainty-Aware Digital Twin

This section first describes the methodology of the contour error prediction using a deterministic digital twin in Section 3.1, and extends it to prediction of contour error and its distribution using an uncertainty-aware digital twin in Section 3.2.

3.1. Overview of Contour Error Prediction Using a Deterministic Digital Twin

A flowchart of the intelligent feedrate optimization using a deterministic digital twin based on the previous work in [11] is depicted in Figure 3. Note that the internal model in [11] is removed. Also note that Figure 3, as well as the discussions that follow in this section, focus on only the x-axis for the sake of brevity. However, the exact same process can be applied to the y-axis and other independent axes of a Cartesian configured manufacturing machine. Small batches (i.e., look-ahead windows) x_d^j of a desired position trajectory x_d are fed into an intelligent feedrate optimizer to produce the optimized motion command, x_c^j, where the superscript $j \in \{0, 1, 2, \cdots\}$ represents the batch index. Here, each batch has a window length of n_w and is defined on the time domain $t \in \{0, T_s, 2T_s, 3T_s, \cdots\}$ which

represents discrete time at sampling interval T_s, as illustrated in Figure 4. The optimized motion commands are sent to the servo system $\boldsymbol{H}_x$ to produce the actual position $\boldsymbol{x}^j$. The servo system is composed of a servo error pre-compensation $\boldsymbol{C}_x$ followed by machine dynamics $\boldsymbol{G}_x$, i.e., $\boldsymbol{H}_x = \boldsymbol{G}_x \boldsymbol{C}_x$.
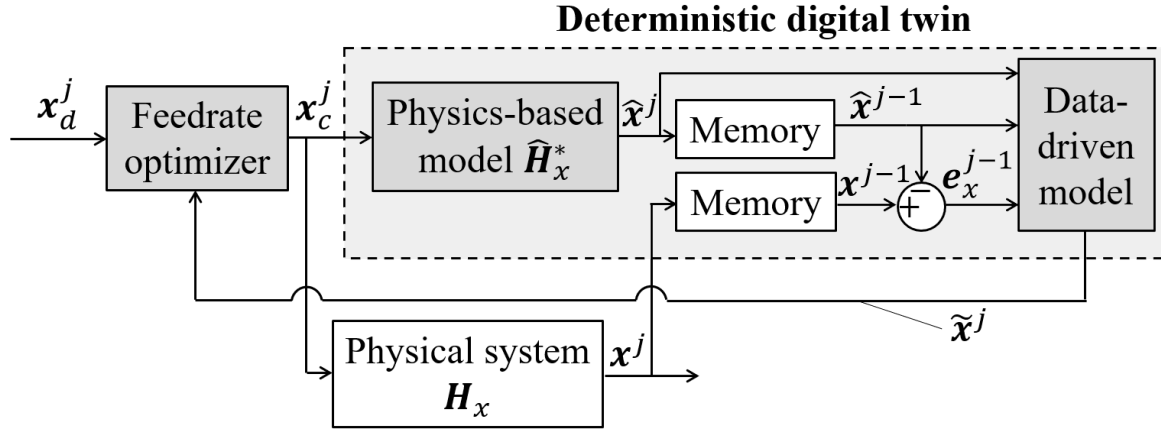
Figure 3. Flowchart of intelligent feedrate optimization using a deterministic digital twin with physics-based and data-driven servo models (y-axis omitted for simplicity).
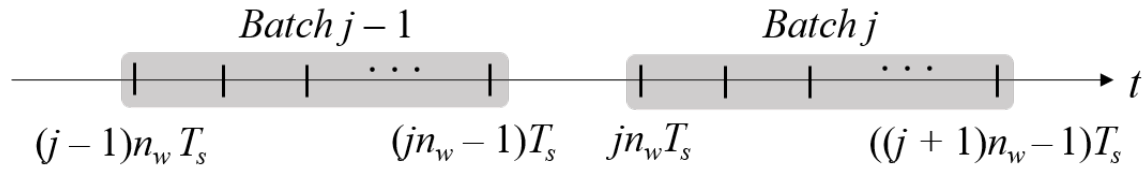
Figure 4. Batches $j-1$ and j defined on a discrete-time domain.

A key requirement for the intelligent feedrate optimization is accurate prediction of the servo error, which is achieved using linear regression built upon the hybrid model presented in [3]. The hybrid model takes input x_c^j and predicts the actual position $\hat{x}^j$ using a stable, nominal (or representative) physics-based model $\hat{\boldsymbol{H}}_x^*$. The predictions $\hat{x}^j$ do not capture the effects of unmodeled dynamics and external disturbances. Therefore, the prediction error of $\hat{\boldsymbol{H}}_x^*$ (delayed by one batch) is computed as $e_x^{j-1} = x^{j-1} - \hat{x}^{j-1}$ and combined with $\hat{x}^j$ and $\hat{x}^{j-1}$ to be fed into a data driven model to generate an improved prediction $\tilde{x}^j$ which is used for constraining contour errors in the feedrate optimization.

Each element $\tilde{x}(t)$ of $\tilde{x}$ is modeled as

$$\tilde{x}(t) = \boldsymbol{\beta}^{\mathrm{T}} \boldsymbol{\psi}_t \tag{1}$$

where $\boldsymbol{\psi}_t$ is the deterministic feature vector and $\boldsymbol{\beta}$ is the weight vector that is learned using linear regression. Note that the superscript j, i.e., batch index, has been removed from $\tilde{x}(t)$ in Eq. (1) to directly define $\tilde{x}$ in time domain. The sub elements of $\boldsymbol{\psi}_t$ are given by

$$\boldsymbol{\psi}_t = [\underbrace{1}_{:=\psi_{t1}}, \quad \underbrace{\hat{x}(t - n_2 T_s), \cdots, \hat{x}(t),}_{:=\psi_{t2}} \\ \underbrace{e_x(t - n_3 T_s), \cdots, e_x(t - T_s)}_{:=\psi_{t3}}]^{\mathrm{T}} \tag{2}$$

The sub-elements $\boldsymbol{\psi}_{t1}$, $\boldsymbol{\psi}_{t2}$ and $\boldsymbol{\psi}_{t3}$ were contained in the hybrid model of [3]. They respectively represent a bias term, the past n_2 and current time steps of $\hat{\boldsymbol{x}}$, and the past n_3 time steps of $\boldsymbol{e}_x$, where n_2 and n_3 are user defined.

The weight $\boldsymbol{\beta}$ is updated recursively in each timestep within a given window, where the final weight is carried over to the next window for prediction, i.e., $\tilde{\boldsymbol{x}}^j$ is predicted based on weight $\boldsymbol{\beta}$ from the previous batch $j-1$. The algorithm for learning $\boldsymbol{\beta}$ is described as follows. For $t=0$, $\boldsymbol{\beta}$ and its covariance matrix $\boldsymbol{P}$ are initialized using ridge regression with a regularization factor λ as

$$
\begin{aligned}
\boldsymbol{\beta} &= (\lambda \mathrm{I} + \boldsymbol{\psi}_t \boldsymbol{\psi}_t^{\mathrm{T}})^{-1} \boldsymbol{\psi}_t x(t) \\
\boldsymbol{P} &= (\lambda \mathrm{I} + \boldsymbol{\psi}_t \boldsymbol{\psi}_t^{\mathrm{T}})^{-1}
\end{aligned}
\tag{3}
$$

For the rest of the timesteps $t \in \{T_s, 2T_s, ...\}$, a recursive least-squares is used to correct $\boldsymbol{\beta}$ and $\boldsymbol{P}$ using a forgetting factor f_0 as

$$
\begin{aligned}
\boldsymbol{\beta} &\leftarrow \boldsymbol{\beta} + \boldsymbol{k}(x(t) - \boldsymbol{\beta}^{\mathrm{T}} \boldsymbol{\psi}_t) \\
\boldsymbol{P} &\leftarrow \frac{1}{f_0}(\boldsymbol{P} - \boldsymbol{k}\boldsymbol{\psi}_t^{\mathrm{T}}\boldsymbol{P}) \\
&\text{where } \boldsymbol{k} = \boldsymbol{P}\boldsymbol{\psi}_t(f_0 + \boldsymbol{\psi}_t^{\mathrm{T}}\boldsymbol{P}\boldsymbol{\psi}_t)^{-1}
\end{aligned}
\tag{4}
$$

Using the final weight in batch $j-1$ to substitute for $\boldsymbol{\beta}$, $\tilde{\boldsymbol{x}}^j$ can be predicted using the feature vector $\boldsymbol{\psi}_t$ formulated by Eq. (2). Since the past sensor data x^{j-1} is provided up to $t = (jn_w - 1)T_s$, for entries in batch j that have unavailable terms in $\boldsymbol{\psi}_{t3}$, $\boldsymbol{e}_x$ for all batches is approximated using the predicted values of $\tilde{\boldsymbol{x}}$, i.e.,

$$
\boldsymbol{e}_x = \boldsymbol{x} - \hat{\boldsymbol{x}} \approx \tilde{\boldsymbol{x}} - \hat{\boldsymbol{x}}
\tag{5}
$$

The same procedure is applied to the y-axis to predict $\tilde{\boldsymbol{y}}$. Lastly, the contour error $\tilde{\varepsilon}$ can be estimated from the predicted axis tracking errors $\tilde{\varepsilon}_x = \boldsymbol{x}_d - \tilde{\boldsymbol{x}}$ and $\tilde{\varepsilon}_y = \boldsymbol{y}_d - \tilde{\boldsymbol{y}}$, using a linear approximation [5] as

$$
\tilde{\varepsilon} = -\sin(\boldsymbol{\theta})\tilde{\varepsilon}_x + \cos(\boldsymbol{\theta})\tilde{\varepsilon}_y
\tag{6}
$$

where $\boldsymbol{\theta}$ is inclination angle of the curve $(\boldsymbol{x}_d, \boldsymbol{y}_d)$.

3.2. Prediction and Uncertainty Quantification of Contour Error Using Uncertainty-Aware Digital Twin

The accuracy of the predictions of the physics-based and data-driven servo models in Section 3.1 can be improved by incorporating the known uncertainty from the physics-based models. To do so, a digital twin that uses physics-informed data-driven servo model is proposed, which is updated on-the-fly via Bayesian approach that is capable of considering uncertainty in the feature vector $\boldsymbol{\psi}_t$. A flowchart of the proposed intelligent feedrate optimization using the uncertainty-aware digital twin is given in Figure 5. The key difference between Figure 3 and Figure 5 is that known uncertainty is included in $\hat{\boldsymbol{H}}_x$, and unknown uncertainty embedded in x^j is learned using the data-driven model and used to predict $\hat{\boldsymbol{x}}^j$ with quantified uncertainty.
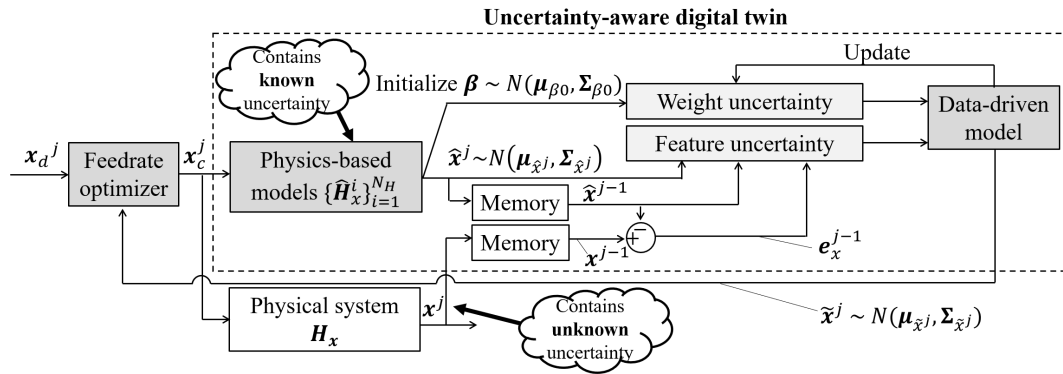
Figure 5. Flowchart of intelligent feedrate optimization using an uncertainty-aware digital twin (with physics-based and data-driven servo models, y-axis omitted).

Each element $\tilde{x}(t)$ of the digital twin's prediction of the output position $\tilde{x}^j$ is modeled as

$$\tilde{x}(t) = \boldsymbol{\beta}^{\mathrm{T}} \boldsymbol{\psi}_t + \epsilon \tag{7}$$

where $\boldsymbol{\psi}_t \sim N(\boldsymbol{\mu}_{\psi_t}, \boldsymbol{\Sigma}_{\psi_t})$ is the feature vector defined as an uncorrelated Gaussian random variable with mean $\boldsymbol{\mu}_{\psi_t}$ and variance $\boldsymbol{\Sigma}_{\psi_t}$ derived from the uncertainty distribution of $\boldsymbol{\psi}_t$ previously defined in Eq. (2). $\boldsymbol{\beta} \sim N(\boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta)$ is the weight vector defined as a correlated Gaussian random variable learned via Bayesian linear regression, and $\epsilon \sim N(0, \sigma_\epsilon^2)$ is the unobserved Gaussian noise. Unlike the deterministic point-estimation of $\tilde{x}^j$ from Section 3.1, a distribution $\tilde{x}^j \sim N(\boldsymbol{\mu}_{\tilde{x}^j}, \boldsymbol{\Sigma}_{\tilde{x}^j})$ is estimated in this section based on the uncertainties of the features and the weights.

This paper proposes that the known uncertainties of the physics-based models are embedded into the feature vector $\boldsymbol{\psi}_t$ in Eq. (7) to enable efficient training of $\boldsymbol{\beta}$. To do so, a set of N_H stable physics-based models $\{\hat{H}_x^i\}_{i=1}^{N_H}$ is obtained, where each model $\hat{H}_x^i$ for $i \in \{1, 2, ..., N_H\}$ is identified in the form of the complex-valued frequency response function (FRF) of the physical system H_x at discrete frequencies ω_k via experiments as

$$\hat{H}_x^i(\omega_k) = a^i(\omega_k) + b^i(\omega_k)j \tag{8}$$

where $\omega_k = k\Delta\omega$, of which $\Delta\omega$ is the increment of frequencies, $k \in \{1, 2, ..., \frac{N_\omega}{2} - 1\}$ where N_ω is the number of discrete negative and positive frequencies at which FRF is identified, and j is the unit imaginary number (which should not be confused with the batch index j used as a superscript elsewhere).

Then, the uncertainty in $\hat{H}_x$ is propagated to the finite impulse response $\hat{h}_x$ as follows. The discrete sets $\{\hat{H}_x^i(\omega_k)\}_{i=1}^{N_H}$ for each k will introduce discrete sets of their real and imaginary coefficients, namely $\{a^i(\omega_k)\}_{i=1}^{N_H}$ and $\{b^i(\omega_k)\}_{i=1}^{N_H}$. For computational efficiency, it is assumed that $\{a^i(\omega_k)\}_{i=1}^{N_H}$ and $\{b^i(\omega_k)\}_{i=1}^{N_H}$ are sampled from Gaussian distributions of $a(\omega_k)$ and $b(\omega_k)$, of which 99.73%, i.e., the 3-sigma range, lie within the minimum and maximum of the identified discrete sets. Then, $a(\omega_k) \sim N(\mu_{a(\omega_k)}, \sigma_{a(\omega_k)}^2)$ can be approximated as

$$\begin{aligned} \mu_{a(\omega_k)} &= \frac{\max\{a(\omega_k)\} + \min\{a(\omega_k)\}}{2} \\ \sigma_{a(\omega_k)} &= \frac{\max\{a(\omega_k)\} - \mu_{a(\omega_k)}}{3} \end{aligned} \tag{9}$$

and $b(\omega_k) \sim N(\mu_{b(\omega_k)}, \sigma_{b(\omega_k)}^2)$ can be approximated using the same procedure as Eq. (9).

Then, the impulse response $\hat{h}_x$ (i.e., $\hat{h}_x[n]$ for $n \in \{1, 2, ..., N_h\}$) of the physics-based model with sampling time T_s can be formulated using discrete inverse Fourier transform as

$$\hat{h}[n] = \frac{1}{N_\omega} \sum_{k=\frac{-N_\omega}{2}}^{\frac{N_\omega}{2}-1} (a(\omega_k) + b(\omega_k)j)e^{\frac{2\pi j(n-1)(k-1)}{N_\omega}} \tag{10}$$

where $N_h = \frac{1}{T_s \Delta \omega}$ is the maximum estimable length of the impulse response. Then, due to the linearity of Eq. (10), $\hat{h}[n]$ follows a Gaussian distribution $\hat{h}[n] \sim N(\mu_{\hat{h}[n]}, \sigma^2_{\hat{h}[n]})$ as

$$\mu_{\hat{h}[n]} = \frac{1}{N_\omega} \sum_{k=1}^{N_\omega} (\mu_{a(\omega_k)} + \mu_{b(\omega_k)}j)e^{\frac{2\pi j(n-1)(k-1)}{N_\omega}}$$

$$\sigma^2_{\hat{h}[n]} = \frac{1}{N_\omega} \sum_{k=1}^{N_\omega} (\sigma^2_{a(\omega_k)} - \sigma^2_{b(\omega_k)})e^{\frac{2\pi j(n-1)(k-1)}{N_\omega}} \tag{11}$$

Next, the uncertainty in $\hat{h}[n]$ can be propagated to $\hat{x}(t) \sim N(\mu_{\hat{x}(t)}, \sigma^2_{\hat{x}(t)})$ as

$$\mu_{\hat{x}(t)} = \sum_{i=1}^{N_h} \mu_{\hat{h}[i]} x_d(t - iT_s)$$

$$\sigma^2_{\hat{x}(t)} = \sum_{i=1}^{N_h} \sigma^2_{\hat{h}[i]} x_d(t - iT_s)^2 \tag{12}$$

Finally, the feature vector follows an uncorrelated multivariate normal distribution $\boldsymbol{\psi}_t \sim N(\boldsymbol{\mu}_{\psi_t}, \boldsymbol{\Sigma}_{\psi_t})$, where

$$\boldsymbol{\mu}_{\psi_t} = [\underbrace{1,}_{:=\boldsymbol{\mu}_{\psi t1}} \underbrace{\mu_{\hat{x}(t-n_2 T_s)}, \cdots, \mu_{\hat{x}(t)},}_{:=\boldsymbol{\mu}_{\psi t2}}$$
$$\underbrace{x(t - n_3 T_s) - \mu_{\hat{x}(t-n_3 T_s)}, \cdots, x(t - T_s) - \mu_{\hat{x}(t-T_s)}}_{:=\boldsymbol{\mu}_{\psi t3}}]^T$$

$$\boldsymbol{\Sigma}_{\psi_t} = \text{diag}(\underbrace{0,}_{:=\boldsymbol{\Sigma}_{\psi t1}} \underbrace{\sigma^2_{\hat{x}(t-n_2 T_s)}, \cdots, \sigma^2_{\hat{x}(t)},}_{:=\boldsymbol{\Sigma}_{\psi t2}}$$
$$\underbrace{\sigma^2_{\hat{x}(t-n_3 T_s)}, \cdots, \sigma^2_{\hat{x}(t-T_s)}}_{:=\boldsymbol{\Sigma}_{\psi t3}})^T \tag{13}$$

The prior over $\boldsymbol{\beta} \sim N(\boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta)$ is initialized using the nominal physics-based model $\mu_{\hat{h}_x}$ from Eq. (11), as described in Appendix A. Now given $\boldsymbol{\beta} \sim N(\boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta)$ and the feature uncertainty $\boldsymbol{\psi}_t \sim N(\boldsymbol{\mu}_{\psi_t}, \boldsymbol{\Sigma}_{\psi_t})$ from Eq. (13), we propose to take a Bayesian route to obtain the posterior over $\boldsymbol{\beta}$ as more data is collected in real time.

Mathematically, the posterior is given as:

$$p(\boldsymbol{\beta}|x(t), \boldsymbol{\mu}_{\psi_t}, \boldsymbol{\Sigma}_{\psi_t}) \propto p(x(t)|\boldsymbol{\beta}, \boldsymbol{\mu}_{\psi_t}, \boldsymbol{\Sigma}_{\psi_t}) \cdot p(\boldsymbol{\beta})$$
$$\propto \int p(x(t), \boldsymbol{\psi}_t | \boldsymbol{\beta}, \boldsymbol{\mu}_{\psi_t}, \boldsymbol{\Sigma}_{\psi_t}) d\boldsymbol{\psi}_t \cdot p(\boldsymbol{\beta})$$
$$\propto \int p(x(t)|\boldsymbol{\psi}_t, \boldsymbol{\beta}) p(\boldsymbol{\psi}_t | \boldsymbol{\mu}_{\psi_t}, \boldsymbol{\Sigma}_{\psi_t}) d\boldsymbol{\psi}_t$$
$$\cdot p(\boldsymbol{\beta})$$
$$\propto p(x(t)|\boldsymbol{\beta}) \cdot p(\boldsymbol{\beta}) \tag{14}$$

where $x(t)$ is the actual position on the x-axis at timestep t defined in Section 3.1. Here, $p(x(t)|\boldsymbol{\beta})$ and $p(\boldsymbol{\beta})$ are given as as

$$
\begin{aligned}
p(x(t)|\boldsymbol{\beta}) = \exp\Big(&-\frac{1}{2}(x(t) - \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\mu}_{\psi_t})^{\mathrm{T}}(\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\Sigma}_{\psi_t}\boldsymbol{\beta} + \sigma_\epsilon^2)^{-1} \\
&(x(t) - \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\mu}_{\psi_t})\Big) \\
p(\boldsymbol{\beta}) = \exp\Big(&-\frac{1}{2}(\boldsymbol{\beta} - \boldsymbol{\mu}_\beta)^{\mathrm{T}}\boldsymbol{\Sigma}_\beta^{-1}(\boldsymbol{\beta} - \boldsymbol{\mu}_\beta)\Big)
\end{aligned}
\tag{15}
$$

As shown above, a central feature of our approach is that it accounts for input uncertainty through $\boldsymbol{\psi}_t \sim N(\boldsymbol{\mu}_{\psi_t}, \boldsymbol{\Sigma}_{\psi_t})$. Unfortunately, the price to pay is the lack of a closed form solution due to the quadratic term in the variance. While monte carlo (MC) sampling approach such as Markov chain MC (i.e. MCMC) or Hamiltonian MC can be used, our goal is to obtain the posterior on the fly to enable real-time control.

To this end, we take a Gaussian Laplacian approximation [8] where we write log of $p(x(t)|\boldsymbol{\beta})$ as a quadratic function of $\boldsymbol{\beta}$. This is shown below:

$$
\begin{aligned}
\ln p(x(t)|\boldsymbol{\beta}) \approx &\underbrace{\ln p(x(t)|\boldsymbol{\beta})\Big|_{\boldsymbol{\beta}=\bar{\boldsymbol{\beta}}}}_{:=c_0} \\
&+ \underbrace{\frac{\mathrm{d}\ln p(x(t)|\boldsymbol{\beta})}{\mathrm{d}\boldsymbol{\beta}}\Big|_{\boldsymbol{\beta}=\bar{\boldsymbol{\beta}}}^{\mathrm{T}}(\boldsymbol{\beta} - \bar{\boldsymbol{\beta}})}_{:=c_1} \\
&+ \frac{1}{2}(\boldsymbol{\beta} - \bar{\boldsymbol{\beta}})^{\mathrm{T}}\underbrace{\frac{\mathrm{d}^2\ln p(x(t)|\boldsymbol{\beta})}{\mathrm{d}\boldsymbol{\beta}^2}\Big|_{\boldsymbol{\beta}=\bar{\boldsymbol{\beta}}}}_{:=C_2}(\boldsymbol{\beta} - \bar{\boldsymbol{\beta}})
\end{aligned}
\tag{16}
$$

where the local point $\bar{\boldsymbol{\beta}}$ is $\boldsymbol{\mu}_\beta$ from the previous batch, i.e., the prior mean. A detailed derivation of the coefficients c_0, c_1 and C_2 in Eq. (16) is provided in Appendix B. Given that $p(x(t)|\boldsymbol{\beta}) \sim N(\boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta)$, now Eq. (14) can re-written in a closed-form solution as

$$
\begin{aligned}
p(\boldsymbol{\beta}|x(t), \boldsymbol{\mu}_{\psi_t}, \boldsymbol{\Sigma}_{\psi_t}) = \exp\Big(&\boldsymbol{\beta}^{\mathrm{T}}(\frac{1}{2}C_2 - \frac{1}{2}\boldsymbol{\Sigma}_\beta^{-1})\boldsymbol{\beta} \\
&+ (c_1^{\mathrm{T}} + \boldsymbol{\mu}_\beta^{\mathrm{T}}\boldsymbol{\Sigma}_\beta^{-1})\boldsymbol{\beta} + \text{constant}\Big)
\end{aligned}
\tag{17}
$$

Thus, the updated variance and mean of the Gaussian posterior $p(\boldsymbol{\beta}|x(t), \boldsymbol{\mu}_{\psi_t}, \boldsymbol{\Sigma}_{\psi_t})$ are

$$
\begin{aligned}
\boldsymbol{\Sigma}_\beta &\leftarrow \mathrm{Var}[\boldsymbol{\beta}|x(t), \boldsymbol{\mu}_{\psi_t}, \boldsymbol{\Sigma}_{\psi_t}] = -\frac{1}{2}\left(\frac{1}{2}C_2 - \frac{1}{2}\boldsymbol{\Sigma}_\beta^{-1}\right)^{-1} \\
\boldsymbol{\mu}_\beta &\leftarrow \mathrm{E}[\boldsymbol{\beta}|x(t), \boldsymbol{\mu}_{\psi_t}, \boldsymbol{\Sigma}_{\psi_t}] = (c_1^{\mathrm{T}} + \boldsymbol{\mu}_\beta^{\mathrm{T}}\boldsymbol{\Sigma}_\beta^{-1})\boldsymbol{\Sigma}_\beta
\end{aligned}
\tag{18}
$$

where the posteriors $\boldsymbol{\mu}_\beta$ and $\boldsymbol{\Sigma}_\beta$ are used as priors for the next batch.

Note that, if the feature $\boldsymbol{\psi}_t$ is assumed to be deterministic (i.e., zero feature uncertainty), no known uncertainties will be included in the Bayesian regression, of which the case will be compared with the proposed method using normally distributed $\boldsymbol{\psi}_t$ in the following Section 5. In the case of deterministic $\boldsymbol{\psi}_t$, the posterior $\boldsymbol{\beta} \sim N(\boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta)$ can be estimated as a closed-form solution using Bayes rule as

$$p(\boldsymbol{\beta}|\boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta)$$
$$= \exp\left(-\frac{1}{2}\boldsymbol{\beta}^{\mathrm{T}}(\sigma_\epsilon^{-2}\boldsymbol{\psi}_t^{\mathrm{T}}\boldsymbol{\psi}_t + \boldsymbol{\Sigma}_\beta^{-1})\boldsymbol{\beta}\right.$$
$$\left. + (\sigma_\epsilon^{-2}\boldsymbol{\psi}_t^{\mathrm{T}}x(t) + \boldsymbol{\Sigma}_\beta^{-1}\boldsymbol{\mu}_\beta)^{\mathrm{T}}\boldsymbol{\beta}\right) \tag{19}$$
$$\therefore \quad \boldsymbol{\Sigma}_\beta \leftarrow (\sigma_\epsilon^{-2}\boldsymbol{\psi}_t^{\mathrm{T}}\boldsymbol{\psi}_t + \boldsymbol{\Sigma}_\beta^{-1})^{-1}$$
$$\boldsymbol{\mu}_\beta \leftarrow \boldsymbol{\Sigma}_\beta(\sigma_\epsilon^{-2}\boldsymbol{\psi}_t^{\mathrm{T}}x(t) + \boldsymbol{\Sigma}_\beta^{-1}\boldsymbol{\mu}_\beta)$$

Lastly, using the trained weight distribution from Eq. (18), the posterior predictive distribution $\tilde{x}(t) \sim N(\mu_{\tilde{x}(t)}, \sigma_{\tilde{x}(t)}^2)$ can be written as

$$\mu_{\tilde{x}(t)} = \boldsymbol{\mu}_\beta^{\mathrm{T}}\boldsymbol{\psi}_t$$
$$\sigma_{\tilde{x}(t)}^2 = \boldsymbol{\psi}_t^{\mathrm{T}}\boldsymbol{\Sigma}_\beta\boldsymbol{\psi}_t + \sigma_\epsilon^2 \tag{20}$$

The same procedures are applied to learn $\boldsymbol{\beta}_y \sim N(\boldsymbol{\mu}_{\beta_y}, \boldsymbol{\Sigma}_{\beta_y})$ and predict $\tilde{y}(t) \sim N(\mu_{\tilde{y}(t)}, \sigma_{\tilde{y}(t)}^2)$ based on y-axis feature vector $\boldsymbol{\psi}_{ty}$ and unobserved noise ϵ_y. Using Eq. (6), the contour error distribution $\tilde{\varepsilon}(t) \sim N(\mu_{\tilde{\varepsilon}(t)}, \sigma_{\tilde{\varepsilon}(t)}^2)$ can be predicted as

$$\mu_{\tilde{\varepsilon}(t)} = -\sin(\theta(t))(x_d(t) - \boldsymbol{\mu}_\beta^{\mathrm{T}}\boldsymbol{\psi}_t)$$
$$+ \cos(\theta(t))(y_d(t) - \boldsymbol{\mu}_{\beta_y}^{\mathrm{T}}\boldsymbol{\psi}_{ty})$$
$$\sigma_{\tilde{\varepsilon}(t)}^2 = \sin(\theta(t))^2(\boldsymbol{\psi}_t^{\mathrm{T}}\boldsymbol{\Sigma}_\beta\boldsymbol{\psi}_t + \sigma_\epsilon^2) \tag{21}$$
$$+ \cos(\theta(t))^2(\boldsymbol{\psi}_{ty}^{\mathrm{T}}\boldsymbol{\Sigma}_{\beta_y}\boldsymbol{\psi}_{ty} + \sigma_{\epsilon_y}^2)$$

4. Methodology for Intelligent Feedrate Optimization with Contour Error Constraints

The feedrate optimization with contour error constraints using the quantified uncertainty from the digital twin is formulated in accordance with authors' previous work [12] using a model predictive control framework. Taking the x-axis, for example, a desired trajectory $\boldsymbol{X}_d = f(\boldsymbol{p})$ is parametrized with respect to a normalized, monotonically increasing path variable $\boldsymbol{p}$, which is a vectorized form of p. Then, $X_d(t)$ is linearized as $x_d(t)$ with respect to $p(t)$ using an estimated linearization point $\bar{p}(t)$ as

$$x_d(t) = -\left.\frac{\partial f}{\partial p}\right|_{p=\bar{p}(t)} \cdot (p(t) - \bar{p}(t)) + f(\bar{p}(t)) \tag{22}$$

The procedure for computing the optimal $\boldsymbol{p}^j$ (corresponding to the optimal feedrate) using the uncertainty-aware digital twin is as follows. The path variable $\boldsymbol{p}^j$ is maximized under monotonicity, maximum feedrate, and axis-acceleration constraints as

$$\max \mathbf{1}^{\mathrm{T}}\boldsymbol{p}^j$$
$$\text{s.t.} \quad p(t-1) \le p(t) \le 1$$
$$D[\boldsymbol{p}^j] \le \boldsymbol{V}_{max}T_s \tag{23}$$
$$\left|D^2[\boldsymbol{x}_d^j]\right| \le \boldsymbol{A}_{max}T_s^2$$

where $\mathbf{1}$ is a ones-vector, D is a difference operator, and $\boldsymbol{V}_{max}$ and $\boldsymbol{A}_{max}$ are the vectorized representations of feedrate and acceleration limits, respectively. In addition, kinematic and dynamic

continuity between adjacent windows is enforced. The process described above for the x-axis can be applied to the y-axis.

The feedrate optimization constrains the contour error under a given tolerance and stringency, using the posterior predictive distribution from Section 3.2. To do so, we show that $\mu_{\tilde{\varepsilon}(t)}$ and $\sigma_{\tilde{\varepsilon}(t)}$ are linear in terms of x_d^j, by showing that the only alterable feature in ψ_t, which is the last term in ψ_{t2} (i.e., $\hat{x}(t)$), is linear in x_d^j.

Let $\Phi_x \in \mathbb{R}^{n_h \times n_h}$ be the matrix (lifted domain) representation of $\mu_{\hat{h}_x}$ truncated by length n_h. The last n_w rows in Φ_x can further be decomposed into two parts: its first $n_h - n_w$ columns $\Phi_{x,p}$ and its last n_w columns $\Phi_{x,c}$ as

$$\Phi_x = \begin{bmatrix} \vdots & \vdots \\ \Phi_{x,p} & \Phi_{x,c} \end{bmatrix} \tag{24}$$

If $x_{c,p}$ represents the last $n_h - n_w$ elements of the x_c at past timesteps, $\hat{x}(t)$ can be re-written as

$$\hat{x}^j = \Phi_{x,c} x_d^j + \Phi_{x,p} x_{c,p}$$
$$\therefore \hat{x}(t) = \underbrace{M_t \Phi_{x,c}}_{:=T_x} x_d^j + \underbrace{M_t \Phi_{x,p} x_{cp}}_{:=T_{0x}} \tag{25}$$

where M_t is a selection matrix that picks the entry at timestep t. Similarly, for y-axis, the alterable term in ψ_{ty} can be derived to be linear in terms of y_d^j, by using a similar notation as Eq. (25), i.e., $\hat{y}(t) = T_y y_d^j + T_{0y}$.

Then, the worst-case out of the η [%] variations of distribution of the contour error $\tilde{\varepsilon}(t)$, where η is a user-defined stringency, is bounded by the tolerance E_{max} as a stochastic constraint by

$$p(\tilde{\varepsilon}(t) \leq E_{max}) \geq \eta \tag{26}$$

For the sake of brevity, the negative stochastic contour error constraint $p(\tilde{\varepsilon}(t) \geq -E_{max}) \geq \eta$ is omitted. Then, inversion the both sides of Eq. (26) becomes

$$\mu_{\tilde{\varepsilon}(t)} \leq E_{max} - \Phi^{-1}(\eta)\sigma_{\tilde{\varepsilon}(t)} \tag{27}$$

where Φ is the cumulative density function of the distribution of $\tilde{\varepsilon}(t)$, which is invertible because $\tilde{\varepsilon}(t)$ follows a Gaussian distribution as was shown in Eq. (21).

Eq. (27) can be rearranged as a linear constraint in terms of x_d^j and y_d^j using Eq. (21), written as

$$U_x x_d^j + U_y y_d^j \leq U_0 \tag{28}$$

where the derivation of coefficients U_x, U_y and U_0 is described in Appendix C. Finally, the contour error constraint is also linear with respect to the decision variable p^j using the relationship in Eq. (22).

The methodology of feedrate optimization described in this section can be broadly considered as model predictive control [22], because it: (1) optimizes manipulatable inputs, e.g., desired trajectory, over a finite, receding horizon using (2) predictions of the dynamical system's behavior through a model that is updated via feedback.

5. Numerical Validation of the Intelligent Feedrate Optimization Using Uncertainty-Aware Digital Twin

This section validates the importance of the uncertainty quantification of the proposed physics-informed data-driven (PIDD) uncertainty-aware digital twin in feedrate optimization, by comparing the method with the following approaches:

1. Conservative method, which is defined as the benchmark generated using a trapezoidal acceleration profile [7] with kinematic limits tuned by trial-and-error to achieve the contour error tolerance with $\eta\%$ stringency, by allowing up to $(100 - \eta)\%$ RMS violation normalized by E_{max} defined in Section 4

2. Physics-based (PB) method, which predicts the output position and its uncertainty using only the known uncertainty obtained from the set of physics-based models $\{\hat{H}_x^i\}_{i=1}^{N_H}$ and $\{\hat{H}_y^i\}_{i=1}^{N_H}$

3. Data-driven (DD) method, which predicts the output position and its uncertainty by learning the unknown uncertainty without incorporating any known uncertainties, i.e., the prior μ_{β_0}, Σ_{β_0}, $\mu_{\beta_y 0}$ and $\Sigma_{\beta_y 0}$ are initialized as zero at the 0-th batch, and β and β_y are learned via Bayesian linear regression for deterministic features in Eq. (19). Note that both the PB and DD methods are subsets of the proposed uncertainty-aware digital twin. However, we have separated them out to highlight the effect of introducing uncertainty in both the PB and the DD models through the PIDD method used in the uncertainty aware digital twin

A Nomad 3 three-axis desktop CNC machine tool is chosen as the simulated system, where its setup is shown in Figure 6. To analyze its known uncertainties, the position commands are generated and commanded by dSPACE DS1007 real-time control board running at 500 Hz sampling rate, connected to DRV8825 stepper motor drivers for the x-, y- and z-axes stepper motors. ADXL335 accelerometers are attached on the x- and y-axis gantries to measure the x and y-axes acceleration. The known uncertainties are identified by measuring FRFs, of which the input is a swept sine acceleration command to the stepper motors, and the output is the relative acceleration between the x- and y-axis using the accelerometers. The operating condition under which the FRFs are measured is varied by modifying the input acceleration amplitude at discrete values: 2 m/s^2, 3 m/s^2 and 4 m/s^2, and 3 FRFs are measured per each acceleration amplitude to collect a total of $N_H = 9$ FRFs per axes.
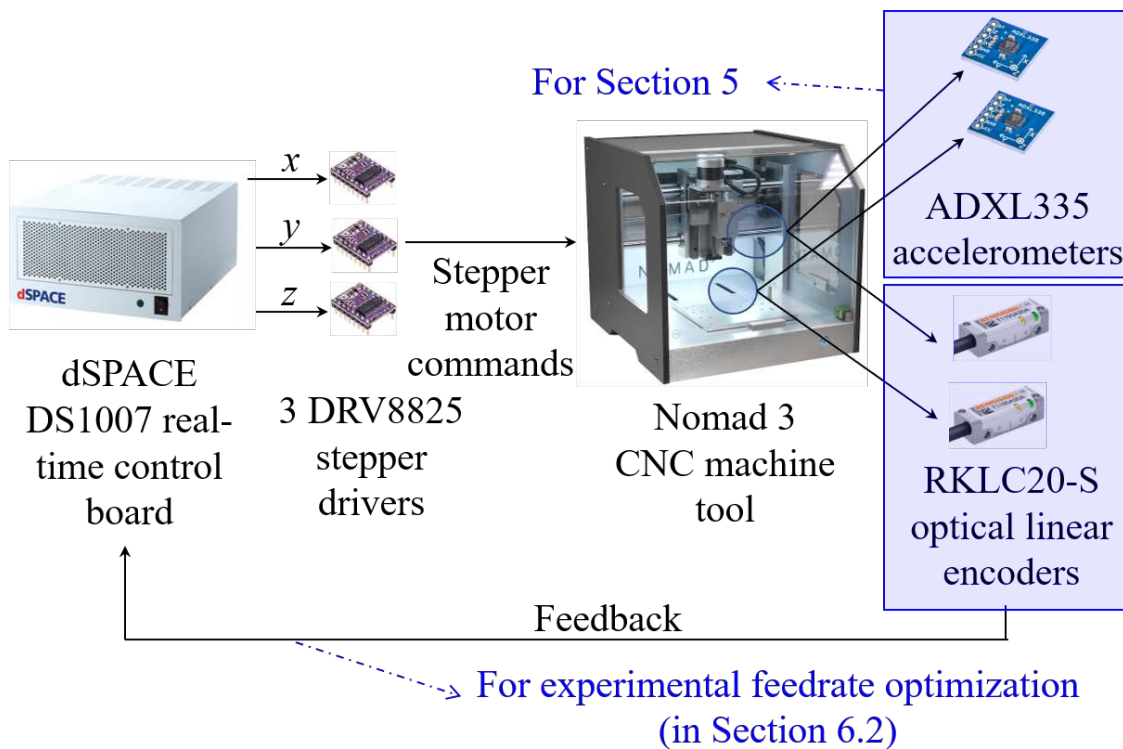
Figure 6. Experimental setup for Sections 5 and 6.2.

The set of FRFs $\{\hat{H}_x^i\}_{i=1}^9$ of the x- and y-axis of the printer are shown in Figures 7 and 8, respectively. The uncertainties in $\hat{H}_x$ are then propagated to $\hat{h}_x \sim N(\mu_{\hat{h}_x}, \Sigma_{\hat{h}_x})$ to initialize μ_β and Σ_β and construct μ_{ψ_t} and Σ_{ψ_t} in the physics-informed data-driven digital twin. To validate the hypothesis

that the FRF coefficients of $\{\hat{H}_x\}_{i=1}^9$ and $\{\hat{H}_y\}_{i=1}^9$ belong to Gaussian distributions, Lilliefors test [17] is performed on every frequency ω_k for $k \in \{1, 2, ..., N_H\}$ of $a(\omega_k)$ and $b(\omega_k)$ in x- and y-axis to compute the test decisions at the 5% significance level and p-values. Figures 9 and 10 show the FRF coefficients $a(\omega_k)$ and $b(\omega_k)$ for x- and y-axis, respectively, in the upper plots. The lower plots of Figures 9 and 10 show p-values and Lilliefors test results, where 0 and 1 represent acceptance and rejection of the hypothesis, respectively. The figures imply that Gaussian hypothesis for both $a(\omega_k)$ and $b(\omega_k)$ is accepted at 90% and 88% of the frequencies for the x- and y-axis, respectively. One way to improve the reliability of Lillifors test result is to gather more data of FRFs. However, for the sake of simplicity and computational efficiency, the Gaussian hypothesis will be assumed valid for all frequencies in the following sections.
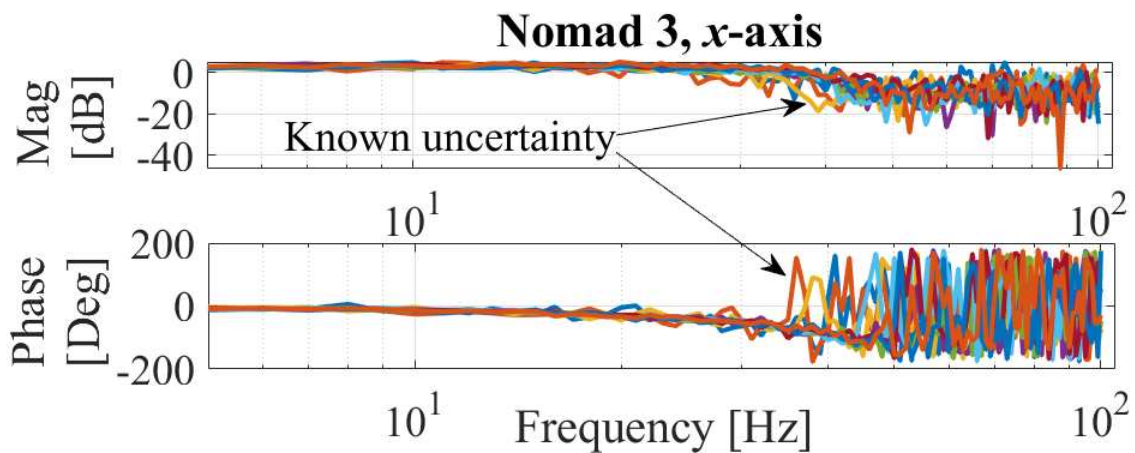
Figure 7. Frequency response functions of the Nomad 3's x-axis showing the known uncertainty obtained under different input acceleration amplitudes.
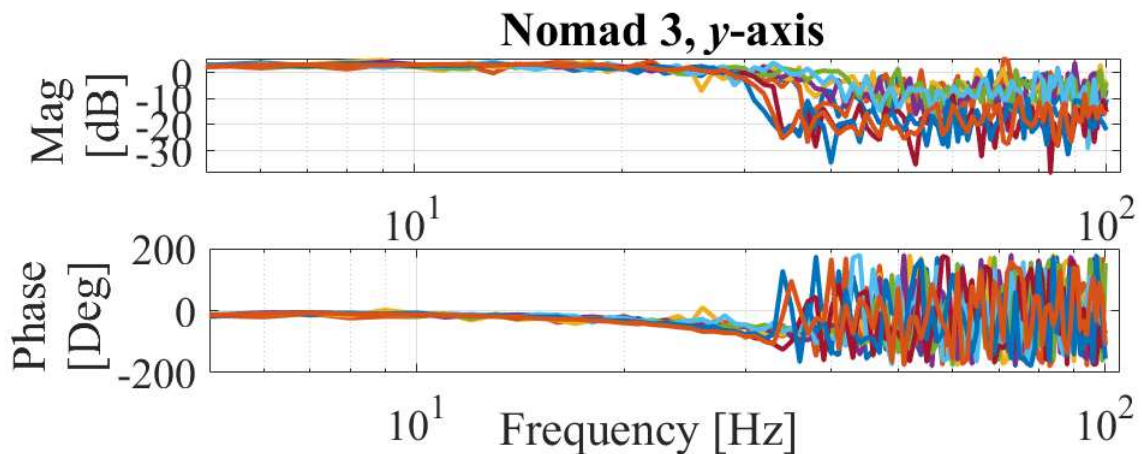
Figure 8. Frequency response functions of the Nomad 3's y-axis showing the known uncertainty obtained under different input acceleration amplitudes.
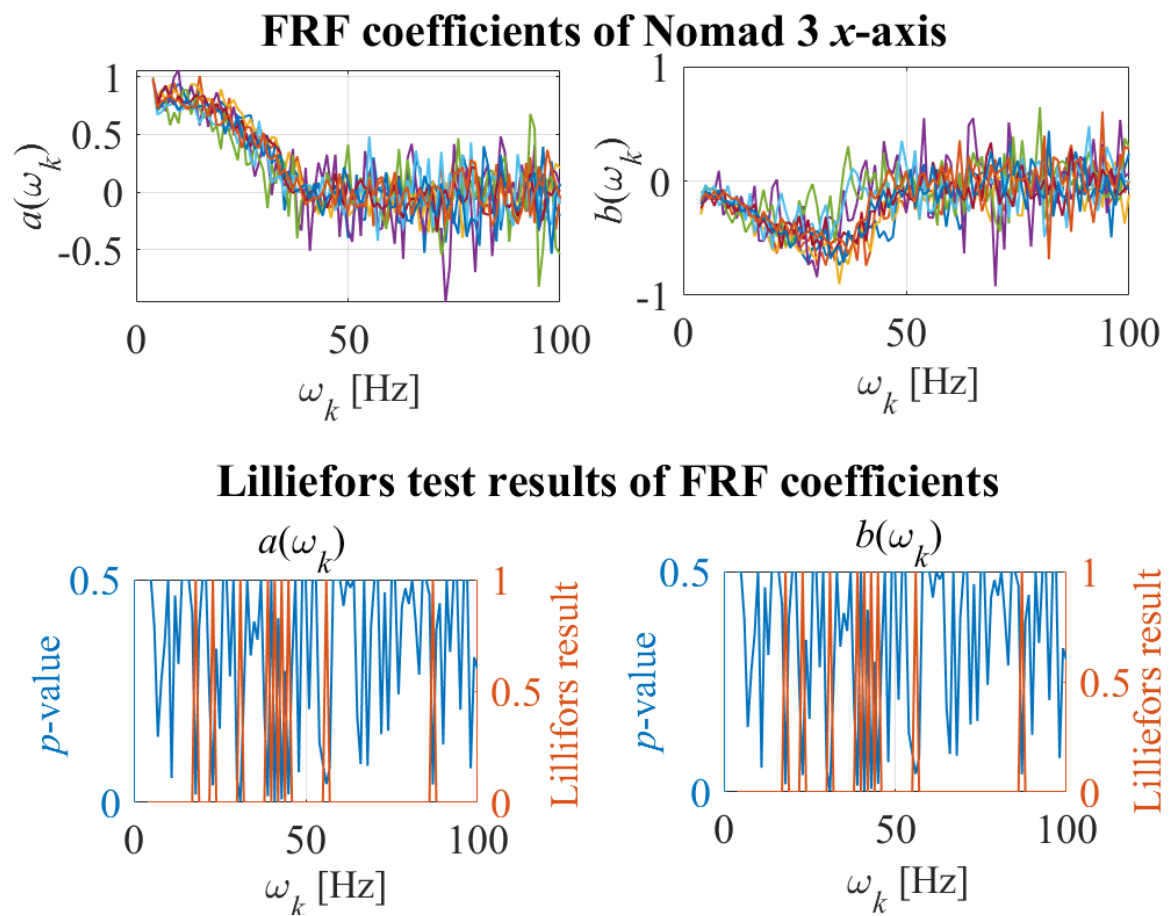
Figure 9. Upper plot: FRF coefficients $a(\omega_k)$ and $b(\omega_k)$ of Nomad 3 x-axis; lower plot: p-value and accept/reject results of Lilliefors test of FRF coefficients.
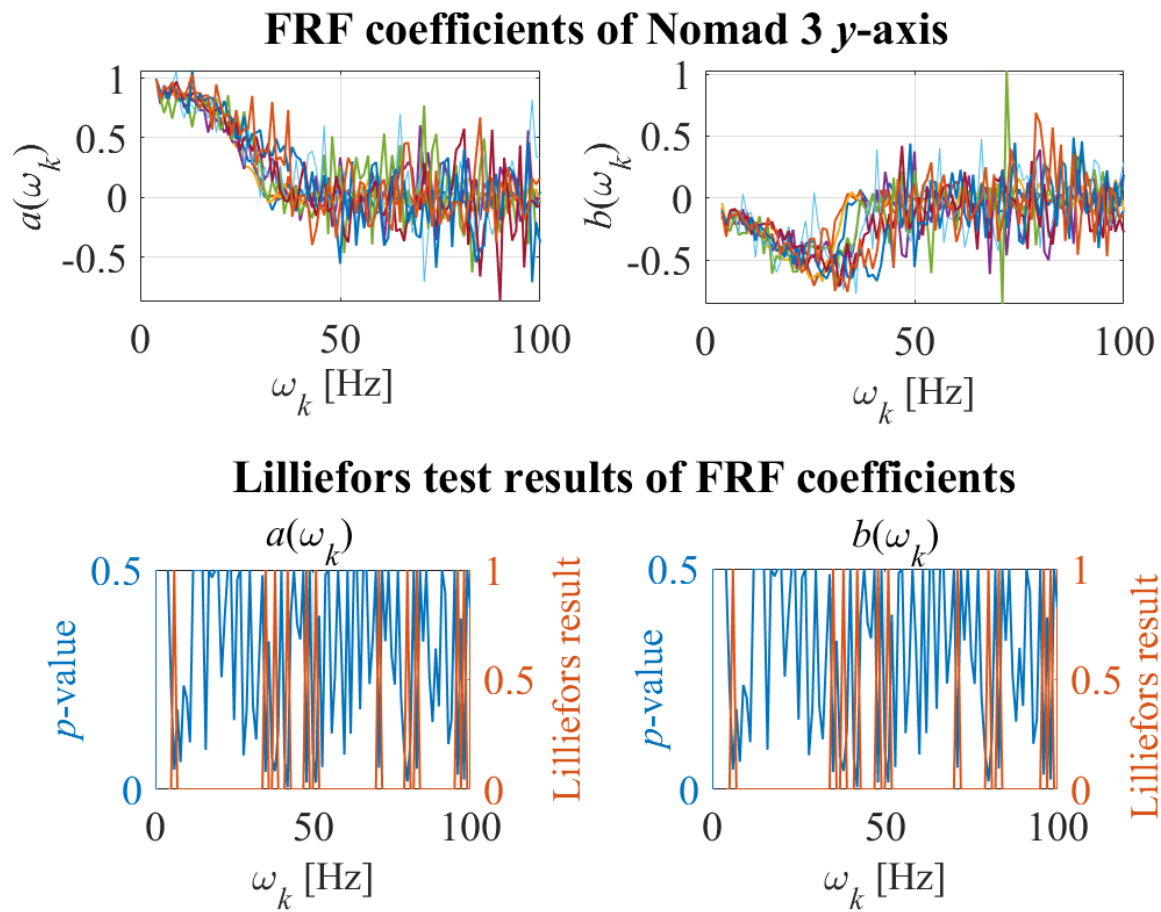
Figure 10. Upper plot: FRF coefficients $a(\omega_k)$ and $b(\omega_k)$ of Nomad 3 y-axis; lower plot: p-value and accept/reject results of Lilliefors test of FRF coefficients.

The output position x is simulated as the sum of motion-induced position x_m and force-induced position x_f, as

$$x = x_m + x_f = \hat{h}_x * x_c + x_f$$
$$\text{where } \hat{h}_x \sim N(\mu_{\hat{h}_x}, \Sigma_{\hat{h}_x}) \text{ is sampled at every } t \tag{29}$$
$$\text{and } x_f(t) = A_f \sin \omega_f t$$

where $A_f = 0.2$ and $\omega_f = 733$ rad/s (7000 rpm) are chosen.

The butterfly trajectory [36] with its contour of the toolpath on the x- and y-axis shown in Figure 11 is selected. For the DD and PIDD methods, $n_w = 10$, $n_2 = 3$, $n_3 = 10$ and $\sigma_\epsilon = 0.01$ are selected. For the DD and PIDD methods, stringency $\eta = 95\%$ is selected. $V_{max} = 30$ mm/s, $A_{max} = 5$ m/s^2, and contour error limit of $E_{max} = 0.4$ mm are selected for the feedrate optimization. The tolerance violation γ, which will be analyzed for each method, is defined as

$$\gamma(t) = \begin{cases} |\varepsilon(t)| - E_{max} & \text{if } |\varepsilon(t)| > E_{max} \\ 0 & \text{otherwise} \end{cases} \tag{30}$$
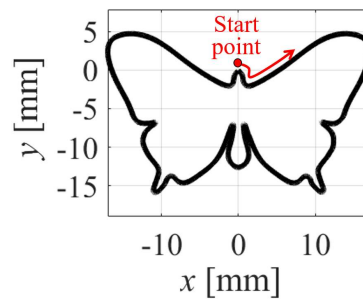
Figure 11. Desired toolpath.

Figure 12 shows the optimized feedrate, acceleration, contour error, tolerance violation and prediction error of all methods. The cycle times and RMS of tolerance violation γ are summarized in Table 1. The PB method is the worst in prediction performance because it is not aware of the unknown uncertainties caused by the force-induced servo error, and hence results in the highest RMS tolerance violation. The DD method improves adherence to the tolerance by learning the unknown uncertainties over time. However, DD method initially suffers from significant prediction error due to its unawareness of known uncertainties. The proposed PIDD method with $\eta = 95\%$ enables restriction of the contour error under the desired stringency by incorporating known uncertainties and learning unknown uncertainties the quickest, which enables it to conservatively stay below the error limit most of the time. Overall, the PIDD method is able to reduce cycle time by 19.3% compared to the conservative approach while maintaining a similar tolerance violation level. To demonstrate the effect of the selection of stringency, Figure 13 compares the commanded feedrate, acceleration, contour error, tolerance violation and prediction error of the PIDD methods using $\eta = 95\%$ and 98%. It is observed that tuning η to a higher level has the effect of making the optimized feedrate more conservative and reducing the error violation.

Note that the proposed PIDD method is not perfect in satisfying the contour error constraints. One reason is that the prediction error is not perfectly zero, and the stringency constraints can only ensure that the worst-case out of $\eta\%$ of contour error distribution is within the tolerance. This issue can be mitigated by increasing η, which will entail more conservative feedrate. Another reason might be due to the nonlinear effects neglected by linearization of the contour error constraint in Eq. (33) and sub-optimal learning in β introduced by Laplace's approximation in Eq. (16). These problems can be addressed by applying nonlinear optimization and non-Gaussian Bayesian regression, at the price of higher computational cost.
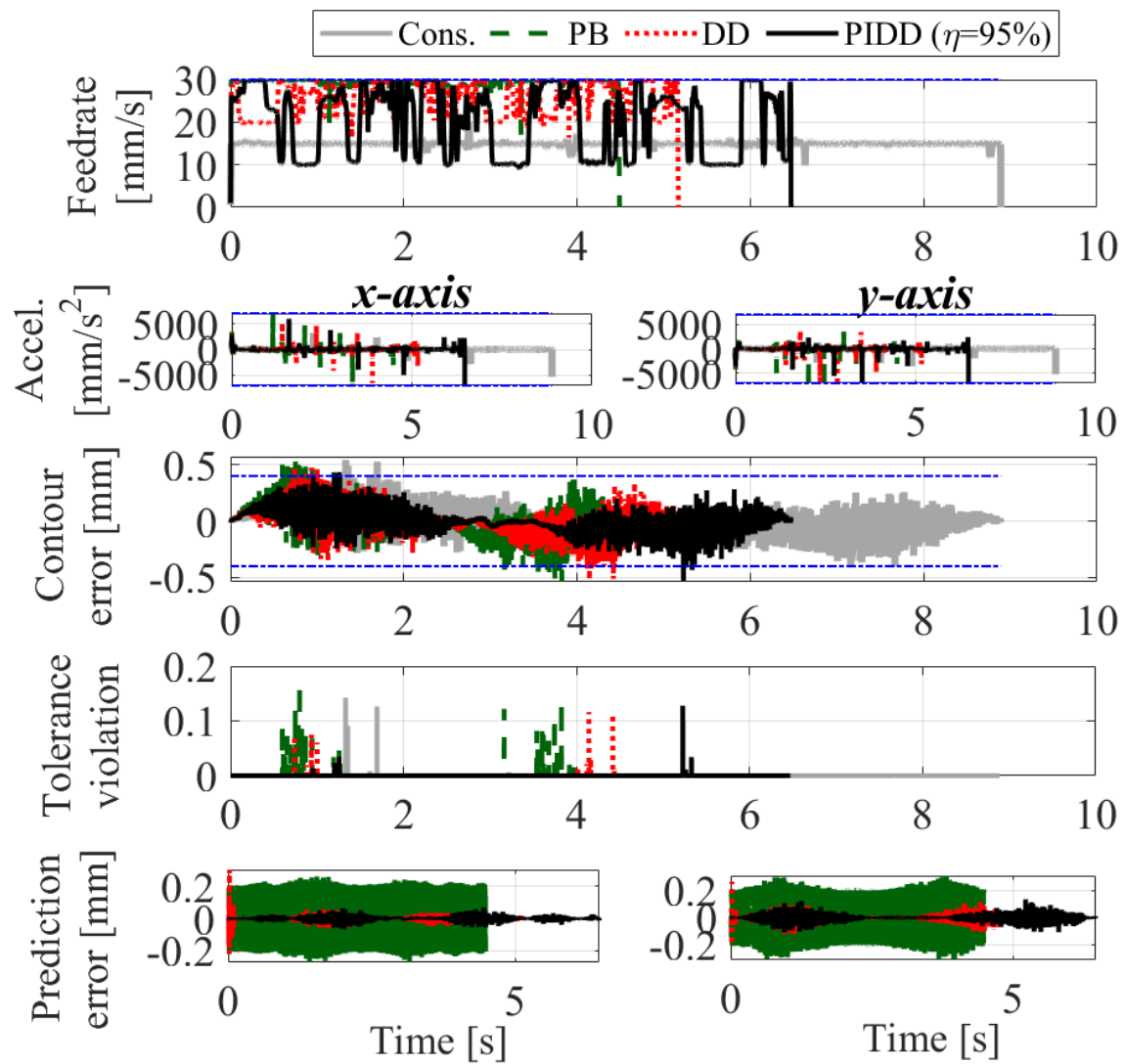
Figure 12. Feedrate, acceleration, contour error, tolerance violation, and prediction error using conservative (Cons.) physics-based (PB), data-driven (DD) and proposed (PIDD) methods with $\eta =$ 95% for numerical validation.
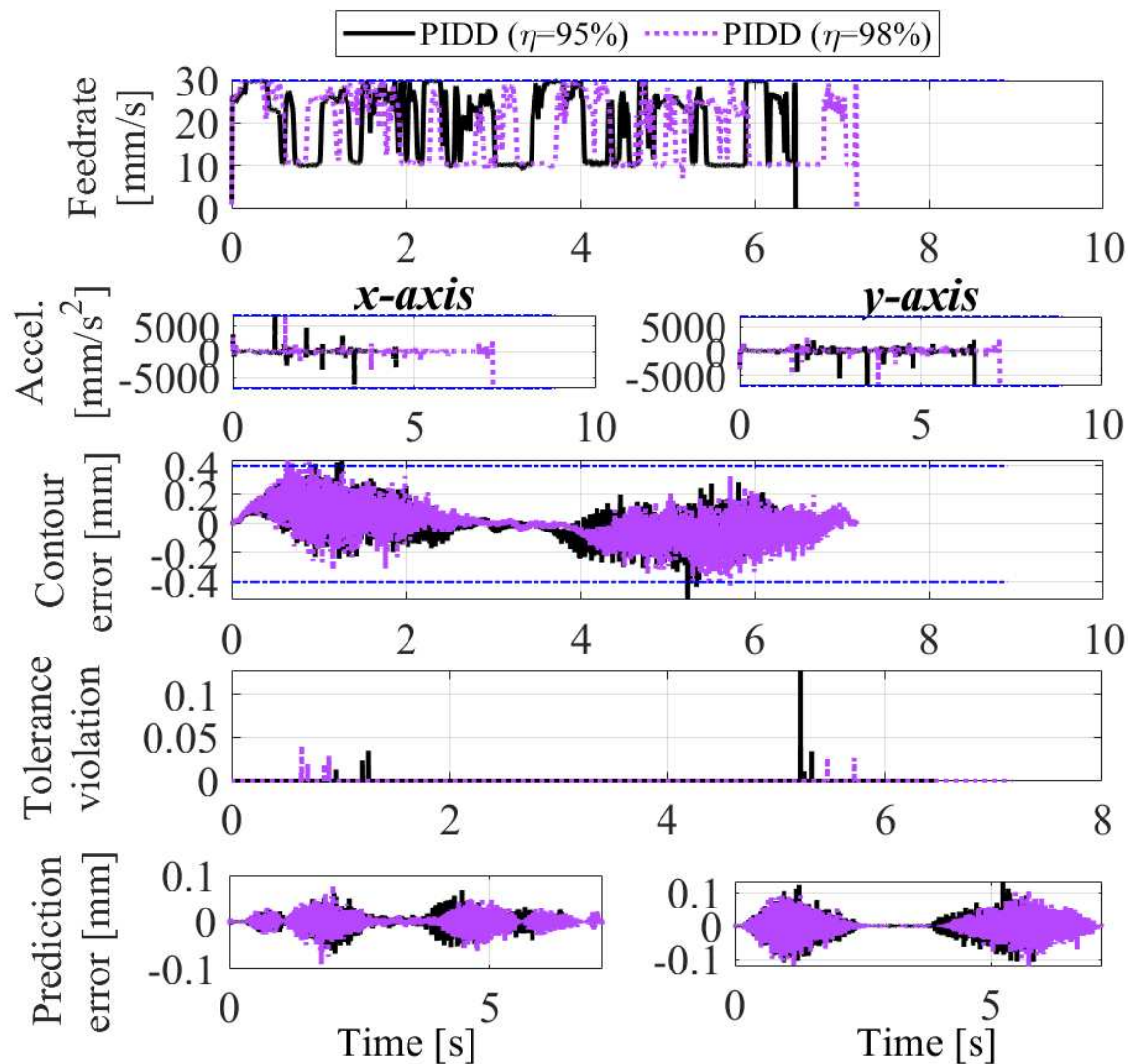
Figure 13. Feedrate, acceleration, contour error, tolerance violation, and prediction error using the proposed (PIDD) methods with η = 95% (from Figure 12 and η = 98% for numerical validation).

Table 1. Cycle times and RMS of tolerance violation γ for conservative (Cons.), physics-based (PB), data-driven (DD) and proposed (PIDD) methods in Figures 12 and 13.

	Cons.	PB	DD	PIDD (η=95%)	PIDD (η=98%)
RMS of γ [μm]	2.2	6.6	3.0	1.7	0.8
Cycle time [s]	8.89	4.49	5.17	6.47	7.17

6. Experimental Validation

For validation of the proposed approach, two experimental setups are used. The first set of experiments, described in Section 6.1, is carried out on an Ender 3 Pro desktop 3D printer, and the second set of experiments, described in Section 6.2, is carried out on a Nomad 3 desktop CNC machine tool used in Section 5. Demonstration of the proposed method on two experimental setups helps to show its versatility.

6.1. Desktop 3D Printer

6.1.1. Experimental Setup

The experimental setup using an Ender 3 Pro desktop 3D printer is shown in Figure 14. The optimization algorithm is implemented on dSPACE 1007 real-time control board running at 500Hz sampling rate, connected to DRV8825 stepper motor drivers for x, y, z and e- axes stepper motors. The measured x and y- axes accelerations from ADXL335 accelerometers are fed back to dSPACE 1007. To deduce the x and y axes displacement from acceleration measurements, a Luenberger state observer [6] is used. The observer gains are chosen such that the dynamics of the observer error (i.e., discrepancy between estimated position using the nominal physics-based model $\mu_{\hat{h}_x}$ and observed position) obtains global asymptotic convergence with an observer frequency $f_o = 15$ Hz.
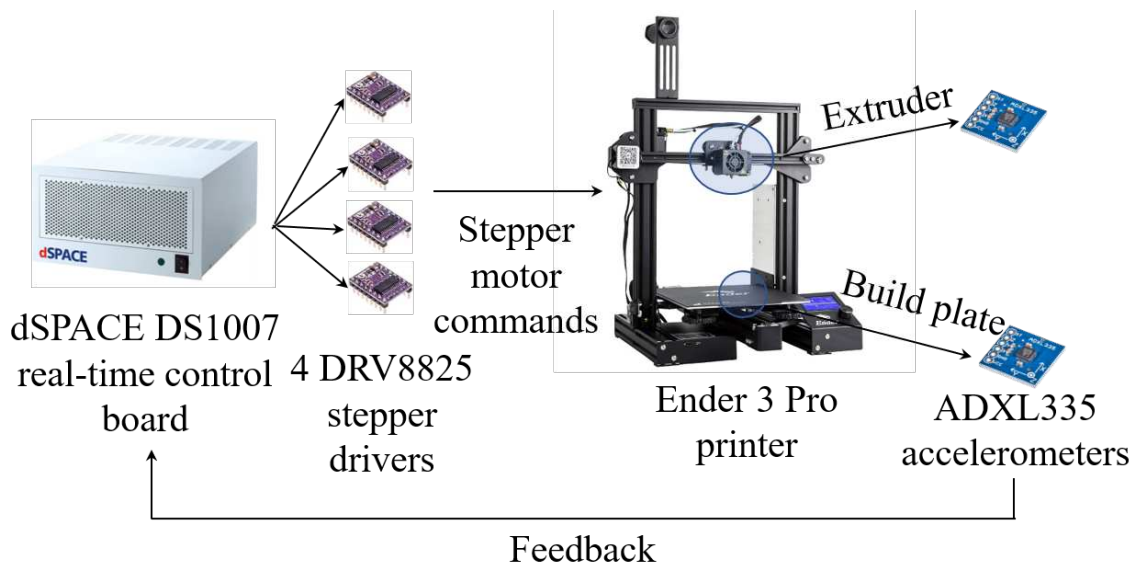
Figure 14. Experimental setup for Section 6.1.

6.1.2. Experimental Results

This section validates the proposed approach experimentally using the desktop 3D printer, by comparing its performance with conservative, PB and DD methods. The butterfly toolpath in Figure 11 is used to optimize the feedrate for air-printing (i.e., no material extrusion) and actual printing of the 3D printer. The known uncertainties of x- and y-axis of the printer are incorporated from FRFs in Figures 15 and 16. Similar to Section 5, Lilliefors test is performed on $a(\omega_k)$ and $b(\omega_k)$ of the FRFs in the x- and y-axis to validate the Gaussian assumption at the 5% significance level and p-values, where the hypothesis acceptance rates are computed as 92% and 91% out of all frequencies for the x- and y-axis, respectively. For the DD and PIDD methods, $n_w = 30$, $n_2 = 10$, $n_3 = 30$ and $\sigma_\epsilon = 0.01$ are used. For the PIDD method, the desired stringency is selected as $\eta = 95\%$. For feedrate optimization, $V_{max} = 70$ mm/s, $A_{max} = 3$ m/s^2 and $E_{max} = 0.1$ mm are chosen.
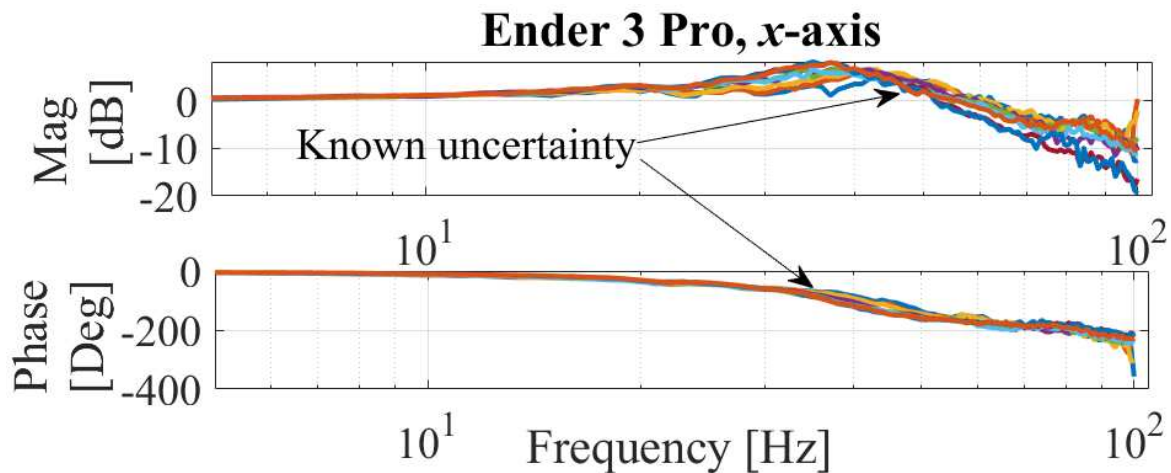
Figure 15. Frequency response functions of the Ender 3 Pro's x-axis showing the known uncertainty obtained under different input acceleration amplitudes.
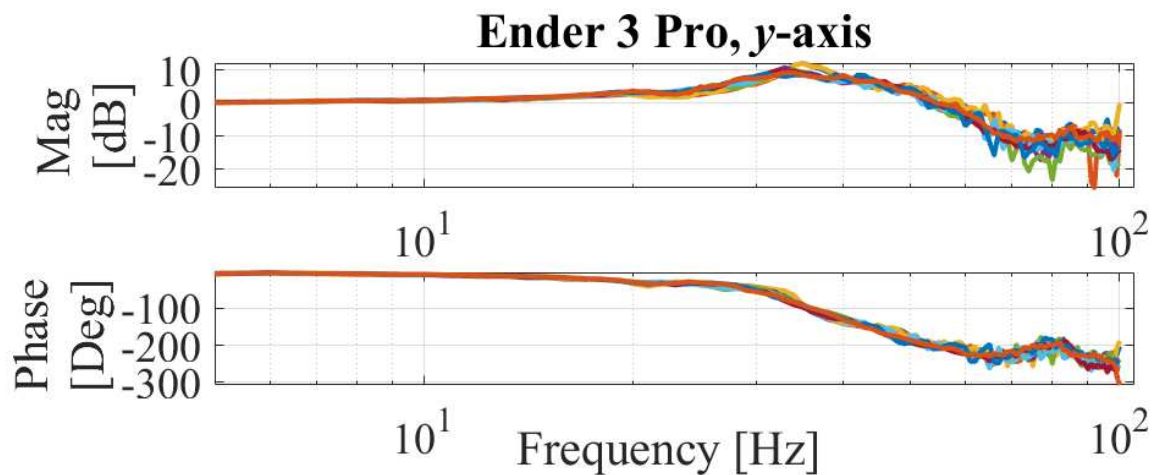
Figure 16. Frequency response functions of the Ender 3 Pro's y-axis showing the known uncertainty obtained under different input acceleration amplitudes.

Figure 17 shows the profiles of the optimized feedrate, acceleration, contour error and prediction error of x- and $y-$ axis using the conservative, PB, DD and PIDD methods. The RMS prediction errors, cycle times and RMS tolerance violation of all methods are reported in Table 2. The PB approach cannot predict the unknown uncertainties, and hence shows the most significant violation in the contour error. The DD method mitigates the violation by learning the unknown uncertainties, and the PIDD method further improves the accuracy by staying the closest to the tolerance with the desired stringency. As a result, the PIDD method completes the motion 17.8% faster than, while yielding similar contour error tolerance satisfaction as the conservative one.
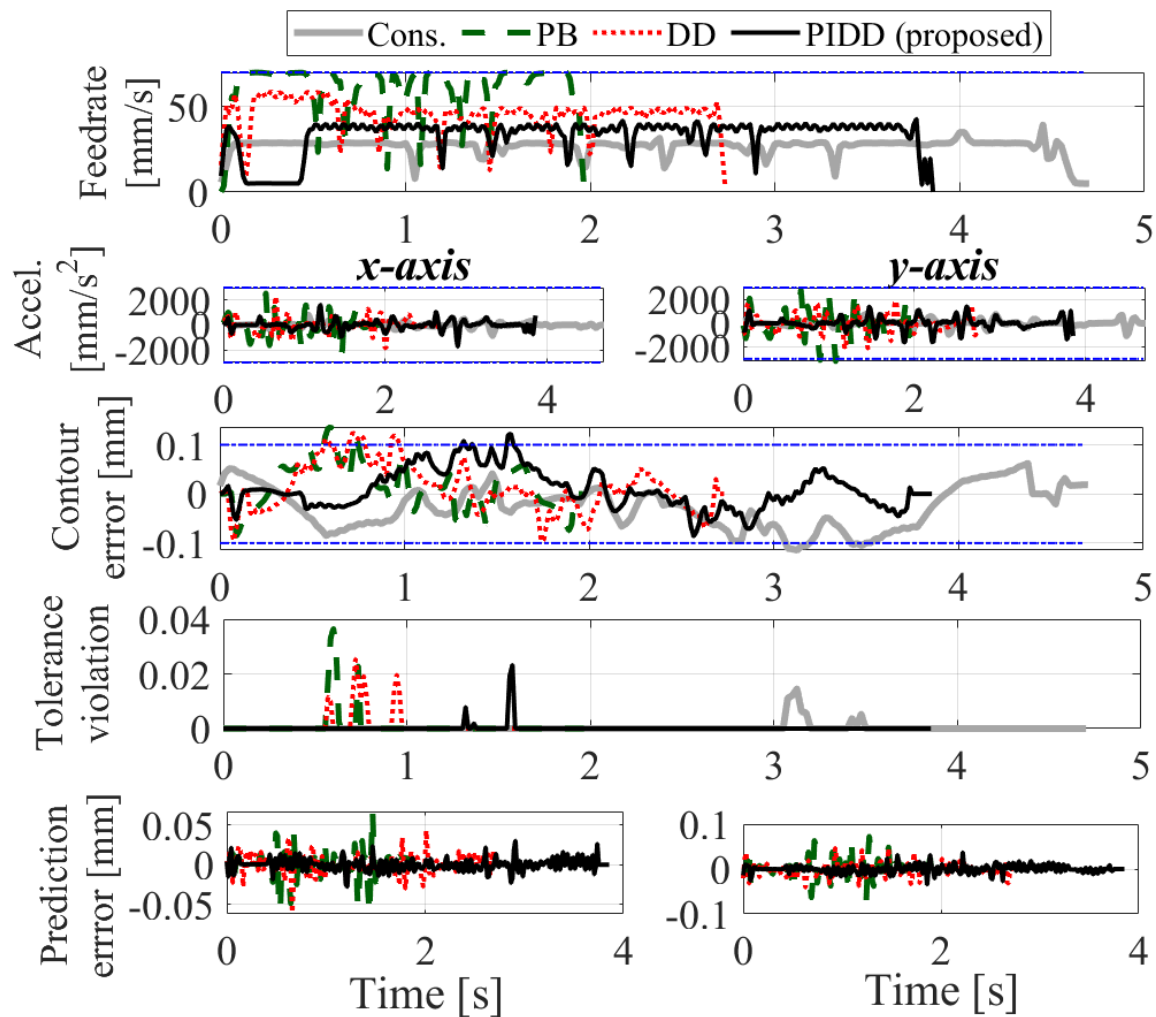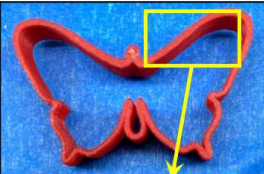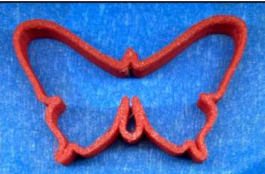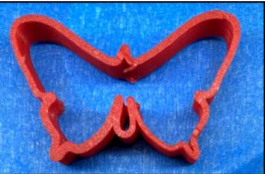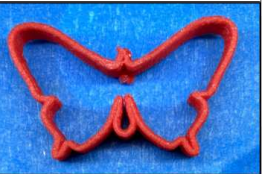
Figure 17. Commanded feedrate, acceleration, contour error, tolerance violation and prediction error using conservative (Cons.), physics-based, data-driven and proposed approaches in air-printing.

Table 2. Comparison of RMS prediction errors, cycle times and RMS of tolerance violation γ using conservative (Cons.), physics-based (Physics.), data-driven (Data.) and proposed methods in air-printing

	Cons.	Physics.	Data.	Proposed
x Pred. Error [μm]	N/A	37.4	22.1	18.1
y Pred. Error [μm]	N/A	31.7	23.9	19.3
Cycle time [s]	4.70	1.97	2.73	3.86
RMS of γ [μm]	1.8	5.5	3.9	1.9

To further validate our findings, a 3D-augmentation of the trajectory in Figure 11 with z-height 8 mm is printed using the same printer. Conservative, PB, DD and PIDD methods are applied at each layer of the print. Figure 18 shows the top and side views of the printed butterflies using the four methods, as well as their print times. The physics-based and data-driven prints show vibration marks in the side view, while the proposed and conservative prints are able to achieve vibration-free surface quality. Overall, the proposed method is able to achieve 15.51% print time reduction compared to the conservative approach while achieving similar print quality.

	Conservative	Physics-based (PB)	Data-driven (DD)	Proposed (PIDD)
Top view				
Side view				
Print time	4 min 30 s	1 min 18 s	1 min 57 s	2 min 38 s

Figure 18. Top and side views of 3D-printed butterfly models using conservative, physics-based, data-driven and proposed approaches and their print times.

6.2. CNC Machine Tool

6.2.1. Experimental Setup

For experimental validation, the same experimental setup with the machine tool in Figure 6 in used. The optimization algorithm is implemented on dSPACE 1007 real-time control board running at 500Hz sampling rate, connected to DRV8825 stepper motor drivers for x, y, and z- axes stepper motors. Renishaw RKLC20-S optical linear encoders are attached to the x and y- axes gantries to measure x and y- axes positions that are fed back to dSPACE 1007.

6.2.2. Experimental Results

This section validates proposed feedrate optimization using the same set of methods for command generation, which are conservative, PB, DD and the proposed PIDD methods. The same desired butterfly trajectory in Figure 11 is used for air cutting and machining an aluminum workpiece with a 3.175 mm diameter flat-end mill and spindle speed of 7000 rpm. Kinematic limits are set as V_{max} = 20 mm/s and A_{max} = 0.5 m/s^2, and contour error bound is chosen as E_{max} = 0.1 mm in the feedrate optimization; n_w = 30, n_2 = 2, n_3 = 30 and σ_ϵ = 0.01 are used in the DD and PIDD methods. The desired stringency is chosen as η = 95% in the DD and PIDD method.

Figure 19 shows the profiles of optimized feedrate, acceleration, contour error, tolerance violation, and prediction error of x- and y-axis in air-cutting using the conservative, PB, DD and PIDD approaches. The PB method frequently violates the tolerance due to unmodeled dynamics, which is caused by the significant prediction error. The DD method slightly improves prediction accuracy, which is further improved in the PIDD method where the contour error is able to be constrained close to the limit using the desired stringency, similar to conservative approach. The proposed PIDD approach completes the motion 38.06% faster than the conservative method, while maintaining a similar level of tolerance adherence. The RMS prediction errors in x- and y-axis, cycle times and RMS tolerance violations of each method are summarized in Table 3.
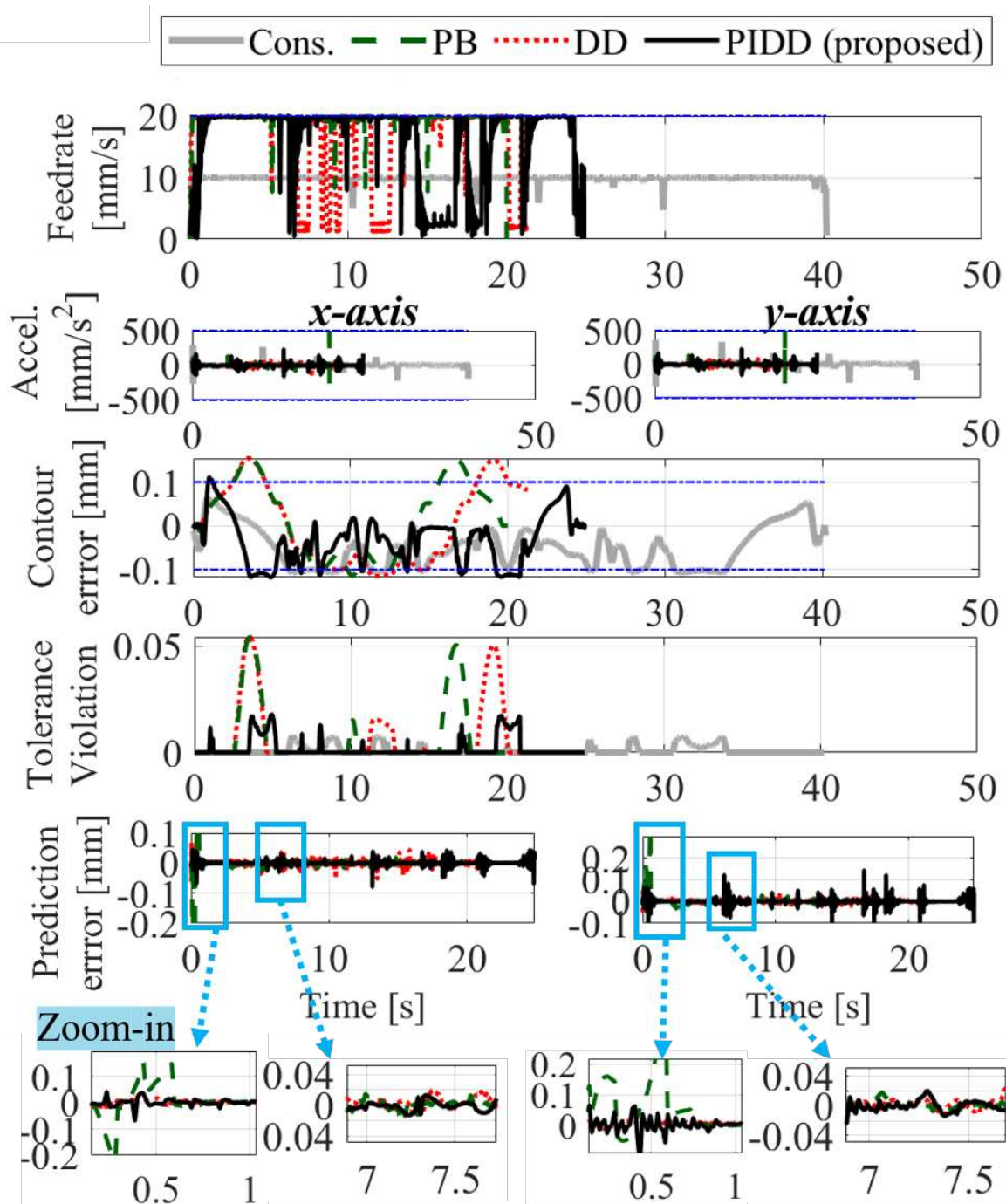
Figure 19. Commanded feedrate, acceleration, contour error, tolerance violation, and prediction error (with its zoomed-in images) of conservative, physics-based (PB), data-driven (DD) and proposed (PIDD) approaches in air-cutting.

Figure 20 shows the profiles of optimized feedrate, acceleration, contour error, tolerance violation, and prediction error of *x*- and *y*-axis in actual cutting using the conservative, PB, DD and PIDD approaches. Similar to air-cutting, the PB method is worst in constraining the contour error due to unmodeled dynamics and/or cutting force. The DD and PIDD methods reduce the prediction error compared to the PB method and are able to constrain the contour error close to the limit using the desired stringency. However, occasionally, PIDD shows worse performance than DD, which may be due to the difference between measured FRFs when the machine tool is not cutting (shown in Figures 7 and 8) and the FRFs while it is cutting. Research has shown that there can be significant

differences between FRFs measured without cutting and those measured while cutting [19]. A possible solution solve this issue is to measure the FRFs and compute known uncertainties during cutting using operational modal analysis, which may be complicated because of FRF's variability on the operating condition of the cutting tool. Another solution is to discard the known uncertainties when they are inaccurate, i.e., use DD only when PIDD may have errors. Overall, the proposed PIDD approach completes the motion 29.02% faster than the conservative method while maintaining a similar level of tolerance adherence. The RMS prediction errors in *x*- and *y*-axis, cycle times and RMS tolerance violations of each method are summarized in Table 3.
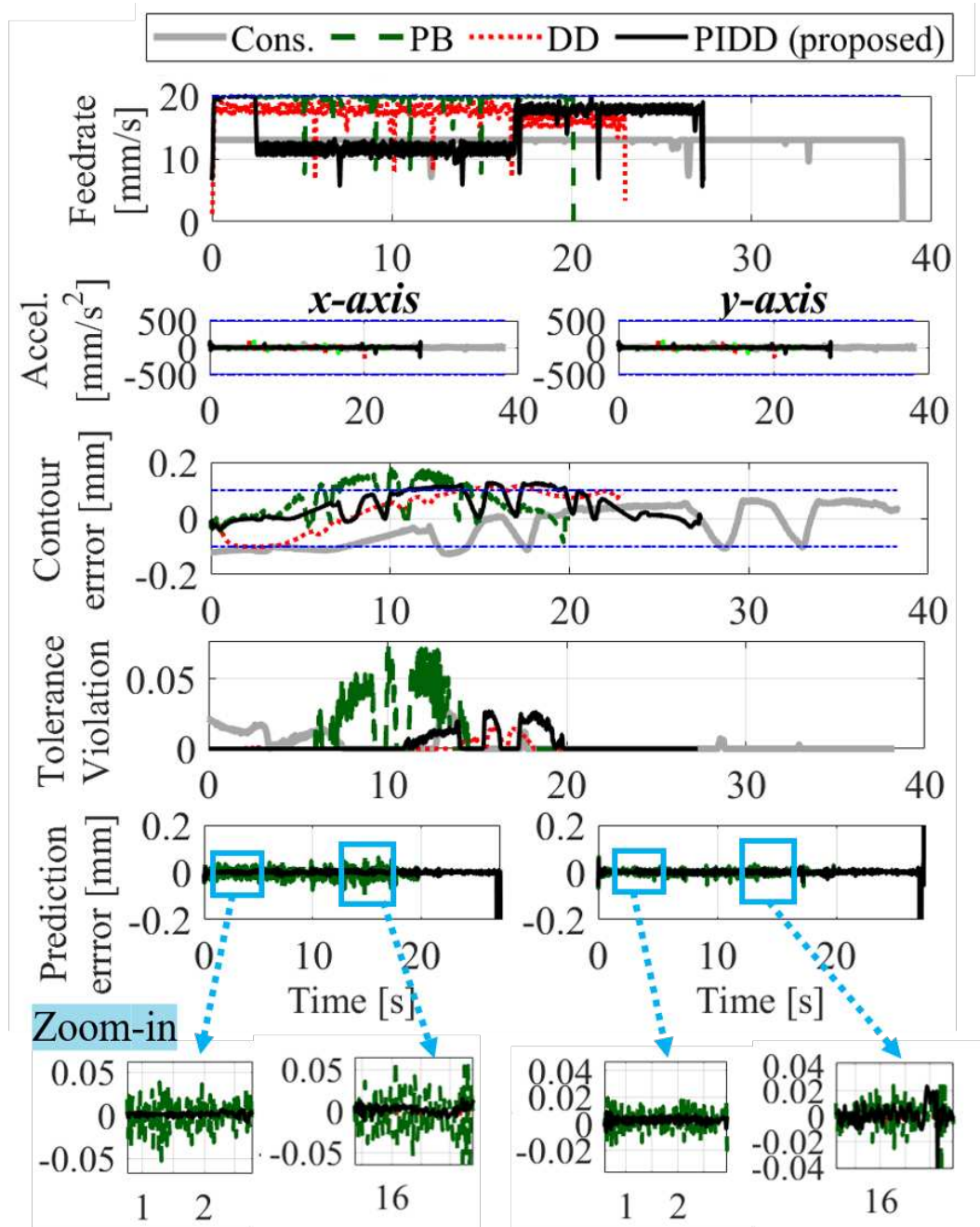
Figure 20. Commanded feedrate, acceleration, contour error, tolerance violation, and prediction error (with its zoomed-in images) of conservative, physics-based (PB), data-driven (DD) and proposed (PIDD) approaches in actual cutting.

Table 3. Comparison of RMS prediction errors, cycle times and RMS of tolerance violation γ using conservative (Cons.), physics-based (PB), data-driven (DD) and proposed (PIDD) methods

		Cons.	PB	DD	PIDD
Air-cutting	x Pred. Error [μm]	N/A	17.8	17.5	10.2
	y Pred. Error [μm]	N/A	25.4	15.7	7.2
	Cycle time [s]	40.2	20.0	21.1	24.9
	RMS of γ [μm]	5.0	16.7	16.5	2.3
Cutting	x Pred. Error [μm]	N/A	47.2	17.3	16.7
	y Pred. Error [μm]	N/A	34.2	17.7	18.1
	Cycle time [s]	39.4	20.0	23.0	27.2
	RMS of γ [μm]	6.8	22.2	3.7	7.4

Figure 21 shows the machined surfaces and their zoomed-in images of upper left wing using the trajectories from Figure 20. The surface machined using PB method shows vibration marks, while the DD and PIDD methods mitigate the vibration and achieve similar quality to that of conservative approach.
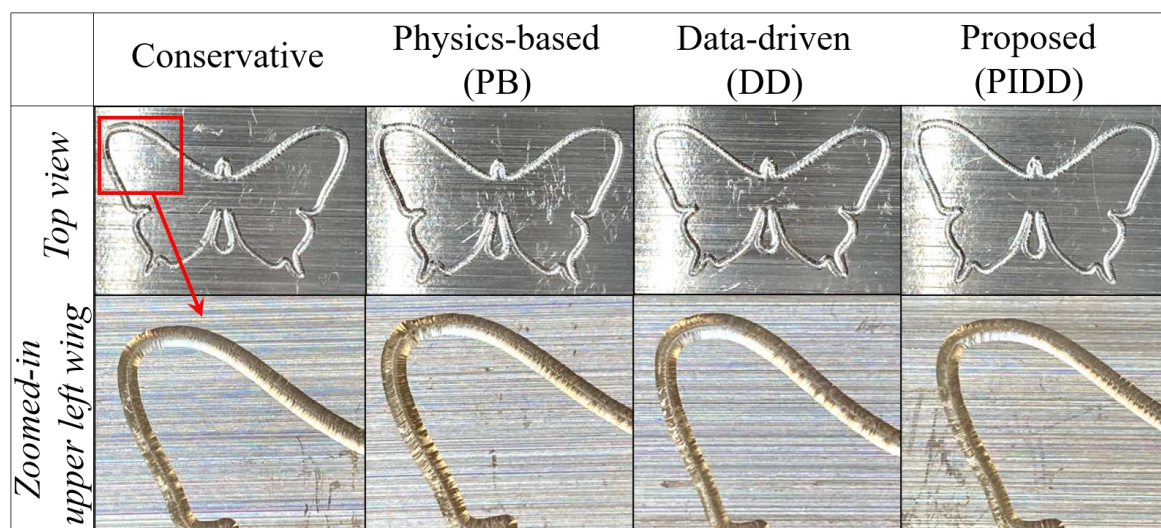
Figure 21. Machined parts and the zoomed-in images of upper left wing using conservative, physics-based (PB), data-driven (DD) and proposed (PIDD) approaches

7. Conclusion and Future Work

This paper presents an MPC framework and methodology for the intelligent feedrate optimization using an uncertainty-aware digital twin. Its key contributions are summarized as follows.

- A novel uncertainty-aware digital twin that predicts contour error is proposed. The digital twin is able to incorporate known uncertainty from physics-based models and learn unknown uncertainty using an online data-driven model to predict contour error's distribution.
- For the first time, a feedrate optimization with constraints on kinematics and contour error using quantified uncertainty is introduced. The contour error's uncertainty using digital twin enables the manufacturer to impose stringency constraints, which can replace trial-and-error approach of tuning the tolerance used in practice.
- We have demonstrated the effectiveness of the intelligent feedrate optimization using uncertainty-aware digital twin, to show up to 38% and 17% cycle time reduction using a desktop CNC machine tool and a desktop 3D printer, respectively, while achieving similar stringency of tolerance to that of the a conservative trial-and-error approach similar to those used in practice.

- The proposed intelligent feedrate optimization is expected to bring impact in achieving desired quality with higher productivity, using less trial-and-error. It is applicable to any machines that use feed drives, such as coordinate measurement machines (CMMs), and precision machine tools.

As a limitation, the proposed method has made several assumptions in the methodology to enable efficient computation, such as Gaussian distribution of frequency response function for computing the known uncertainty and the linearization of contour error constraints for solving the feedrate optimization as a sequential linear programming problem. The future work will explore more sophisticated (non-Gaussian) uncertainty distributions and nonlinear contour error constraints to improve the accuracy of the digital twin, at the expense of higher computational cost and non-closed form solutions. Furthermore, additional forms of learning to the uncertainty-aware digital twin, such as part-to-part or machine-to-machine learning, will be investigated to improve prediction accuracy.

Acknowledgments: This work is partially funded by the National Science Foundation grants #1931950 and #2054715.

Appendix A. Initialization of μ_β and Σ_β in Section 3.2

The distribution of the weight β can be initialized and learned more accurately using the feature uncertainty from Eq. (13), where the procedure is described as follows. First, μ_β and Σ_β are initialized as the priors $\mu_{\beta 0}$ and $\Sigma_{\beta 0}$ in the 0-th batch using the nominal physics-based model $\mu_{\hat{h}_x}$ from Eq. (11). To estimate $\mu_{\beta 0}$ and $\Sigma_{\beta 0}$, maximum likelihood estimation (MLE) method is applied on a dataset created offline using a trial desired trajectory $x_{d,trial}$ with length N_x that traverses a pre-defined path with conservative kinematics used in practice. The nominal physics-based model $\mu_{\hat{h}_x}$ is used to filter $x_{d,trial}$ and formulate $\hat{x}_{trial}$. Then, Section 3.1's framework on deterministic feature vector is borrowed to create multiple datasets consisting of feature vectors and corresponding predictions, i.e., $(\psi_0, x_{trial}(0)), (\psi_{T_s}, x_{trial}(T_s)), ..., (\psi_{N_x T_s}, x_{trial}(N_x T_s))$, assuming ψ_t is deterministic and $\psi_{t3} = 0$ for all t. Finally, $\mu_{\beta 0}$ and $\Sigma_{\beta 0}$ can be optimized using MLE as

$$
\begin{aligned}
&\mu_{\beta 0}, \Sigma_{\beta 0} \\
&= \underset{\mu_\beta \Sigma_\beta}{\operatorname{argmin}} \sum_{i=0}^{N_x} \Big(-\frac{1}{2}(x_{trial}(iT_s) - \mu_\beta^{\mathrm{T}} \psi_{iT_s})^{\mathrm{T}} \\
&\qquad\qquad (\psi_{iT_s}^{\mathrm{T}} \Sigma_\beta \psi_{T_s i} + \sigma_\epsilon^2)^{-1}(x_{trial}(iT_s) - \mu_\beta^{\mathrm{T}} \psi_{iT_s}) \Big)
\end{aligned}
\tag{31}
$$

where $x_{trial}(t)$ is approximated as $\hat{x}_{trial}(t)$.

Appendix B. Derivation of Coefficients c_0, c_1 and C_2 in Eq. (16) in Section 3.2

The coefficients c_0, c_1 and C_2 in Eq. (16) can be derived using 0-th, 1-st and 2-nd derivatives of vector-valued function $\ln p(x(t)|\beta)$ in Eq. (15) with respect to β, respectively, as

$$
\begin{aligned}
c_0 &= -\frac{1}{2}(x(t) - \bar{\beta}^{\mathrm{T}} \mu_{\psi_t})^{\mathrm{T}}(\bar{\beta}^{\mathrm{T}} \Sigma_{\psi_t} \bar{\beta} + \sigma_\epsilon^2)^{-1}(x(t) - \bar{\beta}^{\mathrm{T}} \mu_{\psi_t}) \\
c_1 &= -\frac{1}{2}\Big(2\Sigma_{\psi_t}^{\mathrm{T}} \mu_{\psi_t} x(t) \bar{\beta}^{\mathrm{T}} \bar{\beta} + (2\mu_{\psi_t} \mu_{\psi_t}^{\mathrm{T}} \sigma_\epsilon^2 - 2\Sigma_{\psi_t} x(t)^2)^{\mathrm{T}} \bar{\beta} \\
&\qquad - 2\mu_{\psi_t} x(t) \sigma_\epsilon^2 \Big) \cdot \operatorname{diag}(\bar{\beta}^{\mathrm{T}} \Sigma_{\psi_t}^{\mathrm{T}} \bar{\beta} + \sigma_\epsilon^2)^{-1} \\
C_2 &= -\frac{1}{2}\Big(-4\Sigma_{\psi_t}^2 \mu_{\psi_t} x(t) \bar{\beta}^{\mathrm{T}} \bar{\beta} \bar{\beta}^{\mathrm{T}} + 6\Sigma_{\psi_t}^2 x(t)^2 \bar{\beta} \bar{\beta}^{\mathrm{T}} \\
&\qquad + 12\Sigma_{\psi_t} \mu_{\psi_t} x(t) \sigma_\epsilon^2 \bar{\beta}^{\mathrm{T}} - 6\Sigma_{\psi_t} \mu_{\psi_t} \mu_{\psi_t}^{\mathrm{T}} \sigma_\epsilon^2 \Big) \\
&\qquad \cdot \operatorname{diag}(\bar{\beta}^{\mathrm{T}} \Sigma_{\psi_t}^{\mathrm{T}} \bar{\beta} + \sigma_\epsilon^2)^{-1}
\end{aligned}
\tag{32}
$$

where the local point $\bar{\beta}$ is taken as the prior μ_β from the previous window.

Appendix C. Linearization of Contour Error Constraint in Eq. (27) in Terms of $\tilde{x}_d^j$ and $\tilde{y}_d^j$

The contour error constraint in Eq. (27) is linearized in terms of $\hat{x}_d^j$ and $\hat{y}_d^j$ by linearizing the standard deviation term $\sigma_{\bar{\varepsilon}(t)}$ in Eq. (21) with respect to ψ_t and ψ_{ty} using linearization points $\bar{\psi}_t$ and $\bar{\psi}_{ty}$ as

$$
\begin{aligned}
\sigma_{\bar{\varepsilon}(t)} &\approx S_x \psi_t + S_y \psi_{ty} + S_{00} + S_{01} \\
\text{where } S_{00} &= \sin(\theta(t))^2 (\bar{\psi}_t^T \Sigma_\beta \bar{\psi}_t + \sigma_\epsilon^2) \\
&\quad + \cos(\theta(t))^2 (\bar{\psi}_{ty}^T \Sigma_{\beta_y} \bar{\psi}_{ty} + \sigma_{\epsilon_y}^2) \\
S_{01} &= -2\sin(\theta(t))^2 \Sigma_\beta \bar{\psi}_t - 2\cos(\theta(t))^2 \Sigma_\beta \bar{\psi}_{ty} \\
S_x &= \frac{1}{\sqrt{S_{00}}} \sin(\theta(t))^2 \Sigma_\beta \\
S_y &= \frac{1}{\sqrt{S_{00}}} \cos(\theta(t))^2 \Sigma_{\beta_y}
\end{aligned}
\tag{33}
$$

where $\bar{\psi}_t$ is formulated via generating the terms $\hat{x}(t - n_2 T_s) \cdots \hat{x}(t)$ in $\bar{\psi}_{t2}$ by filtering the linearization point $f(\bar{p})$ with $\mu_{\hat{h}_x}$. Likewise, $\bar{\psi}_{ty}$ is formulated using $\mu_{\hat{h}_y}$.

Finally, let the alterable features in ψ_t be ψ_{ta}, and the weights corresponding to the alterable feature in ψ_t, i.e., $\hat{x}(t)$, be denoted as β_a and that to the unalterable features ψ_{tu} as β_u. The same notations ψ_{tyu}, β_{ya} and β_{yu} will be held for y-axis. Then, by substituting Eq. (21), (25) and (33) into Eq. (27), the contour error constraint be re-written as

$$
\begin{aligned}
&-\sin(\theta(t))(M_t x_d^j - \mu_{\beta_u}^T \psi_{tu} - \mu_{\beta_a}^T (T_x x_d^j + T_{0x})) \\
&+ \cos(\theta(t))(M_t y_d^j - \mu_{\beta_{yu}}^T \psi_{tyu} - \mu_{\beta_{ya}}^T (T_y y_d^j + T_{0y})) \\
&\leq E_{max} - \Phi^{-1}(\eta)\Bigg(S_x \psi_{tu} + S_x (T_x x_d^j + T_{0x}) \\
&\quad + S_y \psi_{tyu} + S_y (T_y y_d^j + T_{0y}) + S_{00} + S_{01} \Bigg)
\end{aligned}
\tag{34}
$$

which can be rearranged as linear in terms of x_d^j and y_d^j as

$$
\begin{aligned}
U_x x_d^j &+ U_y y_d^j \leq U_0 \\
\text{where } U_x &= -\sin(\theta(t))(M_t - \mu_{\beta_a}^T T_x) + \Phi^{-1}(\eta) S_x T_x \\
U_y &= \cos(\theta(t))(M_t - \mu_{\beta_{ya}}^T T_y) - \Phi^{-1}(\eta) S_y T_y \\
U_0 &= \sin(\theta(t))(-\mu_{\beta_u}^T \psi_{tu} - \mu_{\beta_a}^T T_{0x}) \\
&\quad - \cos(\theta(t))(-\mu_{\beta_{yu}}^T \psi_{tyu} - \mu_{\beta_{ya}}^T T_{0y}) \\
&\quad + E_{max} - \Phi^{-1}(\eta)(S_x \psi_{tu} + S_x T_{0x} \\
&\quad + S_y \psi_{tyu} + S_y T_{0y} + S_{00} + S_{01})
\end{aligned}
\tag{35}
$$

Finally, the contour error constraint in Eq. (35) is also linear with respect to the decision variable p^j using the relationship in Eq. (22).

References

1. Bharathi A and Dong J. Feedrate optimization for smooth minimum-time trajectory generation with higher order constraints. *Int J Adv Manuf Technol*, 82(5):1029–1040, 2016.

2. Fussell BK, Jerard RB, and Hemmett JG. Robust feedrate selection for 3-axis NC machining using discrete models. *J. Manuf. Sci. Eng.*, 123(2):214–224, 2001.

3. Chou CH, Duan M, and Okwudire CE. A linear hybrid model for enhanced servo error pre-compensation of feed drives with unmodeled nonlinear dynamics. *CIRP Annals*, 70(1):301–304, 2021.

4. Bort CM, Leonesio M, and Bosetti P. A model-based adaptive controller for chatter mitigation and productivity enhancement in cnc milling machines. *Robotics and Computer-Integrated Manufacturing*, 40:34–43, 2016.

5. Lam D, Manzie C, and Good MC. Model predictive contouring control for biaxial systems. *IEEE Transactions on Control Systems Technology*, 2(1):552–559, 2012.

6. Luenberger DG. An introduction to observers. *IEEE Trans. On Automatic Control*, 16(6):596–602, 1971.

7. Gordon DJ and Erkorkmaz K. Accurate control of ball screw drives using pole-placement vibration damping and a novel trajectory prefilter. *Precision Engineering*, 37(1):308–322, 2013.

8. MacKay DJ. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

9. Erdim H, Lazoglu I, and Ozturk B. Feedrate scheduling strategies for free-form surfaces. *International Journal of Machine Tools and Manufacture*, 46(7–8):747–757, 2006.

10. Erdim H, Lazoglu I, and Kaymakci M. Free-form surface machining and comparing feedrate scheduling strategies. *Machining Science and Technology*, 11(1):117–133, 2007.

11. Kim H and Okwudire C. Intelligent feedrate optimization using a physics-based and data-driven digital twin. *CIRP Annals.*, pages 308–322, 2023.

12. Kim H and Okwudire CE. Simultaneous servo error pre-compensation and feedrate optimization with tolerance constraints using linear programming. *Int J Adv Manuf Technol*, 109(3):809–821, 2020.

13. Kim H and Okwudire CE. Accurate and computationally efficient approach for simultaneous feedrate optimization and servo error pre-compensation of long toolpaths—with application to a 3D printer. *The International Journal of Advanced Manufacturing Technology*, 115(7–8):2069–2082, 2021.

14. Möhring HC, Wiederkehr P, Erkorkmaz K, and Kakinuma Y. Self-optimizing machining systems. *CIRP Annals*, 69(2):740–763, 2020.

15. Park HS, Qi B, Dang DV, and Park DY. Development of smart machining system for optimizing feedrates to minimize machining time. *Journal of Computational Design and Engineering*, 5(3):299–304, 2018.

16. Lee HU and Cho DW. An intelligent feedrate scheduling based on virtual machining. *The International Journal of Advanced Manufacturing Technology*, 22:873–882, 2003.

17. Lilliefors HW. On the kolmogorov-smirnov test for normality with mean and variance unknown. *Journal of the American Statistical Association*, 62(118):339–402, 1967.

18. Feng HY and Su N. Integrated tool path and feed rate optimization for the finishing machining of 3D plane surfaces. *International Journal of Machine Tools and Manufacture*, 40(11):1557–1572, 2000.

19. Zagbani I and Songmene V. Estimation of machine-tool dynamic parameters during machining operation through operational modal analysis. *International Journal of Machine Tools and Manufacture*, 49(12–13):945–957, 2009.

20. Dong J and Stori JA. Optimal feed-rate scheduling for high-speed contouring. *J. Manuf. Sci. Eng.*, 129(1):63–76, 2007.

21. Yang J, Aslan D, and Altintas Y. A feedrate scheduling algorithm to constrain tool tip position and tool orientation errors of five-axis CNC machining under cutting load disturbances. *CIRP journal of Manufacturing Science and Technology*, 23:78–90, 2018.

22. Rawlings JB. Model predictive control technology. volume 1, pages 662–676, 1999.

23. Rawlings JB, Maye DQ, and Diehl M. *Model predictive control: theory, computation, and design*, volume 2. Madison, WI: Nob Hill Publishing.

24. Ko JH, Yun WS, and Cho DW. Off-line feed rate scheduling using virtual cnc based on an evaluation of cutting performance. *Computer-Aided Design*, 35(4):383–393, 2003.

25. Erkorkmaz K, Chen QGC, Beudaert X Zhao MY, and Gao XS. Linear programming and windowing based feedrate optimization for spline toolpaths. *CIRP Annals*, 66(1):393–396, 2017.

26. Rattunde L, Laptev I, Klenske ED, and Möhring HC. Safe optimization for feedrate scheduling of power-constrained milling processes by using Gaussian processes. *Procedia CIRP*, 99:127–132, 2021.

27. Zhang L, Feng J, Wang Y, and Chen M. Feedrate scheduling strategy for free-form surface machining through an integrated geometric and mechanistic model. *International Journal of Advanced Manufacturing Technology*, 40:740–763, 2009.

28. Chen M, Xu J, and Sun Y. Adaptive feedrate planning for continuous parametric tool path with confined contour error and axis jerks. *The International Journal of Advanced Manufacturing Technology*, 89(1):1113–1125, 2017.

29. Richards ND, Fussell BK, and Jerard RB. Efficient NC machining using off-line optimized feedrates and on-line adaptive control. volume 3641, pages 181–191, 2002.

30. Qin P, Wang M, and Sun L. Feed rate variation strategy for semi-conical shell workpiece in ball head end milling process. *Applied Sciences*, 10(24):9135, 2020.

31. Zhang Q and Li SR. Efficient computation of smooth minimum time trajectory for cnc machining. *Int J Adv Manuf Technol*, 68:683–692, 2013.

32. Kontar R, Raskutti G, and Shiyu Z. Minimizing negative transfer of knowledge in multivariate gaussian processes: A scalable and regularized approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10):3508–3522, 2020.

33. Balula S, Liao-McPherson D, Rupenyan A, and Lygeros J. Data-driven reference trajectory optimization for precision motion systems. *arXiv preprint 2205.15694*, 2022.

34. Tajima S and Sencer B. Real-time trajectory generation for 5-axis machine tools with singularity avoidance. *CIRP Annals*, 69(1):349–352, 2020.

35. Tajima S and Sencer B. Online interpolation of 5-axis machining toolpaths with global blending. *International Journal of Machine Tools and Manufacture*, 175:103862, 2022.

36. Yang S, Ghasemi AH, Lu X, and Okwudire CE. Pre-compensation of servo contour errors using a model predictive control framework. *Int J Mach Tool Manu*, 98:50–60, 2015.

37. Merdol SD and Altintas Y. Virtual simulation and optimization of milling applications—part II: optimization and feedrate scheduling.

38. Merdol SD and Altintas Y. Virtual cutting and optimization of three-axis milling processes. *International Journal of Machine Tools and Manufacture*, 48(10):1063–1071, 2008.

39. Fan W, Gao XS, Lee CH, Zhang K, and Zhang Q. Time-optimal interpolation for five-axis cnc machining along parametric tool path based on linear programming. *Int J Adv Manuf Technol*, 69(5–8):1373–1388, 2013.

40. Fan W, Gao XS, Lee CH, Zhang K, and Zhang Q. Time-optimal interpolation for five-axis cnc machining along parametric tool path based on linear programming. *The International Journal of Advanced Manufacturing Technology*, 69(5):1373–1388, 2013.

41. Altintas Y, Yang J, and Kilic ZM. Virtual prediction and constraint of contour errors induced by cutting force disturbances on multi-axis cnc machine tools. *CIRP Annals*, 68(1):377–380, 2019.

42. Altintas Y, Biermann D Kersting P, Budak E, Denkena B, and Lazoglu I. Virtual process systems for part machining operations. *CIRP Annals*, 63(2):585–605, 2014.

43. Altintas Y and Merdol SD. Virtual high performance milling. *CIRP annals*, 56(1):81–84, 2007.

44. Chang YC, Chen CW, and Tsao TC. Near time-optimal real-time path following under error tolerance and system constraints. *CIRP Annals*, 140(7):071004, 2018.