CIRP Annals - Manufacturing Technology 74 (2025) 569–573

Contents lists available at ScienceDirect

CIRP Annals - Manufacturing Technology

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

Feedrate optimization based on part-to-part learning in repeated machining

Cheng-Hao Chou, Chenhui Shao, Chinedum E. Okwudire (2)*

Department of Mechanical Engineering, University of Michigan, Ann Arbor, Michigan, USA

ARTICLE INFO

Article history:
Available online 13 May 2025

Keywords:
Computer numerical control (CNC)
Machine learning
Feedrate optimization

ABSTRACT

Cutting operations often involve machining parts of similar geometry repeatedly, offering opportunities for learning-based improvements. While past studies have focused on enhancing machining accuracy through part-to-part learning, this work shifts the focus to optimizing feedrate under servo error constraints. A data-driven model, trained online on prior machining data, predicts future servo errors and enables iterative feedrate optimization. Confidence in the model improves as more parts are machined, permitting progressively higher feedrates. Experimental results demonstrate significant speed gains within a few iterations, showcasing the potential of part-to-part learning for autonomously achieving faster machining without violating servo error constraints.

1. Introduction

In manufacturing settings, it is common to produce multiple copies of similar or identical parts on the same machine tool. Given the similarity in part geometry and machining conditions, servo errors for successive parts tend to exhibit predictable patterns. This consistency presents an opportunity for learning from past machining operations to improve the performance of future ones. Such part-to-part learning has typically been explored in the context of iterative learning control (ILC) [1], where servo error compensation strategies are refined over successive parts to enhance motion accuracy. There are generally two approaches for performing ILC. The first involves iteratively learning the compensation signal to progressively reduce servo errors [2–4]. However, such methods are only effective for compensating the servo errors of identical part geometries machined under the exact same conditions. The second approach involves iteratively tuning the parameters of a compensation model to progressively reduce servo errors [5–7]. This approach is more versatile in its effectiveness for compensating the servo errors of similar but non-identical parts.

Regardless of the adopted approach, the existing work on part-to-part learning has so far mainly focused on minimizing servo errors. However, in practice, when servo tolerance requirements are already met, further improvements in accuracy yield diminishing returns. In these scenarios, manufacturers are more interested in enhancing productivity while maintaining desired quality levels. Along this direction, Rattunde et al. [8] proposed an approach for optimally scheduling feedrate subject to spindle power constraints by progressively learning a model of spindle power online, using Gaussian process regression, as parts were machined. However, to the authors' best knowledge, there have been no efforts to achieve part-to-part learning where the goal is to optimize feedrate subject to servo error constraints.

Note, however, that numerous studies have investigated optimizing feedrate while considering constraints related to servo errors. Most feedrate optimization methods focus on kinematic limitations such as speed, acceleration, and jerk [9–12]. However, such methods fail to impose direct constraints on servo error. To address servo error explicitly, some feedrate optimization techniques incorporate kinematic constraints along with servo error constraints using physics-based or empirical servo error models that are determined offline [13–17]. The problem is that, because such models are not updated online, they fail to account for unmodeled dynamics or uncertainties in the motion or cutting forces, significantly hindering their accuracy and effectiveness in constraining servo errors.

The shortcomings of offline servo error models have recently led to efforts to leverage online learning for feedrate optimization. For example, Chang et al. [18] utilized a model predictive control (MPC) framework to achieve feedrate optimization subject to servo error and kinematic constraints. The servo error model was updated online through state observers. Their method was experimentally shown to constrain servo errors with reasonable accuracy. Kim and Okwudire [19] presented a digital twin (DT) based framework for feedrate optimization that integrated a physics-based model of servo dynamics with a data-driven component, updated online, to accurately predict and constrain servo errors induced by motion and cutting forces. Furthermore, Kim et al. [20] proposed an approach for performing feedrate optimization using an uncertainty-aware DT that integrated stochastic physics-based and data-driven models. Updated online via sensor feedback, the DT accounted for model uncertainties, enabling feedrate optimization while enforcing servo error constraints with quantified confidence levels. However, none of the methods discussed above addressed how to transfer online learning from one part to another to facilitate feedrate optimization subject to servo error constraints.

To address this shortcoming of prior work, this paper makes the following original contributions:

(1) It proposes a novel method for online part-to-part learning that iteratively optimizes feedrate based on machining data from prior parts of identical or similar geometry.

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

https://doi.org/10.1016/j.cirp.2025.04.043

(2) The part-to-part learning is enabled by a stochastic data-driven servo error model that starts out conservatively with low confidence but progressively builds confidence as it gathers more data that confirms its predictions.

The predicted contour errors are integrated as constraints into an existing feedrate optimization algorithm [21] that increases feedrate as the servo error model's confidence grows. The effectiveness of the proposed method is validated using air-cutting experiments on a 3-axis desktop milling machine, where it demonstrates significant increases in feedrate within a few iterations while enforcing servo error constraints with high accuracy.

The outline of the paper is as follows. Section 2 discusses the proposed approach, including details of the data-driven model for part-to-part learning, the tuning of its hyperparameters, and its integration into an existing feedrate optimization algorithm. Section 3 then presents the experimental validation while conclusions and future work are presented in Section 4.

2. Methodology

2.1. Overview of proposed approach

Fig. 1 provides an overview of the proposed approach for feedrate optimization based on part-to-part learning. It comprises an actual machine tool (MT) and its digital twin (DT). Motion commands for machining identical or similar parts are sent to both the MT and its DT. Each part is parameterized using a path coordinate system with path variable s. Without loss of generality, the servo errors of both the MT and its DT are represented as contour errors derived from the tracking errors of each of its axes. The DT consists of a data-driven contour error model that is iteratively trained online using contour errors measured from the MT. The contour errors predicted by the model are used as constraints in feedrate optimization, and the optimized feedrates are sent to the MT.

Initially, the DT is assumed to have no reliable contour error model. Therefore, conservative feedrates are sent to the MT to cut the first few parts to initialize the model. The contour errors measured after cutting the first few parts are sent to the DT to learn a contour error model along with its confidence in the model. As one would expect, the confidence in the model is initially low, hence the large credible interval. The feedrate optimizer, uses the predicted contour errors and credible intervals to determine the highest feedrate that satisfies the error constraints. The low confidence of the model leads to relatively low feedrates. The process is repeated for subsequent parts and the confidence of the model increases as it observes consistent patterns from the measured data that confirm its model, leading to higher feedrates.

Note that the proposed approach differs in a few ways from the traditional ILC for servo error compensation using part-to-part learning [1–7]. First, even when the geometric path is identical from iteration to iteration, the commanded feedrate is not. Therefore, the motion trajectory changes from iteration to iteration, which could trigger unmodeled speed-dependent dynamics. Second, the potential for iteration-to-iteration variance in servo dynamics necessitates the incorporation of uncertainty into the model to avoid overconfidence in its predictions until a consistent pattern of contour errors is observed and learned. This is in some ways similar to the safe learning approach adopted by Rattunde et al. [8] using Gaussian process regression.

2.2. Data-driven contour error model

When machining similar parts, one expects that the contour errors ϵ of the parts have similar trends with respect to the path variable s. Accordingly, the data-driven contour error model f_ϵ is assumed to be a function of s that is dependent on the features $\boldsymbol{\phi}$ representing feedrate and part geometry, i.e.,

$$\hat{\epsilon}(s) = f_\epsilon(\boldsymbol{\phi}; s) \tag{1}$$

where $\hat{\epsilon}$ (representing the predicted contour errors) can be discretized with sampled path variables s_i, $i = 0, 1, 2, \ldots, N$, and downsampled using the B-spline basis functions $\boldsymbol{\psi}_j$, $j = 0, 1, 2, \ldots, m$, as

$$\hat{\boldsymbol{\epsilon}} = \left[\hat{\epsilon}(s_0) \vdots \hat{\epsilon}(s_N)\right] = \left[\hat{\epsilon}_0 \vdots \hat{\epsilon}_N\right] = \sum_{j=0}^{m} \boldsymbol{\psi}_j \gamma_j = \Psi \boldsymbol{\gamma} \tag{2}$$

Ψ is the concatenated matrix of $\boldsymbol{\psi}_j$, and $\boldsymbol{\gamma}$ is the vector of the coefficients for the basis functions. Therefore, the data-driven model of Eq. (1) reduces to creating a data-driven model for each coefficient γ_j that captures its variation from part to part:

$$\gamma_j = f_{\gamma_j}(\boldsymbol{\varphi}_j) \tag{3}$$

where $\boldsymbol{\varphi}_j$ is the feature vector that is used for the prediction of γ_j, the j-th element of $\boldsymbol{\gamma}$, and is a subset of $\boldsymbol{\phi}$ in Eq. (1). We opt to define f_{γ_j} using a Bayesian linear regression model of the form:

$$\gamma_j = \boldsymbol{\varphi}_j^T \boldsymbol{\beta}_j + \eta \tag{4}$$

where $\boldsymbol{\beta}_j$ is a vector of stochastic weights to be learned from measured contour errors, while the deterministic feature vector $\boldsymbol{\varphi}_j$, is designed as

$$\boldsymbol{\varphi}_j^T = [\dot{s}_j \quad h \quad 1] \tag{5}$$

and η is the error term consisting of measurement noise and unmodeled dynamics. In Eq. (5), $\dot{s}_j$ is the feedrate-related feature, h is the geometry-related feature, represented by the height of the part, and 1 is the bias term.

To implement the Bayesian linear regression for the prediction of the contour error, the prior of the model weights $\boldsymbol{\beta}_j$ are predetermined as $\boldsymbol{\beta}_{j,0} \sim \mathcal{N}(\boldsymbol{\mu}_{\beta_j,0}, \Sigma_{\beta_j,0})$, where $\boldsymbol{\mu}_{\beta_j,0}$ and $\Sigma_{\beta_j,0}$ are the prior mean and prior variance, respectively. The error term η is also assumed to be $\eta \sim \mathcal{N}(0, \sigma_\eta^2)$, which is independent and identically distributed. Given a new measurement of the contour error ϵ from a machined part, the corresponding ground truth $\boldsymbol{\gamma}$ is first obtained by the pseudoinverse $\boldsymbol{\gamma} = \Psi^\dagger \boldsymbol{\epsilon}$. Then, each element γ_j is used to derive the posterior distribution of the corresponding model parameter $\boldsymbol{\beta}_{j,1} \sim \mathcal{N}(\boldsymbol{\mu}_{\beta_j,1}, \Sigma_{\beta_j,1})$ as follows:

$$\begin{aligned} \Sigma_{\beta_j,1} &= \left(\Sigma_{\beta_j,0}^{-1} + \frac{1}{\sigma_\eta^2}\boldsymbol{\varphi}_j \boldsymbol{\varphi}_j^T\right) \\ \boldsymbol{\mu}_{\beta_j,1} &= \Sigma_{\beta_j,1}\left(\Sigma_{\beta_j,0}^{-1}\boldsymbol{\mu}_{\beta_j,0} + \frac{1}{\sigma_\eta^2}\boldsymbol{\varphi}_j \gamma_j\right) \end{aligned} \tag{6}$$

Finally, the predicted $\hat{\gamma}_j \sim \mathcal{N}(\mu_{\gamma_j,1}, \sigma_{\gamma_j,1}^2)$, given the posterior distribution of $\boldsymbol{\beta}_j$, can be computed as

$$\begin{aligned} \mu_{\gamma_j,1} &= \boldsymbol{\varphi}_j^T \boldsymbol{\mu}_{\beta_j,1} \\ \sigma_{\gamma_j,1}^2 &= \boldsymbol{\varphi}_j^T \Sigma_{\beta_j,1} \boldsymbol{\varphi}_j + \sigma_\eta^2 \end{aligned} \tag{7}$$

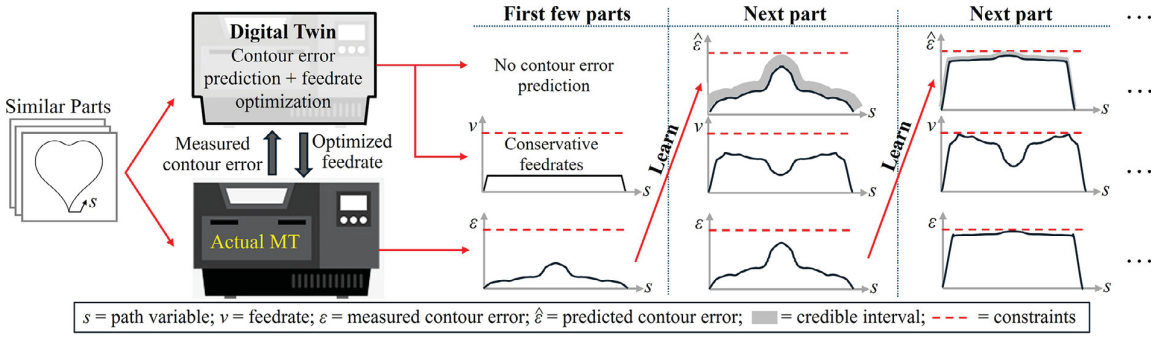

Fig. 1. Overview of proposed approach for feedrate optimization using part-to-part learning in repeated machining.

After determining the predicted mean and the credible intervals of $\hat{\gamma}_j$, defined as $\mu_{\gamma_i,1} \pm 3\sigma_{\gamma_i,1}$ in this paper, the mean and credible intervals of the predicted contour error can then be computed as

$$\hat{\boldsymbol{\epsilon}} \pm \delta\hat{\boldsymbol{\epsilon}} = \Psi\left(\boldsymbol{\mu}_{\boldsymbol{\gamma},1} \pm 3\boldsymbol{\sigma}_{\boldsymbol{\gamma},1}\right) \tag{8}$$

2.3. Determination of hyperparameters

The performance of Bayesian linear regression is highly affected by its hyperparameters, i.e., $\boldsymbol{\mu}_{\boldsymbol{\beta}_j,0}$, $\Sigma_{\boldsymbol{\beta}_j,0}$, and σ_η^2. Here, we provide guidelines for effectively determining the hyperparameters. We start by assuming that $\boldsymbol{\mu}_{\boldsymbol{\beta}_j,0} = \mathbf{0}$ following the ridge regression process. Accordingly, $\Sigma_{\boldsymbol{\beta}_j,0}$ and σ_η^2 are tuned using the contour error data gathered from the first part. Specifically, after collecting contour error measurements from the first part, a set of randomly chosen $\Sigma_{\boldsymbol{\beta}_j,0}$ are applied to Eqs. (6)-(7) to get $\boldsymbol{\mu}_{\boldsymbol{\beta}_j,1}$ and the residual errors $\eta = \gamma_j - \boldsymbol{\varphi}_j^T \boldsymbol{\mu}_{\boldsymbol{\beta}_j,1}$. This process is repeated several times on the same set of initial contour error data. Assuming that each iteration for calculating $\boldsymbol{\mu}_{\boldsymbol{\beta}_j,1}$ and η converges, and is bias free, η can be considered to be the distribution of the measurement noise and the unmodeled dynamics. Therefore, the root mean square of η is used as σ_η. Finally, given any $\boldsymbol{\mu}_{\boldsymbol{\beta}_j,1}$, to determine $\Sigma_{\boldsymbol{\beta}_j,0}$, we assume that $\alpha\%$ of the data population is within the range of the prediction. Accordingly, $\Sigma_{\boldsymbol{\beta}_j,0} = \sigma_{\beta,0}^2$, with $\sigma_{\beta,0}$ is obtained by

$$\mathrm{erf}\left(\frac{\max\{\boldsymbol{\mu}_{\boldsymbol{\beta}_j,1}\}}{\sigma_{\beta,0}\sqrt{2}}\right) = 1 - 0.01\alpha \tag{9}$$

where erf is the error function.

2.4. Contour-error-constrained feedrate optimization

The contour errors and credible intervals predicted from the data-driven model from prior parts are used for the feedrate optimization for the next part. The feedrate optimization is achieved using a sequential linear programming (SLP) method presented in [21]. It formulates feedrate optimization by maximizing a normalized path variable $s(k)$, $0 \le s(k) \le 1$, which represents the distance travelled at each time step k. Constraints are applied to the maximum feedrate $V_{\max}$, acceleration $A_{\max}$, jerk $J_{\max}$, and contour errors $E_{\max}$, i.e.,

$$\begin{aligned} &min_{\boldsymbol{s}} \sum_{k=0}^{K-1} -s(k) = -\mathbf{1}^T\boldsymbol{s} \\ &\text{Subject to :} \\ &s(0) = 0,\ s(K-1) = 1 \\ &s(k-1) \le s(k) \\ &0 \le L\frac{D[\boldsymbol{s}]}{T_s} \le V_{\max} \qquad (\boldsymbol{s}) \le E_{\max} \\ &\left|\frac{D^2[\boldsymbol{x}_d(\boldsymbol{s})]}{T_s^2}\right|, \left|\frac{D^2[\boldsymbol{y}_d(\boldsymbol{s})]}{T_s^2}\right|, \left|L\frac{D^2[\boldsymbol{s}]}{T_s^2}\right| \le A_{\max} \\ &\left|\frac{D^3[\boldsymbol{x}_d(\boldsymbol{s})]}{T_s^3}\right|, \left|\frac{D^3[\boldsymbol{y}_d(\boldsymbol{s})]}{T_s^3}\right|, \left|L\frac{D^3[\boldsymbol{s}]}{T_s^3}\right| \le J_{\max} \\ &\hat{\boldsymbol{\epsilon}} \end{aligned} \tag{10}$$

where $\mathbf{1}$ is a vector with all elements equal to 1, K is the maximum number of timesteps, L is the path length of the entire part, $\boldsymbol{x}_d(s)$ and $\boldsymbol{y}_d(s)$ are the x- and y-axes reference commands, which are functions of s, while D, D^2 and D^3 are the 1st, 2nd and 3rd-order time derivative operators. Note that, contrary to the approach in [21], the vector size of $\boldsymbol{\epsilon}(s)$ can be different from those of $\boldsymbol{x}_d(s)$ and $\boldsymbol{y}_d(s)$. The reference $\boldsymbol{x}_d(s)$ and $\boldsymbol{y}_d(s)$ need to be discretized at timesteps k due to the calculation of feedrate and higher order time derivative terms. However, for $\boldsymbol{\epsilon}(s)$, one can choose any discretization points of interest, e.g., based on the geometry, by adjusting the B-spline matrix Ψ in Eq. (2).

Note that $\boldsymbol{x}_d(s)$ and $\boldsymbol{y}_d(s)$ are typically nonlinear functions of s; thus their derivatives need to be linearized about the equilibrium points s_{eq}. For the double derivative of x-axis, it can be written as

$$\overline{\ddot{\boldsymbol{x}}}_d(k) = \ddot{\boldsymbol{x}}_d(s_{eq}(k)) + \left.\frac{\partial \ddot{\boldsymbol{x}}_d}{\partial s}\right|_{s_{eq}} \cdot \left(s(k) - s_{eq}(k)\right) \tag{11}$$

A similar equation applies to the y-axis as well as the x- and y-axes third-order derivative terms but is omitted here for brevity. The predicted contour error $\hat{\boldsymbol{\epsilon}}(s)$ is also a nonlinear function of s, due to the nonlinear relationship between s and s. Thus, $\boldsymbol{\epsilon}(s)$ also needs to be linearized using s_{eq}, similar to Eq. (11). SLP addresses the use of linearization to approximate nonlinear problems by substituting the solution of s from the last iteration as s_{eq} in the next iteration of linear programming. It repeats the process until convergence between s and s_{eq} is obtained. Hence, to avoid large changes in the solution from one iteration to the other, which can make SLP infeasible, the constraint below is added to Eq. (10):

$$|\boldsymbol{s} - \boldsymbol{s}_{eq}| \le \boldsymbol{\delta}_s \tag{12}$$

where $\boldsymbol{\delta}_s$ is a user defined maximum step size. Lastly, to further reduce the dimension of the SLP, $\boldsymbol{s}$ can further be fitted by another B-spline matrix Ψ_{dn}, i.e.,

$$\boldsymbol{s} = \Psi_{dn}\boldsymbol{p}_s \tag{13}$$

where $\boldsymbol{p}_s$ is the coefficient vector having fewer elements than $\boldsymbol{s}$.

3. Experimental validation

3.1. Experimental setup

To validate the proposed method presented in Section 2, an experimental air-cutting case study is performed on a desktop CNC MT (Carbide 3D Nomad 3) used in [19]. The three axes of the MT are actuated by stepper motors. Two Renishaw RKLC20-S optical linear encoders (with 5 μm resolution) are externally attached to the x- and y-axes to measure the servo position. To collect data for training of the data-driven contour error model and to apply the command from feedrate optimization to the motors, a dSPACE 1007 real-time control board, running at 1 kHz sampling frequency, along with DRV8825 stepper motor drivers (driven at 40 kHz), are used to bypass the stock controller board on the Nomad 3. As shown in Fig. 2 the dSPACE control board is also used to implement an integral feedback controller with $K_i = 50$ to compensate the steady state errors of the stepper motor (i.e., open-loop) controlled motion, where e_x and e_y are respectively the x- and y-axes tracking errors, u_x and u_y are respectively the x- and y-axes motor commands, and z is the discrete time forward shift operator.

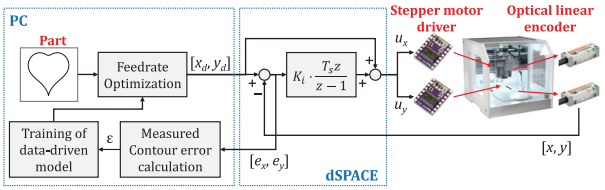

Fig. 2. Experimental setup of the desktop CNC machine tool.

3.2. Experimental validation

The heart shape trajectory shown in Fig. 1 in three different sizes – small (S), medium (M), and large (L) – are used as the desired geometry, where size M and size L are 1.5x and 2x of size S, which has a height, h = 13.6 mm. The contour error constraint is set as $E_{\max}$ = 25 μm, and the kinematic constraints are given by

$$V_{\max} = 30\ mm/s,\ A_{\max} = 1.0\ m/s^2,\ J_{\max} = 50\ m/s^3 \tag{14}$$

The model is initialized using the first two parts of size S (i.e., parts S1 and S2) using conservative kinematic limits of

$$\begin{aligned} &\text{S1}: V_{\max} = 10\ mm/s,\ A_{\max} = 0.3\ m/s^2,\ J_{\max} = 10\ m/s^3 \\ &\text{S2}: V_{\max} = 12\ mm/s,\ A_{\max} = 0.4\ m/s^2,\ J_{\max} = 15\ m/s^3 \end{aligned} \tag{15}$$

Data from two parts are needed to initialize the contour error model along the speed (s_i) component of the model because it is a linear regression, meaning that it needs at least two data points to build a model. Once the feedrate optimization converges for size S, the iteration continues for size M and then size L (i.e., the sequence is given by S1-S8, then M1-M8 and lastly L1-L8). Note that part M1 uses the same kinematic limits as part S1 to initialize the data-driven model along the size (h) component of the model.

For the contour error model, 361 sampled path variables s_i ($N = 361$) and 77 B-spline basis functions $\boldsymbol{\psi}_j$ ($m = 77$), both of which are not uniformly distributed about s but specially selected to emphasize the parts' geometric features, are used. For the feedrate optimization settings, K = 5001 timesteps or the length of $\boldsymbol{s}_{eq}$ from the prior SLP iteration if it is larger than 5001. The variable $\boldsymbol{\delta}_s$ is linearly spaced from

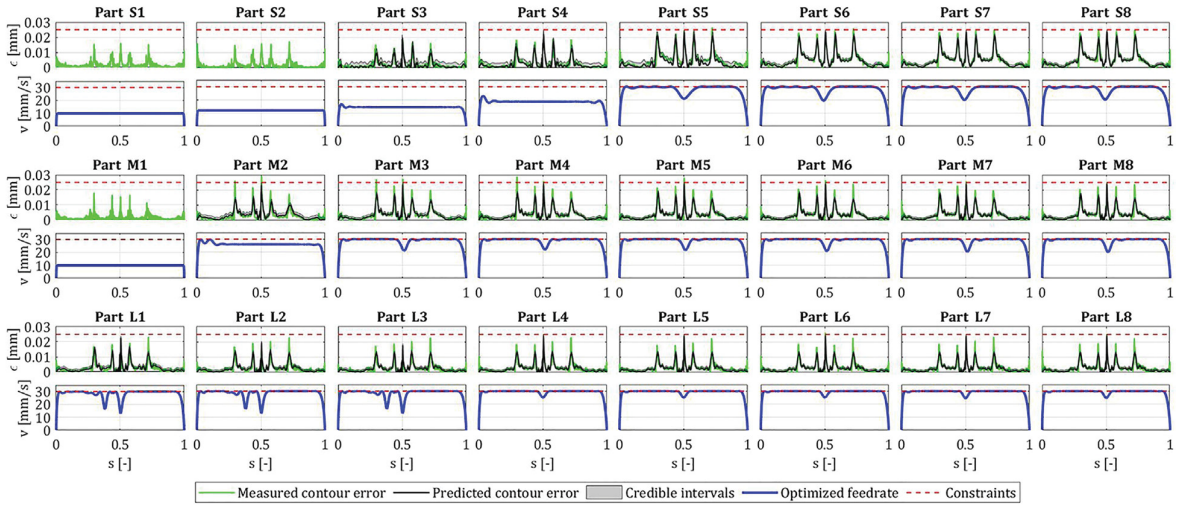

Fig. 3. The optimized feedrate, the corresponding contour errors, and the credible intervals predicted by the data-driven model.

0 to 0.1. The stopping criterion for the SLP is when the RMS value of $s - s_{eq} \leq 10^{-6}$ or, if the former cannot be satisfied, 21 SLP iterations. Then, among the feasible solutions of all the SLP iterations, the one that gives the minimum travel time is selected. For hyperparameter tuning, 100 random values of $\Sigma_{\beta_j,0}$ are trained for 100 iterations to get the distribution of η used to compute σ_η. The variable α in Eq. (9) is selected as 99%.

Fig. 3 shows the optimized feedrate and the corresponding contour error profiles for the parts based on air cutting. One observes that for size S, after initialization using S1 and S2, it takes three more machining iterations (parts) for the feedrate to converge to the optimum. For size M, although the first part (M1) needs to be used for initialization via a conservative feedrate, thanks to the data-driven model developed from training the S-sized parts, the feedrate for the M-sized parts can be quickly optimized starting from part M2. The situation is even better for size L, where a close-to-optimal (not optimal due to linearization difficulty) feedrate profile is obtained from part L1 due to prior training on sizes S and M. These show the benefit of part-to-part learning in rapidly optimizing feedrate. The summary of the travel time for each part can be found in Table 1 and Fig. 4.

Table 1
Summary of the travel time for all parts [unit: s].

Part	1	2	3	4	5	6	7	8
S	4.22	3.52	3.07	2.46	1.67	1.67	1.67	1.67
M	6.29	2.57	2.37	2.37	2.37	2.37	2.37	2.37
L	3.17	3.17	3.17	3.06	3.06	3.06	3.06	3.06

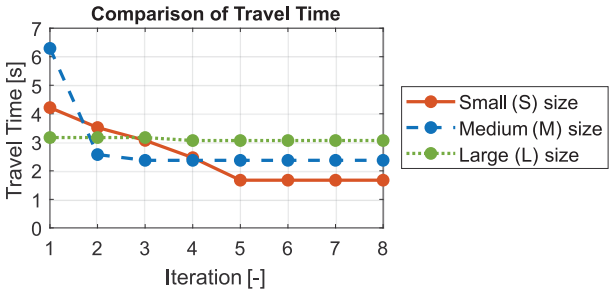

Fig. 4. Evolution of travel time for different part sizes.

Further looking into the contour error predicted by the proposed model, one can also find that the prediction and the measurements are close to each other, except some peaks which occur at quadrant glitches that are not captured due to downsamping using Eq. (2). As a result, the contour error is successfully constrained except for a few minor violations (e.g., in parts M2 and M3). In general, the upper bound of the credible intervals hits the error constraints, which limits the feedrate at the corresponding locations. Interestingly, one can notice that for part S3, the credible intervals are far lower than the error limits. This is caused by the linearization errors in the SLP optimization algorithm.

4. Conclusion and future work

This paper has proposed a framework for a data-driven model to learn contour errors from part to part in repeated machining, and to use the model for feedrate optimization during each iteration. The data-driven model is based on Bayesian linear regression. This allows the model to be trained on very few samples, and its confidence, hence, the optimized feedrate, to increase as more data is gathered. Air cutting experiments are used to show that feedrate profiles can be optimized within a few iterations of one part size and carried over to facilitate the optimization of other part sizes.

There are several avenues for future work. Physics-based servo dynamics models combined with cutting force and friction models could be incorporated for faster and more accurate learning. Cutting experiments will be conducted to further validate the proposed approach. The framework could be extended to part-to-part learning for parts of non-similar geometries, or machine-to-machine learning, where a fleet of MTs cutting similar parts learn from one another to enhance overall outcomes.

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.

CRediT authorship contribution statement

Cheng-Hao Chou: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Chenhui Shao:** Writing – review & editing, Methodology, Conceptualization. **Chinedum E. Okwudire:** Writing – review & editing, Writing – original draft, Visualization, Supervision, Resources, Project administration, Methodology, Funding acquisition, Conceptualization.

Acknowledgments

This work is partially funded by the National Science Foundation grant #2054715.

References

[1] Bristow DA, Tharayil M, Alleyne AG (2006) A survey of iterative learning control. *IEEE Control Systems Magazine* 26(3):96–114.
[2] Lo CC, Hsiao CY (1998) CNC machine tool interpolator with path compensation for repeated contour machining. *Computer-Aided Design* 30(1):55–62.
[3] Barton KL, Alleyne AG (2008) A cross-coupled iterative learning control design for precision motion control. *IEEE Transactions on Control Systems Technology* 16 (6):1218–1231.
[4] Hendrawan YM, Simba KR, Uchiyama N (2018) Iterative learning based trajectory generation for machine tool feed drive systems. *Robotics and Computer-Integrated Manufacturing* 51:230–237.
[5] Bolder J, Oomen T (2014) Rational basis functions in iterative learning control—With experimental verification on a motion system. *IEEE Transactions on Control Systems Technology* 23(2):722–729.
[6] Dumanli A, Sencer B (2019) Pre-compensation of servo tracking errors through data-based reference trajectory modification. *CIRP Annals* 68(1):397–400.
[7] Bahtiyar K, Sencer B, Beudaert X (2024) Data-driven feedforward control of inertial dampers for accuracy improvement. *CIRP Annals* 73(1):317–320.
[8] Rattunde L, Laptev I, Klenske ED, Möhring HC (2021) Safe optimization for feedrate scheduling of power-constrained milling processes by using Gaussian processes. *Procedia CIRP* 99:127–132.
[9] Altintas Y, Erkorkmaz K (2003) Feedrate optimization for spline interpolation in high speed machine tools. *CIRP Annals* 52(1):297–302.
[10] Dong J, Ferreira PM, Stori JA (2007) Feed-rate optimization with jerk constraints for generating minimum-time trajectories. *International Journal of Machine Tools and Manufacture* 47(12–13):1941–1955.
[11] Erkorkmaz K, Chen QGC, Zhao MY, Beudaert X, Gao XS (2017) Linear programming and windowing based feedrate optimization for spline toolpaths. *CIRP Annals* 66(1):393–396.
[12] Erkorkmaz K, Layegh SE, Lazoglu I, Erdim H (2013) Feedrate optimization for freeform milling considering constraints from the feed drive system and process mechanics. *CIRP Annals* 62(1):395–398.
[13] Fan W, Gao XS, Lee CH, Zhang K, Zhang Q (2013) Time-optimal interpolation for five-axis CNC machining along parametric tool path based on linear programming. *The International Journal of Advanced Manufacturing Technology* 69:1373–1388.
[14] Chen M, Xu J, Sun Y (2017) Adaptive feedrate planning for continuous parametric tool path with confined contour error and axis jerks. *The International Journal of Advanced Manufacturing Technology* 89:1113–1125.
[15] Dong J, Stori JA (2007) Optimal feed-rate scheduling for high-speed contouring. *Journal of Manufacturing Science and Engineering* 129(1):63–76.
[16] Kim H, Okwudire CE (2020) Simultaneous servo error pre-compensation and feedrate optimization with tolerance constraints using linear programming. *The International Journal of Advanced Manufacturing Technology* 109:809–821.
[17] Balula S, Liao-McPherson D, Rupenyan A, Lygeros J (2024) Data-driven reference trajectory optimization for precision motion systems. *Control Engineering Practice* 144:105834.
[18] Chang YC, Chen CW, Tsao TC (2018) Near time-optimal real-time path following under error tolerance and system constraints. *Journal of Dynamic Systems, Measurement, and Control* 140(7):071004.
[19] Kim H, Okwudire CE (2023) Intelligent feedrate optimization using a physics-based and data-driven digital twin. *CIRP Annals* 72(1):325–328.
[20] Kim H, Kontar RA, Okwudire CE (2024) Intelligent feedrate optimization using an uncertainty-aware digital twin within a model predictive control framework. *IEEE Access* 12:49947–49961.
[21] Kim H, Okwudire CE (2021) 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):2069–2082.