

# Data-Driven Extrusion Force Control Tuning for 3D Printing

Xavier Guidetti<sup>1,2</sup>, Ankita Mukne<sup>1</sup>, Marvin Rueppel<sup>2</sup>, Yannick Nagel<sup>3</sup>, Efe C. Balta<sup>1,2</sup>, and John Lygeros<sup>1</sup>

**Abstract**—The quality of 3D prints often varies due to different conditions inherent to each print, such as filament type, print speed, and nozzle size. Closed-loop process control methods improve the accuracy and repeatability of 3D prints. However, optimal tuning of controllers for given process parameters and design geometry is often a challenge with manually tuned controllers resulting in inconsistent and suboptimal results. This work employs Bayesian optimization to identify the optimal controller parameters. Additionally, we explore transfer learning in the context of 3D printing by leveraging prior information from past trials. By integrating optimized extrusion force control and transfer learning, we provide a novel framework for closed-loop 3D printing and propose an automated calibration routine that produces high-quality prints for a desired combination of print settings, material, and shape.

## I. INTRODUCTION

Additive Manufacturing (AM), often known as 3D printing, enables manufacturing of complex geometries using a layerwise bottom-up process methodology. A popular method for AM in practice is Fused Filament Fabrication (FFF) with polymers. FFF uses a heated extruder system to melt polymer material and print it in the form of ellipsoidal beads in a layerwise fashion. The recent development of high-performance technical polymers allows FFF printed parts to have high mechanical strength and stiffness, expanding their use in a wide range of application areas [1]. Additionally, fiber-reinforced polymer printing is commonly employed in various advanced manufacturing applications [2].

A critical challenge in FFF is ensuring the reliability and repeatability of the process under changing process conditions. On most current applications, the challenge is partially caused by a lack of in-situ sensors, control-oriented models, and closed-loop control strategies [3]. A possible approach to improve performance is through modeling printed bead outputs as a function of input parameters [4], [5]. Similarly, extrusion dynamics models can be used for feedforward input optimization to improve dimensional accuracy [6]. Even though open-loop model-based approaches are powerful and beneficial for improving process performance, model inaccuracies and run time disturbances remain a challenge. As a result, in-situ measurement of run-time performance for monitoring and control is necessary for high-performance AM in practice.

In-situ monitoring for FFF processes has received attention in the recent literature [7]. Applications of health

management [8], quality monitoring [9], and anomaly detection [10] have been successfully developed for FFF using in-situ measurements. Meanwhile, process control received comparatively less attention. An effective strategy for closed-loop control is using measurements of the printed part between layers. Layer-to-layer measurements of FFF prints have been used for accurate process modeling [3] and parameter optimization [11]. However, layer-to-layer updates are ineffective for rejecting fast-acting run-time disturbances to the extrusion flow in the layer, which has a great influence on the resulting printed material shape [4], [12], [13]. Recent developments consider in-layer measurements of the material flow within the extruder [14]. Such measurements provide valuable run-time feedback and can be used for closed-loop process control. In this work, we characterize the extrusion flow through feedback force measurements and develop a novel in-layer closed-loop control framework for FFF.

Controlling the extrusion force enables an effective way to build controllers that ensure desired flow characteristics. However, developing and tuning such controllers is challenging due to the complex characteristics of the extrusion process coupled with the printing dynamics. Manually tuned controllers cannot guarantee optimal results. Additionally, the nonlinear process dynamics are difficult to model for real-time model-based control. In this work, we employ Bayesian Optimization, a data-driven optimization method, to find optimal control parameters for Force Controlled Printing [15] for FFF and present effective Transfer Learning strategies for knowledge transfer between optimal tunings. The main contributions of this work are:

- The development of a preliminary Force Controlled Printing framework for FFF;
- A continuous Bayesian Optimization method for controller parameter tuning during a print process;
- The use of Transfer Learning on Bayesian Optimization to accelerate convergence rates.

We experimentally demonstrate our results on a custom-built setup to showcase the effectiveness. Our work provides a baseline for future developments in force-controlled FFF.

## II. BACKGROUND

### A. Force Controlled Fused Filament Fabrication

In a conventional FFF extruder, a thermoplastic filament is pushed by a driving gear into a heated nozzle. The extruder is moved over the build plane where the material exiting the nozzle is deposited to form a part. The volume of extruded material is pre-computed based on the filament size, the nozzle diameter, and the extruder motion [5]. In Force Controlled

Research supported by Innosuisse (project №102.617 IP-ENG) and by the Swiss National Science Foundation under NCCR Automation (grant №180545). <sup>1</sup>Automatic Control Laboratory (IfA), ETH Zürich, 8092 Zürich, Switzerland. <sup>2</sup>Control and Automation Group, Inspire AG, 8005 Zürich, Switzerland. <sup>3</sup>NematX AG, 8093 Zürich, Switzerland.  
Corresponding author: E. C. Balta, efe.balta@inspire.ch.

Printing (FCP) for FFF, a sensor is installed in the extruder assembly to measure the force applied onto the filament while extruding it through the nozzle during printing. Unlike the conventional approach, where the filament driver speed is predefined, in FCP the driver speed is continuously adapted by a controller to maintain the measured driving force at a desired and constant reference value (see Fig. 1). The printing performance of this technique strongly depends on the controller’s performance. PID controllers [16] can be used for this application, but only a well-tuned PID controller will properly regulate the extrusion process while dealing with external disturbances.

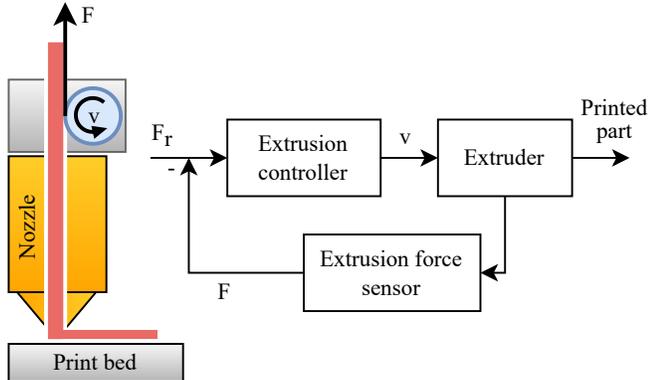


Fig. 1: Simplified diagram of the extrusion process and feedback loop for FCP for FFF

## B. Bayesian Optimization

BO is a data-driven iterative optimization method, used to optimize expensive-to-evaluate black-box functions that can be sampled only in a point-wise fashion. It is best suited for optimization over continuous domains and tolerates stochastic noise in function evaluations [17], [18]. BO has been successfully used in a variety of applications such as machine learning [19], robotics [20], control [21], manufacturing [22], and more; [11] uses BO to tune the printing process parameters of FFF.

Consider a problem of optimizing an unknown objective  $f : \mathcal{X} \rightarrow \mathbb{R}$  over a set of inputs  $\mathcal{X} \subset \mathbb{R}^d$ :

$$\min_{x \in \mathcal{X}} f(x). \quad (1)$$

In each iteration we observe a noisy evaluation  $y_t = f(x_t) + \varphi_t$  for the chosen input  $x_t \in \mathcal{X}$ , where  $\varphi_t \sim \mathcal{N}(0, \rho^2)$  is zero-mean noise, independent across different time steps  $t$ . We collect the past input-output data in the set  $D_t = \{(x_i, y_i)\}_{i=1}^t$ . Using the collected data in  $D_t$ , BO uses a Gaussian Process (GP) [23] to model the unknown function and an acquisition function to choose subsequent samples.

GPs are commonly used in non-parametric modeling as they provide a distribution over functions from limited data. Let  $\mu_t(\cdot)$  and  $\sigma_t^2(\cdot)$ , denote the posterior mean and variance, respectively, given the previous measurements  $\mathbf{y}_t =$

$[y_1, \dots, y_t]^\top$  and kernel  $k(\cdot, \cdot)$ . These are computed by

$$\mu_t(x) = k_t(x)^T (K_t + \rho^2 I)^{-1} \mathbf{y}_t, \quad (2)$$

$$\sigma_t^2(x) = k(x, x) - k_t(x)^\top (K_t + \rho^2 I)^{-1} k_t(x), \quad (3)$$

where  $(K_t)_{i,j} = k(x_i, x_j)$ ,  $k_t(\cdot)^T = [k(x_1, \cdot), \dots, k(x_t, \cdot)]^T$ .

The acquisition function  $\text{acq}_t : \mathcal{X} \rightarrow \mathbb{R}$  is used for understanding how informative an input is for finding the minimizer of (1) given the GP model of  $f$ . A good acquisition function effectively trades off exploration of new uncertain inputs with the exploitation of observed data that leads to better iterates with lower cost. Algorithm 1 provides the standard BO loop.

---

### Algorithm 1 Bayesian Optimization Loop

---

- 1: **Initialize:** Prior  $f \sim GP(0, k)$ ,  $\bar{x} = 0$
  - 2: **for**  $t = 1, \dots, T$  **do**
  - 3:    $x_t \leftarrow \text{argmin}\{\text{acq}_t(x) \mid x \in \mathcal{X}\}$
  - 4:   Observe  $y_t \leftarrow f(x_t) + \varphi_t$
  - 5:   Update the GP posterior (2)-(3) using  $y_t$
  - 6:   **if**  $f(x_t) < f(\bar{x})$  **then**  $\bar{x} \leftarrow x_t$
  - 7: **Output:**  $\bar{x}$
- 

A stopping condition (such as a maximum number of iterations) is commonly used to interrupt the BO. One of the key strengths of BO is its efficiency in finding the optimal solution with a relatively small number of evaluations.

## C. Transfer Learning

Transfer learning (TL) is a machine learning technique in which any knowledge learned on one task is reused to improve performance on a secondary related task. For example, TL is very popular in deep learning, where pre-trained models are used to warm start the training process of a specialized model [24]. TL improves or accelerates the understanding of the current task by relating it to other tasks performed at different periods through a related source domain [25]. TL can be integrated with Bayesian Optimization, by modifying the surrogate model, allowing the outcomes of one optimization task to inform the optimization process of a related task. TL has been used in FFF by [26], where it was combined with a convolutional neural network for surface feature prediction. In our framework, we use TL to accelerate the convergence of BO on new tasks.

## III. PROPOSED FRAMEWORK

### A. Controller Structure

We implement a modified PID controller to regulate the extrusion force. The proportional and integral gains correspond to a common PID structure. However, we modify the derivative part to include two derivative terms. These two terms are designed to compute numerical derivatives of the force measurements at different time resolutions:

- a) Fast (more responsive) derivative accounting for the faster machine motion system with a small time constant;
- b) Slow (more damped) derivative accounting for the slower filament extrusion system, having a larger time constant.

By processing the measurements at the two different time constants, we are able to control the printing process more effectively. Ultimately, our PID controller has four tunable gains:  $K_p$  (the proportional gain),  $K_i$  (the integral gain),  $K_d$  (the fast derivative), and  $K_{dd}$  (the slow derivative).

### B. Continuous Bayesian Optimization

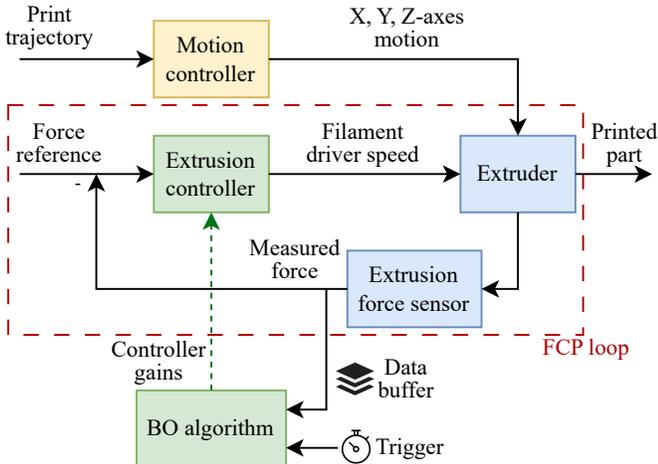


Fig. 2: Flowchart of the continuous Bayesian optimization method

We implement BO (see Algorithm 1) to find the optimal gains of the PID controller regulating the extrusion force. Specifically, to learn the optimal controller for a specific geometry, filament, printing temperature, and nozzle we conduct the BO routine *while* printing an exact copy of the desired part. Unlike most BO applications – where each iteration corresponds to one experiment, such as printing an entire part – we define each iteration as a 10 s long controller deployment executed while printing a part. As manufacturing a part by FFF is a time-consuming process, we can conduct a large number of BO iterations during one print. We call this approach *continuous BO*.

In practice, the process, which we illustrate in Fig. 2, works as follows:

- 1) The printing instructions for a desired geometry (G-Code) are executed, and the printer head moves along a predefined trajectory;
- 2) Extrusion control is initialized by setting the desired force reference and a random initial controller;
- 3) The software records 10 s of printing data with the given controller; the BO routine evaluates the printing performance and immediately deploys a new controller; this point is repeated until completion of the printing instructions or until user interruption.

This procedure continuously employs Algorithm 1 on an FFF system in a single print. The system keeps printing with no interruption during the time needed in between iterations to evaluate the controller performance, update the surrogate models, run the acquisition function, and deploy the next controller. This means that discovering the optimal

controller for a given geometry is achieved with one single print requiring the same printing time as the final part. By the end of a single 10 min experiment, 60 controllers have been tested, and the controller with the best performance will be used to print the desired part in a subsequent print.

Next, we provide the specific formulation of BO used in our framework. The feedback control scheme aims to minimize the error between the measured extrusion force and a desired reference force. Given a reference force  $F_r$  and a sequence of consecutive force measurements  $\{F_i\}_{i=1}^n$ , we compute the force root-mean-square error (RMSE) as

$$\text{RMSE}(\{F_i\}_{i=1}^n, F_r) = \sqrt{\frac{1}{n} \sum_{i=1}^n (F_i - F_r)^2}. \quad (4)$$

We evaluate the performance of a controller by computing the RMSE of the data recorded while the controller is deployed. Since the ideal controller has  $\text{RMSE} = 0$ , we aim to minimize the RMSE. Considering a set of controller gains  $x = \{K_p, K_i, K_d, K_{dd}\}$ , and the force recorded while deploying this controller  $\{F_i(x)\}_{i=1}^n$ , we define the BO black-box objective function as

$$f(x) = \text{RMSE}(\{F_i(x)\}_{i=1}^n, F_r). \quad (5)$$

We select expected improvement (EI) as an acquisition function [27], a popular strategy for information collection which can be computed analytically for a GP surrogate model. EI is defined as

$$\text{EI}(x) = \mathbb{E}[\max\{f(\bar{x}) - f(x) + \xi, 0\}], \quad (6)$$

where  $\bar{x}$  represents the best controller discovered so far and  $\xi$  is a tuning parameter influencing the selections made by the acquisition function. Larger values of  $\xi$  lead to more exploration, while smaller values of  $\xi$  produce more exploitation. By experimenting with the FCP setup, we find a suitable trade-off by setting  $\xi = 1 \times 10^{-3}$ . We bound the inputs by  $0 \leq K_p \leq 50$ ,  $0 \leq K_i \leq 50$ ,  $0 \leq K_d \leq 100$ , and  $0 \leq K_{dd} \leq 100$ . The collected data is modeled using a GP with a Matern kernel having  $\nu = 2.5$ . The hyperparameters of the GP kernel are selected by fitting the GP on the available data and maximizing the log marginal likelihood.

### C. Transfer Learning with Bayesian Optimization

We consider the case where we want to print the same geometry with different reference forces. Naturally, each reference force corresponds to a different optimal controller. The goal of TL is to accelerate and improve the discovery of the optimal controller corresponding to a new reference force by exploiting the data that was previously collected with other reference forces; we refer to the different reference forces as “different tasks”.

To utilize TL in conjunction with BO, we made minor modifications to the formulation of the BO discussed in Section III-B. First, we extend the input vector  $x$  by appending the reference force, leading to  $x = \{K_p, K_i, K_d, K_{dd}, F_r\}$ . Then, to produce a more uniform data set for better task

learning, the objective function is normalized using the force reference to reduce inter-task variations:

$$f(x) = \frac{\text{RMSE}(\{F_i(x)\}_{i=1}^n, F_r)}{F_r}. \quad (7)$$

The acquisition function (EI) remains unchanged.

#### IV. EXPERIMENTAL STUDY

##### A. Hardware Setup and Printing Conditions

Our FCP setup comprises a printer connected to three different sub-systems. The displacement of the extruder in space is controlled by a programmable logic controller (PLC) using Beckhoff<sup>1</sup> software (corresponding to the yellow block in Fig. 2). This unit operates the motors moving the printer head along the three Cartesian axes. The sequence of motions that are executed depends on a printing instructions file loaded into the Beckhoff software. The second unit has a force sensor and a Raspberry Pi board actuating the filament extruder stepper motor (corresponding to the blue blocks in Fig. 2). The last unit is a computer running ROS2 [28] (corresponding to the green blocks in Fig. 2), which closes the feedback loop for FCP and runs the PID controller in run-time. The force sensor embedded in the extruder streams real-time data to the ROS2 unit. Based on the measurements and the control and/or optimization algorithms, ROS2 streams new extrusion values or parameters to the Raspberry Pi, which in turn updates the extruder drive speed.

All experiments were conducted by printing the *tower* shells shown in Fig. 3, hollow cubes with no top or bottom, made of a single and continuous contour line. This geometry was selected as the motion and printing conditions repeat identically throughout the print, making the learning and optimization tasks more effective. Prints were made at an axis speed of 100 mm/s, temperature of 300 °C, layer height of 50 μm, using a 1.75 mm nozzle, and while extruding a liquid crystal polymer produced by NematX AG<sup>2</sup>.

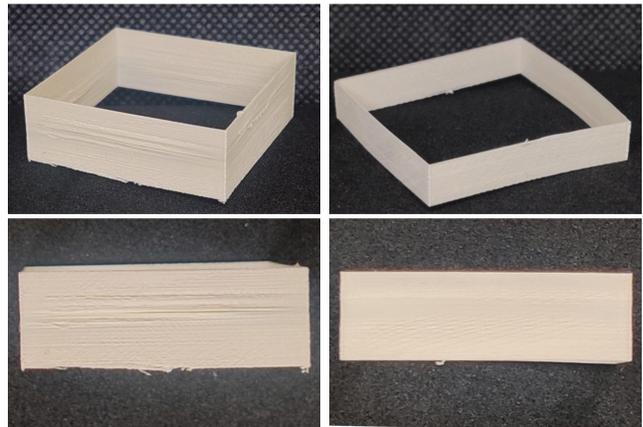
##### B. Results

1) *Single-Task Bayesian Optimization*: The *tower* was printed while running single-task continuous BO in four different experiments, each with a different reference force. The results are summarized in Table I. In all cases, the performance of a controller manually tuned by a practitioner was used to initialize the optimization. Then continuous BO was executed, and the performance of the best-found controller was recorded. The results show a significant reduction in RMSE for all cases, which corresponds to an increase in print quality. This observation can be confirmed qualitatively by analyzing Fig. 3. The figure shows two towers printed under identical conditions at a force reference of 0.3 N before and after controller optimization. In Fig. 3a, it is clearly visible how the side walls of the part have a rough surface. This is because machine’s abrupt velocity and direction changes at the corners of the part disturb the extrusion process. As the manually tuned controller is unable to properly reject these disturbances, numerous defects appear in the *tower* walls.

After BO, however, the extrusion controller performance is drastically increased. In Fig. 3b, the walls show practically no surface defects, since the controller regulates the extrusion process more effectively.

TABLE I: Comparison of controller performance before and after Bayesian optimization

Reference [N]	RMSE before BO [N]	RMSE after BO [N]
0.1	$4.029 \times 10^{-3}$	$3.477 \times 10^{-3}$
0.2	0.289	0.038
0.3	0.284	0.047
0.4	0.200	0.011



(a) Before BO

(b) After BO

Fig. 3: Printed *tower* shells on the experimental setup with force feedback extrusion control. Both parts are continuously printed until user interruption, with force reference 0.3 N.

2) *Transfer Learning*: To study the effect of TL on BO, we conducted a set of experiments where controllers were tuned for a new force reference while having access to data belonging to the optimization conducted previously for another force reference. In particular, we were interested in observing whether the use of TL would produce a controller with better performance and reduce the number of iterations required to optimize the process. In the three experiments we conducted, we *a)* used optimization data from a force reference of 0.4 N to warm-start the optimization for a force reference of 0.3 N, *b)* used optimization data from a force reference of 0.2 N to warm-start the optimization for a force reference of 0.3 N, and *c)* used optimization data from force references of 0.4 N and 0.3 N to warm-start the optimization for a force reference of 0.2 N.

Figures 4a and 4b show the results for experiments *a)* and *b)*, comparing the convergence of the optimization using TL for a target task of reference 0.3 N with the results produced without TL. In both cases, the initial iterations produce worse performance, as commonly observed in TL. The optimization process trusts excessively the structure learned in the training task, which does not necessarily generalize to the target

<sup>1</sup> <https://www.beckhoff.com/> <sup>2</sup> <https://nematx.com/>

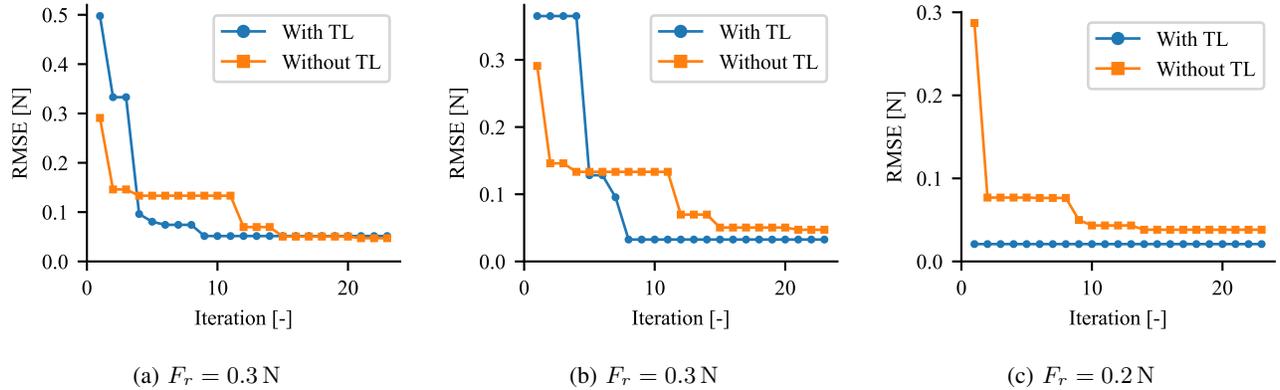


Fig. 4: Comparison of convergence plots from controller optimization for: (a) force reference of 0.3 N with TL using data from the optimization of 0.4 N, (b) force reference of 0.3 N with TL using data from the optimization of 0.2 N, (c) force reference of 0.2 N with TL using data from the optimization of both 0.3 N and 0.4 N.

task. However, a few samples collected in the target task are enough for the TL approach to capitalize on the inter-task similarities, leading to faster convergence. In addition, the performance of the optimal controller is either similar to the one produced without TL (Fig. 4a), or better (Fig. 4b). The main metrics are compared quantitatively in Table II.

TABLE II: Comparison of cost (RMSE) and required iterations between different controller tuning strategies, for a target force reference of 0.3 N

	Best RMSE [N]	Iterations to convergence
Without TL	0.047	21
With TL (from 0.4 N)	0.052	9
With TL (from 0.2 N)	0.033	8

In Fig. 4c we show the results from experiment *c*), where data from optimization for references 0.4 N and 0.3 N are used to warm-start BO for reference 0.2 N. Note that the target task is outside the range of tasks used in the training, requiring the TL to extrapolate. We compare the convergence of the optimizations conducted with and without TL. The results clearly show that the model learned from two training tasks predicts exceptionally well the target task: the optimal controller is discovered at the first iteration and it significantly outperforms the controller discovered by conventional BO. Quantitative metrics are reported in Table III.

TABLE III: Comparison of cost (RMSE) and required iterations between different controller tuning strategies, for a target force reference of 0.2 N

	Best RMSE [N]	Iterations to convergence
Without TL	0.038	14
With TL (from 0.4 N and 0.3 N)	0.020	1

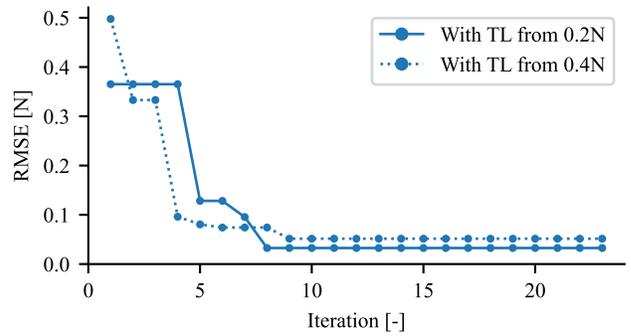


Fig. 5: Comparison of convergence plots from controller optimization for a force reference of 0.3 N

### C. Discussion

Our results in Section IV-B.1 demonstrate how for the force references of 0.1 N, 0.2 N, 0.3 N, and 0.4 N the RMSE of manually tuned controllers was reduced by 13.7%, 86.9%, 83.5%, and 94.5% respectively by using standard BO. This translates directly into better-quality parts. It appears from the results that BO is a powerful method for automated controller tuning in FCP. In the experiment with reference 0.1 N, BO appears produce subpar results, as the RMSE improvement is not as significant as in the other tasks. This is due to the fact that the manually tuned controller used for benchmarking was already producing excellent results, since regulating the extrusion force at low references appears to be relatively simple. Nonetheless, BO was able to efficiently discover a better controller.

The results of Section IV-B.2 show how TL in conjunction with BO can further improve performance, significantly accelerating the optimization and leading to better controllers. The improvements are summarized in Table IV. In only one case, TL produced a marginally worse controller. This was however compensated by a drastic reduction in the number of iterations required for convergence. In Fig. 5 we compare the two cases where TL was conducted with the same target

task (0.3N reference) but with two different training tasks (0.2N and 0.4N). Using data from the 0.2N reference, TL converges marginally faster and to a better controller. We believe this is due to the characteristics of the controllers found in the training data set. In particular, the data set generated with a lower reference (0.2N) contains controllers with lower proportional and integral gains (data not shown). This seems to skew the controller exploration in favor of less aggressive control gains, making the optimization more efficient. Conversely, the optimization using the higher reference (0.4N) data set was observed to test numerous unstable controllers, which affected the optimization negatively.

TABLE IV: Change in cost (RMSE) and iterations produced by different transfer learning strategies, when compared to controller tuning without transfer learning

	Best RMSE	Iterations to convergence
−0.3 N from −0.4 N	10.6 %	−57.1 %
−0.3 N from −0.2 N	−29.8 %	−61.9 %
−0.2 N from −0.4 N and −0.3 N	−47.4 %	−92.8 %

## V. CONCLUSION AND FUTURE WORK

We introduced a framework for force controlled FFF and demonstrated experimentally how continuous BO can be used to efficiently tune extrusion controllers that optimize the printing performance. The method requires a single print and reliably produces high-performance controllers. We have also utilized TL in conjunction with BO and shown that the optimization is significantly faster and leads to better controllers when transferring information from past experiments. Future work focuses on characterizing the mechanical properties of parts printed with optimized FCP and on extending the TL framework to changes in part geometry.

## REFERENCES

- [1] X. Guidetti, E. C. Balta, Y. Nagel, H. Yin, A. Rupenyan, and J. Lygeros, “Stress flow guided non-planar print trajectory optimization for additive manufacturing of anisotropic polymers,” *Additive Manufacturing*, p. 103628, 2023.
- [2] G. D. Goh, Y. L. Yap, S. Agarwala, and W. Y. Yeong, “Recent progress in additive manufacturing of fiber reinforced polymer composite,” *Advanced Materials Technologies*, vol. 4, no. 1, p. 1800271, 2019.
- [3] E. C. Balta, D. M. Tilbury, and K. Barton, “Layer-to-layer stability of linear layerwise spatially varying systems: Applications in fused deposition modeling,” *IEEE Transactions on Control Systems Technology*, vol. 29, no. 6, pp. 2517–2532, 2021.
- [4] E. C. Balta and A. Altinkaynak, “Numerical and experimental analysis of bead cross-sectional geometry in fused filament fabrication,” *Rapid Prototyping Journal*, vol. 28, no. 10, pp. 1882–1894, 2022.
- [5] D. Aksoy, E. C. Balta, D. M. Tilbury, and K. Barton, “A control-oriented model for bead cross-sectional geometry in fused deposition modeling,” in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 474–480.
- [6] P. Wu, C. Qian, and C. E. Okwudire, “Modeling and feedforward control of filament advancement and retraction in material extrusion additive manufacturing,” *Additive Manufacturing*, vol. 78, p. 103850, 2023.

- [7] P. K. Rao, J. Liu, D. Roberson, Z. Kong, and C. Williams, “Online real-time quality monitoring in additive manufacturing processes using heterogeneous sensors,” *Journal of Manufacturing Science and Engineering*, vol. 137, no. 6, p. 061007, 2015.
- [8] J. S. Kim, C. S. Lee, S.-M. Kim, and S. W. Lee, “Development of data-driven in-situ monitoring and diagnosis system of fused deposition modeling (fdm) process based on support vector machine algorithm,” *International Journal of Precision Engineering and Manufacturing-Green Technology*, vol. 5, pp. 479–486, 2018.
- [9] S. R. Chhetri, S. Faezi, A. Canedo, and M. A. A. Faruque, “Quilt: Quality inference from living digital twins in iot-enabled manufacturing systems,” in *Proceedings of the international conference on internet of things design and implementation*, 2019, pp. 237–248.
- [10] E. C. Balta, D. M. Tilbury, and K. Barton, “A digital twin framework for performance monitoring and anomaly detection in fused deposition modeling,” in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2019, pp. 823–829.
- [11] X. Guidetti, M. Kühne, Y. Nagel, E. C. Balta, A. Rupenyan, and J. Lygeros, “Data-driven process optimization of fused filament fabrication based on in situ measurements,” *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 4713–4718, 2023.
- [12] M. P. Serdeczny, R. Comminal, D. B. Pedersen, and J. Spangenberg, “Experimental validation of a numerical model for the strand shape in material extrusion additive manufacturing,” *Additive Manufacturing*, vol. 24, pp. 145–153, 2018.
- [13] —, “Numerical simulations of the mesostructure formation in material extrusion additive manufacturing,” *Additive Manufacturing*, vol. 28, pp. 419–429, 2019.
- [14] T. J. Coogan and D. O. Kazmer, “In-line rheological monitoring of fused deposition modeling,” *Journal of Rheology*, vol. 63, no. 1, pp. 141–155, 2019.
- [15] X. Guidetti, N. Mingard, R. Cruz-Oliver, Y. Nagel, M. Rueppel, A. Rupenyan, E. C. Balta, and J. Lygeros, “Force controlled printing for material extrusion additive manufacturing,” *arXiv*, 2024.
- [16] A. Visioli, *Practical PID control*. Springer Science & Business Media, 2006.
- [17] P. I. Frazier, “A tutorial on bayesian optimization,” *arXiv preprint arXiv:1807.02811*, 2018.
- [18] R. Garnett, *Bayesian optimization*. Cambridge University Press, 2023.
- [19] J. Snoek, H. Larochelle, and R. P. Adams, “Practical bayesian optimization of machine learning algorithms,” *Advances in neural information processing systems*, vol. 25, 2012.
- [20] F. Berkenkamp, A. Krause, and A. P. Schoellig, “Bayesian optimization with safety constraints: safe and automatic parameter tuning in robotics,” *Machine Learning*, vol. 112, no. 10, pp. 3713–3747, 2023.
- [21] M. Khosravi, V. N. Behrunani, P. Myszkowski, R. S. Smith, A. Rupenyan, and J. Lygeros, “Performance-driven cascade controller tuning with bayesian optimization,” *IEEE Transactions on Industrial Electronics*, vol. 69, no. 1, pp. 1032–1042, 2021.
- [22] X. Guidetti, A. Rupenyan, L. Fassl, M. Nabavi, and J. Lygeros, “Advanced manufacturing configuration by sample-efficient batch bayesian optimization,” *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 886–11 893, 2022.
- [23] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006, vol. 2, no. 3.
- [24] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, “A survey on deep transfer learning,” in *Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part III 27*. Springer, 2018, pp. 270–279.
- [25] A. Hosna, E. Merry, J. Gyalmo, Z. Alom, Z. Aung, and M. A. Azim, “Transfer learning: a friendly introduction,” *Journal of Big Data*, vol. 9, no. 1, p. 102, 2022.
- [26] J. Zhu, Z. Su, Q. Wang, Z. Lan, F. S.-f. Chan, Z. Han, Z. Wang, S. W.-f. Wong, and A. C.-f. Ngan, “Surface quality prediction and quantitative evaluation of process parameter effects for 3d printing with transfer learning-enhanced gradient-boosting decision trees,” *Expert Systems with Applications*, vol. 237, p. 121478, 2024.
- [27] D. Zhan and H. Xing, “Expected improvement for expensive optimization: a review,” *Journal of Global Optimization*, vol. 78, no. 3, pp. 507–544, 2020.
- [28] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, “Robot operating system 2: Design, architecture, and uses in the wild,” *Science Robotics*, vol. 7, no. 66, p. eabm6074, 2022.