

**Intelligent Feedrate Optimization using Uncertainty-aware Physics-based and Data-driven
Servo Dynamic Models**

by

Heejin Kim

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Mechanical Engineering)
in the University of Michigan
2023

Doctoral Committee:

Professor Chinedum E. Okwudire, Chair
Professor Shorya Awtar
Professor Kira Barton
Assistant Professor Raed Al Kontar

Heejin Kim

heejink@umich.edu

ORCID iD: 0000-0002-8967-9675

© Heejin Kim 2023

DEDICATION

To my family

ACKNOWLEDGMENTS

I would like to thank those who provided assistance in this research and support throughout my journey towards PhD degree. First and foremost, I would like to express my deepest gratitude to my advisor, Professor Chinedum Okwudire, for his guidance of my research and genuine support. Having Professor Okwudire as my PhD advisor is one of the greatest privilege of my life. I have learned tremendously from his keen insight and his practical mindset. Besides research, he has always inspired me to become enthusiastic, open-minded, and caring, which significantly influenced me as a person.

I would also like to thank my committee members Profs. Kira Barton, Shorya Awtar, and Raed Al Kontar for their thoughtful support and feedback. Professor Barton has given me invaluable lessons and inspirations when I was her teaching assistant for three semesters. I am thankful for Professor Awtar for his excellent courses where I learned a great deal of knowledge that I will continue to use in the future. Lastly, I deeply thank Professor Al Kontar for always willing to provide me with his kind and invaluable feedback whenever I asked questions on research.

I would like to thank all my past and present colleagues at the Smart and Sustainable Automation Research Lab. Their warm and intellectual support has always motivated me to become a better researcher, and I will continue to cherish our collegueship. I thank Dr. Keval Ramani, Dr. Nosakhare Edoimioya and Cheng-Hao Chou for their willingness to discuss and provide me with valuable feedback. I would like to gratefully acknowledge the National Science Foundation (NSF) and Cisco Inc. for financially supporting this research. Last but not the least, I would like to thank my mom, dad and my sister, for their endless support and patience.

PREFACE

This doctoral dissertation consists of five chapters. Chapter 1 introduces the reader to the background, motivation, objectives and contributions of this dissertation. The following three chapters (i.e., Chapter 2, 3 and 4) are self-contained discussions and can be read independently. Chapter 2 and Chapter 3 are based on articles [1] and [2] published in the International Journal of Advanced Manufacturing Technology in 2020 and 2021, respectively. Chapter 4 is partially based on article [3] published in the CIRP Annals - Manufacturing Technology in 2023, as well as a manuscript in preparation. Chapter 5 summarizes the conclusions and suggestions for future research emanating from this dissertation.

TABLE OF CONTENTS

DEDICATION	ii
ACKNOWLEDGMENTS	iii
PREFACE	iv
LIST OF FIGURES	vii
LIST OF TABLES	xi
ABSTRACT	xii
CHAPTER	
1 Introduction and Literature Review	1
1.1 Background and motivation	1
1.2 Literature review	4
1.2.1 Feedrate optimization with servo error constraints	4
1.2.2 Feedrate optimization with computational efficiency	6
1.3 Dissertation contributions and outline	6
2 Feedrate Optimization with Servo Error Constraints using Physics-based Servo Dynamics Model	9
2.1 Overview	9
2.2 Time-based vs path-based LP for feedrate optimization	10
2.3 Simultaneous feedrate optimization and servo error pre-compensation using time-based LP	14
2.3.1 Framework of feedrate optimization and servo error pre-compensation using time-based LP	14
2.3.2 Realization of servo error pre-compensation in FO+SEP using filtered B splines	15
2.4 Experimental validation	17
2.4.1 Desktop 3D printer	17
2.4.2 Precision motion stage	21
2.5 Conclusions	25
3 Feedrate Optimization using Physics-based Servo Dynamics Model for Long Toolpaths	26

3.1	Overview	26
3.2	Motivation	27
3.3	A new approach for FO+SEP using windowed sequential linear programming (Win-SLP)	28
3.3.1	Formulation of FO+SEP using SLP	28
3.3.2	Formulation of FO+SEP using Win-SLP	29
3.3.3	Guaranteeing feasibility of Win-SLP	33
3.4	Validation of proposed approach via simulation	36
3.5	Experimental validation	39
3.5.1	Experimental setup	39
3.5.2	Benchmarking to determine E_{max}	40
3.5.3	Optimization results using FO+SEP and FO then SEP using Win-SLP	42
3.5.4	3D print results	44
3.6	Conclusions	45
4	Intelligent Feedrate Optimization using an Uncertainty-aware Physics-informed Data-driven Digital Twin	47
4.1	Framework of intelligent feedrate optimization using digital twin with quantified uncertainty	48
4.2	Methodology for contour error prediction using an uncertainty-aware digital twin	50
4.2.1	Overview of prediction of contour error using a deterministic digital twin	50
4.2.2	Prediction and uncertainty quantification of contour error using physics-informed data-driven digital twin	53
4.3	Methodology of intelligent feedrate optimization with contour error constraints	58
4.4	Numerical validation of the intelligent feedrate optimization using physics-informed data-driven digital twin	62
4.5	Experimental validation	67
4.5.1	CNC machine tool	67
4.5.2	Desktop 3D printer	70
4.6	Conclusion and future work	73
5	Summary, Conclusions and Future Research	75
5.1	Summary and conclusions	75
5.2	Practical considerations for implementation on general feed drive servo systems	76
5.3	Suggestions for future research	77
	APPENDIX	79
	BIBLIOGRAPHY	90

LIST OF FIGURES

FIGURE

1.1	Feed drive mechanism with a ball screw drive [4]	2
1.2	Examples of feed drives in (a) 3D printing (b) face milling (c) wafer scanning; dynamic servo errors in feed drives manifested as (d) surface profile of 3D-printed block [5] (e) chatter marks in face milling [6] (f) vibration response of ultra precision machines [7]	2
1.3	Control diagram of feed drives	3
1.4	Categorization of feedrate optimization methods and the associated approaches for each chapter and appendix in the dissertation	4
1.5	Diagram of intelligent feedrate optimization using uncertainty-aware physics-based and data-driven servo dynamic models	7
2.1	Parametric planar curve as function of path variable, s	10
2.2	Desired toolpath	12
2.3	Feedrate, axis acceleration and axis jerk profiles of path- and time-based LP with and without jerk limits imposed	13
2.4	Block diagram of the proposed simultaneous SEP and FO method (y -component of SEP and servo dynamics omitted for simplicity)	13
2.5	Contouring error ϵ_c and its linear approximation $\hat{\epsilon}_c$	16
2.6	Experimental set up	17
2.7	Measured and curve fitted FRFs of x and y axes of 3D printer	18
2.8	(a). Commanded feedrate and (b) simulated (approximated) and actual (measured) CE profiles of conservative and aggressive TAP motion commands	19
2.9	Feedrate, acceleration and jerk profiles of trajectories generated by FO and FO+SEP using aggressive kinematic limits and $\hat{E}_{max} = 14 \mu\text{m}$	20
2.10	Simulated (approximated) and actual (measured) contour error (CE) profiles of FO and FO+SEP	20
2.11	CAD model of cylinder	21
2.12	Side and top view of printed cylinders using Conservative TAP, Aggressive TAP, FO and FO+SEP on a support brim	22
2.13	Experimental setup	22
2.14	Measured and curve fitted FRFs of x and y axes of planar motion stage	23
2.15	(a) Commanded feedrate and (b) simulated (approximated) and actual (measured) CE profile of TAP motion commands	23
2.16	Feedrate, acceleration and jerk profiles of trajectories generated by FO and FO+SEP using aggressive kinematic limits and $\hat{E}_{max} = 13 \mu\text{m}$	24

2.17	Simulated (approximated) and actual (measured) CE profiles of FO and FO+SEP . . .	25
3.1	Illustration of Win-SLP applied on (a). the current window $j - 1$ with control interval N_c and preview interval N_p , and its augmentation of the final solution s^* by N_c points at the end to serve as a complete initial solution s_0 for (b). the next window j	30
3.2	Block diagram of FO+SEP using LP (Note: y -component of SEP and servo dynamics are omitted for simplicity)	33
3.3	Overview of scheme to guarantee feasibility of Win-SLP consisting of: (a) generation of backup solution in window $j - 1$; and (b) adoption of first N_p points of backup solution (i.e., ③) if no optimal solution ④ or backup solution ⑤ can be found in window j	34
3.4	Flow chart of scheme for guaranteeing feasibility of FO+SEP using Win-SLP discussed in Section 3.3	35
3.5	Feedrate, acceleration and jerk profiles of trajectories generated by FO+SEP using LP, SLP and Win-SLP (proposed) for short circular toolpath ($R = 5$ mm). The switching of the Win-SLP solution to the backup solution at the marked instances enables it to guarantee feasibility at the cost of optimality (as long as the starting solution, s_c , is feasible)	37
3.6	Simulated axis tracking error profiles of LP, SLP, and Win-SLP (proposed) showing the inaccuracy of LP and the accuracy of SLP and Win-SLP in enforcing constraints	38
3.7	Computation time vs. radius R of the circular trajectory for FO+SEP using SLP and Win-SLP (proposed) showing the superior computational efficiency of Win-SLP relative to SLP for long toolpaths	38
3.8	Experimental setup: Lulzbot Taz 6 3D Printer	39
3.9	Measured and curve fitted FRFs of x and y axes of 3D printer	40
3.10	Desired path of butterfly curve	41
3.11	Commanded feedrate, acceleration and jerk profiles of conservative TAP	41
3.12	Commanded feedrate, acceleration and jerk profiles of FO then SEP, FO+SEP (Proposed), and TAP with aggressive kinematics (Aggr.)	42
3.13	Simulated tracking error profiles of FO then SEP, FO+SEP (proposed), and TAP with aggressive kinematics (Aggr.)	43
3.14	Simulated tracking error of conservative TAP on x and y axis and the approximate error limit E_{max}	43
3.15	CAD model of the butterfly plate of height 1.2 mm with outer contour defined by the path in Fig. 2.11	44
3.16	Top view of printed butterfly plate using Conservative TAP, FO then SEP, and FO+SEP. Both FO then SEP and FO+SEP were computed using the proposed Win-SLP approach	45
4.1	Diagram of intelligent feedrate optimization using 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	49

4.2	(a) Need for a tolerance range due to violation of error tolerance in presence of uncertainties, (b) Proposed method of feedrate optimization with desired tolerance stringency using quantified uncertainty	50
4.3	Flowchart of intelligent feedrate optimization using deterministic digital twin with physic-based and data-driven servo models (<i>y</i> -axis omitted for simplicity)	50
4.4	Flowchart of intelligent feedrate optimization using an uncertainty-aware digital twin (with physics-based and data-driven servo models, <i>y</i> -axis omitted)	52
4.5	Experimental setup for Sections 4.4 and 4.5.1	62
4.6	Frequency response functions of the Nomad 3's <i>x</i> -axis showing the known uncertainty obtained under different input acceleration amplitudes	63
4.7	Frequency response functions of the Nomad 3's <i>y</i> -axis showing the known uncertainty obtained under different input acceleration amplitudes	63
4.8	Desired toolpath	64
4.9	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	65
4.10	Feedrate, acceleration, contour error, tolerance violation, and prediction error using the proposed (PIDD) methods with $\eta = 95\%$ (from Figure 4.9) and $\eta = 98\%$ in numerical validation	66
4.11	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	68
4.12	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	69
4.13	Machined parts and the zoomed-in images of upper left wing using conservative, physics-based (PB), data-driven (DD) and proposed (PIDD) approaches	70
4.14	Experimental setup for Section 4.5.2	71
4.15	Known uncertainty of Ender 3 Pro's <i>x</i> -axis	71
4.16	Known uncertainty of Ender 3 Pro's <i>y</i> -axis	71
4.17	Commanded feedrate, acceleration, contour error and prediction error using conservative (Cons.), physics-based, data-driven and proposed approaches in air-printing	72
4.18	Top and side views of 3D-printed butterfly models using conservative, physics-based, data-driven and proposed approaches and their print times	73
A.1	Diagram of intelligent feedrate optimization framework using a digital twin	80
A.2	Flowchart of intelligent feedrate optimization using digital twin (<i>y</i> -axis omitted for simplicity)	80
A.3	Velocity of desired trajectory on <i>x</i> -axis)	84
A.4	Prediction error using DT without and with the internal model	84
A.5	Desired toolpath	85
A.6	Experimental Setup	86
A.7	Experimentally identified disturbance frequencies for internal model	86
A.8	Feedrate, acceleration, contour error and prediction error using conservative, physics-based and DT-based feedrate optimization in air-cutting	86

A.9	Feedrate, acceleration, contour error and prediction error using conservative, physics-based and DT-based feedrate optimization in machining	88
A.10	Machined surfaces using the three approaches investigated	88

LIST OF TABLES

TABLE

2.1	Cycle time and computation time for feedrate optimization (FO) using path-based and time-based LP, without and with jerk constraints	14
2.2	Comparison of cycle and computation time of FO and FO + SEP	21
2.3	Comparison of cycle and computation time of FO and FO + SEP	25
3.1	Cycle time and constraint satisfaction accuracy of FO+SEP using LP, SLP and Win-SLP for short toolpath ($R = 5$ mm)	38
3.2	Cycle and computation time of FO then SEP and FO+SEP using Win-SLP approach .	43
3.3	Computation time for FO then SEP and FO+SEP for the butterfly plate in Fig. 3.16 . .	45
4.1	Cycle times and RMS of tolerance violation γ for conservative (Cons.), physics-based (PB), data-driven (DD) and proposed (PIDD) methods in Figures 4.9 and 4.10	66
4.2	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	68
4.3	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	72

ABSTRACT

Intelligent manufacturing machines envisioned for the future must be able to autonomously select process parameters that maximize their speed (productivity) while adhering to quality specifications. One of the major sources of quality degradation in feed drives of the manufacturing machines is motion-induced servo error, which can be caused by several aspects such as limited bandwidth of feedback controllers, flexible structures, or nonlinear friction. Another source of servo error is force-induced servo error, which is caused by process force such as cutting force. Given that motion- and force-induced servo errors tend to increase with higher motion speeds, there is a keen interest in maximizing the speed of motion while respecting the tolerances on servo errors. To enable this, numerous works in the feedrate optimization aim to maximize machine speed subject to servo error constraints. However, the vast majority of the available methods only consider constraints on kinematics and/or static models of servo error without any incorporation of the dynamic servo models, which leads to sub-optimal feedrates. Moreover, they do not quantify the uncertainty of the servo error predictions, and hence may not effectively adhere to constraints in the presence of high uncertainty due to the model inaccuracy.

To address these shortcomings, this dissertation proposes the framework and a set of methodologies for intelligent feedrate optimization approach enabled by uncertainty-aware physics-based and data-driven servo dynamic models. First, it proposes feedrate optimization with constraints on kinematics and servo error using physics-based servo dynamics. The optimization is formulated with respect to a time-based path parameter, which enables the linear dynamic model to be included in the optimization solved using linear programming. The integration of the servo dynamic model enables dynamic components of the servo error to be incorporated in the feedrate optimization, such as servo error pre-compensation, which allows for faster motions without violating tolerance constraints. Furthermore, the accuracy and computational efficiency of the feedrate optimization is improved using windowed sequential linear programming. Numerical feasibility is guaranteed by imposing smooth switching between the feedrate-optimal trajectory and a conservative backup trajectory. The performance of the feedrate optimization using physics-based servo model is validated using a desktop 3D printer and a precision motion stage to demonstrate reduction in cycle time while achieving similar quality to that of conservative approach used in the status quo.

Second, the dissertation augments the physics-based servo models with a data-driven servo

model to form an uncertainty-aware digital twin in the feedrate optimization to correct for inaccuracies introduced by dynamic uncertainties that are not modeled in the physics-based models. Known uncertainty is incorporated via a set of physics-based models, while unknown uncertainty is learned online by training the data-driven models via sensor measurements. The uncertainty-aware digital twin predicts the distribution of servo error, which is used in the feedrate optimization to constrain the servo error under desired tolerance and stringency. The proposed intelligent feedrate optimization is validated using a desktop 3D printer and a CNC machine tool prototype to demonstrate cycle time reduction while achieving similar tolerance stringency with conservative approach.

The broader impact of this dissertation is to achieve desired quality and higher productivity with less trial-and-error. It is expected to be applicable to any manufacturing machines that use feed drives.

CHAPTER 1

Introduction and Literature Review

This doctoral dissertation proposes a framework and a set of methodologies for intelligent feedrate optimization subject to tolerance constraints using uncertainty-aware physics-based and data-driven servo dynamic models. Section 1.1 provides background and motivation for the study, Section 1.2 reviews the literature on servo error compensation and feedrate optimization, and Section 1.3 discusses the contributions and content outline of the dissertation.

1.1 Background and motivation

Quality and productivity are two important and often competing attributes in manufacturing. Therefore, manufacturers often seek to maximize productivity subject to quality (tolerance) constraints. In practice, this goal is often achieved by trial-and-error, which is time-consuming, making it impractical in some applications such as low-volume production (e.g., 3D printing). Recently, there is a push for self-optimizing (intelligent) manufacturing machines that are capable of, among other things, autonomously optimizing their process parameters to maximize productivity while maintaining desired quality levels, without need for trial-and-error [8].

A feed drive is a controlled actuator/motor that is used to position the machine tools to the desired location. They are used in a wide variety of manufacturing machines, including computer numerically controlled (CNC) machine tools [9], wafer scanners [10], 3D printers [11], semiconductor manufacturing equipment, etc (see Figure 1.2(a)-(c)). The servomotors in feed drives can either be linear motors or rotary motors with a ball screw and nut assembly, which are used to move the drive train, such as machine tool table on a guide. The motors are driven by electronic amplifiers, which are connected to CNC systems that generate position commands at discrete time intervals. Figure 1.1 shows an example of a feed drive that uses ball screw [4], which is most frequently used in machine tool applications.

The feed drive control structure for a single axis can be modeled as a closed-loop servo system consisting of mechanics and servo controller, as illustrated in Figure 1.3. The desired motion is

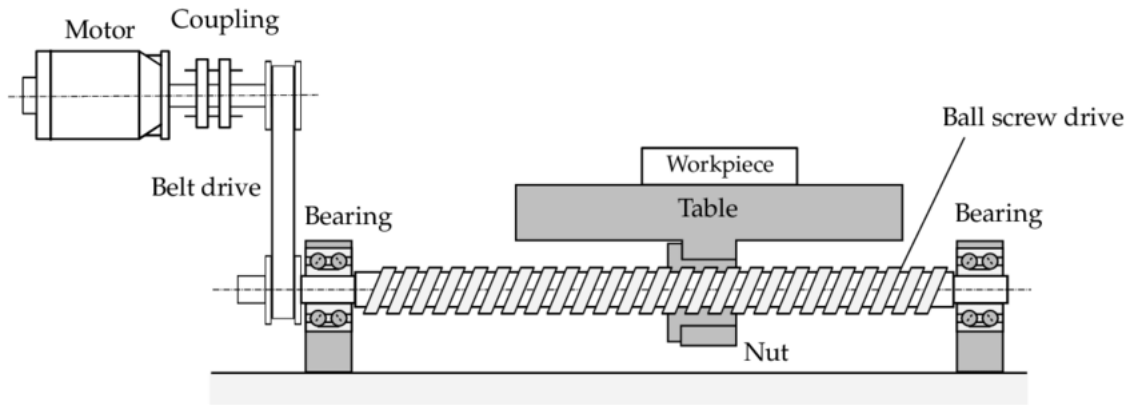
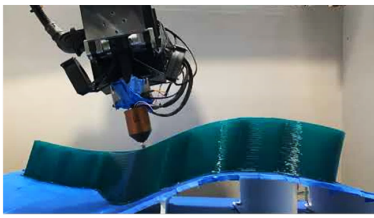
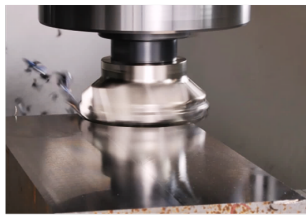


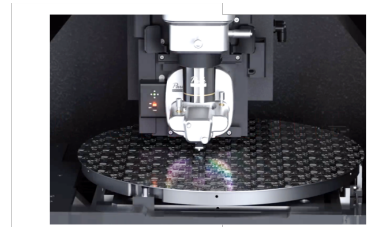
Figure 1.1: Feed drive mechanism with a ball screw drive [4]



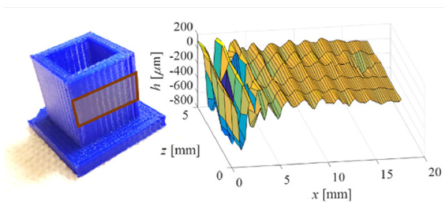
(a)



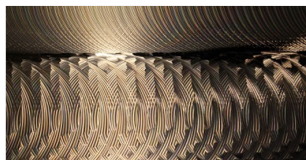
(b)



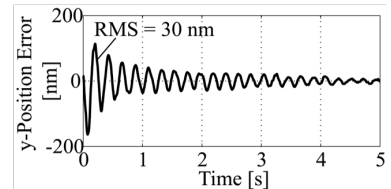
(c)



(d)



(e)



(f)

Figure 1.2: Examples of feed drives in (a) 3D printing (b) face milling (c) wafer scanning; dynamic servo errors in feed drives manifested as (d) surface profile of 3D-printed block [5] (e) chatter marks in face milling [6] (f) vibration response of ultra precision machines [7]

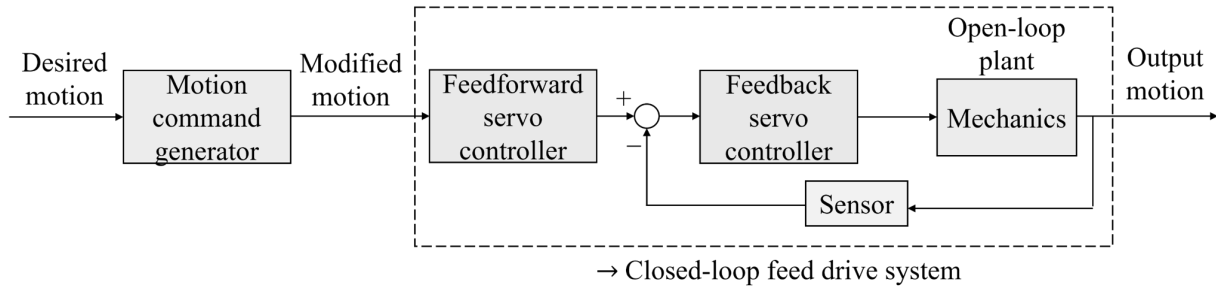


Figure 1.3: Control diagram of feed drives

modified using motion command generator, then is fed into the closed-loop system with feedback and/or feedforward servo controller. The servo error is defined as the difference between desired and output motion. Servo errors in machine tools can be caused by quasi-static errors (such as thermal and geometric error) and/or dynamic errors. Dynamic servo errors are a major source of inaccuracy in feed drives. One major source of dynamic servo error in feed drives is motion-induced servo errors. They can result from the limited bandwidth caused by nonlinear flexible structures or high-frequency components in command inputs. Another source of dynamic servo error is disturbance forces such as manufacturing process forces (e.g., cutting force), friction, etc.

Motion-induced and disturbance-induced servo errors typically increase with motion speed, because of the limited bandwidth of servo controllers, mechanical components inside servo loop, and transmission between disturbance and servo drives. Hence, dynamic servo errors are very important in determining the trade-off between speed and accuracy. The servo errors lead to the actual toolpath deviating from reference toolpath, which creates unwanted inaccuracy in manufacturing parts, e.g., excessive surface roughness or form (contouring) errors, as illustrated in examples of 3D printing [5], face milling [6] and ultra precision machines [7] in Fig. 1.2(d)-(f). So it is of interest to maximize speed subject to constraints on servo error by designing a motion command generator with given servo controllers and mechanics.

The goal to maximize feedrate subject to servo error constraints has been extensively researched under the topic of feedrate optimization. However, servo error is driven by dynamic effects as well as uncertainties. Dynamic effects are caused by several aspects such as the mechanics (e.g., flexible modes), feedback controller or feedforward servo controller (i.e., servo error pre-compensation), whereas the uncertainties may be caused by nonlinear friction, unmodeled cutting forces, geometric errors, etc. To be able to perform effective feedrate optimization with servo error constraints, this dissertation focuses on feedrate optimization that can account for servo dynamics and uncertainties by incorporating servo dynamic models.

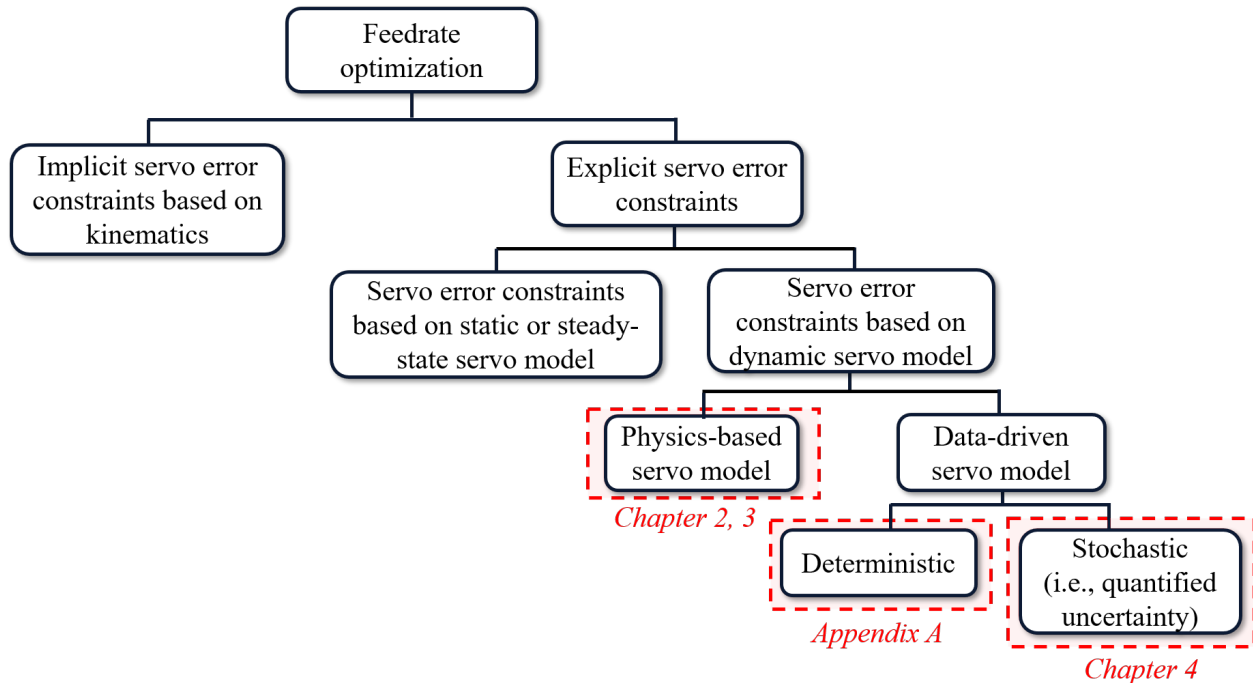


Figure 1.4: Categorization of feedrate optimization methods and the associated approaches for each chapter and appendix in the dissertation

1.2 Literature review

This section addresses literature review on feedrate optimization. Various feedrate optimization methods are discussed from the perspective of servo error constraints in Section 1.2.1, whereas they are discussed from the perspective of computational efficiency in Section 1.2.2.

1.2.1 Feedrate optimization with servo error constraints

Extensive research has been conducted in the field of feedrate optimization with the objective of maximizing the feedrate while respecting servo error constraints, and can be categorized as Figure 1.4. The majority of feedrate optimization techniques primarily focus on maximizing the feedrate while considering kinematic limits such as speed, acceleration, and jerk [12–19]. The kinematic limits help to avoid saturation of the drives and act as an implicit (indirect) method to achieve dynamic positioning accuracy, by limiting high frequency motion commands. However, the existing studies in [12–15] cannot incorporate servo error dynamics (including feedforward and feedback controller dynamics) and cutting force, resulting in the need for a cautious selection of kinematic limits to indirectly meet the requirements on dynamic parameters. This indirect approach is necessary due to the nonlinear relationship between kinematics and dynamic parameters, which often

leads to sub-optimal feedrates chosen by trial-and-error.

In order to directly constrain dynamic components, certain feedrate optimization methods incorporate limits on the servo error, in addition to kinematics, by using steady-state [20] or static [21, 22] 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 [23–35]. Some feedrate scheduling techniques maximize feedrate while regulating machining error due to tool deflection [36–42] or force-induced servo error [43, 44] under desired tolerance in CNC machine tools. However, the works in [23–44] 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.

To effectively optimize feedrate with existence of uncertainties, data-driven approaches can be used to provide more-accurate predictions of process or servo dynamics for feedrate optimization using measured data from sensors. For example, some feedrate scheduling and optimization techniques utilize artificial neural networks (ANNs) to predict and constrain dynamic parameters like spindle power [45] or contour error [46]. However, the existing data-driven models used in [45, 46] are trained offline through numerous experiments, which renders them unable to adapt to uncertainties encountered in real-time that are not captured in the offline data.

For the data-driven models to adapt to uncertainties encountered in-situ for the feedrate optimization, one approach is to estimate parameters of the physics-based models online. For example, cutting force model parameters are corrected online using virtual simulations [47] to accurately constrain tool deflection, torque, etc. in the feedrate scheduling. A model predictive control (MPC) is exploited [48] to update initial states of servo dynamics model in real time via an observer, which is used to accurately predict and constrain the servo error. Another approach is to exploit online machine learning models. For example, ANN [49] can be trained real-time to predict cutting force based on process parameters, and feedrate can be scheduled to regulate cutting force with a slow update frequency. However, the works in [47–49] do not quantify or exploit the uncertainty of the prediction. Hence, they may not effectively adhere to constraints in the presence of high uncertainty due to lack of training data or sudden change of operating conditions.

To quantify uncertainties and impose robustness, there exist studies on maximizing feedrate while regulating spindle power, where the spindle power is modeled as Gaussian Process Regression (GPR) [50]. 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 [50] is updated cycle-by-cycle, which can be slow and may not effectively predict the spindle power for non-repetitive processes. 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 the feedrate optimization, especially for toolpaths with high curvatures that can create significant structural vibration.

1.2.2 Feedrate optimization with computational efficiency

The vast majority of feedrate optimization methods are formulated with respect to a path-based parameter discretized on the trajectory arclength domain. However, because of the nonlinear relationship between feedrate (which is path-level) and axis-level constraints (e.g., jerk or servo error constraints), it is numerically challenging to solve feedrate optimization. Several works use gradient-based methods such as sequential quadratic programming (SQP) [16]. However, SQPs are nonlinear optimization solvers that require heavy computational load. In order to solve the issue of computational efficiency, some works use heuristic algorithm [51, 52] such as bidirectional scanning method [21] or greedy algorithm [53]. On the other hand, several works approximate nonlinear jerk constraints using pseudo-jerk [12, 13, 18, 54], i.e., replacement of nonlinear terms in jerk by sub-optimal path-parameter terms. However, both heuristics and pseudo-jerk approaches sacrifice optimality for efficiency. Overall, most feedrate optimization methods cannot achieve optimality and computational efficiency at the same time.

1.3 Dissertation contributions and outline

To address the shortcomings faced by feedrate optimization methods in state-of-the-art, this dissertation (based on publications [1–3]) proposes a novel intelligent feedrate optimization using uncertainty-aware physics-based and data-driven servo dynamic models. The proposed framework is depicted in Figure 1.5. First, a manufacturer submits a part together with the desired dimensions and servo 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 uncertainty-aware servo dynamic models that predict the servo error, which the machine can exploit for feedrate optimization with servo error constraints.

The uncertainty-aware servo dynamic models use the known uncertainty from the physics-based models and trains the data-driven model using the machine’s sensor measurements to learn the unknown uncertainty. The servo dynamic models predict and quantify the uncertainty of the servo error, which is used in the feedrate optimization with desired tolerance on the servo error. Together

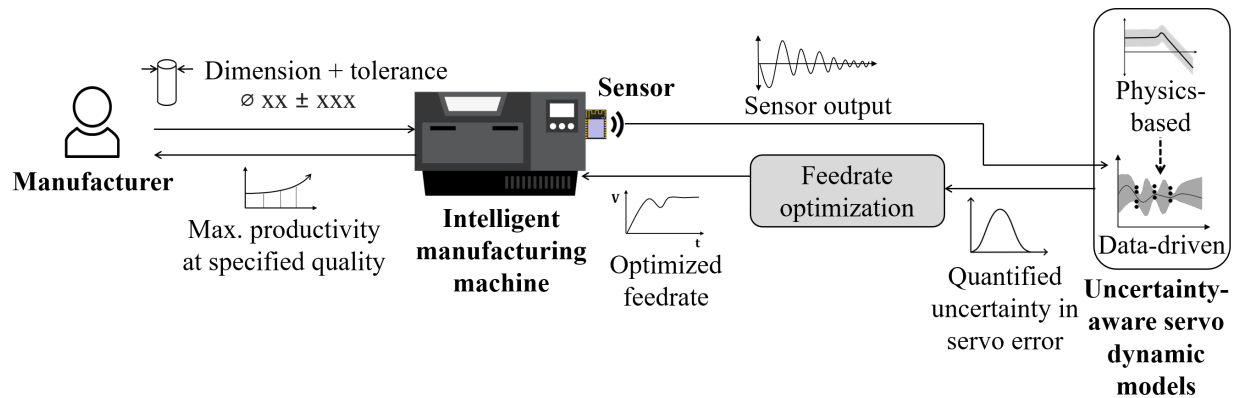


Figure 1.5: Diagram of intelligent feedrate optimization using uncertainty-aware physics-based and data-driven servo dynamic models

with the uncertainty-aware servo dynamic models, the feedrate optimization determines the fastest feedrate to run the machine while respecting the limits for the servo 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 data-driven model and optimization algorithm in the next iteration of the feedrate optimization.

Each chapter of this dissertation makes the following contributions as a step towards the intelligent framework:

C.2 Chapter 2 proposes a time-based linear programming approach for feedrate optimization that is able to incorporate linear servo dynamic models, such as physics-based servo model and servo error pre-compensation, in the servo error constraints. The servo error constraint uses predictions from physics-based model and is able to include servo error pre-compensation, thus optimally executing simultaneous feedrate optimization and servo error pre-compensation. It is demonstrated in a 3D printer and a precision motion stage that the proposed feedrate optimization can improve the cycle time up to 43% and 47%, respectively, while closely adhering to the error tolerance, compared to conservative approaches or feedrate optimization methods without incorporation of pre-compensation.

C.3 Chapter 3 builds upon Chapter 2 by improving the accuracy and computational efficiency of feedrate optimization for long toolpaths via windowed sequential linear programming within smaller batches. Feasibility is guaranteed by smooth switching between optimal solution and conservative back-up solution. It is demonstrated in 3D printing that the studied approach mitigates the linearization error and computational bottleneck of feedrate optimization from Chapter 2, while still improving cycle time compared to conservative approach up to 25%.

C.4 Finally, based on the foundations of Chapter 2 and 3, Chapter 4 proposes an intelligent feedrate optimization which incorporates physics-informed data-driven servo dynamic models, i.e., digital twin. Physics-based models are incorporated to identify the known uncertainties, while the data-driven model is used to learn the unknown uncertainties encountered in-situ using sensor measurements. The physics-informed data-driven model predicts the servo error and its uncertainty, which is used in the feedrate optimization to constrain the servo error under the desired tolerance and stringency. It is demonstrated in CNC milling and 3D printing that the intelligent feedrate optimization approach reduces the cycle time by 38% and 17%, respectively, compared to the conservative approach, while staying close to the error tolerance.

Chapter 5 summarizes the dissertation, presents conclusions and suggests recommendations for future research.

CHAPTER 2

Feedrate Optimization with Servo Error Constraints using Physics-based Servo Dynamics Model

2.1 Overview

This chapter studies feedrate optimization (FO) with servo error constraints using physics-based servo dynamics model. The systematic incorporation of servo dynamic models (including error pre-compensation, i.e., SEP) into servo error constraints expands the feasible region of feedrate optimization, thus allowing for faster motions without violating tolerance constraints. Performance of the method is validated using 3D printer and precision motion stage, and cycle time and tolerance adherence are compared with conservative approaches and feedrate optimization without incorporating error pre-compensation.

This chapter is organized as follows: Section 2.2 contrasts time-based linear programming (LP) and available path-based LP [19, 55] approaches for feedrate optimization. It shows that time-based LP is similar to path-based LP in terms of computational efficiency but is superior to path-based LP in handling jerk constraints and incorporating servo dynamics into feedrate optimization. Section 2.3 presents the proposed approach for simultaneous feedrate optimization and servo error pre-compensation (i.e., FO+SEP) using time-based LP. Section 2.4 validates the effectiveness of the proposed FO+SEP approach in simulations and experiments carried out on a 3D printer and planar motion stage. Conclusions are presented in Section 2.5.

This chapter is based on the following publication:

- Kim H, Okwudire CE. *Simultaneous servo error pre-compensation and feedrate optimization with tolerance constraints using linear programming*. The International Journal of Advanced Manufacturing Technology. 2020; 109 (1); 809-821.

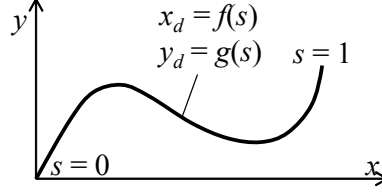


Figure 2.1: Parametric planar curve as function of path variable, s

2.2 Time-based vs path-based LP for feedrate optimization

Fig. 2.1 illustrates an arbitrary, curved path in the x - y plane with path parameter $s \in [0, 1]$. Note that s is a function of time t (i.e., $s = s(t)$). Let $x_d = f(s)$ and $y_d = g(s)$ denote a pair of parametric equations in s , representing the x and y components of desired position, respectively.

An increasingly popular approach is to perform feedrate optimization via path-based LP, i.e., using s as the independent variable [19, 55]. In path-based LP, it is assumed that s represents arc length, i.e., distance travelled along the curve, which is normalized by total travel length L . Let x_d and y_d be already known in the defined domain of s . Then, the kinematic limits F_{max} , A_{max} , and J_{max} on feedrate, axis acceleration and axis jerk respectively, can be imposed as

$$\begin{aligned}
 L |\dot{s}| &\leq F_{max} \\
 \left| \frac{d^2 x_d(s)}{dt^2} \right| &= |x_d''(s) \dot{s}^2 + x_d'(s) \ddot{s}| \leq A_{max} \\
 \left| \frac{d^3 x_d(s)}{dt^3} \right| &= |x_d'''(s) \dot{s}^3 + 3x_d''(s) \dot{s} \ddot{s} + x_d'(s) \ddot{\ddot{s}}| \leq J_{max}
 \end{aligned} \tag{2.1}$$

for $\forall s$, where $x_d'(s)$, $x_d''(s)$, and $x_d'''(s)$ denote geometric derivatives of $x_d(s)$ with respect to s ; \dot{s} , \ddot{s} , and $\ddot{\ddot{s}}$ are tangential velocity, acceleration, and jerk, respectively. The y -axis acceleration and jerk limits are imposed in the same manner. Note that, instead of imposing feedrate limits as in Eq. (2.1), axis velocity limits could be imposed in addition to axis acceleration and jerk limits [19, 55].

To facilitate path-based LP, a new parameter $q = \dot{s}^2$ is introduced to remove the nonlinearity in Eq. (2.1). With q , the following substitutions hold:

$$\begin{aligned}
 \dot{s} &= \sqrt{q}, \dot{s}^2 = q, \dot{s}^3 = q\sqrt{q} \\
 \ddot{s} &= \frac{1}{2}q', \ddot{\ddot{s}} = \frac{1}{2}q''\sqrt{q}, \dot{s}\ddot{s} = \frac{1}{2}q'\sqrt{q}
 \end{aligned} \tag{2.2}$$

With Eq. (2.2), the feedrate optimization with the same kinematic constraints is formulated as

Eq. (2.3) for $\forall s$:

$$\begin{aligned}
& \min_q - \int_0^1 q(s) ds \\
& \text{s.t. } L\sqrt{q(s)} \leq F_{max} \\
& \left| x_d''(s)q(s) + \frac{1}{2}x_d'(s)q'(s) \right| \leq A_{max} \\
& \left| x_d'''(s)q(s) + \frac{3}{2}x_d''(s)q'(s) + \frac{1}{2}x_d'(s)q''(s) \right| \sqrt{q(s)} \\
& \leq J_{max}
\end{aligned} \tag{2.3}$$

Here, the feedrate constraint can be linearized by squaring both sides, and the $q'(s)$ and $q''(s)$ terms in acceleration can be linearized by B-spline parametrization of $q(s)$ with respect to s [19, 55]. However, because the $\sqrt{q(s)}$ term in the jerk limit is still nonlinear, $q(s)$ is replaced by a precomputed upper bound $q^*(s)$. One candidate for $q^*(s)$ is the solution obtained with only velocity and acceleration constraints [19, 55] in Eq. (2.3). Then, the jerk constraint in Eq. (2.3) is reformulated using pseudo jerk $\bar{j}(s)$ as:

$$\begin{aligned}
& \left| x_d'''(s)q(s) + \frac{3}{2}x_d''(s)q'(s) + \frac{1}{2}x_d'(s)q''(s) \right| \sqrt{q^*(s)} \\
& = |\bar{j}(s)| \leq J_{max}
\end{aligned} \tag{2.4}$$

Although path-based LP is capable of imposing linear feedrate and axis acceleration constraints on FO, it cannot impose linear axis jerk constraints without the use of pseudo jerk, at the cost of optimality [19, 55]. Moreover, because path-based LP uses q as the independent variable, it is limited in its ability to accommodate servo dynamics, which uses time t as the independent variable.

Therefore, this dissertation formulates a time-based LP approach for feedrate optimization, adapted from the model predictive contour control framework proposed by Lam et al. [56]. Let $s(t)$ be discretized with fixed sampling interval T_s and expressed as a vector $\mathbf{s} = \{s(0), s(1), \dots, s(N-1)\}^T$. Then, FO can be formulated as:

$$\begin{aligned}
& \min_{\mathbf{s}} - \sum_{k=0}^{N-1} s(k) \\
& \text{s.t. } s(k-1) \leq s(k) \leq 1, \quad \forall k = 1, 2, \dots, N-1
\end{aligned} \tag{2.5}$$

The idea of Eq. (2.5) is that to minimize total time, the sum of $s(k)$ over N time steps must be maximized – i.e., the path from $s(0) = 0$ to $s(N-1) = 1$ should be traversed as fast as possible,

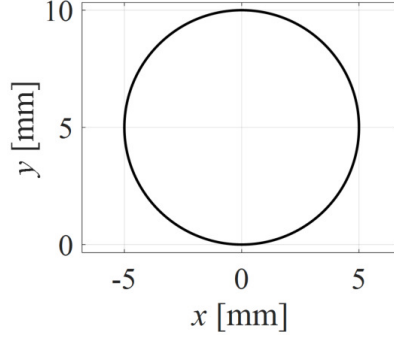


Figure 2.2: Desired toolpath

while satisfying the monotonicity and endpoint constraints on s in Eq. (2.5). In addition, it should satisfy kinematic constraints F_{max} , A_{max} , and J_{max} on feedrate, axis acceleration and axis jerk, respectively:

$$\begin{aligned}
 L \frac{D[s]}{T_s} &\leq \mathbf{F}_{max} \\
 \left| \frac{D^2[\hat{\mathbf{x}}_d]}{T_s^2} \right|, \left| \frac{D^2[\hat{\mathbf{y}}_d]}{T_s^2} \right| &\leq \mathbf{A}_{max} \\
 \left| \frac{D^3[\hat{\mathbf{x}}_d]}{T_s^3} \right|, \left| \frac{D^3[\hat{\mathbf{y}}_d]}{T_s^3} \right| &\leq \mathbf{J}_{max}
 \end{aligned} \tag{2.6}$$

Here, D denotes finite difference operator, while \mathbf{F}_{max} , \mathbf{A}_{max} , and \mathbf{J}_{max} are vectorized representations of the corresponding kinematic limits, and $\hat{\mathbf{x}}_d$ and $\hat{\mathbf{y}}_d$ are vectorized versions of $\hat{x}_d(k)$ and $\hat{y}_d(k)$, respectively, similar to \mathbf{s} . This notation is maintained hereinafter. The terms $x_d = f(s)$ and $y_d = g(s)$ are generally nonlinear in s . Thus, at each time step k , they are linearized with linearization points $s^e(k)$ estimated from an initial unoptimized trajectory as

$$\hat{x}_d(k) = \left. \frac{\partial f(s)}{\partial s} \right|_{s=s^e(k)} \cdot (s(k) - s^e(k)) + f(s^e(k)) \tag{2.7}$$

and $\hat{y}_d(k)$ is obtained by linearizing $g(s)$ in the same manner.

To compare time-based and path-based LP, a circular toolpath with radius, $R = 5$ mm is employed, as illustrated in Fig. 3. The kinematic constraints are $F_{max} = 30$ mm/s, $A_{max} = 0.5$ m/s², and $J_{max} = 5$ m/s³. For both path-based and time-based LP, \mathbf{s} is discretized and represented using B-splines [54, 55] as

$$\mathbf{s} = \mathbf{N}_s \mathbf{p}_s \tag{2.8}$$

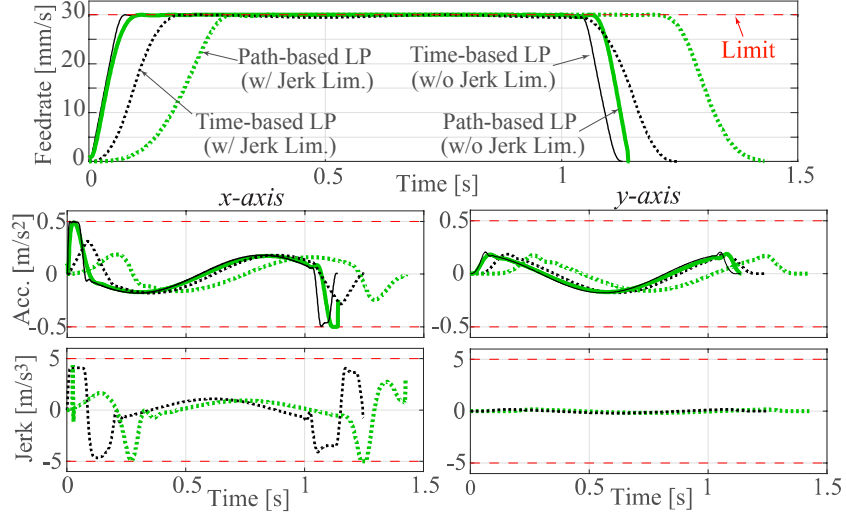


Figure 2.3: Feedrate, axis acceleration and axis jerk profiles of path- and time-based LP with and without jerk limits imposed

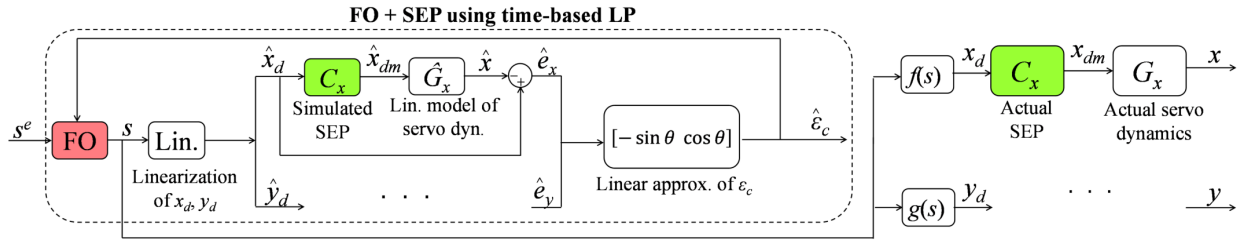


Figure 2.4: Block diagram of the proposed simultaneous SEP and FO method (y-component of SEP and servo dynamics omitted for simplicity)

where \mathbf{p}_s is the control point vector of length n_p ; \mathbf{N}_s is the basis function matrix. By using \mathbf{p}_s as the optimization variable in place of \mathbf{s} , the problem size is substantially reduced because $n_p \ll N$. Here, a 5th degree B-spline with uniform knot vector and $n_p = 40$ control points are used.

Time-based LP is initialized using an unoptimized trajectory generated using trapezoidal acceleration profile (TAP) [57] with the just-given kinematic limits. First, only the feedrate and axis acceleration limits are imposed on path-based and time-based LP. In this case, they yield almost the same feedrate profile, as shown in Fig. 2.3. As a result, their cycle time and computation time (using MATLAB[®] R2019a on a Windows PC with Intel Core i7-8750H CPU and 16 GB RAM) is similar, as summarized in Table 2.1. This shows that both methods have similar computational efficiency, under similar conditions. Next, the constraint on axis jerk and pseudo-jerk of $J_{max} = 5$ m/s³ is introduced. Fig. 2.3 and Table 2.1 shows that the cycle time becomes 1.42 s for path-based LP and 1.25 s for time-based LP. This discrepancy shows the sub-optimality introduced by the pseudo-jerk relaxation. The computation time is also summarized in Table 2.1.

Table 2.1: Cycle time and computation time for feedrate optimization (FO) using path-based and time-based LP, without and with jerk constraints

	FO algorithm	Cycle time [s]	Computation time [s]
w/o jerk constraints	Time-based LP	1.13	0.75
	Path-based LP	1.14	0.77
w/ jerk constraints	Time-based LP	1.25	1.89
	Path-based LP	1.42	1.35

Time-based LP is also superior to path-based LP because it can incorporate any linear servo dynamics into FO. Conversely, path-based LP can only accommodate servo dynamics that are linear with regard to velocity and acceleration [54, 58], without need for approximation. Given these advantages, the time-based LP formulated in this section is selected for the simultaneous servo error pre-compensation and feedrate optimization approach proposed in the next section.

2.3 Simultaneous feedrate optimization and servo error pre-compensation using time-based LP

2.3.1 Framework of feedrate optimization and servo error pre-compensation using time-based LP

Fig. 2.4 shows a block diagram of the proposed simultaneous feedrate optimization and servo error pre-compensation, or FO+SEP. The idea is to impose contour error (tolerance) constraints on feedrate optimization taking servo error pre-compensation into account. Contour error, denoted as CE hereinafter, has been selected as the accuracy index in feedrate optimization because it directly impacts the ability of part quality to meet tolerance specifications in manufacturing [59–62]. However, because the proposed approach uses LP, CE must be estimated using linear dynamics. To do this, linearized desired x -axis position, \hat{x}_d , is used to generate modified position command \hat{x}_{dm} using a servo error pre-compensation process, represented by C_x . A linear model, \hat{G}_x , of the actual servo dynamics, G_x , is used to estimate the x -axis position as \hat{x} and tracking error as $\hat{e}_x = \hat{x}_d - \hat{x}$. A similar process is followed to obtain $\hat{e}_y = \hat{y}_d - \hat{y}$, using C_y and \hat{G}_y .

CE is defined as the orthogonal distance between an actual trajectory point at time k and the reference toolpath [63], denoted as $\epsilon_c(k)$ in Fig. 2.5. Approximate CE, denoted as $\hat{\epsilon}_c(k)$, can be

computed from the axis tracking errors using a linear estimation [62] as

$$\begin{aligned}
\hat{e}_c(k) &= -\sin \theta(k)\hat{e}_x(k) + \cos \theta(k)\hat{e}_y(k) \\
&= -\sin \theta(k)(1 - \hat{G}_x C_x)\hat{x}_d(k) \\
&\quad + \cos \theta(k)(1 - \hat{G}_y C_y)\hat{y}_d(k)
\end{aligned} \tag{2.9}$$

where $\theta(k)$ is the angle of incline of the curve (\hat{x}_d, \hat{y}_d) at time step k . The linear approximation of CE \hat{e}_c (including the effects of servo error pre-compensation using C_x and C_y) is imposed as an additional constraint on the time-based LP formulation of Section 2 as:

$$\begin{aligned}
|\hat{e}_c| &= |-\sin \theta(\mathbf{I} - \hat{G}_x C_x)\hat{x}_d \\
&\quad + \cos \theta(\mathbf{I} - \hat{G}_y C_y)\hat{y}_d| \leq \hat{E}_{max}
\end{aligned} \tag{2.10}$$

where \hat{E}_{max} is the vectorized form of the maximum allowable approximate (i.e., linearized) CE, \hat{E}_{max} ; C_x , \hat{G}_x , C_y and \hat{G}_y are matrix (lifted) versions of the corresponding system dynamics [64]; and \mathbf{I} is the identity matrix. The implication is that a model of servo error pre-compensation is incorporated into feedrate optimization, yielding FO+SEP. The optimized x_d and y_d from FO+SEP are then applied to the actual servo dynamics, G_x and G_y , after being pre-compensated using C_x and C_y , respectively. Note that if $C_x = C_y = 1$, then it means that no servo error pre-compensation is considered in feedrate optimization.

2.3.2 Realization of servo error pre-compensation in FO+SEP using filtered B splines

It is worth pointing out that C_x and C_y can be any linear servo error pre-compensation (feed-forward tracking control) method, e.g., [5, 57, 59, 60, 63, 65, 66]. However, among the available linear SEP methods, the filtered B spline (FBS) approach [5, 66] stands out because of its effectiveness and versatility in handling any type of linear system dynamics [5]. Therefore, it is selected for servo error pre-compensation in this chapter.

The FBS approach parameterizes modified command \mathbf{x}_{dm} (see Fig. 2.4) using B splines as $\mathbf{x}_{dm} = N_x \mathbf{p}_x$, where N_x is the basis function matrix of degree m and \mathbf{p}_x is a vector of n control

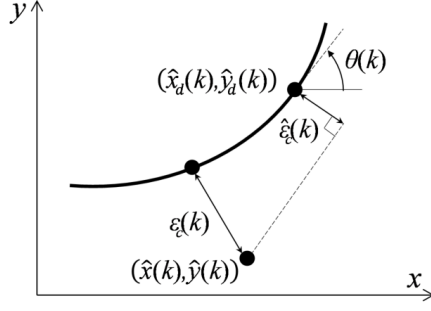


Figure 2.5: Contouring error ϵ_c and its linear approximation $\hat{\epsilon}_c$

points as

$$\begin{aligned}
 & \underbrace{\begin{bmatrix} x_{dm}(0) \\ x_{dm}(1) \\ \vdots \\ x_{dm}(N-1) \end{bmatrix}}_{=\mathbf{x}_{dm}} \\
 &= \underbrace{\begin{bmatrix} N_{0,m}(\xi_0) & N_{1,m}(\xi_0) & \cdots & N_{n-1,m}(\xi_0) \\ N_{0,m}(\xi_1) & N_{1,m}(\xi_1) & \cdots & N_{n-1,m}(\xi_1) \\ \vdots & \vdots & \ddots & \vdots \\ N_{0,m}(\xi_{N-1}) & N_{1,m}(\xi_{N-1}) & \cdots & N_{n-1,m}(\xi_{N-1}) \end{bmatrix}}_{=\mathbf{N}_x} \underbrace{\begin{bmatrix} p_x(0) \\ p_x(1) \\ \vdots \\ p_x(n-1) \end{bmatrix}}_{=\mathbf{p}_x}
 \end{aligned} \tag{2.11}$$

where $\xi \in [0, 1]$ is the spline parameter, representing normalized time; it is uniformly discretized into $\xi_0, \xi_1, \dots, \xi_{N-1}$. Each basis function $N_{j,m}(\xi)$ is defined as

$$\begin{aligned}
 N_{j,m}(\xi) &= \frac{\xi - \bar{g}_j}{\bar{g}_{j+m} - \bar{g}_j} N_{j,m-1}(\xi) \\
 &\quad + \frac{\bar{g}_{j+m+1} - \xi}{\bar{g}_{j+m+1} - \bar{g}_{j+1}} N_{j+1,m-1}(\xi) \\
 N_{j,0}(\xi) &= \begin{cases} 1 & \bar{g}_j \leq \xi \leq \bar{g}_{j+1} \\ 0 & \text{otherwise} \end{cases}
 \end{aligned} \tag{2.12}$$

where $j = 0, 1, \dots, n-1$, and $\bar{g} = [\bar{g}_0, \bar{g}_1, \dots, \bar{g}_{m+n}]$ is a normalized uniformly-spaced knot vector defined in $[0, 1]$. Accordingly, the system output is expressed as $\mathbf{x} \approx \hat{\mathbf{G}}_x \mathbf{x}_{dm} = \tilde{\mathbf{N}}_x \mathbf{p}_x$, where $\tilde{\mathbf{N}}_x = \tilde{\mathbf{G}}_x \mathbf{N}_x$ (i.e., \mathbf{N}_x filtered by $\tilde{\mathbf{G}}_x$). The tracking error is modeled as Eq. (2.13):

$$\mathbf{e}_x = \mathbf{x}_d - \mathbf{x} \approx \mathbf{x}_d - \tilde{\mathbf{N}}_x \mathbf{p}_x \tag{2.13}$$

Then, the least-squares solution for minimizing $e_x^T e_x$ yields optimal coefficients \mathbf{p}_x^* as:

$$\mathbf{p}_x^* = (\tilde{\mathbf{N}}_x^T \tilde{\mathbf{N}}_x)^{-1} \tilde{\mathbf{N}}_x \mathbf{x}_d = \tilde{\mathbf{N}}_x^\dagger \mathbf{x}_d \quad (2.14)$$

where \dagger represents pseudoinverse. Therefore, $\mathbf{x}_{dm} = \mathbf{N}_x \mathbf{p}_x^* = \mathbf{N}_x \tilde{\mathbf{N}}_x^\dagger \mathbf{x}_d$, which leads to $\mathbf{C}_x = \mathbf{N}_x \tilde{\mathbf{N}}_x^\dagger$. The same process is applied to \mathbf{y}_d . Accordingly, $\mathbf{C}_x = \mathbf{N}_x \tilde{\mathbf{N}}_x^\dagger$ and $\mathbf{C}_y = \mathbf{N}_y \tilde{\mathbf{N}}_y^\dagger$ are substituted into Eq. (10) to realize FO+SEP using the FBS approach.

2.4 Experimental validation

For validation of the proposed FO+SEP approach, two experimental setups are used. The first set of experiments, described in Section 4.1, is carried out on a desktop 3D printer commonly used for rapid prototyping. The second set of experiments, described in Section 4.2, is carried out on a linear motor driven planar motion stage typically used in industry for precision positioning. Demonstration of the proposed method on two experimental setups helps to show its versatility.

2.4.1 Desktop 3D printer

2.4.1.1 Experimental setup

A Lulzbot Taz 6 3D printer is used, as shown in Fig. 2.6. The optimization algorithms are implemented on dSPACE DS1202 real-time control board running at 1 kHz sampling rate, connected to DRV8825 stepper motor drivers for x , y , z , and e - (extruder) axes stepper motors. ADXL335 accelerometers are attached on the build plate and extruder to measure x , y -axes acceleration.

In this section, FO denotes feedrate optimization without incorporating servo error pre-compensation in the servo loop, i.e., $C_x = C_y = 1$, and is compared with FO+SEP. To execute FO and FO+SEP with error constraints, the x and y axis servo dynamics of the printer must be

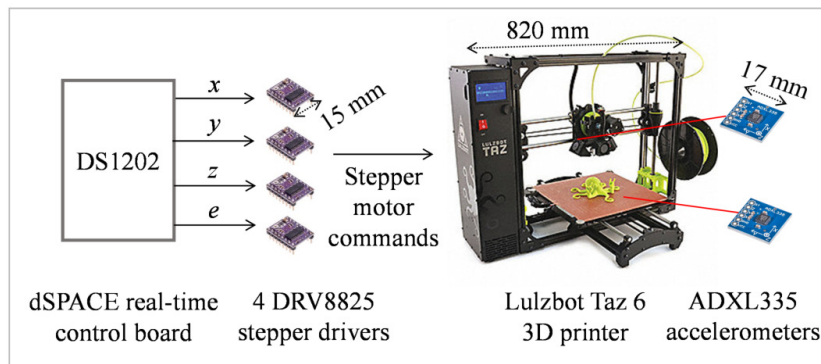


Figure 2.6: Experimental set up

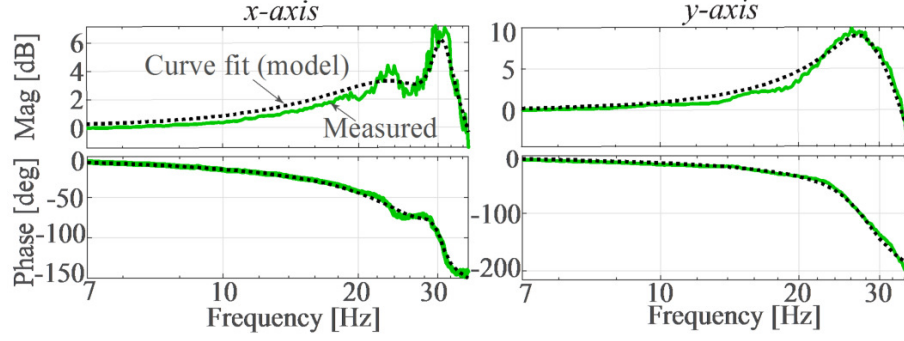


Figure 2.7: Measured and curve fitted FRFs of x and y axes of 3D printer

measured in the form of frequency response functions (FRFs) and modeled, via curve fitting, as \hat{G}_x and \hat{G}_y . Fig. 2.7 shows the measured and modeled FRFs of the x - and y -axes of the printer. The input of each FRF are swept sine acceleration commands to the stepper motor, and the output is relative acceleration between the build plate and nozzle measured using the two ADXL335 accelerometers. The discrete-time transfer function representation of \hat{G}_x and \hat{G}_y are shown in Eq. (2.15), where the open-loop bandwidth is located around 30 Hz for both axes.

$$\begin{aligned} \hat{G}_x &= \frac{0.021z^5 - 0.061z^4 + 0.044z^3 + 0.033z^2 - 0.056z + 0.012}{z^6 - 5.627z^5 + 13.38z^4 - 17.2z^3 + 12.6z^2 - 4.994z + 0.836} \\ \hat{G}_y &= \frac{0.018z^5 - 0.053z^4 + 0.038z^3 + 0.027z^2 - 0.048z + 0.017}{z^6 - 5.648z^5 + 13.48z^4 - 17.4z^3 + 12.8z^2 - 5.093z + 0.856} \end{aligned} \quad (2.15)$$

Moreover, in recovery of x , y axes displacement from acceleration measurements, a Luenberger state observer [67] is used. Observer gains are chosen such that the dynamics of the observer error (i.e., difference between estimated position using the linear system model in Eq. (2.15) and observed position) obtains global asymptotic convergence with observer frequency $f = 10$ Hz.

2.4.1.2 Benchmarking to determine approximate CE limit

Unoptimized position commands generated using trapezoidal acceleration profile (TAP) [57] are used for benchmarking to determine suitable approximate CE limit to traverse a circle of 5 mm radius. Two sets of kinematic limits are used. They are:

- Conservative: $F_{max} = 30$ mm/s, $A_{max} = 0.5$ m/s², $J_{max} = 5$ m/s³;
- Aggressive: $F_{max} = 50$ mm/s, $A_{max} = 10$ m/s², $J_{max} = 5000$ m/s³

Fig. 2.8(a) shows the TAP feedrate profile generated using the conservative and aggressive kinematic limits; the acceleration and jerk profiles are omitted for the sake of brevity. Fig. 2.8(b)

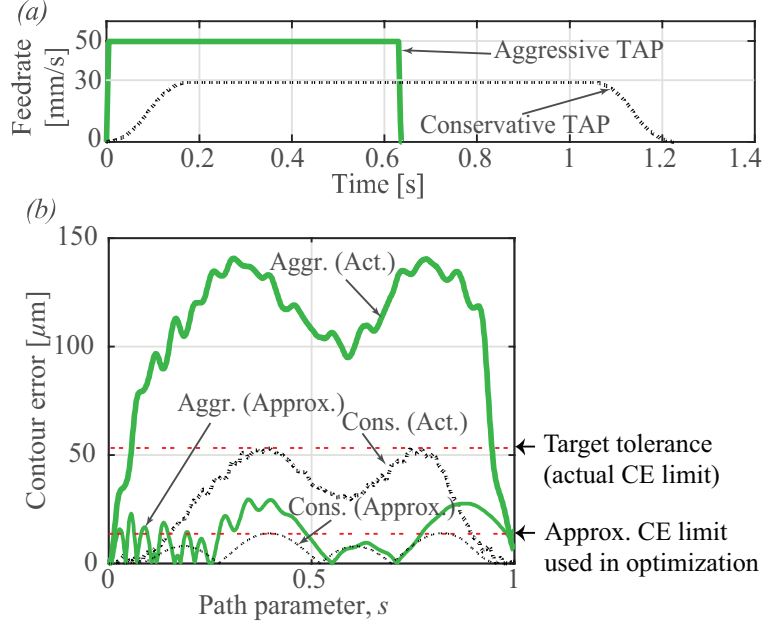


Figure 2.8: (a). Commanded feedrate and (b) simulated (approximated) and actual (measured) CE profiles of conservative and aggressive TAP motion commands

shows simulated (approximated) and actual (measured) CE profiles of the conservative and aggressive TAP position commands applied to the 3D printer. The simulations are performed using the curve fit linear dynamic model in Eq. (2.15). Conservative TAP yields maximum simulated and actual CEs of $14 \mu\text{m}$ and $54 \mu\text{m}$, respectively. Conversely, Aggressive TAP yields maximum simulated and actual CEs of $30 \mu\text{m}$ and $141 \mu\text{m}$, respectively. The reason for the discrepancy between the simulated and actual CEs is due to dynamics like friction and geometric errors not included in the linear model, as well as errors due to linear approximation of the CE in Eq. (2.9) and constraint equations in Eq. (2.6) and (2.10). As a result of these discrepancies between the linear dynamics/approximations and the actual dynamics, it is very important to determine the approximate CE limits (\hat{E}_{max}), used in the proposed FO+SEP, that correspond to acceptable tolerance (i.e., actual CE). Prior work [68] has shown that the aggressive TAP results in poor print quality, while the conservative TAP (with maximum actual CE of $54 \mu\text{m}$) yields acceptable print quality on the Taz 6 printer. Therefore, $\hat{E}_{max} = 14 \mu\text{m}$ is selected as the approximate CE for LP-based optimization to help keep actual CEs close to the target $54 \mu\text{m}$ in reality.

2.4.1.3 Optimization results using FO and FO+SEP

In this section, FO and the proposed FO+SEP are compared with a goal to achieve similar accuracy as Conservative TAP in Fig. 2.8(b) with the shortest cycle time. To do this, the aggressive kinematic limits in Section 4.1.2 are imposed on both FO and FO+SEP, together with an approxi-

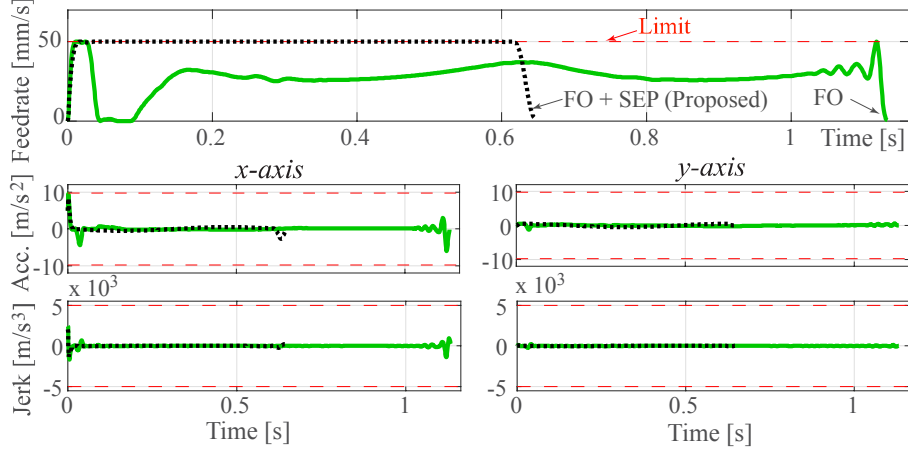


Figure 2.9: Feedrate, acceleration and jerk profiles of trajectories generated by FO and FO+SEP using aggressive kinematic limits and $\hat{E}_{max} = 14 \mu\text{m}$

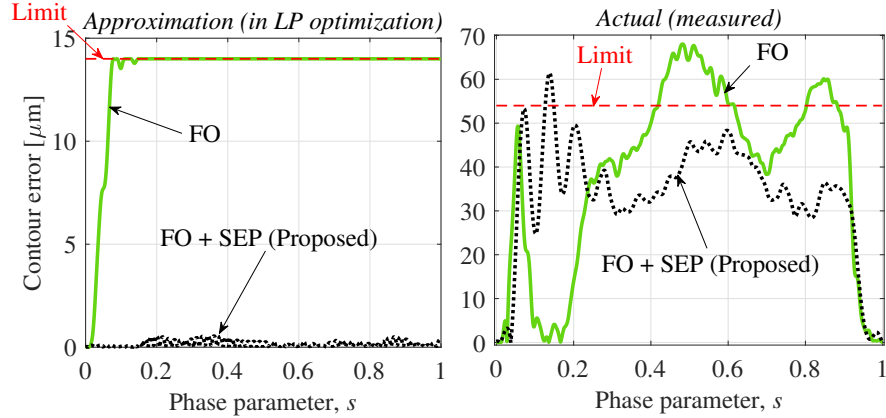


Figure 2.10: Simulated (approximated) and actual (measured) contour error (CE) profiles of FO and FO+SEP

mate CE limit of $\hat{E}_{max} = 14 \mu\text{m}$ using Eq. (2.10). For FO, $C_x = C_y = \mathbf{I}$ (i.e., tolerance constraints are imposed without SEP). However, for FO+SEP, C_x and C_y are generated via the FBS approach described in Section 3.2. using a 5th degree B-spline with uniform knot vector and $n = 40$ control points. Another 5th degree B-spline with uniform knot vector and $n_p = 40$ control points is used to parametrize s to reduce the problem size, as explained in Section 2. Both the FO and FO+SEP cases are initialized using unoptimized TAP trajectories.

Fig. 2.9 shows the commanded feedrate, acceleration, and jerk profiles of FO and FO+SEP. Fig. 2.10 shows the simulated (approximated) and actual (measured) CE profiles. Both FO and FO+SEP enforce the approximate CE limit in the LP optimization, leading to the system staying close to the target in experiments. However, FO has to slow down because it hits the approximate CE limit while FO+SEP is able to stay very close to the maximum speed throughout the motion.

Table 2.2: Comparison of cycle and computation time of FO and FO + SEP

	Cycle time [s]	Computation time [s]
FO	1.13	37.5
FO + SEP (Proposed)	0.64	1.8

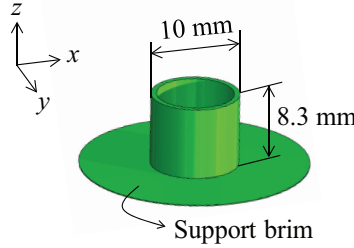


Figure 2.11: CAD model of cylinder

As a result, FO+SEP completes the motion in 0.64 s, which is 43% faster than FO at 1.13 s, as summarized in Table 4.3. Note that implementing SEP after FO (i.e., independent approach) would not lower the cycle time of FO; it would only reduce the CE, which has little or no practical value if the desired tolerance has already been met. The computation time for FO+SEP is 1.8 s; FO's is much higher at 37.5 s because it is operating very close to the imposed error constraint.

To further validate the findings, a cylinder of height 8.3 mm consisting of three concentric circular toolpath of radii 4.39 mm, 4.69 mm and 5 mm are printed using the same 3D printer, as shown in Fig. 2.11. Conservative and Aggressive TAP as well as FO and FO+SEP, as discussed above, are applied to each circular toolpath at each layer of the print. Fig. 2.12 shows the side and top view of the printed cylinders for the four cases. FO and FO+SEP save 10.9% and 50.5% in cycle time, respectively, compared to Conservative TAP, while maintaining similar surface quality. However, Aggressive TAP results in poor surface quality, though it takes a similar length of time as FO+SEP to print.

2.4.2 Precision motion stage

The performance of FO+SEP is also validated without the use of state observer, by testing on a precision motion stage with direct encoder feedback. This is to show that FO+SEP is applicable to various types of servo systems that suffer from limited bandwidth.

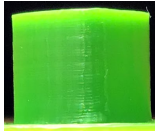
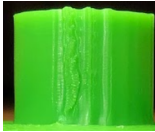
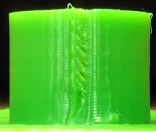
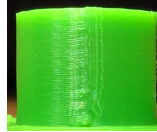
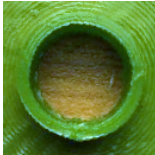
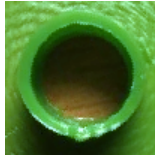
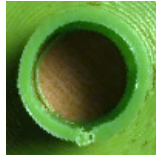
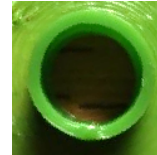
Algorithm	Conservative TAP	Aggressive TAP	FO	FO + SEP (Proposed)
Side view				
Top view				
Time (min)	11:06	5:24	9:53	5:30

Figure 2.12: Side and top view of printed cylinders using Conservative TAP, Aggressive TAP, FO and FO+SEP on a support brim

2.4.2.1 Experimental setup

A biaxial linear-motor-driven motion stage (Aerotech ALS 25010) is used, as shown in Fig. 2.13. The optimization algorithms of FO and FO+SEP are implemented on dSPACE DS1103 real-time control board running at 1 kHz sampling rate, connected to Soloist CP controller/drive. Each axis is controlled by pre-tuned closed-loop P-PI controller and velocity feedforward. The planar motion stage is equipped with optical linear encoders with resolution $0.1 \mu\text{m}$ to provide position feedback on each axis.

As with the 3D printer, the servo dynamics of the x and y axis are measured in the form of FRFs and fitted as transfer functions. Fig 2.14 shows the measured and modeled FRFs of each axis of the planar motion stage. The input of each FRF are position commands constructed by swept sine acceleration to the servomotor, and output is the position measured by encoders on each axis.

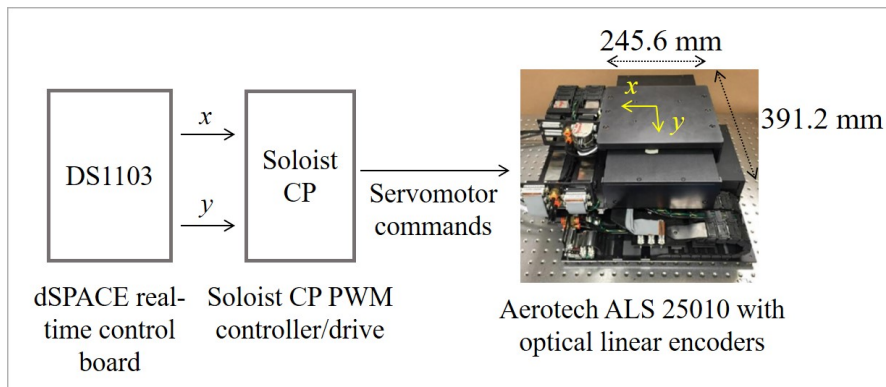


Figure 2.13: Experimental setup

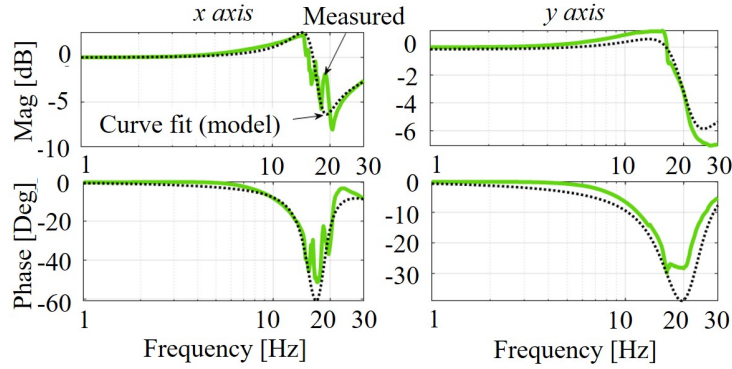


Figure 2.14: Measured and curve fitted FRFs of x and y axes of planar motion stage

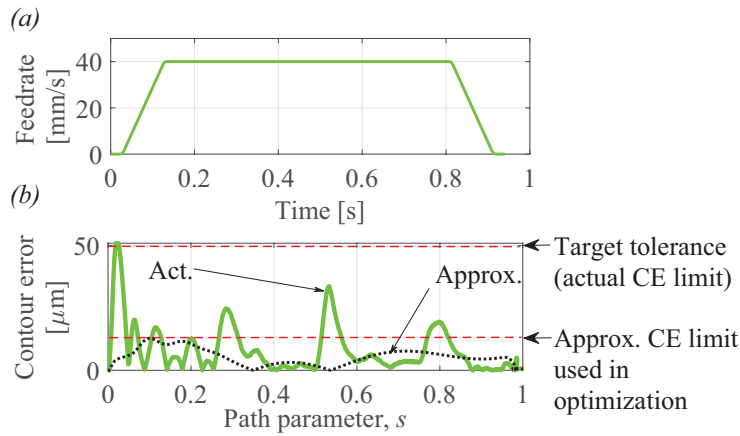


Figure 2.15: (a) Commanded feedrate and (b) simulated (approximated) and actual (measured) CE profile of TAP motion commands

2.4.2.2 Benchmarking to determine approximate CE limit

Unoptimized position commands generated using a Conservative TAP [57] are used for benchmarking to determine suitable approximate CE error in traversing a circle of 5 mm radius. Fig. 2.15(a) shows the TAP feedrate profile generated using conservative kinematic limits of $F_{max} = 40$ mm/s, $A_{max} = 0.4$ m/s², $J_{max} = 4$ m/s³; the acceleration and jerk profiles are omitted for the sake of brevity.

Fig. 2.15(b) shows simulated (approximated) and actual (measured) CE profiles of the TAP position commands applied to the planar motion stage. The simulations are performed using the curve fit linear dynamic model of Fig. 2.14. The conservative TAP yields maximum simulated and actual CE of 13 μm and 50 μm , respectively. Because the actual contouring accuracy of the TAP is considered to be satisfactory, $\hat{E}_{max} = 13$ μm is selected as the approximate CE limit for LP-based optimization, to help keep CE close to the target 50 μm in reality.

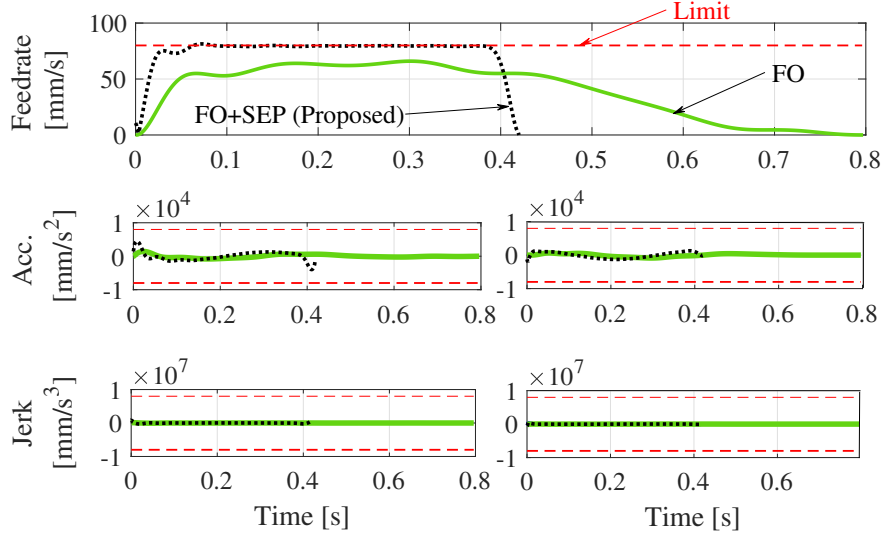


Figure 2.16: Feedrate, acceleration and jerk profiles of trajectories generated by FO and FO+SEP using aggressive kinematic limits and $\hat{E}_{max} = 13 \mu\text{m}$

2.4.2.3 Optimization results using FO and FO+SEP

We compare FO and the proposed FO+SEP with a goal to achieve similar accuracy as Conservative TAP in Fig. 2.15(b) with the shortest cycle time. To do this, aggressive kinematic limits are imposed on both FO and FO+SEP as: $F_{max} = 80 \text{ mm/s}$, $A_{max} = 8 \text{ m/s}^2$, $J_{max} = 8,000 \text{ m/s}^3$. In addition, approximate CE limit of $\hat{E}_{max} = 13 \mu\text{m}$ is imposed. In FO, $C_x = C_y = \mathbf{I}$, and in FO+SEP, C_x and C_y are generated via the FBS approach described in Section 3.2 using a 5th degree B-spline with uniform knot vector and $n = 30$ control points. To reduce the problem size, another 5th degree B-spline with uniform knot vector and $n_p = 30$ control points is used to parametrize s . Both the FO and FO+SEP are initialized using unoptimized TAP trajectories.

Fig. 2.16 shows the commanded feedrate, acceleration, and jerk profiles of FO and FO+SEP. Fig. 2.17 shows the simulated (approximated) and actual (measured) CE profiles. Both FO and FO+SEP enforce the tolerance in simulations, which enforces the experimental error close to the target. However, as was in the experiment with the 3D printer, FO has to slow down because it hits the CE limit while FO+SEP is able to stay very close to the maximum speed throughout the motion. Consequently, FO+SEP completes the motion in 0.42 s, which is 47% faster than FO at 0.79 s, as summarized in Table 2.3. Note that the cycle time of FO is only 0.15 s (i.e., 16.0%) faster than the conservative TAP. The computation time for FO+SEP is 0.04 s; FO's is higher at 0.12 s because it is operating very close to the approximate error limit imposed.

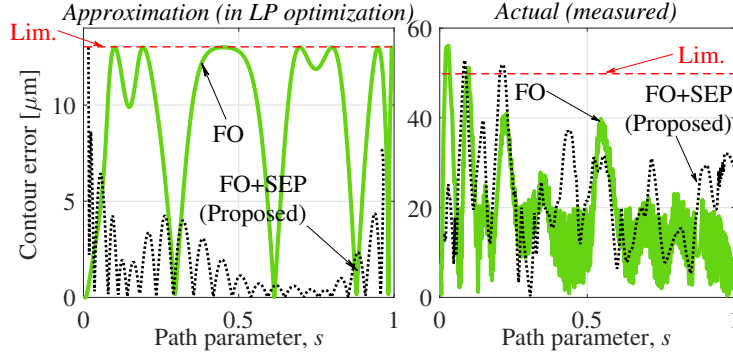


Figure 2.17: Simulated (approximated) and actual (measured) CE profiles of FO and FO+SEP

Table 2.3: Comparison of cycle and computation time of FO and FO + SEP

	Cycle time [s]	Computation time [s]
FO	0.79	0.12
FO + SEP (Proposed)	0.42	0.04

2.5 Conclusions

This chapter has introduced a new concept of simultaneous feedrate optimization and servo error pre-compensation (i.e., FO+SEP), and proposed a novel approach for realizing FO+SEP using time-based LP.

A time-based LP approach, which uses time as the independent variable, is formulated and compared with commonly-used path-based LP. Time-based LP is preferable to path-based LP in two aspects: (1) axis jerk constraints can be imposed without the use pseudo-jerk approximation, and (2) any general linear dynamics constraints can be incorporated. It is shown in the simulations that time-based LP provides an elegant and computationally efficient approach for FO+SEP.

Compared to the standard practice of performing feedrate optimization and servo error pre-compensation independently, FO+SEP relaxes the error tolerance constraints imposed on feedrate optimization, allowing shorter cycle times without violating tolerance constraints. A 3D printer and precision motion stage yielded up to 43% and 47% reduction, respectively, in cycle time using FO+SEP compared to FO, subject to the same tolerance and kinematic constraints.

CHAPTER 3

Feedrate Optimization using Physics-based Servo Dynamics Model for Long Toolpaths

3.1 Overview

Chapter 2 studied a new concept of feedrate optimization where servo error pre-compensation is integrated into feedrate optimization (i.e., FO+SEP), yielding large reductions in motion time without sacrificing positioning accuracy relative to independent approach. However, FO+SEP was achieved using linear programming (LP). As a result, it had two major shortcomings: (i) inaccuracy in enforcing nonlinear constraints due to linearization errors; and (ii) poor computational efficiency for long trajectories because it processed the full motion trajectory in one batch.

This chapter addresses the problem by proposing a new approach for FO+SEP using windowed sequential linear programming (Win-SLP), where LP is iteratively applied to FO+SEP in small overlapping batches, thus significantly improving its accuracy and computational efficiency. This chapter also addresses the potential for infeasibility in achieving FO+SEP using Win-SLP by imposing smooth switching between the optimal Win-SLP solution and a conservative backup solution, thus guaranteeing the existence of a feasible solution. Finally, using the proposed Win-SLP, this chapter demonstrates the practical benefits of FO+SEP on long toolpaths in experiments on a 3D printer, leading to 25% reduction in cycle time without sacrificing motion accuracy compared to independent approach of feedrate optimization and servo error pre-compensation.

In Section 3.2, the proposed approach for FO+SEP using Win-SLP is presented. A series of simulations are carried out in Section 3.4 to validate the effectiveness of the proposed approach with regard to accuracy, computational efficiency and feasibility. Then, in Section 3.5, the practical benefit of the proposed Win-SLP approach for FO+SEP is demonstrated in experiments on a 3D printer with long tool paths, as an example of a servo system that suffer from limited bandwidth. Conclusions are presented in Section 3.6.

This chapter is based on the following publication:

- Kim H, 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. 2021; 115 (7-8); 2069-2082.

3.2 Motivation

A major problem with the LP-based FO+SEP approach presented in Chapter 2 is that the accuracy of the solution highly depends on the trajectory s^e in Eq. (2.7), estimated from an initial unoptimized trajectory, as it is used to initialize the solution and linearize the feedrate optimization's kinematic constraints in Eq. (3.1) and servo error constraints in Eq. (3.2). As s tends to s^e , the linearization error converges to zero; however, when s moves further away from s^e , depending on the polynomial order of the nonlinearity in Eq. (3.1) and (3.2), the linearization error diverges. Another issue with the approach in Chapter 2 is that it processes all N points of the motion trajectory in one shot. This approach is reasonable for short trajectories where N is small [1], but is impractical for long trajectories (which are typical in manufacturing) due to the curse of dimensionality. Hence, an approach to address these two issues together with a scheme to guarantee feasible solutions is needed.

$$\begin{aligned}
\min_s \quad & \sum_{k=0}^{N-1} -s(k) \\
\text{s.t.} \quad & s(k-1) \leq s(k) \leq 1 \quad \forall k = 1, 2, \dots, N-1; \\
& L \frac{D[s]}{T_s} \leq \mathbf{F}_{max}; \\
& \left| \frac{D^2[\hat{\mathbf{x}}_d]}{T_s^2} \right|, \left| \frac{D^2[\hat{\mathbf{y}}_d]}{T_s^2} \right| \leq \mathbf{A}_{max}
\end{aligned} \tag{3.1}$$

$$|\hat{\mathbf{e}}_x| = \left| (\mathbf{I} - \hat{\mathbf{G}}_x \mathbf{C}_x) \hat{\mathbf{x}}_d \right| \leq \mathbf{E}_{max} \tag{3.2}$$

3.3 A new approach for FO+SEP using windowed sequential linear programming (Win-SLP)

3.3.1 Formulation of FO+SEP using SLP

Sequential linear programming (SLP) is an optimization technique for solving nonlinear optimization problems iteratively using LP [69]. Given an estimate of the optimal solution, a sequence of first-order approximations (i.e., linearization) of the problem is executed. In other words, given a constrained nonlinear programming problem with decision variable \mathbf{q} , cost function $J(\mathbf{q})$, and a set of inequality constraints $\mathbf{l}(\mathbf{q})$:

$$\begin{aligned} \min_{\mathbf{q}} \quad & J(\mathbf{q}) \\ \text{s.t.} \quad & \mathbf{l}(\mathbf{q}) \leq \mathbf{0} \end{aligned} \quad (3.3)$$

an initial set of linearization points \mathbf{q}_0 is given to render the problem in Eq. (3.3) into Eq. (3.4).

$$\begin{aligned} \min_{\mathbf{q}} \quad & J(\mathbf{q}_0) + \left. \frac{\partial J}{\partial \mathbf{q}} \right|_{\mathbf{q}=\mathbf{q}_0} \cdot (\mathbf{q} - \mathbf{q}_0) \\ \text{s.t.} \quad & \mathbf{l}(\mathbf{q}_0) + \nabla \mathbf{l}(\mathbf{q}_0)^T \cdot (\mathbf{q} - \mathbf{q}_0) \leq \mathbf{0} \end{aligned} \quad (3.4)$$

Solving Eq. (3.4) using LP gives the optimal solution \mathbf{q}_1 at the 1st iteration. Then, \mathbf{q}_1 is used to define a new set of linearization points to compute optimal solution \mathbf{q}_2 at the 2nd iteration, and this process is repeated i times until an acceptable level of accuracy in the optimal solution $\mathbf{q}^* = \mathbf{q}_i$ is attained.

In a similar manner, using SLP, first FO+SEP uses $\mathbf{s}_0 = \mathbf{s}^e$ at the 1st iteration to linearize the nonlinear term $\hat{\mathbf{x}}_d, \hat{\mathbf{y}}_d$ to solve for \mathbf{s}_1 ; then \mathbf{s}_1 is used as linearization points in the 2nd iteration, and so on until the optimal solution $\mathbf{s}^* = \mathbf{s}_i$ is obtained. Accordingly, Eq. (3.1), (3.2) can be reformulated as the pseudocode in Eq. (3.5).

- 1: **Initialize:**
- $i = 0$
- $\mathbf{s}_0 = \mathbf{s}^e$
- 2:
- 3: **do**
- 4: $i = i + 1;$
- 5: $\min_{\mathbf{s}} \sum_{k=0}^{N-1} -s_i(k)$
- 6: **s.t.** $\forall k \in [1, N - 1],$ (3.5)
- 7: $s_i(k - 1) \leq s_i(k) \leq 1;$
- 8: $L \frac{D[\mathbf{s}_i]}{T_s} \leq \mathbf{F}_{max};$
- 9: $\left| \frac{D^2[\hat{\mathbf{x}}_{d,i-1}]}{T_s^2} \right|, \left| \frac{D^2[\hat{\mathbf{y}}_{d,i-1}]}{T_s^2} \right| \leq \mathbf{A}_{max};$
- 10: $\left| (\mathbf{I} - \hat{\mathbf{G}}_x \mathbf{C}_x) \hat{\mathbf{x}}_{d,i-1} \right|, \left| (\mathbf{I} - \hat{\mathbf{G}}_y \mathbf{C}_y) \hat{\mathbf{y}}_{d,i-1} \right| \leq \mathbf{E}_{max}$
- 11: **while** $\left| \sum_{k=0}^{N-1} s_i(k) - \sum_{k=0}^{N-1} s_{i-1}(k) \right| \geq tol$
- 12: $\mathbf{s}^* = \mathbf{s}_i$

Here, tol represents the maximum allowable difference between previous and current values of the cost function; Furthermore, $\hat{\mathbf{x}}_{d,i-1}$ and $\hat{\mathbf{y}}_{d,i-1}$ in Eq. (3.5) respectively represent $\hat{\mathbf{x}}_d$ and $\hat{\mathbf{y}}_d$ of Eq. (2.7) evaluated using \mathbf{s}_{i-1} instead of \mathbf{s}^e .

3.3.2 Formulation of FO+SEP using Win-SLP

Notice that SLP is more computationally expensive than LP because it involves repeated executions of LP. To address this problem, a windowed SLP (Win-SLP) scheme is implemented as illustrated in Fig. 3.1(a). In Win-SLP, rather than optimizing $s_i(k)$ over $k \in [0, N - 1]$ for $i = 1, 2, \dots$, the SLP optimization discussed in Eq. (3.5) is applied within a window j defined over $k \in [jN_c, jN_c + N_p - 1]$, where $j = 0, 1, 2, \dots$, is the window index, N_p is the length of the preview interval and $N_c < N_p$ is the length of control interval over $k \in [jN_c, jN_c + N_c - 1]$. Upon completion of the optimization, the window j advances to window $j + 1$ by N_c time steps. This process is repeated until $s_i(jN_c + N_c - 1) = 1$. In mathematical terms, FO+SEP using Win-SLP at iteration i and window j can be formulated as Eq. (3.6), (3.7):

$$\min_{\mathbf{s}} \sum_{jN_c}^{jN_c + N_p - 1} -s_i(k) \quad (3.6)$$

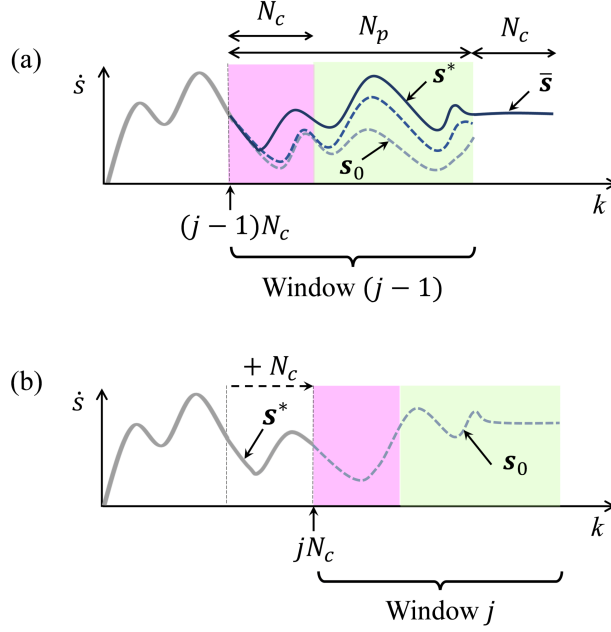


Figure 3.1: Illustration of Win-SLP applied on (a). the current window $j - 1$ with control interval N_c and preview interval N_p , and its augmentation of the final solution \mathbf{s}^* by N_c points at the end to serve as a complete initial solution \mathbf{s}_0 for (b). the next window j

$$\begin{aligned}
 \text{s.t. } \quad & \forall k \in [jN_c + 1, jN_c + N_p - 1], \\
 & s_i(k-1) \leq s_i(k) \leq 1; \\
 & L \frac{D[\mathbf{s}_i]}{T_s} \leq \mathbf{F}_{max}; \\
 & \left| \frac{D^2[\hat{\mathbf{x}}_{d,i-1}]}{T_s^2} \right|, \left| \frac{D^2[\hat{\mathbf{y}}_{d,i-1}]}{T_s^2} \right| \leq \mathbf{A}_{max}
 \end{aligned} \tag{3.7}$$

For $j = 0$, $\mathbf{s}_0 = \mathbf{s}^e$ is used for determining $\hat{\mathbf{x}}_{d0}$ and $\hat{\mathbf{y}}_{d0}$. However, starting from $j = 1$, \mathbf{s}_0 is calculated as

$$\mathbf{s}_0(k) = \begin{cases} \mathbf{s}^*(k) & k \in [jN_c, jN_c + N_p - N_c - 1] \\ \bar{\mathbf{s}}(k) & k \in [jN_c + N_p - N_c, jN_c + N_p - 1] \end{cases} \tag{3.8}$$

where \mathbf{s}^* as shown in Fig. 3.1(a) indicates the optimal solution obtained by applying Win-SLP until window $j - 1$; $\bar{\mathbf{s}}$ is defined as:

$$\begin{aligned}\bar{s}(k) &= s^*(jN_c + N_p - N_c - 1) \\ &\quad + \Delta_{end}(k - (jN_c + N_p - N_c - 1))\end{aligned}\tag{3.9}$$

where

$$\begin{aligned}\Delta_{end} &= s^*(jN_c + N_p - N_c - 1) \\ &\quad - s^*(jN_c + N_p - N_c - 2)\end{aligned}$$

The implication of Eq. (3.8) is that the unused portion of s^* from window $j - 1$ is used to partially initialize the optimization in window j . To make up for the missing N_c points used up in window $j - 1$, \bar{s} is defined in Eq. (3.9) such that the speed of the last point of s^* in window $j - 1$ is maintained in the last N_c points of s_0 in window j , as illustrated in Fig. 3.1(a).

Also, at $k = jN_c$, kinematic and monotonicity constraints must be enforced between the solution from window $j - 1$ and the solution being computed by Win-SLP in window j . This is achieved as follows (for $k = jN_c$):

$$\begin{aligned}s^*(k - 1) \leq s_i(k) \leq s^*(k - 1) + \frac{F_{max}}{L}T_s \\ \left| \frac{\hat{x}_{d,i-1}(k) - 2\hat{x}_d^*(k - 1) + \hat{x}_d^*(k - 2)}{T_s^2} \right| \leq A_{max} \\ \left| \frac{\hat{y}_{d,i-1}(k) - 2\hat{y}_d^*(k - 1) + \hat{y}_d^*(k - 2)}{T_s^2} \right| \leq A_{max} \\ \left| \frac{\hat{x}_{d,i-1}(k + 1) - 2\hat{x}_{d,i-1}(k) + \hat{x}_d^*(k - 1)}{T_s^2} \right| \leq A_{max} \\ \left| \frac{\hat{y}_{d,i-1}(k + 1) - 2\hat{y}_{d,i-1}(k) + \hat{y}_d^*(k - 1)}{T_s^2} \right| \leq A_{max}\end{aligned}\tag{3.10}$$

Note that $\hat{x}_d^*(k)$ and $\hat{y}_d^*(k)$ are respectively defined as $\hat{x}_d(k)$ and $\hat{y}_d(k)$ evaluated using $s^*(k)$. The same method in Eq. (3.10) can be also used to ensure axis jerk continuity. The ten inequalities in Eq. (3.10) can be concatenated as \mathbf{l}_{cont} and \mathbf{m}_{cont} in Eq. (3.11) by using the relationship in Eq. (2.7) for s_i on $k \in [jN_c, jN_c + 1]$:

$$\mathbf{l}_{cont,j} \cdot \mathbf{s}_i \leq \mathbf{m}_{cont,j}\tag{3.11}$$

Furthermore, let $N_{sys,x}$ and $N_{sys,y}$ respectively represent the lengths of the finite impulse response of \hat{G}_x and \hat{G}_y . Then, $N_{sys} = \max(N_{sys,x}, N_{sys,y})$ determines the number of time steps needed for perturbations in both x, y axis to decay to negligible levels. Let $\hat{e}_{x,i}(k)$ be defined as $\hat{e}_x(k)$ evaluated using $s_i(k)$. Then, the domain of k that the inequality $\hat{e}_{x,i}(k) \leq E_{max}$ is evaluated

at should be $k \in [jN_c - N_{sys}, jN_c + N_p - 1]$. In other words, it also includes the N_{sys} time steps preceding the beginning of the current window j at $k = jN_c$ that contribute to $\hat{e}_{x,i}(k)$. Thus, the constraints on $\hat{e}_{x,i-1}$ are formulated as Eq. (3.12):

$$|\hat{e}_{x,i-1}| = \left| \underbrace{(\mathbf{I} - \hat{\mathbf{G}}_x \mathbf{C}_x)}_{\Gamma_x} \begin{bmatrix} \hat{\mathbf{x}}_d^* \\ \hat{\mathbf{x}}_{d,i-1} \end{bmatrix} \right| \leq \mathbf{E}_{max} \quad (3.12)$$

It can be re-written as Eq. (3.13):

$$\left| \begin{bmatrix} \Gamma_{x,p} & \Gamma_{x,c} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{x}}_d^* \\ \hat{\mathbf{x}}_{d,i-1} \end{bmatrix} \right| \leq \mathbf{E}_{max} \quad (3.13)$$

$$\therefore -\mathbf{E}_{max} - \Gamma_{x,p} \hat{\mathbf{x}}_d^* \leq \Gamma_{x,c} \hat{\mathbf{x}}_{d,i-1} \leq \mathbf{E}_{max} - \Gamma_{x,p} \hat{\mathbf{x}}_d^*$$

where $\Gamma_{x,p}$ and $\Gamma_{x,c}$ represents rows of Γ_x that correspond to $\hat{\mathbf{x}}_d^*$ (past N_{sys} points from jN_c) and $\hat{\mathbf{x}}_{d,i-1}$ (current N_p points from jN_c), respectively; here, $\hat{\mathbf{x}}_d^*$ and $\hat{\mathbf{x}}_{d,i-1}$ are $\hat{x}_d^*(k) \forall k \in [jN_c - N_{sys}, jN_c - 1]$ and $\hat{x}_{d,i-1}(k) \forall k \in [jN_c, jN_c + N_p - 1]$, respectively. A similar equation with $\Gamma_{y,p}$, $\Gamma_{y,c}$, $\hat{\mathbf{y}}_d^*$ and $\hat{\mathbf{y}}_{d,i-1}$ can be written to constrain $\hat{e}_{y,i-1}$.

Finally, FO+SEP using Win-SLP can be represented as the pseudocode in Eq. (3.14):

```

1: Initialize:
    $i = 0$ 
    $j = 0$ 
2: do
3:   if  $j = 0$  then
4:      $\mathbf{s}_0 = \mathbf{s}^e$ 
5:   else
6:     Eq.(9)
7:   end if
8:   do
9:      $i = i + 1$ ;
10:     $\min_{\mathbf{s}} \sum_{k=jN_c}^{jN_c+N_p-1} -s_i(k)$ 
11:    s.t. Eq.(8),(12),(14)
12:    while  $\left| \sum_{k=jN_c}^{jN_c+N_p-1} s_i(k) - \sum_{k=jN_c}^{jN_c+N_p-1} s_{i-1}(k) \right| \geq tol$ 
13:       $\mathbf{s}^*(k) = s_i(k) \forall k \in [jN_c, (j+1)N_c - 1]$ 
14:       $j = j + 1$ ;
15:    while  $s(jN_c + N_c + 1) < 1$ 

```

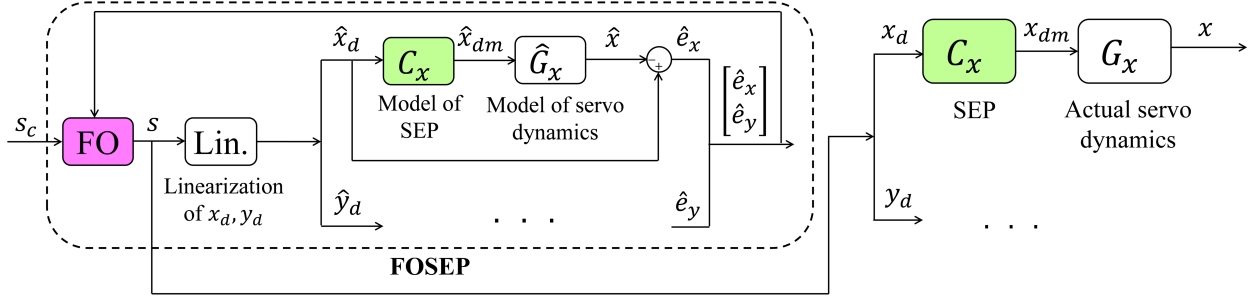


Figure 3.2: Block diagram of FO+SEP using LP (Note: y -component of SEP and servo dynamics are omitted for simplicity)

3.3.3 Guaranteeing feasibility of Win-SLP

Since Eq. (3.14) is defined as a finite horizon optimization with preview length $N_p \ll N$, the solution determined at window j may lead to infeasibility in future windows, with no recourse to generate a feasible solution [70]. Infeasibility could be catastrophic in online implementation of FO+SEP where it is impossible to go back to the past to re-compute a feasible solution. Therefore, to guarantee feasibility using Win-SLP, we develop a scheme whereby at window j , solution s_i must have a complementary backup solution that is feasible $\forall k \in [jN_c, N - 1]$.

The proposed approach is illustrated in Fig. 3.3 and summarized as a flow chart in Fig. 3.4. In Fig. 3.3(a), an optimal trajectory $s^*(k) \forall k \in [(j-1)N_c, (j-1)N_c + N_p - 1]$ (represented as ①), as well as a feasible backup solution $\tilde{s}(k) \forall k \in [(j-1)N_c + N_p, (j-1)N_c + N_p + N_{sys} - 1]$ (represented as ②) is created to smoothly patch onto ①. When the window recedes by N_c as in Fig. 3.3(b) and k is updated from $k = (j-1)N_c$ to jN_c , the trajectory transitions to the previously-generated backup (represented as ③) if either of the below two attempts encounters infeasibility:

- **Attempt 1:** Optimization of $s_i(k) \forall k \in [jN_c, jN_c + N_p - 1]$ (represented as ④) for any SLP step i ;
- **Attempt 2:** Generation of a feasible backup solution $\tilde{s}(k) \forall k \in [jN_c + N_p, jN_c + N_p + N_{sys} - 1]$ (represented as ⑤)

This iterative process is possible by the definition of backup solution s^e in Eq. (2.7) in Chapter 2; at $k = 0$, s^e serves as a kinematically conservative profile that satisfies constraints in Eq. (3.14).

The reason for generating the backup solution \tilde{s} up to N_{sys} time steps beyond the end of window j is that the transition from ① to ③ can create a transient effect on the actual response of the system dynamics x and y in Fig. 3.2. These transients can cause \hat{e}_x and \hat{e}_y to violate their tolerance limit E_{max} . However, this perturbation will die down in both axis within N_{sys} time steps; thus, if there exists a feasible trajectory $s^*(k) \forall k \in [jN_c + N_p, jN_c + N_p + N_{sys} - 1]$, there also will exist

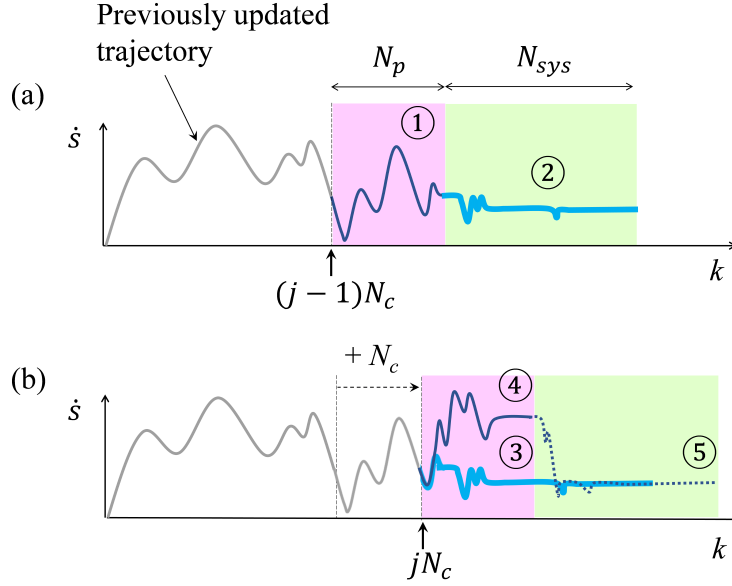


Figure 3.3: Overview of scheme to guarantee feasibility of Win-SLP consisting of: (a) generation of backup solution in window $j - 1$; and (b) adoption of first N_p points of backup solution (i.e., ③) if no optimal solution ④ or backup solution ⑤ can be found in window j

a feasible trajectory on the rest of the trajectory, i.e., $\forall k \in [jN_c + N_p + N_{sys}, N - 1]$, due to the availability of s_c which is the ultimate backup solution. Therefore, guaranteeing a feasible solution on N_{sys} backup points is enough to guarantee feasibility within the entire trajectory $s^*(k)$ $k \in [0, N - 1]$.

Furthermore, the smoothness of the transition from ① to ③ can be guaranteed by how \tilde{s} , as well as s^e , is generated: First, the values of the final displacement, feedrate, and acceleration at $s^*((j-1)N_c + N_p - 1)$ (the last point of ①) are identified and represented as s_{in} , f_{in} , and a_{in} , respectively. Then, a standard trapezoidal acceleration profile (TAP) [71] is created from $s = s^*((j-1)N_c + N_p - 1)$ to $s = 1$. TAP chooses a conservative set of limits on the feedrate and acceleration as $F_{max,c}$ and $A_{max,c}$, as well as initial boundary conditions as s_{in} , f_{in} , and a_{in} , respectively, to create \tilde{s} . Also, to ensure smoothness at the axis level, the axis velocity profiles of $\hat{\mathbf{x}}_d$ and $\hat{\mathbf{y}}_d$ evaluated using \tilde{s} are filtered with a moving average filter $H(z)$ [72] with time constant $\tau = F_{max,c}/A_{max,c}$. By choosing the first N_{sys} points of \tilde{s} , $\tilde{s}_k \forall k \in [(j-1)N_c + N_p, (j-1)N_c + N_p + N_{sys} - 1]$ is generated such that it connects to ① smoothly. Then, a check is made on whether $\tilde{s}(k) \forall k \in [(j-1)N_c + N_p, (j-1)N_c + N_p + N_{sys} - 1]$ respects the limits to determine whether Attempt 2 (described above) is feasible in window $j - 1$ as in Eq. (3.15):

$$-E_{max} - \Gamma_{x,p}\hat{\mathbf{x}}_d^* \leq \Gamma_{x,c}\tilde{\mathbf{x}}_d \leq E_{max} - \Gamma_{x,p}\hat{\mathbf{x}}_d^* \quad (3.15)$$

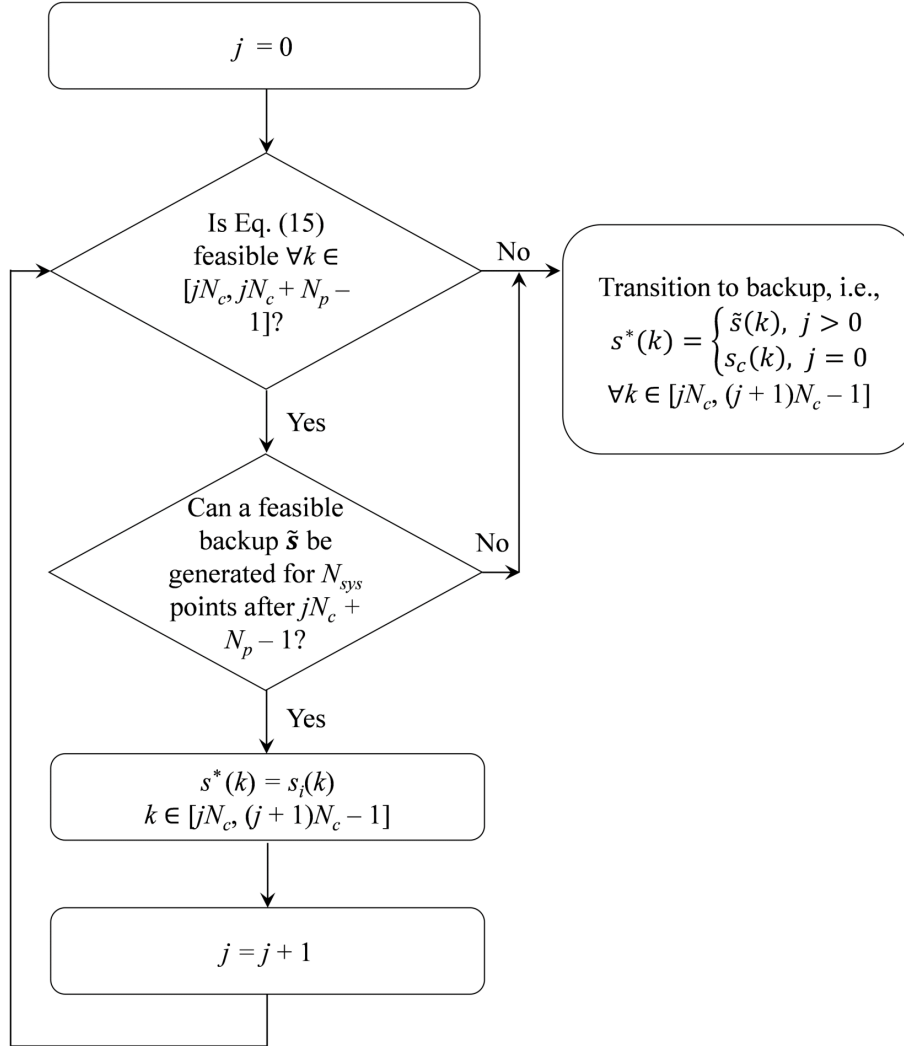


Figure 3.4: Flow chart of scheme for guaranteeing feasibility of FO+SEP using Win-SLP discussed in Section 3.3

where $\tilde{\mathbf{x}}_d$ represents $\hat{\mathbf{x}}_d$ defined at $\tilde{s}(k) \forall k \in [(j-1)N_c + N_p, (j-1)N_c + N_p + N_{sys} - 1]$ with $H(z)$ applied, and $\hat{\mathbf{x}}_d^*$ represents $\hat{\mathbf{x}}_d^*(k) \forall k \in [(j-1)N_c + N_p - N_{sys}, (j-1)N_c + N_p - 1]$.

3.4 Validation of proposed approach via simulation

The goal of this section is to validate the accuracy, computational efficiency and guaranteed feasibility of the proposed Win-SLP approach for FO+SEP via a series of simulations. For this purpose, a 2nd order linear system dynamics in the x and y axis has been selected, as shown in Eq. (3.16):

$$G_x = G_y = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (3.16)$$

Its parameters are chosen as $\omega_n = 2\pi \times 50 = 314.16$ rad/s and $\zeta = 0.1$. Then, G_x and G_y are discretized with sampling time $T_s = 1$ ms and approximated as an FIR filter by stacking its truncated impulse response as Eq. (3.17):

$$\mathbf{G}_x \approx \begin{bmatrix} \mathbf{x}_{imp} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{x}_{imp} & \vdots \\ \vdots & \vdots & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & & \mathbf{x}_{imp} \end{bmatrix} \quad (3.17)$$

where \mathbf{x}_{imp} is the truncated impulse response column vector of \mathbf{G}_x with length N_{sys} , which is identified as 400 for the system in Eq. (3.16). For SEP, $\mathbf{C}_x = \mathbf{C}_y$ is generated via the filtered B spline (FBS) approach [73] because of its effectiveness and versatility in handling any type of linear system dynamics [74].

A 5th degree B spline with uniform knot vector is used to generate \mathbf{C}_x using the FBS method. The number of trajectory points in the B spline is selected as the length of s_c , and the ratio of number of B spline basis functions to the length of the trajectory is 1:20. In Win-SLP, $N_p = 50$ and $N_c = 15$ are used so that adjacent windows are overlapped by 35 points; $tol = 10^{-3}$ is used for SLP termination threshold.

All simulations are conducted using circles of radius R . The simulations are conducted using MATLAB[®] R2019a on a Windows PC with Intel Core i7-8750H CPU and 16 GB RAM. In each simulation, the initialization trajectory, s_c , is generated using trapezoidal acceleration profile (TAP) [71] with conservative kinematic limits as $F_{max,c} = 30$ mm/s, $A_{max,c} = 0.5$ m/s², $J_{max,c} = 5$ m/s³. The following algorithms are used implement FO+SEP:

- Case A: LP (performed in one batch);

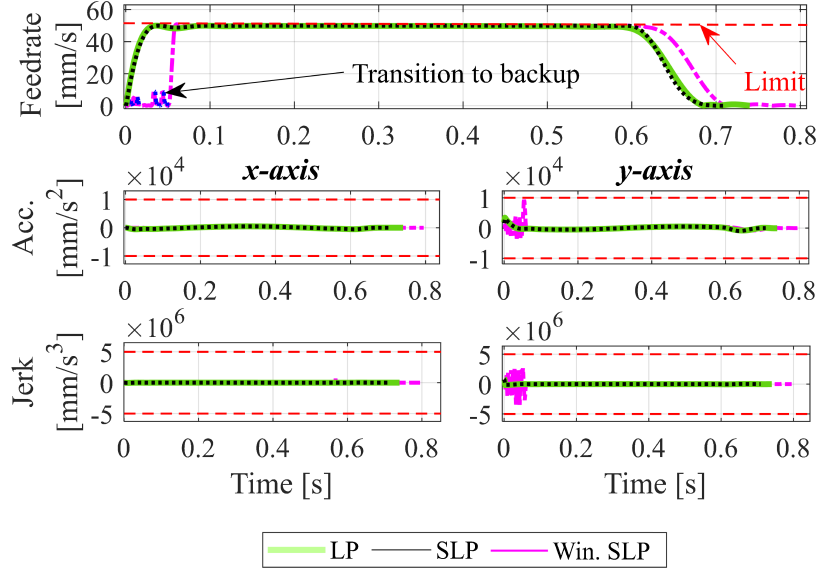


Figure 3.5: Feedrate, acceleration and jerk profiles of trajectories generated by FO+SEP using LP, SLP and Win-SLP (proposed) for short circular toolpath ($R = 5$ mm). The switching of the Win-SLP solution to the backup solution at the marked instances enables it to guarantee feasibility at the cost of optimality (as long as the starting solution, s_c , is feasible)

- Case B: SLP (performed in one batch); and
- Case C: Win-SLP (with smooth switching)

In all cases, $F_{max} = 50$ mm/s, $A_{max} = 10$ m/s², $J_{max} = 5000$ m/s³, which are borrowed from the prior work [1]. For the first set of simulations, a short circular toolpath ($R = 5$ mm) with tight tracking error constraints of $E_{max} = 3$ μ m are used. Note that, with $R = 5$ mm, s_c yields maximum tracking error of 1.24 μ m with a cycle time of 1.58 s, meaning that it is feasible under all limits of F_{max} , A_{max} , J_{max} , and E_{max} .

Fig. 3.5 shows the commanded feedrate, acceleration, and jerk profiles of the three cases. Fig. 3.6 shows the simulated tracking error profiles in both x and y axis, \hat{e}_x and \hat{e}_y , which are simulated using the discretized version of the dynamics in Eq. (3.16). LP violates the tracking error tolerance in both x and y axis due to linearization errors, because the LP solution is linearized with s_c which is significantly different from the optimal solution. Conversely, SLP and Win-SLP satisfy all the kinematic and tracking error constraints, which highlights their accuracy relative to LP, and why LP is considered unacceptable for FO+SEP.

The importance of smooth switching in guaranteeing the feasibility of Win-SLP can also be seen from Fig. 3.5 and 3.6. Observe that the Win-SLP solution has to switch several times between the optimal and backup solution in order to maintain feasibility. The implication is that, without the backup solution, Win-SLP would fail to yield a feasible solution. The cycle time of the LP, SLP

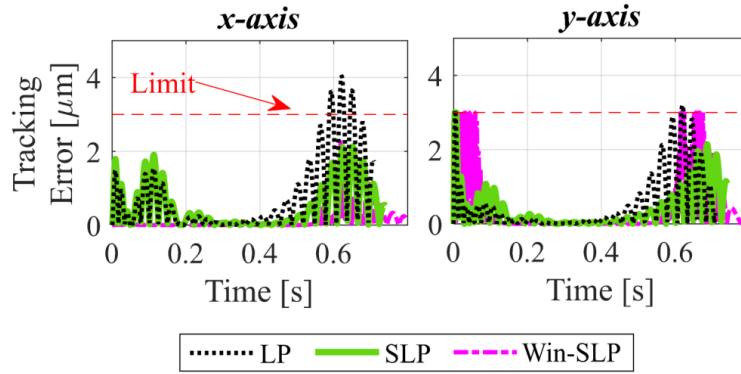


Figure 3.6: Simulated axis tracking error profiles of LP, SLP, and Win-SLP (proposed) showing the inaccuracy of LP and the accuracy of SLP and Win-SLP in enforcing constraints

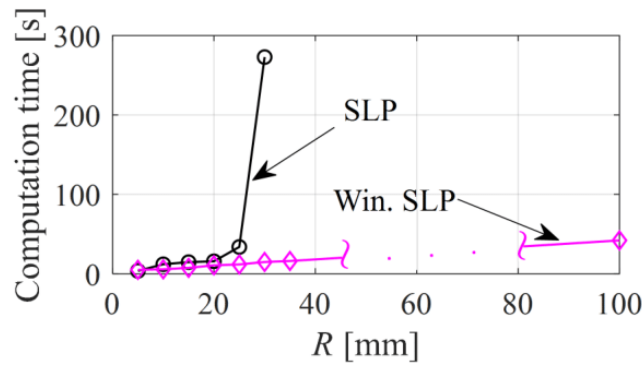


Figure 3.7: Computation time vs. radius R of the circular trajectory for FO+SEP using SLP and Win-SLP (proposed) showing the superior computational efficiency of Win-SLP relative to SLP for long toolpaths

Table 3.1: Cycle time and constraint satisfaction accuracy of FO+SEP using LP, SLP and Win-SLP for short toolpath ($R = 5$ mm)

	LP	SLP	Win-SLP (Proposed)
Cycle time [s]	0.704	0.737	0.795
Satisfies Constraints Accurately?	No	Yes	Yes

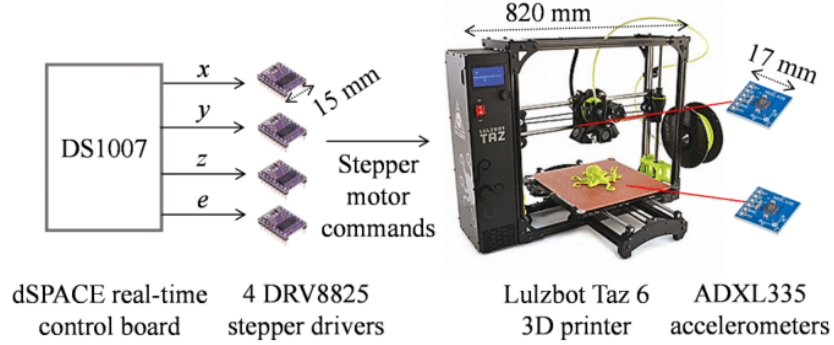


Figure 3.8: Experimental setup: Lulzbot Taz 6 3D Printer

and Win-SLP solutions for FO+SEP are summarized in Table 3.1, along with their accuracy in enforcing constraints. Though LP provides the shortest cycle time, it is inaccurate. SLP and Win-SLP are accurate but the cost of switching multiple times to the backup solution is that Win-SLP has slightly (7.9%) longer cycle time than SLP.

The advantage of Win-SLP manifests itself as the length of the toolpath grows, as is typical in practice. The computational efficiency of SLP relative to Win-SLP degrades rapidly with increasing toolpath length. To demonstrate this, a second set of simulations is carried out on circular toolpaths with R ranging from 5 mm to 100 mm; $E_{max} = 3 \mu\text{m}$ in all cases. As shown in Fig. 2.8, the computation time for SLP increases exponentially as R grows larger (which in return increases N), whereas Win-SLP achieves nearly linear relationship because of its fixed window size N_p . As R reaches 35 mm, SLP fails due to the computer running out of memory, whereas Win-SLP is able to carry out the optimization up to $R = 100$ mm (and beyond, not shown). Together, these two sets of simulations demonstrate the accuracy and feasibility guarantees of the proposed Win-SLP approach for FO+SEP, together with its computational efficiency for long toolpaths.

3.5 Experimental validation

3.5.1 Experimental setup

Long toolpaths are commonplace in manufacturing. Therefore, through experiments carried out in this section, we seek to demonstrate the benefit of the proposed Win-SLP approach to FO+SEP on improving productivity (compared to independent approach of feedrate optimization and servo error pre-compensation, i.e., FO then SEP) when both are applied to long toolpaths. A Lulzbot Taz 6 3D printer shown in Fig. 3.8, is used for the experiments. The optimization algorithms are implemented offline on a dSPACE 1007 real-time control board running at 1 kHz sampling rate, connected to DRV8825 stepper motor drivers for x , y , z , and e - (extruder) axes stepper motors.

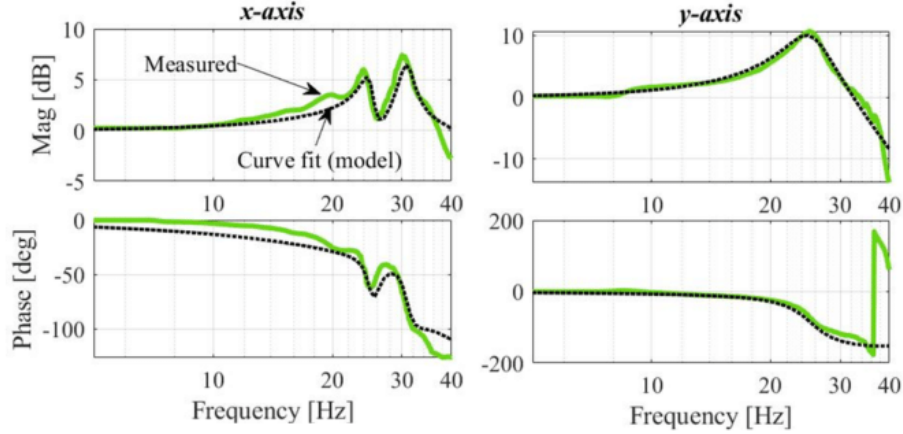


Figure 3.9: Measured and curve fitted FRFs of x and y axes of 3D printer

The setting is used for conducting experiments in the following Section 3.5.2 and 3.5.3.

To execute both FO+SEP and FO then SEP, the x and y axis servo dynamics of the printer must be measured in the form of frequency response functions (FRFs) and modeled, via curve fitting, as \hat{G}_x and \hat{G}_y . Fig. 2.10 shows the measured and modeled FRFs of the x - and y -axes of the printer. The input of each FRF is swept sine acceleration commands to the stepper motor, and the output is relative acceleration between the build plate and nozzle measured using two ADXL335 tri-axial accelerometers. The discrete-time transfer function representation of \hat{G}_x and \hat{G}_y is shown in Eq. (3.18).

$$\begin{aligned}
 & \hat{G}_x \\
 &= \frac{0.026z^5 - 0.078z^4 + 0.055z^3 + 0.042z^2 - 0.069z + 0.023}{z^6 - 5.652z^5 + 13.4z^4 - 17.07z^3 + 12.31z^2 - 4.768z + 0.775} \\
 & \hat{G}_y \\
 &= \frac{0.199z^3 - 0.349z^2 + 0.174z - 2.14 \times 10^{-27}}{z^4 - 1.934z^3 + 0.958z^2 - 2.733 \times 10^{-17}z + 1.947 \times 10^{-34}}
 \end{aligned} \tag{3.18}$$

3.5.2 Benchmarking to determine E_{max}

The desired path is chosen as a butterfly curve [75] shown in Fig. 3.10, where \hat{x}_d and \hat{y}_d are parameterized in s using quintic spline interpolation with minimal feedrate fluctuation [76]. Similar to Section 4, s^e is selected as a TAP position trajectory with conservative kinematic limits as $F_{max,c} = 30$ mm/s, $A_{max,c} = 0.5$ m/s², and $J_{max,c} = 5$ m/s³; it is smoothed at the axis level using a digital filter $H(z)$ with time constant $\tau = 0.06$ s. These conservative kinematic limits are known from prior work [1, 77] to give acceptable quality. Therefore, we use the conservative

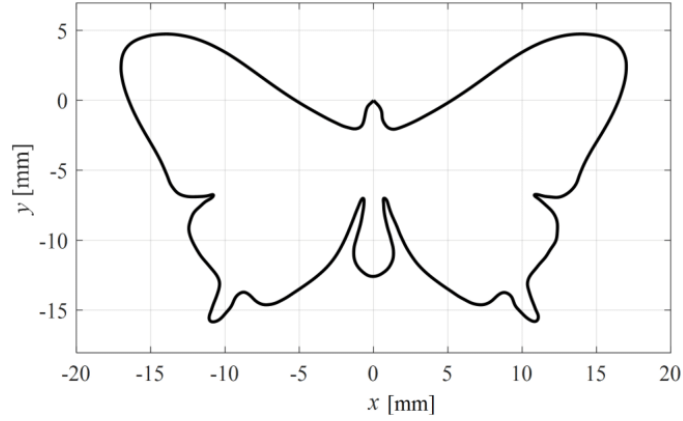


Figure 3.10: Desired path of butterfly curve

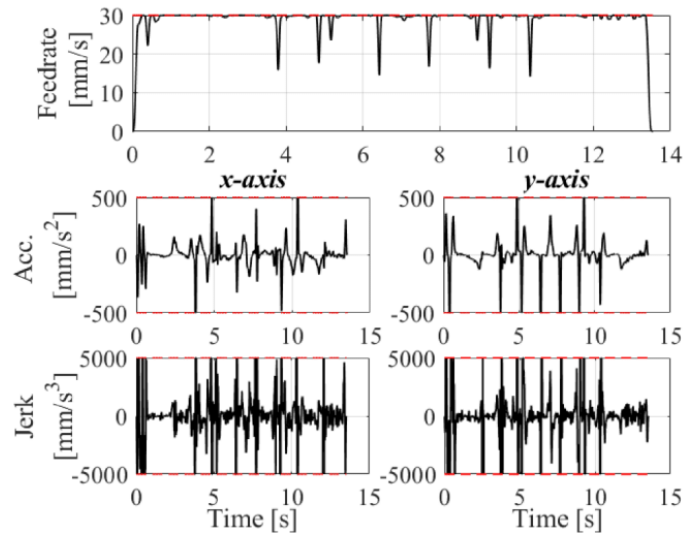


Figure 3.11: Commanded feedrate, acceleration and jerk profiles of conservative TAP

TAP trajectory generated using them to determine E_{max} for FO+SEP and FO then SEP. Fig. 3.11 shows the feedrate, axis acceleration, and axis jerk profiles of the conservative TAP command. Fig. 3.14 shows the simulated x and y axis tracking errors obtained by applying the conservative TAP command to the transfer functions of the 3D printer given in Eq. (3.18). The conservative TAP yields maximum tracking error of $127.4 \mu\text{m}$ and $55.2 \mu\text{m}$ for the x and y axis, respectively. Therefore, $E_{max} = 127.4 \mu\text{m}$ is chosen as the tolerance limit that must be satisfied by FO+SEP and FO then SEP.

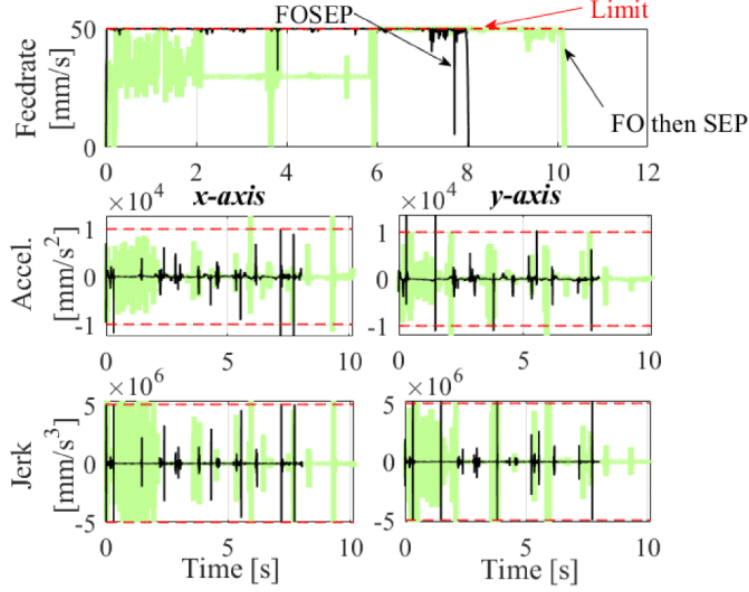


Figure 3.12: Commanded feedrate, acceleration and jerk profiles of FO then SEP, FO+SEP (Proposed), and TAP with aggressive kinematics (Aggr.)

3.5.3 Optimization results using FO+SEP and FO then SEP using Win-SLP

We compare FO then SEP and the proposed FO+SEP using Win-SLP with a goal to achieving $E_{max} = 127.4 \mu\text{m}$ with the shortest cycle time. To do this, aggressive constraints from our prior work [1], namely, $F_{max} = 50 \text{ mm/s}$, $A_{max} = 10 \text{ m/s}^2$, and $J_{max} = 5000 \text{ m/s}^3$ are imposed on both FO+SEP and FO then SEP. In FO then SEP, $C_x = C_y = \mathbf{I}$ are selected (i.e., tolerance constraints are imposed without SEP). On the other hand, in FO+SEP, C_x and C_y are generated using FBS approach [73, 74], where a 5th degree B-spline with uniform knot vector, and $n = 500$ control points are used. Both FO+SEP and FO then SEP are initialized at $i = 1$ and $j = 0$ by using the conservative TAP in Fig. 3.11 as s^e ; $N_p = 50$, $N_c = 20$ and $tol = 10^{-3}$ are used for Win-SLP on both FO+SEP and FO then SEP, and N_{sys} is identified as 450 for the system dynamics in Eq. (3.18). Note that in both FO+SEP and FO then SEP, servo errors are compensated after the optimization using the C_x and C_y generated using FBS.

Fig. 3.12 shows the commanded feedrate, acceleration and jerk profiles of FO+SEP and FO then SEP. Fig. 3.13 shows the resulting tracking error simulated based on the system dynamics in Eq. (3.18). As a reference, the aggressive kinematic limits are used to generate an aggressive TAP trajectory shown in Fig. 3.12. The resultant tracking errors after SEP are also shown in Fig. 3.13. Notice that, because the aggressive TAP trajectory is not optimized, it results in violations of the tracking error limit. It is this inability to guarantee that tolerance limits will be respected that leads to the use of conservative TAP profiles in practice. Conversely, notice that both FO+SEP and

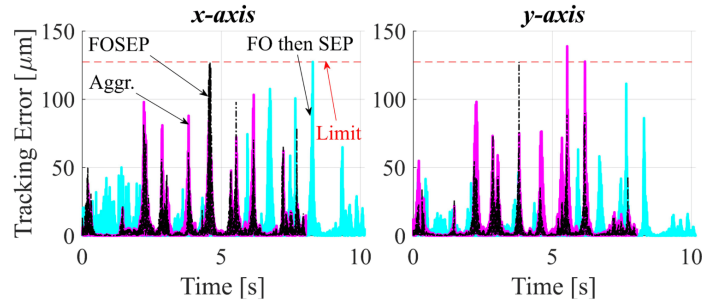


Figure 3.13: Simulated tracking error profiles of FO then SEP, FO+SEP (proposed), and TAP with aggressive kinematics (Aggr.)

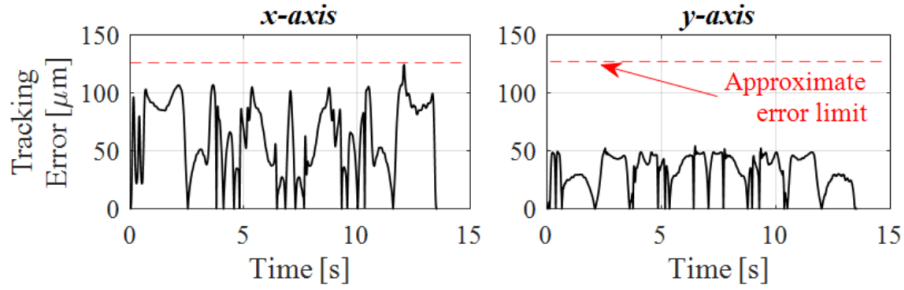


Figure 3.14: Simulated tracking error of conservative TAP on x and y axis and the approximate error limit E_{max}

FO then SEP enforce the kinematic and tracking error constraints. However, FO then SEP has to slow down by transitioning to the backup solution many times from 0 s to 2 s because the error constraint in Eq. (3.13) with $C_x = C_y = \mathbf{I}$ has narrower feasible region than FO+SEP due to the independent application of FO and SEP. On the other hand, FO+SEP is able to stay close to the maximum feedrate throughout the motion with only one transition to the backup at 7.7 s. As a result, FO+SEP completes the motion in 8.02 s, which is 21% faster than FO then SEP at 10.15 s, as summarized in Table 3.2. The computation time for FO+SEP and FO then SEP are 37.76 s and 34.91 s, respectively.

Table 3.2: Cycle and computation time of FO then SEP and FO+SEP using Win-SLP approach

	FO then SEP	FO+SEP (Proposed)
Cycle time [s]	10.15	8.02
Computation time [s]	34.91	37.76

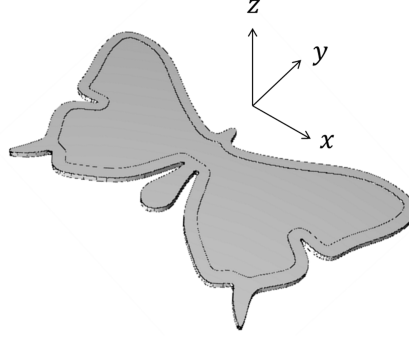


Figure 3.15: CAD model of the butterfly plate of height 1.2 mm with outer contour defined by the path in Fig. 2.11

3.5.4 3D print results

To further validate our findings, a butterfly-shaped 3D part whose CAD model [78] is shown in Fig. 3.15 is printed using the Taz 6 3D printer of Fig. 3.8. The butterfly curve shown in Fig. 3.10 is used to define the outer contour for the part. To parameterize the entire trajectory in s , which consists of curves and lines stacked in z -axis, first the CAD model in Fig. 3.15 is converted to an STL model using a commercial slicing firmware. Then, each layer in z axis is further divided into curves and lines, where the curves are parameterized in s using the same method described in Section 5.2 and lines are indexed as $w = 1, 2, \dots$ and parameterized individually by identifying the slope $l_m(w)$ and y -intercept $l_y(w)$ using Eq. (3.19).

$$\begin{aligned}\hat{\mathbf{x}}_d &= \frac{1}{\sqrt{l_m(w)^2 + 1}} \mathbf{s} \\ \hat{\mathbf{y}}_d &= \frac{l_m(w)}{\sqrt{l_m(w)^2 + 1}} \mathbf{s} + l_y(w)\end{aligned}\quad (3.19)$$

However, Eq. (3.19) only maintains axis-level continuity within a given line w ; as soon as it shifts to the next line $w + 1$, axis acceleration or jerk may violate their limits of A_{max} , J_{max} due to the sharp corner at junction. Therefore, feedrate, or $L \frac{D[s]}{T_s}$, is lowered between two adjacent lines until discontinuity at axis level disappears. Then, all sets of curves and lines are optimized using FO+SEP and FO then SEP using Win-SLP, as discussed in Eq. (3.14), with the same set of limits, F_{max} , A_{max} , J_{max} , E_{max} , and Win-SLP parameters, N_p , N_c , tol , as Section 5.3. As a benchmark, a conservative TAP which uses the same set of slow kinematic limits and moving average filter $H(z)$ as in Section 3.5.2 is applied for comparison.

Fig. 3.16 shows the printed results of using conservative TAP, as well as FO+SEP and FO then SEP both computed using the proposed Win-SLP. The computational time for the three methods




Algorithm	Conservative	FO then SEP	FOSEP (Proposed)
Top view			
Print time (min)	18:30	15:36	11:45

Figure 3.16: Top view of printed butterfly plate using Conservative TAP, FO then SEP, and FO+SEP. Both FO then SEP and FO+SEP were computed using the proposed Win-SLP approach

Table 3.3: Computation time for FO then SEP and FO+SEP for the butterfly plate in Fig. 3.16

	FO then SEP	FO+SEP (Proposed)
Computation time [s]	974.98	835.51

are summarized in Table 3. FO+SEP saves 25% in cycle time compared to FO then SEP while both maintaining similar quality as the part printed using conservative TAP. The ability to print a very long toolpaths using FO+SEP and FO then SEP demonstrates the practicality of the proposed Win-SLP approach. Moreover, the reduction in cycle time highlights the benefits of applying Win-SLP to FO+SEP rather than to FO then SEP.

3.6 Conclusions

This chapter has introduced a method to improve the accuracy in enforcing nonlinear constraints and the computational efficiency of simultaneous FO and SEP (i.e., FO+SEP) applied to long toolpaths. The proposed method, dubbed Win-SLP, achieved windowed sequential linear programming optimization, with feasibility guarantees using smooth transition between the optimal solution and a conservative backup solution.

Win-SLP, which sequentially optimizes the path parameter within a limited horizon length, is formulated and compared with full-preview (i.e., one-shot) LP and SLP. While both SLP and Win-SLP improve accuracy in nonlinear axis-level constraints, Win-SLP shows superiority over SLP in handling longer toolpaths. It is shown in the simulations that Win-SLP achieves both accurate and computationally efficient approach for FO+SEP, while guaranteeing feasibility.

Furthermore, compared to the standard practice of sequential FO then SEP, FO+SEP relaxes the error tolerance constraints in FO, allowing shorter cycle time without violating tolerance constraints. Experiments carried out on a 3D printer using the proposed Win-SLP approach yielded

up to 25% cycle time reduction using FO+SEP compared to FO then SEP, subject to the same tolerance and kinematic constraints.

CHAPTER 4

Intelligent Feedrate Optimization using an Uncertainty-aware Physics-informed Data-driven Digital Twin

Chapters 2 and 3 studied feedrate optimization with servo error constraints using physics-based servo models to improve cycle time while satisfying tolerance constraints. However, they cannot accurately constrain actual servo error when unmodeled dynamics or uncertainties exist in motion dynamics or external disturbances. Hence, their efficacy to maximize feedrate with servo error constraints is very limited. Moreover, they do not estimate the uncertainty of the prediction, which may cause the servo error constraints to be violated in the presence of high uncertainty due to lack of training data or sudden change of operating conditions. This chapter addresses the issue by using an uncertainty-aware physics-informed data-driven digital twin, which is a virtual representation of physical system built on a bi-directional link between simulation and actual data, to optimize feedrate with servo error constraints in a robust way.

Section 4.1 introduces the framework of the proposed intelligent feedrate optimization using digital twin with quantified uncertainty. Section 4.2 describes the methodology of the proposed digital twin for predicting contour error distribution. Section 4.3 provides a formulation for the feedrate optimization with contour error constraint with desired stringency. Section 4.4 numerically validates the proposed method via a desktop 3D printer. Section 4.5 experimentally validates the proposed method via a desktop 3D printer and a CNC machine tool prototype. Section 4.6 concludes the chapter and discusses future work.

Note that there are nomenclature changes made in this chapter compared to previous Chapters 2 and 3, which are listed as follows:

- The desired trajectory and its linearized version are denoted as x_d and \hat{x}_d respectively in Section 2.2 for Chapters 2 and 3, whereas they are denoted as X_d and x_d in Section 4.3 for Chapter 4

- The optimized desired trajectory generated by feedrate optimization is denoted as \hat{x}_d^* in Section 3.3.2 for Chapter 3, whereas it is denoted as x_c in Section 4.2 for Chapter 4
- The servo error is denoted as \hat{e}_x in Section 2.3 for Chapters 2 and 3, whereas a similar notation e_x is used to denote physics-based prediction error in Section 4.2.1 for Chapter 4
- The path parameter and its estimated linearization points are denoted as s and s^e respectively in Section 2.2 for Chapters 2 and 3, whereas they are denoted as p and \bar{p} in Section 4.3 for Chapter 4
- The lifted (matrix) representation of system dynamics corresponding to the current and past trajectories are denoted as $\Gamma_{x,c}$ and $\Gamma_{x,p}$ respectively in Section 3.3.2 for Chapter 3, whereas they are denoted as $\Phi_{x,c}$ and $\Phi_{x,p}$ in Section 4.3 for Chapter 4
- Lastly, the lengths of window (batch) and the finite impulse response of system dynamics are denoted as N_p and N_{sys} respectively in Section 3.3.2 for Chapter 3, whereas they are denoted as n_w and n_h in Section 4.2 for Chapter 4

This chapter is based on the following manuscript in preparation:

- Kim H, Kontar RA, Okwudire CE. Intelligent Feedrate Optimization using a Physics-informed Data-driven Digital Twin with Quantified Uncertainty

4.1 Framework of intelligent feedrate optimization using digital twin with quantified uncertainty

The framework for the proposed intelligent feedrate optimization using digital twin with quantified uncertainty is depicted in Figure 4.1. 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.

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 violation of the contour error tolerance, hence the part quality, as illustrated in Figure 4.2(a). Understanding 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

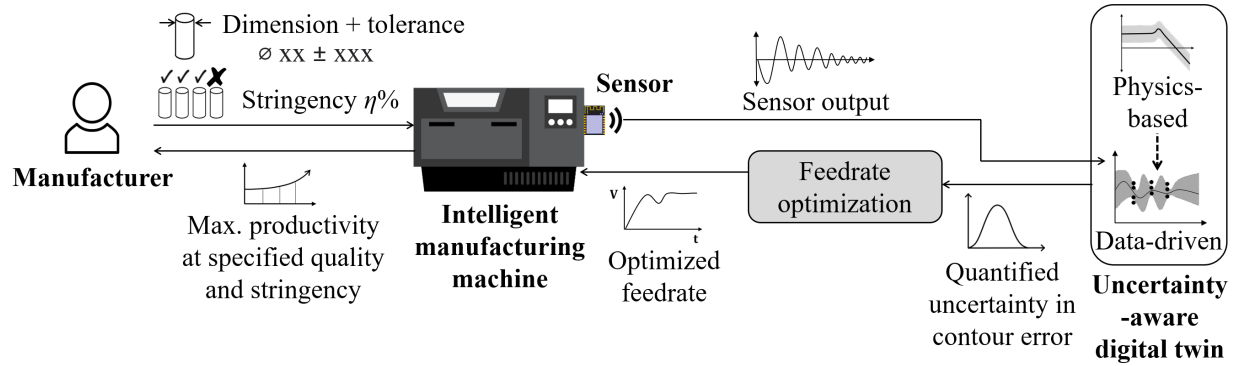


Figure 4.1: Diagram of intelligent feedrate optimization using uncertainty-aware digital twin. A manufacturer provides a part tolerance and stringency (i.e., the tolerance for quality constraint violation under uncertainty). The intelligent machine leverages an uncertainty-aware digital twin to optimize feedrate while satisfying the tolerance and stringency requirements

constraints. while another may be fine with 95% of the parts satisfying the same constraints. This reflects their tolerance for quality constraint violations under uncertainty, which we term stringency in this chapter. We propose that, instead of relying on trial-and-error, manufacturers can impose a desired stringency $\eta\%$ of the given tolerance by incorporating the uncertainty of the contour error prediction as shown in Figure 4.2(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 the data-driven model using the machine's sensor measurements to learn the unknown uncertainty. The digital twin predicts and quantifies the uncertainty of the contour error, which is used in the 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 the feedrate optimization.

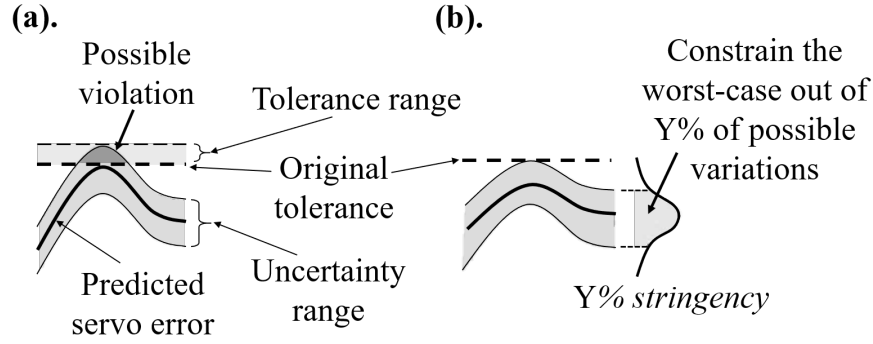


Figure 4.2: (a) Need for a tolerance range due to violation of error tolerance in presence of uncertainties, (b) Proposed method of feedrate optimization with desired tolerance stringency using quantified uncertainty

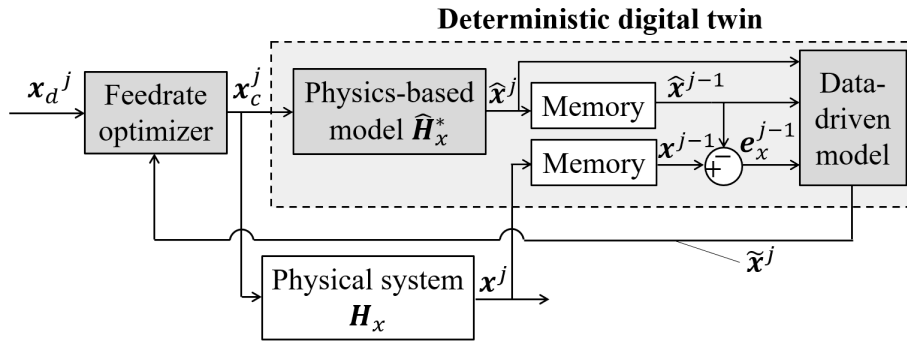


Figure 4.3: Flowchart of intelligent feedrate optimization using deterministic digital twin with physic-based and data-driven servo models (y -axis omitted for simplicity)

4.2 Methodology for contour error prediction using an uncertainty-aware digital twin

4.2.1 Overview of prediction of contour error using a deterministic digital twin

A flowchart of the intelligent feedrate optimization using a deterministic digital twin based on the previous work in [3] is depicted in Figure 4.3. Note that the internal model in [3] is removed in this chapter. Small batches (i.e., look-ahead windows) x_d^j of a desired position trajectory, with window length n_w , are fed into an intelligent feedrate optimizer to produce the optimized motion command, x_c^j where $j = 0, 1, 2, \dots$, represents the batch index. The optimized motion commands are sent to the servo system H_x to produce actual position x^j . The servo system is composed of a servo error pre-compensation C_x followed by machine dynamics G_x , i.e., $H_x = G_x C_x$.

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 [79]. The hybrid model takes input \mathbf{x}_c^j and predicts the actual position $\hat{\mathbf{x}}^j$ using a stable, nominal (or representative) physics-based model $\hat{\mathbf{H}}_x^*$. The predictions $\hat{\mathbf{x}}^j$ do not capture the effects of unmodeled dynamics and external disturbances. Therefore, the prediction error (delayed by one batch) of $\hat{\mathbf{H}}_x^*$ is computed as $\mathbf{e}_x^{j-1} = \mathbf{x}^{j-1} - \hat{\mathbf{x}}^{j-1}$ and combined with $\hat{\mathbf{x}}^j$ and $\hat{\mathbf{x}}^{j-1}$ to be fed into a data driven model to generate an improved prediction $\tilde{\mathbf{x}}^j$ which is used for constraining contour errors in the feedrate optimization.

Each element $\tilde{x}(t)$ of $\tilde{\mathbf{x}}$, where $t = 0, T_s, 2T_s, 3T_s, \dots$ represents discrete time at sampling interval T_s , is modeled as

$$\tilde{x}(t) = \boldsymbol{\beta}^T \boldsymbol{\psi}_t \quad (4.1)$$

where $\boldsymbol{\psi}_t$ is the deterministic feature vector and $\boldsymbol{\beta}$ is the weight vector that is learned using linear regression. The sub elements of $\boldsymbol{\psi}_t$ are given by

$$\boldsymbol{\psi}_t = \left[\underbrace{1}_{:=\psi_{t1}} \quad \underbrace{\hat{x}(t - n_2 T_s) \cdots \hat{x}(t)}_{:=\psi_{t2}} \quad \underbrace{e_x(t - n_3 T_s) \cdots e_x(t - T_s)}_{:=\psi_{t3}} \right]^T \quad (4.2)$$

The sub elements ψ_{t1} , ψ_{t2} and ψ_{t3} were contained in the hybrid model of [79]. They respectively represent a bias term, the past n_2 and current time steps of $\hat{\mathbf{x}}$, and the past n_3 time steps of \mathbf{e}_x , where n_2 and n_3 are user defined.

$\tilde{\mathbf{x}}^j$ at the j -th batch is predicted based on weight $\boldsymbol{\beta}$ from the previous batch $j - 1$, which is trained as follows. For the 0-th batch, i.e., $t = 0, T_s, \dots, (n_w - 1)T_s$, the weight vector $\boldsymbol{\beta}$ and its covariance matrix \mathbf{P} are initialized using ridge regression with a regularization factor λ as

$$\begin{aligned} \boldsymbol{\beta} &= (\lambda \mathbf{I} + \boldsymbol{\psi}_t \boldsymbol{\psi}_t^T)^{-1} \boldsymbol{\psi}_t x(t) \\ \mathbf{P} &= (\lambda \mathbf{I} + \boldsymbol{\psi}_t \boldsymbol{\psi}_t^T)^{-1} \end{aligned} \quad (4.3)$$

For the rest of the batches $j = 1, 2, \dots$, i.e., $t = n_w T_s, (n_w + 1)T_s, \dots$, $\boldsymbol{\beta}$ and \mathbf{P} are corrected via recursive least-squares using a forgetting factor f_0 as

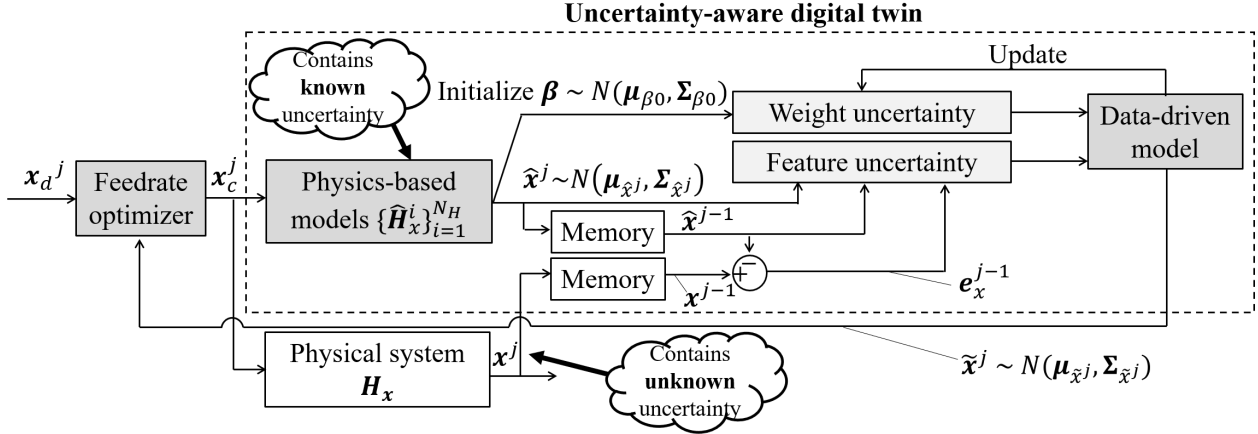


Figure 4.4: Flowchart of intelligent feedrate optimization using an uncertainty-aware digital twin (with physics-based and data-driven servo models, y -axis omitted)

$$\begin{aligned}
 \beta &\leftarrow \beta + \mathbf{k}(x(t) - \beta^T \psi_t) \\
 \mathbf{P} &\leftarrow \frac{1}{f_0} (\mathbf{P} - \mathbf{k} \psi_t^T \mathbf{P}) \\
 \text{where } \mathbf{k} &= \mathbf{P} \psi_t (f_0 + \psi_t^T \mathbf{P} \psi_t)^{-1}
 \end{aligned} \tag{4.4}$$

Using the final weight in batch $j - 1$ to substitute for β , \tilde{x}^j can be predicted using the feature vector ψ_t formulated by Eq. (4.2). 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 ψ_{t3} , e_x is approximated using predicted values \tilde{x} , i.e.,

$$e_x = x - \hat{x} \approx \tilde{x} - \hat{x} \tag{4.5}$$

The same procedure is applied to y -axis to predict \tilde{y} . Lastly, the contour error $\tilde{\epsilon}$ can be estimated from the predicted axis tracking errors $\tilde{\epsilon}_x = x_d - \tilde{x}$ and $\tilde{\epsilon}_y = y_d - \tilde{y}$, using a linear approximation [56] as

$$\tilde{\epsilon} = -\sin(\theta)\tilde{\epsilon}_x + \cos(\theta)\tilde{\epsilon}_y \tag{4.6}$$

where θ is inclination angle of the curve (x_d, y_d) .

4.2.2 Prediction and uncertainty quantification of contour error using physics-informed data-driven digital twin

The accuracy of the predictions of the physics-based and data-driven servo models in Section 4.2.1 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 exploited. A flowchart of the proposed intelligent feedrate optimization using the uncertainty-aware digital twin is given in Figure 4.4. The key difference between Figure 4.3 and Figure 4.4 is that known uncertainty is included in $\hat{\mathbf{H}}_x$, and unknown uncertainty embedded in \mathbf{x}^j is learned using the data-driven model and used to predict $\hat{\mathbf{x}}^j$ with uncertainty.

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

$$\tilde{x}(t) = \boldsymbol{\beta}^T \boldsymbol{\psi}_t + \epsilon \quad (4.7)$$

where $\boldsymbol{\psi}_t \sim N(\boldsymbol{\mu}_{\boldsymbol{\psi}_t}, \boldsymbol{\Sigma}_{\boldsymbol{\psi}_t})$ is the feature vector defined as an uncorrelated Gaussian random variable, $\boldsymbol{\beta} \sim N(\boldsymbol{\mu}_{\boldsymbol{\beta}}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}})$ 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 point-estimation of $\tilde{\mathbf{x}}^j$ from Section 4.2.1, a distribution $\tilde{\mathbf{x}}^j \sim N(\boldsymbol{\mu}_{\tilde{\mathbf{x}}^j}, \boldsymbol{\Sigma}_{\tilde{\mathbf{x}}^j})$ is estimated in this section based on the uncertainties of the features and the weights. This chapter proposes that the known uncertainties of the physics-based models are embedded into the feature vector $\boldsymbol{\psi}_t$ in Eq. (4.7) to enable efficient training of $\boldsymbol{\beta}$. To do so, a set of N_H stable physics-based models $\{\hat{\mathbf{H}}_x^i\}_{i=1}^{N_H}$ is obtained, where each model $\hat{\mathbf{H}}_x^i$ for $i = 1, 2, \dots, N_H$ is identified in the form of the complex-valued frequency response function (FRF) of the physical system \mathbf{H}_x at discrete frequencies ω_k via experiments as

$$\hat{\mathbf{H}}_x^i(\omega_k) = a^i(\omega_k) + b^i(\omega_k)j \quad (4.8)$$

Here $\omega_k = k\Delta\omega$, where $\Delta\omega$ is the increment of frequencies and $k = 1, 2, \dots, \frac{N_\omega}{2} - 1$ and j is the unit imaginary number (which should not be confused with the batch index j used as superscript elsewhere).

Then, the uncertainty in $\hat{\mathbf{H}}_x$ is propagated to finite impulse response $\hat{\mathbf{h}}_x$ as follows. The discrete sets $\{\hat{\mathbf{H}}_x^i(\omega_k)\}_{i=1}^{N_H}$ 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., 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

$$\begin{aligned}\mu_{a(\omega_k)} &= \frac{\max \{a(\omega_k)\} + \min \{a(\omega_k)\}}{2} \\ \sigma_{a(\omega_k)} &= \frac{\max \{a(\omega_k)\} - \min \{a(\omega_k)\}}{3}\end{aligned}\tag{4.9}$$

and $b(\omega_k) \sim N(\mu_{b(\omega_k)}, \sigma_{b(\omega_k)}^2)$ can be approximated using the same procedure as Eq. (4.9).

Then, the impulse response $\hat{\mathbf{h}}_x$ (where $\hat{h}_x[n]$ and $n = 1, 2, \dots, N_h$) of the physics-based model with sampling time T_s can be formulated using discrete inverse Fourier transform as

$$\hat{h}[n] = \frac{1}{N_\omega} \sum_{k=-\frac{N_\omega}{2}}^{\frac{N_\omega}{2}-1} (a(\omega_k) + b(\omega_k)j) e^{\frac{2\pi j(n-1)(k-1)}{N_\omega}}\tag{4.10}$$

where $N_h = \frac{1}{T_s \Delta\omega}$. Then, due to the linearity of Eq. (4.10), $\hat{h}[n]$ follows a Gaussian distribution $\hat{h}[n] \sim N(\mu_{\hat{h}[n]}, \sigma_{\hat{h}[n]}^2)$ as

$$\begin{aligned}\mu_{\hat{h}[n]} &= \frac{1}{N_\omega} \sum_{k=1}^{N_\omega} (\mu_{a(\omega_k)} + \mu_{b(\omega_k)}j) e^{\frac{2\pi j(n-1)(k-1)}{N_\omega}} \\ \sigma_{\hat{h}[n]}^2 &= \frac{1}{N_\omega} \sum_{k=1}^{N_\omega} (\sigma_{a(\omega_k)}^2 - \sigma_{b(\omega_k)}^2) e^{\frac{2\pi j(n-1)(k-1)}{N_\omega}}\end{aligned}\tag{4.11}$$

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

$$\begin{aligned}\mu_{\hat{x}(t)} &= \sum_{i=1}^{N_h} \mu_{\hat{h}[i]} x_d(t - iT_s) \\ \sigma_{\hat{x}(t)}^2 &= \sum_{i=1}^{N_h} \sigma_{\hat{h}[i]}^2 x_d(t - iT_s)^2\end{aligned}\tag{4.12}$$

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

$$\begin{aligned}
\boldsymbol{\mu}_{\psi_t} &= \left[\underbrace{1}_{:=\boldsymbol{\mu}_{\psi_{t1}}} \quad \underbrace{\mu_{\hat{x}(t-n_2T_s)} \cdots \mu_{\hat{x}(t)}}_{:=\boldsymbol{\mu}_{\psi_{t2}}} \right. \\
&\quad \left. \underbrace{x(t-n_3T_s) - \mu_{\hat{x}(t-n_3T_s)} \cdots x(t-T_s) - \mu_{\hat{x}(t-T_s)}}_{:=\boldsymbol{\mu}_{\psi_{t3}}} \right]^T \\
\boldsymbol{\Sigma}_{\psi_t} &= \left[\underbrace{0}_{:=\boldsymbol{\Sigma}_{\psi_{t1}}} \quad \underbrace{\sigma_{\hat{x}(t-n_2T_s)}^2 \cdots \sigma_{\hat{x}(t)}^2}_{:=\boldsymbol{\Sigma}_{\psi_{t2}}} \right. \\
&\quad \left. \underbrace{\sigma_{\hat{x}(t-n_3T_s)}^2 \cdots \sigma_{\hat{x}(t-T_s)}^2}_{:=\boldsymbol{\Sigma}_{\psi_{t3}}} \right]^T
\end{aligned} \tag{4.13}$$

As for the weight uncertainty, the priors, which are $\boldsymbol{\mu}_\beta$ and $\boldsymbol{\Sigma}_\beta$, are initialized as $\boldsymbol{\mu}_{\beta_0}$ and $\boldsymbol{\Sigma}_{\beta_0}$ in the 0-th batch using the nominal physics-based model $\boldsymbol{\mu}_{\hat{h}_x}$ from Eq. (4.11). To do so, a trial desired trajectory $x_d(t), t = 0, T_s, \dots, N_x T_s$, is chosen such that it traverses a pre-defined path with conservative kinematics used in practice. The nominal physics-based model $\boldsymbol{\mu}_{\hat{h}_x}$ is used to filter x_d and formulate \hat{x} . Then, Section 4.2.1's framework on deterministic feature vector is borrowed to create multiple datasets consisting of feature vectors and corresponding predictions, i.e., $(\boldsymbol{\psi}_0, x(0)), (\boldsymbol{\psi}_{T_s}, x(T_s)), \dots, (\boldsymbol{\psi}_{N_x T_s}, x(N_x T_s))$, assuming $\boldsymbol{\psi}_t$ is deterministic and $\boldsymbol{\psi}_{t3} = \mathbf{0}$ for all t . Finally, $\boldsymbol{\mu}_{\beta_0}$ and $\boldsymbol{\Sigma}_{\beta_0}$ can be optimized using the Maximum Likelihood Estimation (MLE) as

$$\begin{aligned}
&\boldsymbol{\mu}_{\beta_0}, \boldsymbol{\Sigma}_{\beta_0} \\
&= \underset{\boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta}{\operatorname{argmin}} \sum_{i=0}^{N_x} \left(-\frac{1}{2} (x(iT_s) - \boldsymbol{\mu}_\beta^T \boldsymbol{\psi}_{iT_s})^T (\boldsymbol{\psi}_{iT_s}^T \boldsymbol{\Sigma}_\beta \boldsymbol{\psi}_{iT_s} + \sigma_\epsilon^2)^{-1} \right. \\
&\quad \left. (x(iT_s) - \boldsymbol{\mu}_\beta^T \boldsymbol{\psi}_{iT_s}) \right)
\end{aligned} \tag{4.14}$$

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

Finally, the posterior can be estimated via an error-in-variables (EIV) Bayesian regression [80], based on the weight uncertainty $\boldsymbol{\beta} \sim N(\boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta)$, i.e., the prior from Eq. (4.14), and the feature uncertainty $\boldsymbol{\psi}_t \sim N(\boldsymbol{\mu}_{\psi_t}, \boldsymbol{\Sigma}_{\psi_t})$ from Eq. (4.13), using Bayes rule [81] as

$$\begin{aligned}
p(\boldsymbol{\beta}|x(t), \boldsymbol{\mu}_{\psi_t}, \boldsymbol{\Sigma}_{\psi_t}) &\propto p(x(t), \boldsymbol{\mu}_{\psi_t}, \boldsymbol{\Sigma}_{\psi_t}|\boldsymbol{\beta}) \cdot p(\boldsymbol{\beta}) \\
&\propto \int p(x(t), \boldsymbol{\mu}_{\psi_t}, \boldsymbol{\Sigma}_{\psi_t}, \boldsymbol{\psi}_t|\boldsymbol{\beta}) d\boldsymbol{\psi}_t \cdot p(\boldsymbol{\beta}) \\
&\propto \int p(x(t)|\boldsymbol{\psi}_t, \boldsymbol{\beta}) p(\boldsymbol{\psi}_t|\boldsymbol{\mu}_{\psi_t}, \boldsymbol{\Sigma}_{\psi_t}) d\boldsymbol{\psi}_t \\
&\quad \cdot p(\boldsymbol{\beta}) \\
&\propto p(x(t)|\boldsymbol{\beta}) \cdot p(\boldsymbol{\beta})
\end{aligned} \tag{4.15}$$

Here, $p(x(t)|\boldsymbol{\beta})$ and $p(\boldsymbol{\beta})$ can be written as

$$\begin{aligned}
p(x(t)|\boldsymbol{\beta}) &= \exp\left(-\frac{1}{2}(x(t) - \boldsymbol{\beta}^T \boldsymbol{\mu}_{\psi_t})^T (\boldsymbol{\beta}^T \boldsymbol{\Sigma}_{\psi_t} \boldsymbol{\beta} + \sigma_\epsilon^2)^{-1} \right. \\
&\quad \left. (x(t) - \boldsymbol{\beta}^T \boldsymbol{\mu}_{\psi_t})\right) \\
p(\boldsymbol{\beta}) &= \exp\left(-\frac{1}{2}(\boldsymbol{\beta} - \boldsymbol{\mu}_\beta)^T \boldsymbol{\Sigma}_\beta^{-1} (\boldsymbol{\beta} - \boldsymbol{\mu}_\beta)\right)
\end{aligned} \tag{4.16}$$

For efficient computation, the non-Gaussian probability density function $p(x(t)|\boldsymbol{\beta})$ in Eq. (4.16) is approximated as Gaussian using Laplace's approximation [82], such that the log of $p(x(t)|\boldsymbol{\beta})$ becomes a quadratic function of $\boldsymbol{\beta}$ as

$$\begin{aligned}
\ln p(x(t)|\boldsymbol{\beta}) &\approx \underbrace{\ln p(x(t)|\boldsymbol{\beta})\Big|_{\boldsymbol{\beta}=\bar{\boldsymbol{\beta}}}}_{:=c_0} \\
&\quad + \underbrace{\frac{d \ln p(x(t)|\boldsymbol{\beta})}{d\boldsymbol{\beta}}\Big|_{\boldsymbol{\beta}=\bar{\boldsymbol{\beta}}}}_{:=c_1} (\boldsymbol{\beta} - \bar{\boldsymbol{\beta}}) \\
&\quad + \underbrace{\frac{d^2 \ln p(x(t)|\boldsymbol{\beta})}{d\boldsymbol{\beta}^2}\Big|_{\boldsymbol{\beta}=\bar{\boldsymbol{\beta}}}}_{:=c_2} (\boldsymbol{\beta} - \bar{\boldsymbol{\beta}})^2
\end{aligned} \tag{4.17}$$

where the local point $\bar{\boldsymbol{\beta}}$ is $\boldsymbol{\mu}_\beta$ from the previous batch, i.e., the prior mean. The coefficients c_0 , c_1 and c_2 in Eq. (4.17) can be derived using 0-th, 1-st and 2-nd derivatives of vector-valued function $\ln p(x(t)|\boldsymbol{\beta})$ in Eq. (4.16) with respect to $\boldsymbol{\beta}$, respectively, as

$$\begin{aligned}
c_0 &= -\frac{1}{2}(x(t) - \bar{\boldsymbol{\beta}}^T \boldsymbol{\mu}_{\psi_t})^T (\bar{\boldsymbol{\beta}}^T \boldsymbol{\Sigma}_{\psi_t} \bar{\boldsymbol{\beta}} + \sigma_\epsilon^2)^{-1} (x(t) - \bar{\boldsymbol{\beta}}^T \boldsymbol{\mu}_{\psi_t}) \\
c_1 &= -\frac{1}{2} \left(2 \boldsymbol{\Sigma}_{\psi_t}^T \boldsymbol{\mu}_{\psi_t} x(t) \bar{\boldsymbol{\beta}}^T \bar{\boldsymbol{\beta}} + (2 \boldsymbol{\mu}_{\psi_t}^2 \sigma_\epsilon^2 - 2 \boldsymbol{\Sigma}_{\psi_t} x(t)^2)^T \bar{\boldsymbol{\beta}} \right. \\
&\quad \left. - 2 \boldsymbol{\mu}_{\psi_t} x(t) \sigma_\epsilon^2 \right) \cdot \text{diag}(\boldsymbol{\Sigma}_{\psi_t}^T \bar{\boldsymbol{\beta}}^2 + \sigma_\epsilon^2)^{-1} \\
c_2 &= -\frac{1}{2} \left(\text{diag}(-4 \boldsymbol{\Sigma}_{\psi_t}^2 \boldsymbol{\mu}_{\psi_t} x(t) \bar{\boldsymbol{\beta}}^3) + \text{diag}(6 \boldsymbol{\Sigma}_{\psi_t}^2 x(t)^2) \bar{\boldsymbol{\beta}}^2 \right. \\
&\quad \left. \text{diag}(12 \boldsymbol{\Sigma}_{\psi_t} \boldsymbol{\mu}_{\psi_t} x(t) \sigma_\epsilon^2 \bar{\boldsymbol{\beta}}) - 6 \boldsymbol{\mu}_{\psi_t}^T \boldsymbol{\Sigma}_{\psi_t} \boldsymbol{\mu}_{\psi_t} \sigma_\epsilon^2 \right) \\
&\quad \cdot \text{diag}(\boldsymbol{\Sigma}_{\psi_t}^T \bar{\boldsymbol{\beta}}^2 + \sigma_\epsilon^2)^{-1}
\end{aligned} \tag{4.18}$$

Using the simplified notation, Eq. (4.15) can be re-written as

$$\begin{aligned}
p(\boldsymbol{\beta}|x(t), \boldsymbol{\mu}_{\psi_t}, \boldsymbol{\Sigma}_{\psi_t}) &= \exp \left((c_2 - \frac{1}{2} \boldsymbol{\Sigma}_\beta^{-1}) \boldsymbol{\beta}^2 \right. \\
&\quad \left. + (c_1 + \boldsymbol{\mu}_\beta^T \boldsymbol{\Sigma}_\beta^{-1}) \boldsymbol{\beta} + \text{constant} \right)
\end{aligned} \tag{4.19}$$

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

$$\begin{aligned}
\boldsymbol{\Sigma}_\beta &\leftarrow \text{Var}[\boldsymbol{\beta}|x(t), \boldsymbol{\mu}_{\psi_t}, \boldsymbol{\Sigma}_{\psi_t}] = -\frac{1}{2} (c_2 - \frac{1}{2} \boldsymbol{\Sigma}_\beta^{-1})^{-1} \\
\boldsymbol{\mu}_\beta &\leftarrow \mathbb{E}[\boldsymbol{\beta}|x(t), \boldsymbol{\mu}_{\psi_t}, \boldsymbol{\Sigma}_{\psi_t}] = (c_1 + \boldsymbol{\mu}_\beta^T \boldsymbol{\Sigma}_\beta^{-1}) \boldsymbol{\Sigma}_\beta
\end{aligned} \tag{4.20}$$

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) such that no known uncertainties are included in the Bayesian regression, the posterior $\boldsymbol{\beta} \sim N(\boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta)$ can be estimated as a closed-form solution using Bayes rule as

$$\begin{aligned}
& p(\boldsymbol{\beta} | \boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta) \\
&= \exp\left(-\frac{1}{2}\boldsymbol{\beta}^\top(\sigma_\epsilon^{-2}\boldsymbol{\psi}_t^\top\boldsymbol{\psi}_t + \boldsymbol{\Sigma}_\beta^{-1})\boldsymbol{\beta}\right. \\
&\quad \left.+ (\sigma_\epsilon^{-2}\boldsymbol{\psi}_t^\top x(t) + \boldsymbol{\Sigma}_\beta^{-1}\boldsymbol{\mu}_\beta)^\top\boldsymbol{\beta}\right) \\
&\therefore \boldsymbol{\Sigma}_\beta \leftarrow (\sigma_\epsilon^{-2}\boldsymbol{\psi}_t^\top\boldsymbol{\psi}_t + \boldsymbol{\Sigma}_\beta^{-1})^{-1} \\
&\quad \boldsymbol{\mu}_\beta \leftarrow \boldsymbol{\Sigma}_\beta(\sigma_\epsilon^{-2}\boldsymbol{\psi}_t^\top x(t) + \boldsymbol{\Sigma}_\beta^{-1}\boldsymbol{\mu}_\beta)
\end{aligned} \tag{4.21}$$

Lastly, using the trained weight distribution from Eq. (4.20), the posterior predictive distribution $\tilde{x}(t) \sim N(\mu_{\tilde{x}(t)}, \sigma_{\tilde{x}(t)}^2)$ can be written as

$$\begin{aligned}
\mu_{\tilde{x}(t)} &= \boldsymbol{\mu}_\beta^\top \boldsymbol{\psi}_t \\
\sigma_{\tilde{x}(t)}^2 &= \boldsymbol{\psi}_t^\top \boldsymbol{\Sigma}_\beta \boldsymbol{\psi}_t + \sigma_\epsilon^2
\end{aligned} \tag{4.22}$$

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 $\boldsymbol{\psi}_{t_y}$ and unobserved noise ϵ_y . Using Eq. (A.13), the contour error distribution $\tilde{\epsilon}(t) \sim N(\mu_{\tilde{\epsilon}(t)}, \sigma_{\tilde{\epsilon}(t)}^2)$ can be predicted as

$$\begin{aligned}
\mu_{\tilde{\epsilon}(t)} &= -\sin(\theta(t))(x_d(t) - \boldsymbol{\mu}_\beta^\top \boldsymbol{\psi}_t) \\
&\quad + \cos(\theta(t))(y_d(t) - \boldsymbol{\mu}_{\beta_y}^\top \boldsymbol{\psi}_{t_y}) \\
\sigma_{\tilde{\epsilon}(t)}^2 &= \sin(\theta(t))^2(\boldsymbol{\psi}_t^\top \boldsymbol{\Sigma}_\beta \boldsymbol{\psi}_t + \sigma_\epsilon^2) \\
&\quad + \cos(\theta(t))^2(\boldsymbol{\psi}_{t_y}^\top \boldsymbol{\Sigma}_{\beta_y} \boldsymbol{\psi}_{t_y} + \sigma_{\epsilon_y}^2)
\end{aligned} \tag{4.23}$$

4.3 Methodology of 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 [1]. Taking the x -axis, for example, a desired trajectory $\mathbf{X}_d = f(p)$ is parametrized with respect to a normalized, monotonically increasing path variable 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

$$x_d(t) = - \left. \frac{\partial f}{\partial p} \right|_{p=\bar{p}(t)} \cdot (p(t) - \bar{p}(t)) + f(\bar{p}(t)) \quad (4.24)$$

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

$$\begin{aligned} & \max \mathbf{1}^T \mathbf{p}^j \\ & \text{s.t. } p(t-1) \leq p(t) \leq 1 \\ & \quad D[\mathbf{p}^j] \leq \mathbf{V}_{max} T_s \\ & \quad |D^2[\mathbf{x}_d^j]| \leq \mathbf{A}_{max} T_s^2 \end{aligned} \quad (4.25)$$

where $\mathbf{1}$ is a ones-vector, D is a difference operator, and \mathbf{V}_{max} and \mathbf{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 4.2.2. To do so, we show that $\mu_{\tilde{e}(t)}$ and $\sigma_{\tilde{e}(t)}$ are linear in terms of \mathbf{x}_d^j , by showing that the only alterable feature in ψ_t , which is the last term in ψ_{t_2} (i.e., $\hat{\mathbf{x}}(t)$), is linear in \mathbf{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

$$\Phi_x = \begin{bmatrix} \vdots & \vdots \\ \Phi_{x,p} & \Phi_{x,c} \end{bmatrix} \quad (4.26)$$

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

$$\begin{aligned} \hat{\mathbf{x}}^j &= \Phi_{x,c} \mathbf{x}_d^j + \Phi_{x,p} \mathbf{x}_{c,p} \\ \therefore \hat{\mathbf{x}}(t) &= \underbrace{\mathbf{M}_t \Phi_{x,c}}_{:=\mathbf{T}_x} \mathbf{x}_d^j + \underbrace{\mathbf{M}_t \Phi_{x,p} \mathbf{x}_{cp}}_{:=\mathbf{T}_{0x}} \end{aligned} \quad (4.27)$$

where \mathbf{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 \mathbf{y}_d^j , by using a similar notation as Eq. (A.8), i.e., $\hat{\mathbf{y}}(t) = \mathbf{T}_y \mathbf{y}_d^j + \mathbf{T}_{0y}$.

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

$$p(\tilde{\varepsilon}(t) \leq E_{max}) \geq \eta \quad (4.28)$$

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. (4.28) becomes

$$\mu_{\tilde{\varepsilon}(t)} \leq E_{max} - \Phi^{-1}(\eta)\sigma_{\tilde{\varepsilon}(t)} \quad (4.29)$$

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. (4.23).

In order to show that the contour error constraint in Eq. (4.29) is linear in terms of $\hat{\mathbf{x}}_d^j$ and $\hat{\mathbf{y}}_d^j$, the standard deviation term $\sigma_{\tilde{\varepsilon}(t)}$ in Eq. (4.23) is linearized with respect to $\boldsymbol{\psi}_t$ and $\boldsymbol{\psi}_{ty}$ using linearization points $\bar{\boldsymbol{\psi}}_t$ and $\bar{\boldsymbol{\psi}}_{ty}$ as

$$\begin{aligned} \sigma_{\tilde{\varepsilon}(t)} &\approx \mathbf{S}_x \boldsymbol{\psi}_t + \mathbf{S}_y \boldsymbol{\psi}_{ty} + \mathbf{S}_{00} + \mathbf{S}_{01} \\ \text{where } \mathbf{S}_{00} &= \sin(\theta(t))^2 (\bar{\boldsymbol{\psi}}_t^T \boldsymbol{\Sigma}_\beta \bar{\boldsymbol{\psi}}_t + \sigma_\epsilon^2) \\ &\quad + \cos(\theta(t))^2 (\bar{\boldsymbol{\psi}}_{ty}^T \boldsymbol{\Sigma}_{\beta_y} \bar{\boldsymbol{\psi}}_{ty} + \sigma_{\epsilon_y}^2) \\ \mathbf{S}_{01} &= -2 \sin(\theta(t))^2 \boldsymbol{\Sigma}_\beta \bar{\boldsymbol{\psi}}_t - 2 \cos(\theta(t))^2 \boldsymbol{\Sigma}_\beta \bar{\boldsymbol{\psi}}_{ty} \\ \mathbf{S}_x &= \frac{1}{\sqrt{\mathbf{S}_{00}}} \sin(\theta(t))^2 \boldsymbol{\Sigma}_\beta \\ \mathbf{S}_y &= \frac{1}{\sqrt{\mathbf{S}_{00}}} \cos(\theta(t))^2 \boldsymbol{\Sigma}_{\beta_y} \end{aligned} \quad (4.30)$$

where $\bar{\boldsymbol{\psi}}_t$ is formulated via generating the terms $\hat{x}(t - n_2 T_s) \cdots \hat{x}(t)$ in $\bar{\boldsymbol{\psi}}_{t_2}$ by filtering the linearization point $f(\bar{\mathbf{p}})$ with $\boldsymbol{\mu}_{\hat{h}_x}$. Likewise, $\bar{\boldsymbol{\psi}}_{ty}$ is formulated using $\boldsymbol{\mu}_{\hat{h}_y}$.

Finally, let the unalterable features in $\boldsymbol{\psi}_t$ be $\boldsymbol{\psi}_{tu}$, and the weights corresponding to the alterable feature in $\boldsymbol{\psi}_t$, i.e., $\hat{x}(t)$, be denoted as $\boldsymbol{\beta}_a$ and that to the unalterable features $\boldsymbol{\psi}_{tu}$ as $\boldsymbol{\beta}_u$. The same notations $\boldsymbol{\psi}_{tyu}$, $\boldsymbol{\beta}_{ya}$ and $\boldsymbol{\beta}_{yu}$ will be held for y -axis. Then, by substituting Eq. (4.23), (A.8) and (4.30) into Eq. (4.29), the contour error constraint be re-written as

$$\begin{aligned}
& -\sin(\theta(t))(\mathbf{M}_t \mathbf{x}_d^j - \boldsymbol{\mu}_{\beta_u}^T \boldsymbol{\psi}_{tu} - \boldsymbol{\mu}_{\beta_a}^T (\mathbf{T}_x \mathbf{x}_d^j + \mathbf{T}_{0x})) \\
& + \cos(\theta(t))(\mathbf{M}_t \mathbf{y}_d^j - \boldsymbol{\mu}_{\beta_{yu}}^T \boldsymbol{\psi}_{tyu} - \boldsymbol{\mu}_{\beta_{ya}}^T (\mathbf{T}_y \mathbf{y}_d^j + \mathbf{T}_{0y})) \\
& \leq E_{max} - \Phi^{-1}(\eta) \left(\mathbf{S}_x \boldsymbol{\psi}_{tu} + \mathbf{S}_x (\mathbf{T}_x \mathbf{x}_d^j + \mathbf{T}_{0x}) \right. \\
& \quad \left. + \mathbf{S}_y \boldsymbol{\psi}_{tyu} + \mathbf{S}_y (\mathbf{T}_y \mathbf{y}_d^j + \mathbf{T}_{0y}) + \mathbf{S}_{00} + \mathbf{S}_{01} \right)
\end{aligned} \tag{4.31}$$

which can be rearranged as linear in terms of \mathbf{x}_d^j and \mathbf{y}_d^j as

$$\begin{aligned}
& \mathbf{U}_x \mathbf{x}_d^j + \mathbf{U}_y \mathbf{y}_d^j \leq \mathbf{U}_0 \\
& \text{where } \mathbf{U}_x = -\sin(\theta(t))(\mathbf{M}_t - \boldsymbol{\mu}_{\beta_a}^T \mathbf{T}_x) + \Phi^{-1}(\eta) \mathbf{S}_x \mathbf{T}_x \\
& \mathbf{U}_y = \cos(\theta(t))(\mathbf{M}_t - \boldsymbol{\mu}_{\beta_{ya}}^T \mathbf{T}_y) - \Phi^{-1}(\eta) \mathbf{S}_y \mathbf{T}_y \\
& \mathbf{U}_0 = \sin(\theta(t))(-\boldsymbol{\mu}_{\beta_u}^T \boldsymbol{\psi}_{tu} - \boldsymbol{\mu}_{\beta_a}^T \mathbf{T}_{0x}) \\
& \quad - \cos(\theta(t))(-\boldsymbol{\mu}_{\beta_{yu}}^T \boldsymbol{\psi}_{tyu} - \boldsymbol{\mu}_{\beta_{ya}}^T \mathbf{T}_{0y}) \\
& \quad + E_{max} - \Phi^{-1}(\eta)(\mathbf{S}_x \boldsymbol{\psi}_{tu} + \mathbf{S}_x \mathbf{T}_{0x} \\
& \quad + \mathbf{S}_y \boldsymbol{\psi}_{tyu} + \mathbf{S}_y \mathbf{T}_{0y} + \mathbf{S}_{00} + \mathbf{S}_{01})
\end{aligned} \tag{4.32}$$

Finally, the contour error constraint in Eq. (4.32) is also linear with respect to the decision variable \mathbf{p}^j using the relationship in Eq. (4.24).

Note that the methodology of feedrate optimization described in Section 4.3 can be broadly considered as model predictive control (MPC) [83], because it: (1) optimizes manipulatable inputs, e.g., desired trajectory, over a finite, receding horizon using (2) prediction of the dynamical system's behavior through a model that is updated via feedback. There are existing works on feedrate optimization using MPC. For example, MPC can be formulated with an objective function that is a weighted-sum of cycle time, lag and contour errors and control inputs, along with kinematic constraints [56, 84] and error tolerance [85]. On the other hand, MPC in [48] uses an objective function based on a tunable index of how far away an unattainable target position is from the current position. However, all of the methods in [48, 56, 84, 85] use objective functions that require trial-and-error to balance quality and productivity, which is philosophically different from the aim of our dissertation, that is to autonomously maximize productivity subject to quality constraints. Moreover, none of the existing works quantify the uncertainty of servo error prediction, which is one of the main contributions of this dissertation.

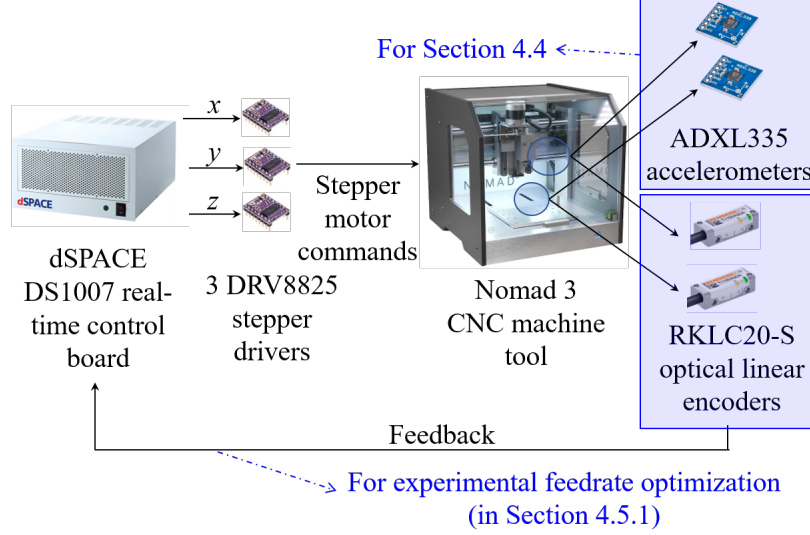


Figure 4.5: Experimental setup for Sections 4.4 and 4.5.1

4.4 Numerical validation of the intelligent feedrate optimization using physics-informed data-driven 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 [86] 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 4.3
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 $\{\hat{\mathbf{H}}_x^i\}_{i=1}^{N_H}$ and $\{\hat{\mathbf{H}}_y^i\}_{i=1}^{N_H}$
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 $\boldsymbol{\mu}_{\beta_0}$, $\boldsymbol{\Sigma}_{\beta_0}$, $\boldsymbol{\mu}_{\beta_y,0}$ and $\boldsymbol{\Sigma}_{\beta_y,0}$ are initialized as zero at the 0-th batch, and $\boldsymbol{\beta}$ and $\boldsymbol{\beta}_y$ are learned via Bayesian linear regression for deterministic features in Eq. (4.21). Note that both the PB and DD methods are subsets of the 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

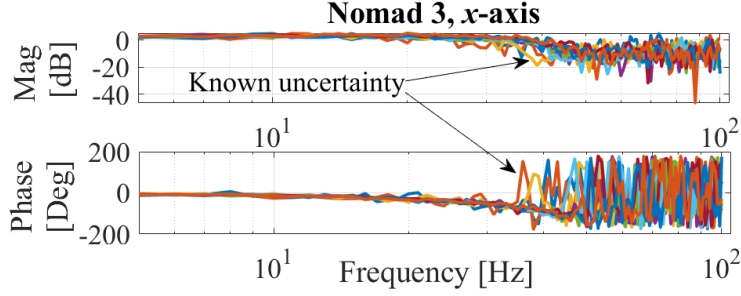


Figure 4.6: Frequency response functions of the Nomad 3's x -axis showing the known uncertainty obtained under different input acceleration amplitudes

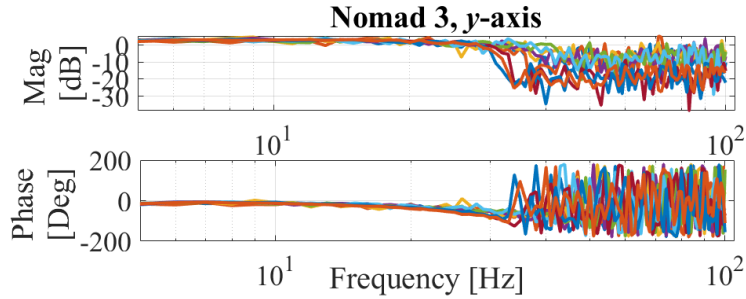


Figure 4.7: Frequency response functions of the Nomad 3's y -axis showing the known uncertainty obtained under different input acceleration amplitudes

A Nomad 3 three-axis desktop CNC machine tool is chosen as the simulated system, where its setup is shown in Figure A.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 of the FRFs is varied by modifying the input acceleration amplitude at discrete values: 2 m/s², 3 m/s² and 4 m/s², and 3 FRFs are measured per each acceleration amplitude to collect a total of $N_H = 9$ FRFs per axes.

The set of FRFs $\{\hat{\mathbf{H}}_x^i\}_{i=1}^9$ of the x - and y -axis of the printer are shown in Figure 4.6 and 4.7, respectively. The uncertainties in $\hat{\mathbf{H}}_x$ are then propagated to $\hat{\mathbf{h}}_x \sim N(\boldsymbol{\mu}_{\hat{\mathbf{h}}_x}, \boldsymbol{\Sigma}_{\hat{\mathbf{h}}_x})$ to initialize $\boldsymbol{\mu}_\beta$ and $\boldsymbol{\Sigma}_\beta$ and construct $\boldsymbol{\mu}_{\psi_t}$ and $\boldsymbol{\Sigma}_{\psi_t}$ in the physics-informed data-driven digital twin.

The output position \mathbf{x} is simulated as the sum of motion-induced position \mathbf{x}_m and force-induced position \mathbf{x}_f , as

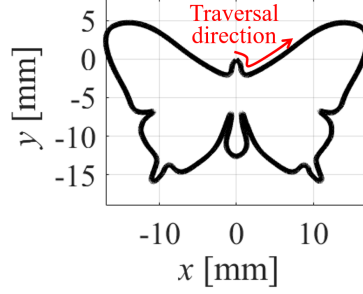


Figure 4.8: Desired toolpath

$$\begin{aligned}
 \mathbf{x} &= \mathbf{x}_m + \mathbf{x}_f = \hat{\mathbf{h}}_x * \mathbf{x}_c + \mathbf{x}_f \\
 \text{where } \hat{\mathbf{h}}_x &\sim N(\boldsymbol{\mu}_{\hat{\mathbf{h}}_x}, \boldsymbol{\Sigma}_{\hat{\mathbf{h}}_x}) \text{ is sampled at every } t \\
 \text{and } x_f(t) &= A_f \sin \omega_f t
 \end{aligned} \tag{4.33}$$

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

The butterfly trajectory [75] with its contour of the toolpath on the x - and y -axis shown in Figure 4.8 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/s², 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

$$\gamma(t) = \begin{cases} ||\epsilon(t)| - E_{max}| & \text{if } |\epsilon(t)| > E_{max} \\ 0 & \text{otherwise} \end{cases} \tag{4.34}$$

Figure 4.9 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 4.1. 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

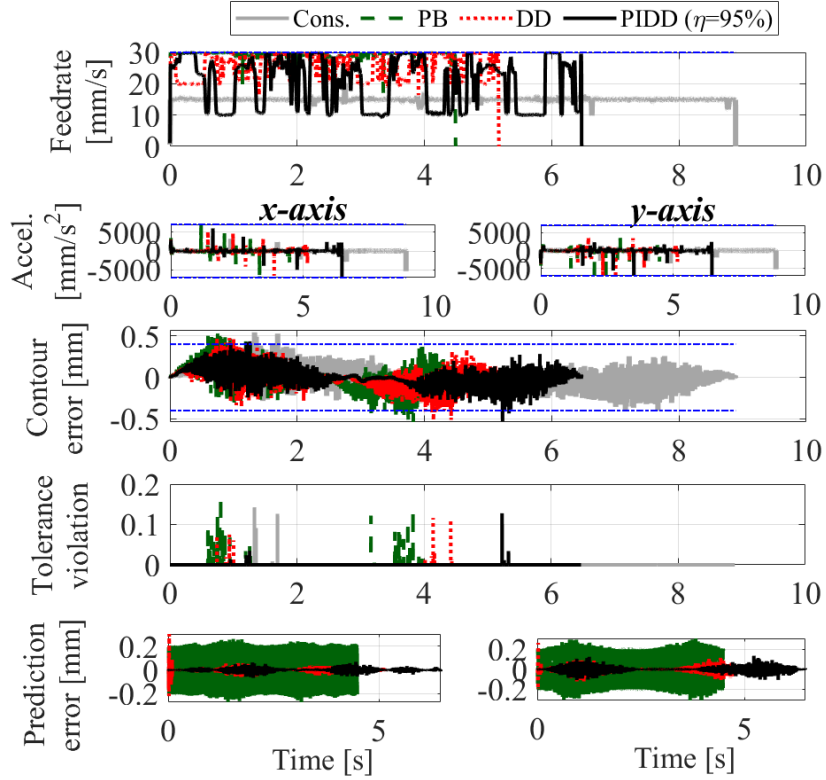


Figure 4.9: Feedrate, acceleration, contour error, tolerance violation, and prediction error using conservative (Cons.) physics-based (PB), data-driven (DD) and proposed (PIDD) methods with $\eta = 95\%$ for numerical validation

level. To demonstrate the effect of the selection of stringency, Figure 4.10 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. (4.30) and sub-optimal learning in β introduced by Laplace's approximation in Eq. (4.17). These problems can be addressed by applying nonlinear optimization and non-Gaussian EIV Bayesian regression, at the price of higher computational cost.

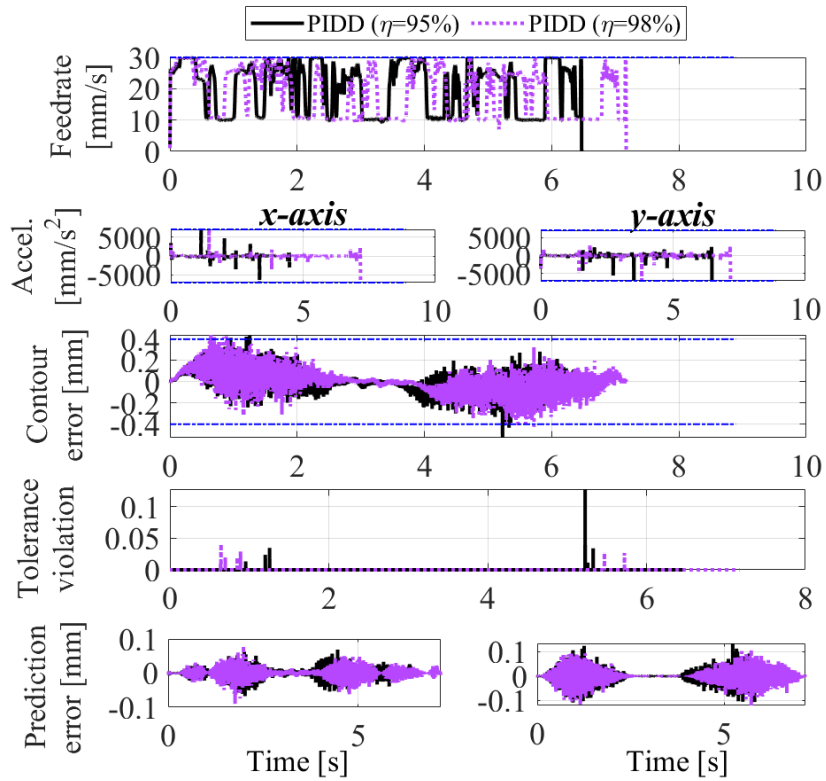


Figure 4.10: Feedrate, acceleration, contour error, tolerance violation, and prediction error using the proposed (PIDD) methods with $\eta = 95\%$ (from Figure 4.9) and $\eta = 98\%$ in numerical validation

Table 4.1: Cycle times and RMS of tolerance violation γ for conservative (Cons.), physics-based (PB), data-driven (DD) and proposed (PIDD) methods in Figures 4.9 and 4.10

	Cons.	PB	DD	PIDD ($\eta=95\%$)	PIDD ($\eta=98\%$)
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

4.5 Experimental validation

For validation of the proposed approach, two experimental setups are used. The first set of experiments, described in Section 4.5.1, is carried out on a Nomad 3 desktop CNC machine tool used in Section 4.4, and the second set of experiments, described in Section 4.5.2, is carried out on an Ender 3 Pro desktop 3D printer. Demonstration of the proposed method on two experimental setups helps to show its versatility.

4.5.1 CNC machine tool

4.5.1.1 Experimental setup

For experimental validation, the same experimental setup with the machine tool in Figure A.6 is 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.

4.5.1.2 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 4.8 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/s², 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 $\sigma_\epsilon = 0.01$ are used in the DD and PIDD methods. The desired stringency is chosen as $\eta = 95\%$ in the DD and PIDD method.

Figure 4.11 and 4.12 show the profiles of optimized feedrate, acceleration, contour error, tolerance violation, and prediction error of x - and y -axis using the conservative, PB, DD and PIDD approaches in air-cutting and actual cutting, respectively. The PB method frequently violates the tolerance due to unmodeled dynamics and/or cutting force, which results in significant prediction error. The DD and PIDD methods reduce the prediction error compared to the physics-based method, where the PIDD method is able to constrain the contour error closer to the limit using the desired stringency, which allows the tolerance to be satisfied closely, similar to conservative approach. The proposed PIDD approach completes the motion 38.06% and 29.02% faster than the conservative method in air-cutting and machining, respectively, 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 in air-cutting and actual cutting are summarized in Table 4.2.

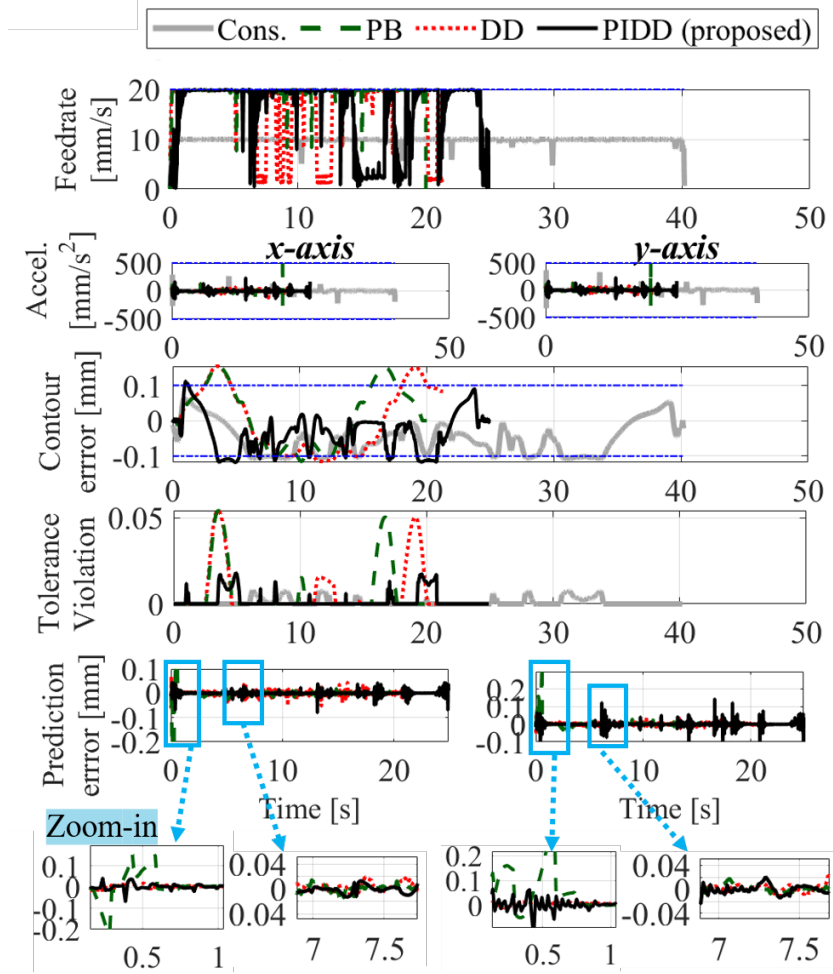


Figure 4.11: Commanded feedrate, acceleration, contour error, tolerance violation, and prediction error (with its zoomed-in images) of conservative, physics-based (PB), data-driven (DD) and proposed (PIDD) approaches in air-cutting

Table 4.2: 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	15.7	18.3	17.8
	Cycle time [s]	40.2	20.00	21.12	24.90
	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	18.11
	Cycle time [s]	39.39	20.04	22.96	27.25
	RMS of γ [μm]	6.8	22.2	3.7	7.4

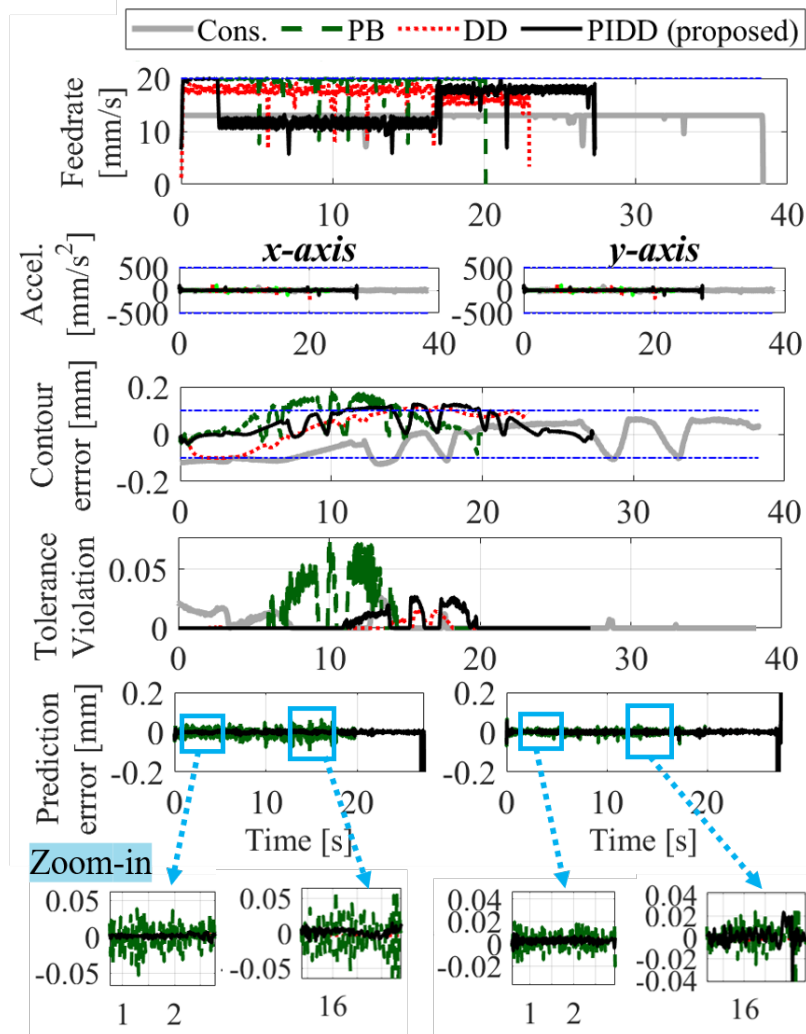


Figure 4.12: Commanded feedrate, acceleration, contour error, tolerance violation, and prediction error (with its zoomed-in images) of conservative, physics-based (PB), data-driven (DD) and proposed (PIDD) approaches in actual cutting

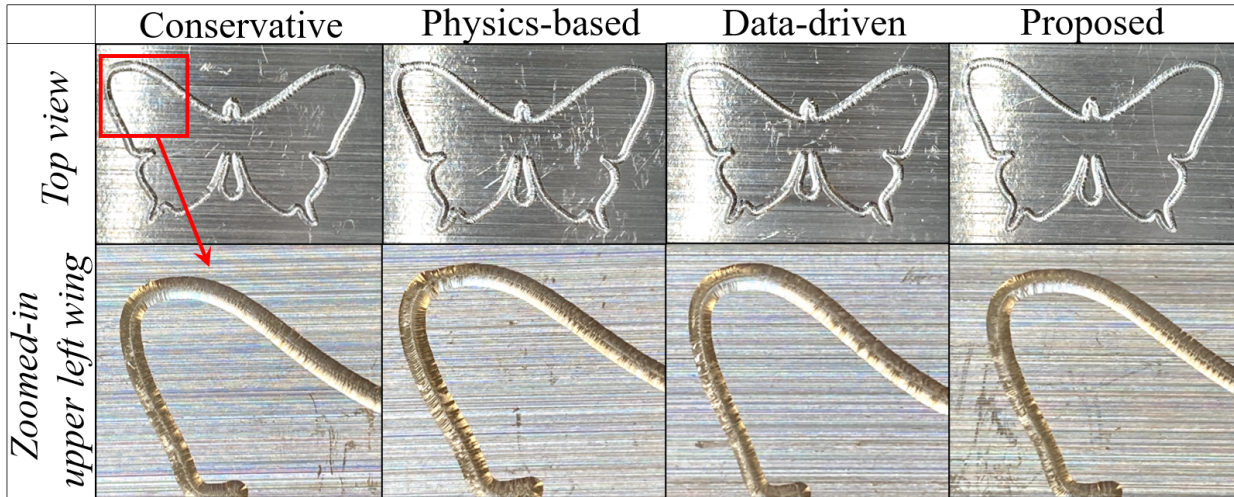


Figure 4.13: Machined parts and the zoomed-in images of upper left wing using conservative, physics-based (PB), data-driven (DD) and proposed (PIDD) approaches

Figure 4.13 shows the machined surfaces and their zoomed-in images of upper left wing using the trajectories from Figure 4.12. 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.

4.5.2 Desktop 3D printer

4.5.2.1 Experimental setup

The experimental setup using an Ender 3 Pro desktop 3D printer is shown in Figure 4.14. 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. In recovery of x and y axes displacement from acceleration measurements, a Luenberger state observer [87] 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.

4.5.2.2 Experimental validation of intelligent feedrate optimization

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

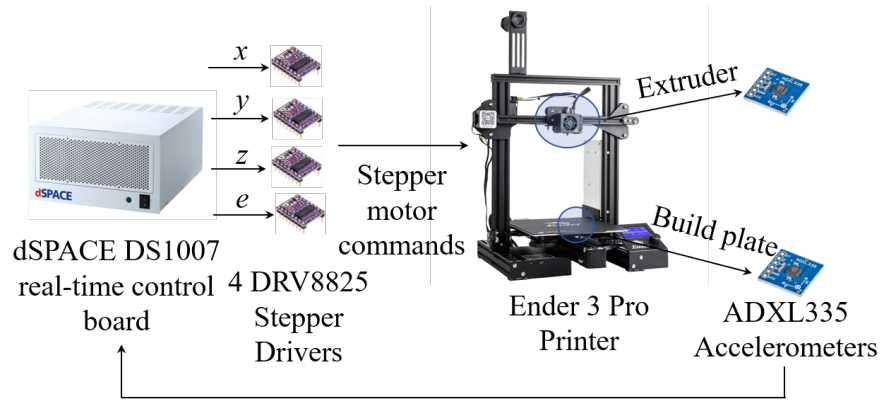


Figure 4.14: Experimental setup for Section 4.5.2

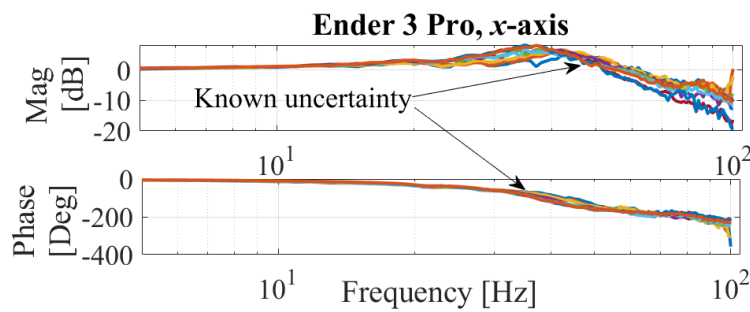


Figure 4.15: Known uncertainty of Ender 3 Pro's x -axis

Figure 4.8 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 4.15 and 4.16. 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/s² and $E_{max} = 0.1$ mm are chosen.

Figure 4.17 shows the profiles of optimized feedrate, acceleration, contour error and prediction error of x - and y using conservative, PB, DD and PIDD methods. The RMS prediction errors, cycle

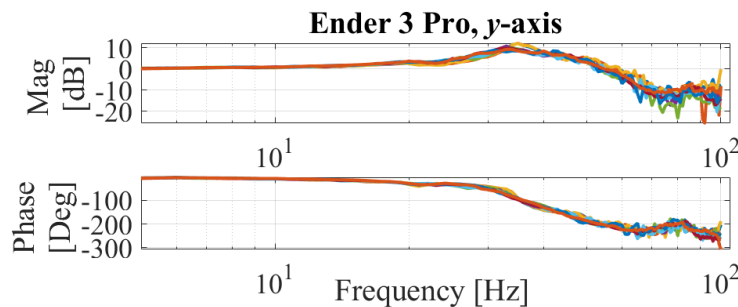


Figure 4.16: Known uncertainty of Ender 3 Pro's y -axis

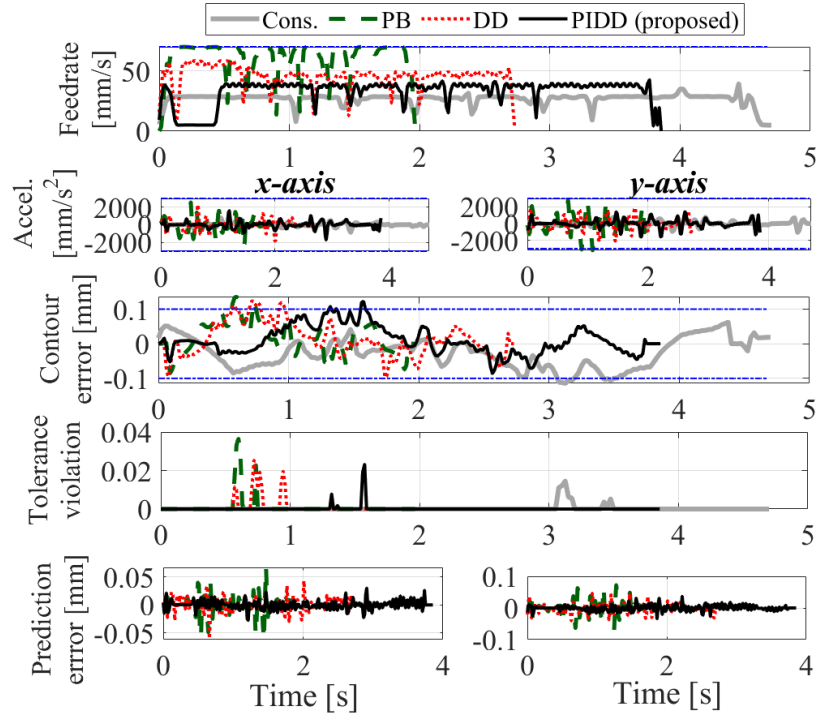


Figure 4.17: Commanded feedrate, acceleration, contour error and prediction error using conservative (Cons.), physics-based, data-driven and proposed approaches in air-printing

times and RMS tolerance violation of all methods are reported in Table 4.3. 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 while yielding contour error tolerance satisfaction as the conservative one.

To further validate our findings, a 3D-augmentation of the trajectory in Figure 4.8 with z -height 8 mm is printed using the same printer. Conservative, PB, DD and PIDD methods are applied at

Table 4.3: 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

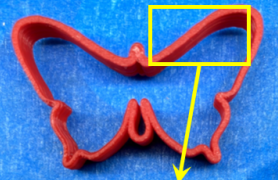
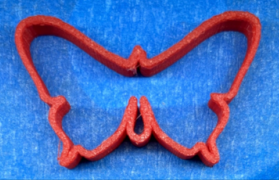

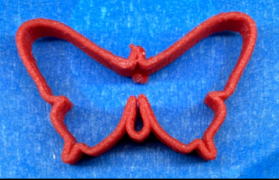




	Conservative	Physics-based (PB)	Data-driven (DD)	Proposed (PIDD)
<i>Top view</i>				
<i>Side view</i>				
<i>Print time</i>	3 min 7 s	1 min 18 s	1 min 57 s	2 min 38 s

Figure 4.18: Top and side views of 3D-printed butterfly models using conservative, physics-based, data-driven and proposed approaches and their print times

each layer of the print. Figure 4.18 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 good quality. Overall, the proposed method is able to achieve 15.51% print time reduction compared to the conservative approach while achieving similar quality.

4.6 Conclusion and future work

This chapter presents the framework and the methodology for the intelligent feedrate optimization using physics-informed data-driven digital twin with quantified uncertainty. The key contributions of this chapter are summarized as follows.

- A novel physics-informed data-driven digital twin that predicts contour error and its uncertainty 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 digital twin with quantified uncertainty, 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 conservative approach.

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 linear programming. The future work will explore more sophisticated (non-Gaussian) uncertainty distributions and nonlinear contour error constraints, 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.

CHAPTER 5

Summary, Conclusions and Future Research

5.1 Summary and conclusions

This dissertation proposes the framework and techniques for maximizing productivity of feed drive systems while satisfying the desired quality, by optimizing feedrate subject to kinematic and error tolerance constraints. The proposed intelligent feedrate optimization uses an uncertainty-aware physics-based and data-driven servo dynamic models and is solved using linear programming to effectively and efficiently optimize feedrate while accurately meeting the constraints by learning the uncertainties from the sensor feedback on-the-fly.

Chapter 2 proposes feedrate optimization technique formulated using time-based parameter, as opposed to using path-based parameter in the vast majority of feedrate optimization such as [13, 88, 89]. This enables the proposed approach to incorporate servo dynamic models, such as servo error pre-compensation, which feedrate optimization can benefit from by allowing for faster motion without violating the tolerances. Moreover, the use of time-based parameter allows the optimization to be efficiently solved using linear programming. We validate the effectiveness of the proposed method using a desktop 3D printer and precision motion stage, and show significant cycle time reduction while closely meeting the tolerances. The proposed approach yield more optimal (faster) feedrate than feedrate optimization without servo dynamic models, allowing up to reduce cycle time by 50% compared to conservative approach.

Chapter 3 extends the feedrate optimization introduced in Chapter 2 to long and nonlinear toolpaths, which introduce inaccuracy and computational inefficiency. To address the issue of inaccuracy due to linearization of desired trajectory using estimated linearization point, the feedrate optimization is executed multiple times, i.e., is solved using sequential linear programming, where the current iteration uses constraints linearized by optimal solution from the previous iteration, hence able to achieve local optimality in the final solution. Second, the issue of computation is solved by optimizing feedrate within smaller dividend of trajectories, i.e., windows, to perform so-called windowed sequential linear programming. Numerical feasibility is guaranteed by generating

conservative backup trajectory at every window, thus avoiding halt of optimization due to infeasible solution. The proposed approach demonstrates up to 25% cycle time reduction compared to conservative approach.

Chapter 4 augments the physics-based servo dynamic model used in Chapters 2 and 3 with a data-driven model to robustly optimize feedrate in the presence of uncertainties. We propose that by quantifying the uncertainty of servo error, the worst-case servo error can be constrained based on the desired tolerance stringency. To effectively learn the uncertainty, the data-driven model uses a feature vector composed of physics-based predictions and their prediction errors, updated from sensor feedback, that embeds the known uncertainty of the servo system, such as variability of the servo dynamics using different operating conditions. The weights in the data-driven model is learned via Bayesian regression, which is used to learn the unknown uncertainty and predict distribution of servo error. The proposed approach is validated using a desktop CNC machine tool and 3D printer to demonstrate up to 38% cycle time reduction while showing similar tolerance violation level to the conservative approach, where the kinematic limits are tuned by trial-and-error to achieve desired tolerance stringency.

The proposed intelligent feedrate optimization is demonstrated to show significant cycle time reduction compared to conservative approach using various types of machines, including 3D printers, precision motion stages and CNC machine tools. It is further expected to be applicable to any types of machines that use feed drives. The broader impact this dissertation aims to bring is to achieve desired quality and higher productivity, with less trial-and-error for wide variety of manufacturing machines.

5.2 Practical considerations for implementation on general feed drive servo systems

For readers who wish to implement the proposed methodology of feedrate optimization on general feed drive servo systems with any desired part geometry, this section discusses a guideline for practitioners.

The first step in implementing the proposed feedrate optimization is to decompose the desired part geometry into a set of toolpaths, then fit each toolpath in terms of trajectory arclength. However, because of the nonlinear relationship between a general curve parameter (e.g., spline parameter) and arclength, techniques such as quintic spline interpolation with minimal feedrate fluctuation [76] can be used to ensure smooth feedrate profile.

The second step is to determine appropriate kinematic limits and identify the known uncertainty of the machine. For the kinematic limits, users can refer to the servo drive's datasheet to find out

the maximum possible actuator velocity, acceleration and jerk limits that are within the saturation level (linear behavior) and operational range [16, 17]. To identify the known uncertainties, one way to do it is to manually measure the machine models in the form of frequency response function, if there are no available information about variability of machine dynamics. The measurement process requires: (1) an accelerometer to measure the machine’s accelerations, (2) a trajectory command composed to sine sweep acceleration signals at several frequencies, and (3) and post-processing tool to analyze the accelerometer data to generate a frequency response function. This process should be repeated multiple times with different operating conditions to collect a set of frequency response functions, such as by varying the magnitude of acceleration used in sine sweep signal or nominal position of the machine.

The third step is to tune the parameters for feedrate optimization, namely: (1) window length (denoted as N_p in Chapter 3), (2) feature lengths (denoted as n_2 and n_3 in Chapter 4), and (3) standard deviation of the unobserved noise (denoted as σ_ϵ in Chapter 4). As a rule-of-thumb, N_p can be tuned such that it is at least longer than the time required for the slowest open-loop response to reach 95% of the steady-state [90]. The feature lengths n_2 and n_3 can be tuned accordingly to the user’s desired balance between prediction accuracy and computational cost. The trade-off can also be determined by other limiting factors, such as computing power of the controller hardware. Higher values of n_2 and n_3 will lead to more accurate predictions, but will increase computation time because of the larger size of ψ_t and the corresponding weight β , and hence the size of the matrix Σ_β which needs to be inverted during Bayesian regression. Finally, one way of tuning σ_ϵ is to use maximum likelihood estimation [81] to determine the optimal value of σ_ϵ at initial batch $j = 0$. Another approach is to jointly learn σ_ϵ along with $\beta \sim N(\mu_\beta, \Sigma_\beta)$ and $\psi_t \sim N(\mu_{\psi_t}, \Sigma_{\psi_t})$ at every batch of the Bayesian regression, at the cost of higher computational complexity.

Finally, for real-time implementation of feedrate optimization, users may encounter computational bottleneck in the application stage and may need to improve the computational efficiency of the executable script, depending on the controller hardware’s computing power. One technique is to tune certain optimization parameters inside the solvers by referring to manuals such as [91, 92]. Efficient numerical techniques such as QR decomposition to invert matrices [93] or down-sampling can also be used to reduce computational load.

5.3 Suggestions for future research

The objective of this dissertation was to develop a framework and a set of methodologies for feedrate optimization subject to servo error constraints using uncertainty-aware physics-based and data-driven servo dynamics model. This section provides the knowledge gaps and suggested directions remaining for possible future research.

First, the feedback loop created by the data-driven model used for feedrate optimization can suffer from prediction instability. Although the kinematic limits inside the feedrate optimization indirectly prevent motion instability, the trajectory may never be able to reach the end because the servo error constraints can be active at every remaining window after reaching prediction instability due to inaccurate predictions. Towards ensuring stability, the use of spectral radius [94] to detect instability for slowly-varying linear time-varying systems can be explored. Stability theories developed from data-driven model predictive control approaches [95] can also be applied to guarantee stability in our work.

Second, the uncertainty-aware servo dynamic models make several assumptions and approximations, such as: (1) assuming a Gaussian distribution form of machine dynamics' known uncertainty, (2) approximating the posterior distribution of data-driven models' weights as Gaussian using Laplace's Approximation, and (3) linearization of contour error with respect to a feature vector estimated using a nominal physics-based model. To extend our dissertation to general cases with non-Gaussian uncertainty where (1) and (2) may not hold, hierarchical Bayesian regression methods [96] or Gibbs sampling techniques [80] can be used. For cases with inaccurate nominal physics-based model or highly nonlinear contour error constraints where (3) may not hold, sophisticated nonlinear optimization solvers may be used.

Finally, additional forms of learning, such as part-to-part or machine-to-machine learning, will be investigated to improve prediction accuracy in the framework of uncertainty-aware servo dynamic models.

APPENDIX A

Intelligent Feedrate Optimization using Physics-based and Data-driven Digital Twin with an Internal Model

This appendix describes the framework and the methodology for intelligent feedrate optimization using physics-based and data-driven digital twin, which is briefly discussed in Section 4.2.1 in Chapter 4. The materials in this appendix are based on the following publication:

- Kim H, Okwudire CE. *Intelligent feedrate optimization using a physics-based and data-driven digital twin*. CIRP Annals. 2023.

A.1 Framework for intelligent feedrate optimization

The framework for the proposed intelligent feedrate optimization is depicted in Figure A.1. First, an operator submits a part together with the desired contour error tolerance to an intelligent manufacturing machine. The goal of the machine is to autonomously produce the part as quickly as possible while respecting the given error tolerance and other machine constraints. The machine is equipped with a digital twin (DT) comprising a physics-based model of its servo dynamics together with a data-driven model that is trained on data gathered from the machine's sensors. The intelligent feedrate optimization algorithm uses the servo error predictions from the DT to determine the fastest feedrate to run the machine while respecting the acceptable limits for the servo errors (and the kinematic limits of the machine). The measured sensor output is compared with the predicted output and used to adjust the data-driven model and optimization algorithm in the next iteration of the feedrate optimization.

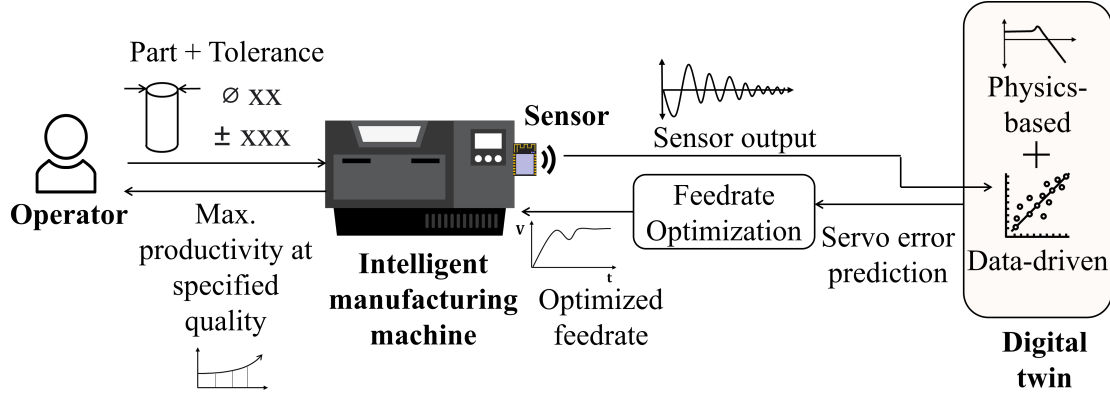


Figure A.1: Diagram of intelligent feedrate optimization framework using a digital twin

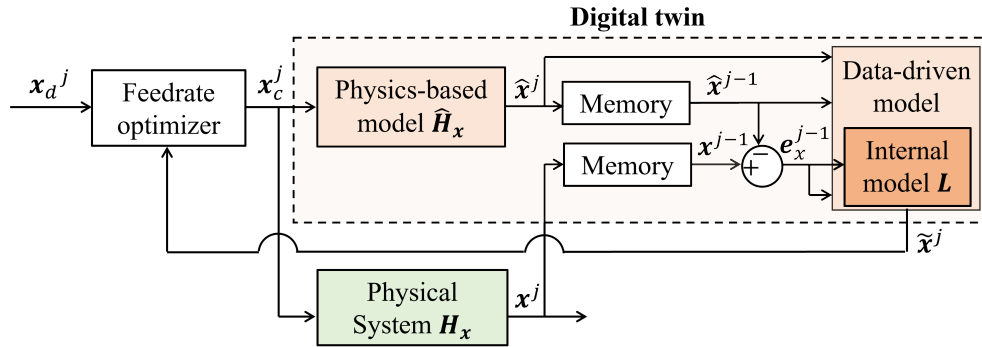


Figure A.2: Flowchart of intelligent feedrate optimization using digital twin (y-axis omitted for simplicity)

A.2 DT-based servo error prediction including cutting force effects

A flowchart of the proposed intelligent feedrate optimization is given in Figure A.2 for the x -axis of a machine tool. Small batches (windows) x_d^j of a desired position trajectory are fed into an intelligent feedrate optimizer to produce the optimized motion command, x_c^j where $j = 0, 1, 2, \dots$, represents the batch index. The optimized motion commands are sent to the servo system H_x to produce actual position x^j . The servo system is composed of a servo error pre-compensation C_x followed by machine dynamics G_x , i.e., $H_x = G_x C_x$.

A key requirement for the feedrate optimization is accurate prediction of servo errors which is achieved using a DT based on the hybrid model presented in [79] augmented with a periodic internal model to facilitate prediction of cutting-force-induced servo errors. The hybrid model takes input x_c^j and predicts the actual position \hat{x}_c^j using a physics-based model \hat{H}_x of H_x . The predictions \hat{x}_c^j do not capture the effects of unmodeled dynamics and cutting force disturbances.

Therefore, the prediction error (delayed by one batch) of $\hat{\mathbf{H}}_x$ is computed as $\mathbf{e}_x^{j-1} = \mathbf{x}^{j-1} - \hat{\mathbf{x}}^{j-1}$ and combined with $\hat{\mathbf{x}}^j$ and $\hat{\mathbf{x}}^{j-1}$ then fed into a data driven model to generate an improved prediction $\tilde{\mathbf{x}}^j$ which is used for constraining servo errors in the feedrate optimization.

Each element $\tilde{x}(t)$ of $\tilde{\mathbf{x}}^j$ (where $t = 0, T_s, 2T_s, 3T_s, \dots$ represents discrete time at sampling interval T_s) is computed as

$$\tilde{\mathbf{x}}(t) = \boldsymbol{\psi}_t \mathbf{W} \quad (\text{A.1})$$

where $\boldsymbol{\psi}_t$ is a row feature vector at time t and \mathbf{W} is the corresponding column weight vector. The sub elements of $\boldsymbol{\psi}_t$ are given by

$$\boldsymbol{\psi}_t = \left[\underbrace{1}_{:=\boldsymbol{\psi}_{t1}} \quad \underbrace{\hat{x}(t - n_2 T_s) \cdots \hat{x}(t)}_{:=\boldsymbol{\psi}_{t2}} \quad \underbrace{e_x(t - n_3 T_s) \cdots \hat{x}(t - T_s)}_{:=\boldsymbol{\psi}_{t3}} \quad \underbrace{\hat{e}_x(t - n_3 T_s) \cdots \hat{e}_x(t - T_s)}_{:=\boldsymbol{\psi}_{t4}} \right] \quad (\text{A.2})$$

The first three elements $\boldsymbol{\psi}_{t1}$, $\boldsymbol{\psi}_{t2}$ and $\boldsymbol{\psi}_{t3}$ were contained in the hybrid model of [79]. They respectively represent a bias term, the past n_2 and current time steps of \hat{x} and the past n_3 time steps of e_x , where n_2 and n_3 are user defined. The fourth element $\boldsymbol{\psi}_{t4}$ is new in the proposed hybrid model. It consists of $\boldsymbol{\psi}_{t3}$ filtered by an internal model L that contains information about dominant frequency components of the cutting force. Specifically, the internal model, in Laplace domain, is a filter of the form

$$L(s) = \sum_i \frac{\omega_i^2}{s^2 + \omega_i^2} \quad (\text{A.3})$$

where s is the Laplace operator and ω_i (rad/s) are key harmonic frequencies contained in the cutting forces. Notice that the filter in Eq. (A.3) introduces infinite gain (poles) at each ω_i , hence ensuring that the data-driven model emphasizes dynamics occurring at ω_i .

$\hat{\mathbf{x}}^j$ at the j -th batch is predicted based on weight \mathbf{W} from the previous batch $j-1$, which is trained as follows. Given that the length of each batch is n_w , for the 0-th batch, i.e., $t = 0, T_s, \dots, (n_w-1)T_s$, the weight vector \mathbf{W} and its covariance matrix \mathbf{P} are initialized using ridge regression with regularization factor λ

$$\begin{aligned} \mathbf{W} &= (\lambda \mathbf{I} + \boldsymbol{\psi}_t^T \boldsymbol{\psi}_t)^{-1} \boldsymbol{\psi}_t^T x(t) \\ \mathbf{P} &= (\lambda \mathbf{I} + \boldsymbol{\psi}_t^T \boldsymbol{\psi}_t)^{-1} \end{aligned} \quad (\text{A.4})$$

For the rest of batches $j = 1, 2, \dots$, i.e., $t = n_w T_s, (n_w + 1)T_s, \dots$, \mathbf{W} and \mathbf{P} are corrected via recursive least-squares using a forgetting factor f_0 as

$$\begin{aligned}
W &\leftarrow \mathbf{W} + k(x(t) - \boldsymbol{\psi}_t \mathbf{W}), \text{ where } k = \mathbf{P} \boldsymbol{\psi}_t^T (f_0 + \boldsymbol{\psi}_t \mathbf{P} \boldsymbol{\psi}_t^T)^{-1} \\
P &\leftarrow \frac{\mathbf{P} - k \boldsymbol{\psi}_t \mathbf{P}}{f_0}
\end{aligned} \tag{A.5}$$

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

$$e_x = \mathbf{x} - \hat{\mathbf{x}} \approx \mathbf{x} - \tilde{\mathbf{x}} \tag{A.6}$$

and $\boldsymbol{\psi}_{t4}$ can be similarly expressed using the approximated $\boldsymbol{\psi}_{t3}$.

Finally, we show that $\tilde{\mathbf{x}}(t)$ is linear in terms of \mathbf{x}_d^j , by showing that the only alterable feature in $\boldsymbol{\psi}_t$, which is the last term in $\boldsymbol{\psi}_{t2}$ (i.e., $\hat{\mathbf{x}}(t)$), is linear in \mathbf{x}_d^j . Let $\Phi_x \in \mathbb{R}^{n_h \times n_h}$ be the matrix (lifted domain) representation of $\hat{\mathbf{H}}_x$ 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

$$\Phi_x = \begin{bmatrix} \vdots & \vdots \\ \Phi_{x,p} & \Phi_{x,c} \end{bmatrix} \tag{A.7}$$

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

$$\begin{aligned}
\hat{\mathbf{x}}^j &= \Phi_{x,c} \mathbf{x}_d^j + \Phi_{x,p} \mathbf{x}_{c,p} \\
\hat{\mathbf{x}}(t) &= \mathbf{M}_t \Phi_{x,c} \mathbf{x}_d^j + \mathbf{M}_t \Phi_{x,p} \mathbf{x}_{cp}
\end{aligned} \tag{A.8}$$

where \mathbf{M}_t is a selection matrix that picks the entry at timestep t .

Let the alterable and unalterable features in Eq. (A.2) be $\boldsymbol{\psi}_{ta}(= \hat{\mathbf{x}}(t))$ and $\boldsymbol{\psi}_{tu}$, and their corresponding sub-weights be \mathbf{W}_a and \mathbf{W}_u respectively. Then, using Eq. (A.8), $\tilde{\mathbf{x}}(t)$ is linearly related to \mathbf{x}_d^j by

$$\begin{aligned}
\tilde{\mathbf{x}}(t) &= \boldsymbol{\psi}_{ta} \mathbf{W}_a + \boldsymbol{\psi}_{tu} \mathbf{W}_u \\
&= \underbrace{\mathbf{M}_t \Phi_{x,c} \mathbf{W}_a}_{:=T_{xt}} \mathbf{x}_d^j + \underbrace{\mathbf{M}_t \Phi_{x,p} \mathbf{x}_{cp} \mathbf{W}_a + \boldsymbol{\psi}_{tu} \mathbf{W}_u}_{:=T_{0t}}
\end{aligned} \tag{A.9}$$

A.3 Feedrate optimization using DT

The feedrate optimization using the DT is formulated in accordance with the previous Chapter 3. Taking the x -axis, for example, a desired trajectory $\mathbf{X}_d = f(p)$ is parametrized with respect to a normalized, monotonically increasing path variable p . Then, $X_d(t)$ is linearized as $x_d(t)$ with respect to $p(t)$ using an estimated linearization point $p_e(t)$ as

$$x_d(t) = - \left. \frac{\partial f}{\partial p} \right|_{p=p_e(t)} \cdot (p(t) - p_e(t)) + f(p_e(t)) \quad (\text{A.10})$$

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

$$\begin{aligned} \min \mathbf{1}^T \mathbf{p}^j \\ \text{s.t. } p(t-1) \leq p(t) \leq 1 \\ D[\mathbf{p}^j] \leq \mathbf{V}_{max} T_s \\ |D^2[\mathbf{x}_d^j]| \leq \mathbf{A}_{max} T_s^2 \end{aligned} \quad (\text{A.11})$$

where $\mathbf{1}$ is a ones-vector, D is a difference operator, and \mathbf{V}_{max} and \mathbf{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 position predictions $\tilde{\mathbf{x}}^j$ of the DT are integrated into the feedrate optimization by using them to constrain the servo errors. To do this, we leverage the fact that the proposed hybrid model is a linear operator, because its physics-based and data-driven components are both linear operators. Therefore, the predicted servo error $\tilde{\epsilon}_x(t)$ for $t = jn_w T_s, \dots, ((j+1)n_w - 1)T_s$ in the j -th batch can be expressed using $\tilde{\mathbf{x}}(t)$ from Eq. (A.9) as

$$\tilde{\epsilon}_x(t) = x_d(t) - \tilde{\mathbf{x}}(t) = (\mathbf{M}_t - \mathbf{T}_{xt}) \mathbf{x}_d^j - \mathbf{T}_{0t} \quad (\text{A.12})$$

The processes described above are repeated for the y -axis to compute $\tilde{\epsilon}_y^j$. Finally, contour error ϵ can be estimated from DT-predicted axis tracking errors $\tilde{\epsilon}_x$ and $\tilde{\epsilon}_y$ and constrained under tolerance \mathbf{E}_{max} using a linear approximation [56] as

$$|\epsilon| = -|\sin(\theta)\tilde{\epsilon}_x + \cos(\theta)\tilde{\epsilon}_y| \leq \mathbf{E}_{max} \quad (\text{A.13})$$

where θ is inclination angle of the curve $(\mathbf{x}_d, \mathbf{y}_d)$, and \mathbf{E}_{max} is the vectorised representation of

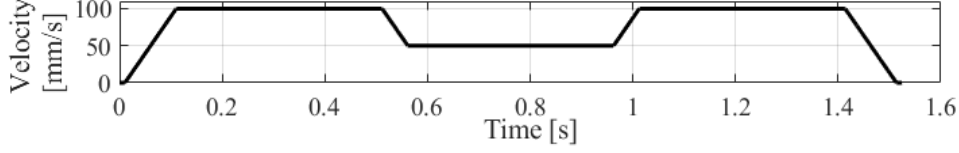


Figure A.3: Velocity of desired trajectory on x -axis)

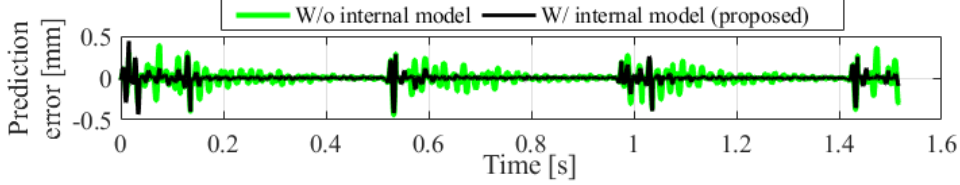


Figure A.4: Prediction error using DT without and with the internal model

E_{max} .

Overall, the DT-based intelligent feedrate optimization becomes a linear programming in terms of the decision variable p^j .

A.4 Numerical validation of DT-based servo error prediction

A simple simulation case study is presented here to highlight the importance of augmenting the hybrid model in [79] with an internal model to enable accurate prediction of motion-induced and cutting-force-induced servo errors. To do so, a desired trajectory on x -axis with its velocity shown in Figure A.3 is selected with maximum velocity as 100 mm/s and acceleration as 1 m/s². The simulated output x is modeled as sum of motion-induced position x_m and cutting-force-induced position x_f as

$$\begin{aligned} \mathbf{x} &= \mathbf{x}_m + \mathbf{x}_f = \mathbf{H}_x \mathbf{x}_d + \mathbf{x}_f \\ x_f(t) &= 0.01 \dot{x}_m(t) \sin(\omega_f t) \end{aligned} \quad (\text{A.14})$$

where $\omega_f = 523.6$ rad/s (5000 rpm) is used. For the DT, $n_w = 20$, $n_2 = 2$, $n_3 = 20$, $\lambda = 0.01$ and $f_0 = 1$ are used; for the internal model, $\omega_i = \omega_f$ is used. The data-driven model is trained for one cycle of the trajectory in Figure A.3 prior to testing. The physics-based model $\hat{\mathbf{H}}_x$ is defined as a 2nd order system with $\omega_n = 157$ rad/s and $\zeta = 0.05$; \mathbf{H}_x is modeled with mismatch in coefficients by 10% deviation as $\omega'_n = 1.1\omega_n$ and $\zeta' = 0.9\zeta$ i.e.,

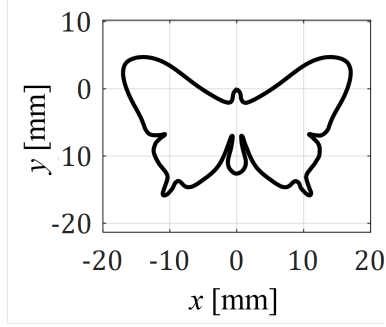


Figure A.5: Desired toolpath

$$\begin{aligned}
 \mathbf{H}_x &= \frac{2\zeta\omega_n s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \\
 \tilde{\mathbf{H}}_x &= \frac{2\zeta'\omega'_n s + \omega_n'^2}{s^2 + 2\zeta'\omega'_n s + \omega_n'^2}
 \end{aligned} \tag{A.15}$$

Figure A.4 shows the prediction error using the DT without and with the internal model. Using the internal model, the RMS prediction error of x is reduced by 27.7%. The following Sections A.5 and A.6 will highlight the performance of the DT-based feedrate optimization using the internal model in experiments.

A.5 Experimental setup

For experimental validation, Nomad3 three-axis CNC machine tool prototype is used, as shown in Figure A.6. 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 on the x and y - axes gantries to measure x and y - axes positions that are fed back to dSPACE 1007.

A frequency response function (FRF) is measured for the x and y axis of the machine to identify the machine dynamic component of physics-based model, $\hat{\mathbf{G}}_x$ and $\hat{\mathbf{G}}_y$. The input of each FRF measurement is a swept sine acceleration command to the stepper motors, and the output is the relative acceleration between the x and y axis using two PCB 393B05 shear accelerometers. Then, the measured FRFs are modelled via curve fitting. The discrete-time transfer function of $\hat{\mathbf{G}}_x$ and $\hat{\mathbf{G}}_y$ with $T_s = 2\text{ms}$ are

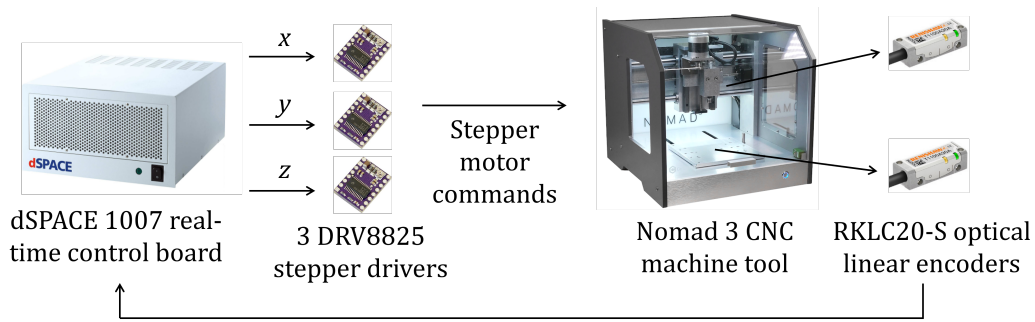


Figure A.6: Experimental Setup

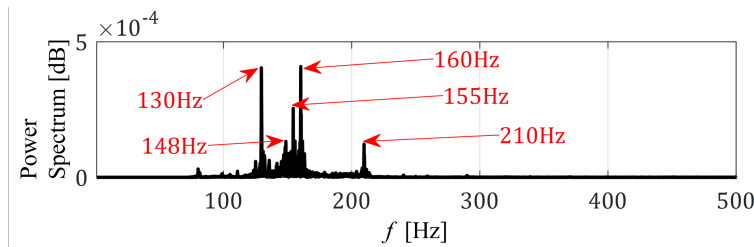


Figure A.7: Experimentally identified disturbance frequencies for internal model

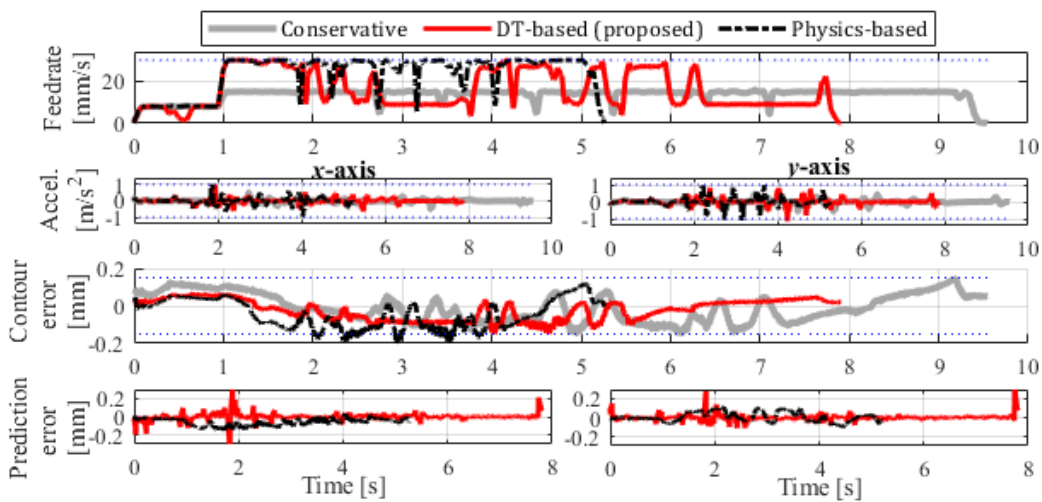


Figure A.8: Feedrate, acceleration, contour error and prediction error using conservative, physics-based and DT-based feedrate optimization in air-cutting

$$\begin{aligned}\hat{G}_x &= \frac{0.487z^3 - 0.8471z^2 + 0.7827z - 0.3768}{z^4 - 2.149z^3 + 2.037z^2 - 0.9917z + 0.1495} \\ \hat{G}_y &= \frac{0.4378z^3 + 0.4994z^2 - 0.0569z + 0.0128}{z^4 - 0.1282z^3 + 0.0217z^2 - 0.001z + 0.0001}\end{aligned}\tag{A.16}$$

The servo error pre-compensation C_x and C_y are formulated using \hat{G}_x and \hat{G}_y via limited-preview filtered B-spline method in [74].

A.6 Experimental validation of DT-based feedrate optimization

The proposed DT-based approach is validated by comparing its performance with physics-based approach (without data-driven model) and conservative method generated using a trapezoidal acceleration profile with kinematic limits tuned by trial-and-error to achieve the servo error tolerances. A butterfly toolpath [75] in Figure A.9 is used for air cutting and machining an aluminum workpiece with 3.175 mm diameter flat-end mill and spindle speed of 7000 rpm. Kinematic limits are set as $V_{max} = 30$ mm/s and $A_{max} = 1$ m/s² in the feedrate optimization; $n_w = 30$, $n_2 = 2$, $n_3 = 30$, $\lambda = 0.01$ and $f_0 = 1$ are used in the data-driven model, which is pre-trained using one cycle of physics-based feedrate optimization. The frequencies ω_i of L in Eq. (A.3) are experimentally identified as in Figure A.7, based on encoder data measured during cutting. A safety factor of 25% is applied to the desired tolerance, i.e., a contour error bound of $E_{max} = 0.12$ mm is applied in optimization to achieve error tolerance of 0.15 mm in experiments.

Figures A.8 and A.9 show the profiles of commanded feedrate, acceleration, contour error, and prediction error of x and y using conservative, physics-based, and DT-based approaches in air-cutting and machining, respectively. The DT-based approach reduces the prediction error in x -axis by 47.3% and 45.7%, and in y -axis by 4.0% and 34.6% during air-cutting and machining, respectively, compared to the physics-based method, which allows the tolerance to be satisfied. However, the physics-based method violates the tolerance due to unmodeled dynamics and/or cutting force. As a result, the proposed approach completes the motion 35.0% and 17.2% faster than the conservative method in air-cutting and machining, respectively, without sacrificing contouring accuracy. The DT-based algorithm runs in real-time by computing the entire trajectory for machining within 2.04 s. Figure A.10 shows the machined surfaces using trajectories from Figure A.9. The surface quality of the proposed approach is improved compared to that of physics-based approach, while staying similar to that of conservative part.

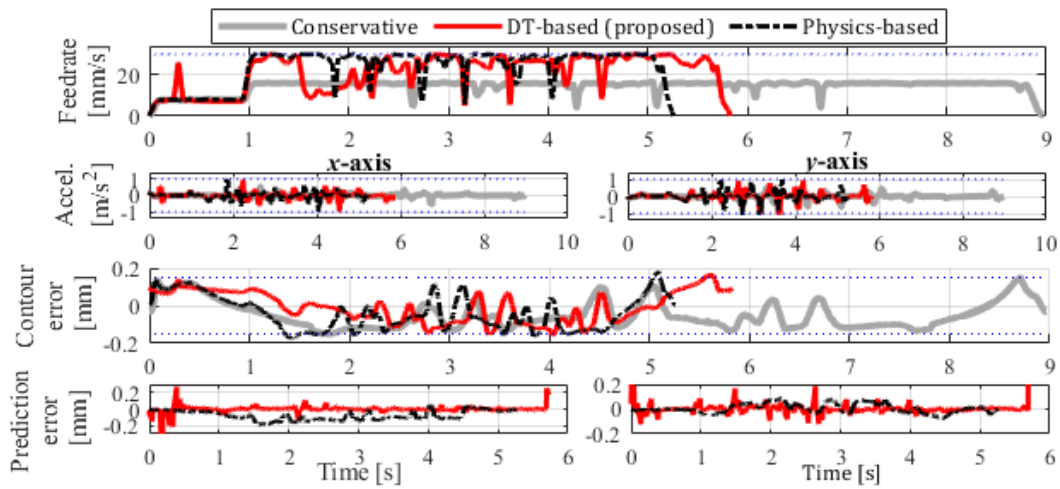


Figure A.9: Feedrate, acceleration, contour error and prediction error using conservative, physics-based and DT-based feedrate optimization in machining

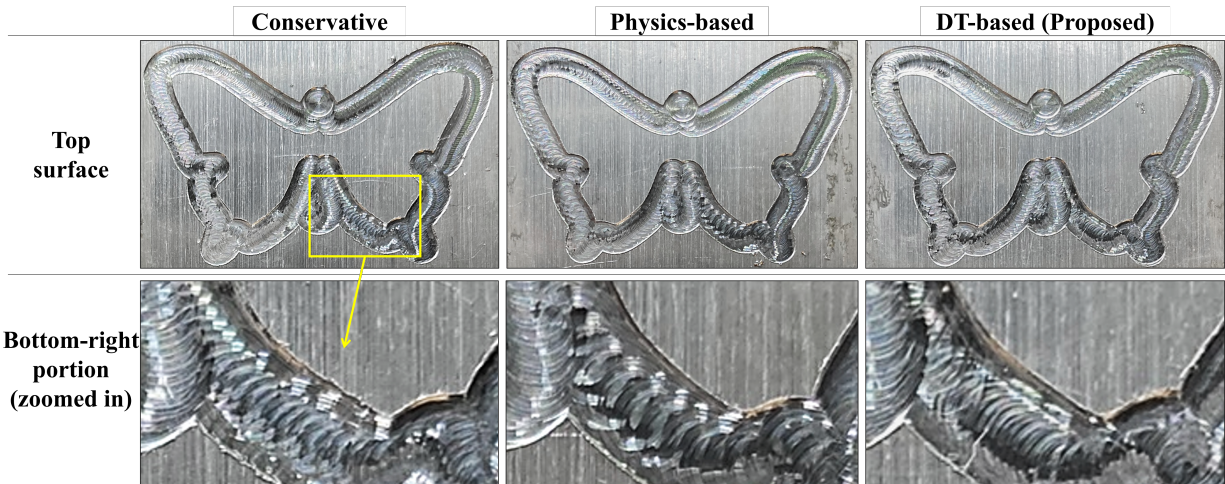


Figure A.10: Machined surfaces using the three approaches investigated

A.7 Conclusions

This appendix discusses the framework and approach for intelligent feedrate optimization using a DT that allows for a machine to produce parts with desired part quality specifications as quickly as possible. The DT is first built on a physics-based dynamics model. Then, a data-driven model with an internal model is updated on-the-fly to adapt to unknown dynamics and cutting force disturbances. Using experiments on a CNC machine tool prototype, the proposed approach showed its performance in accurately constraining the contour error while reducing cycle time by up to 35% compared to conservative approach.

BIBLIOGRAPHY

- [1] 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.
- [2] 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.
- [3] Kim H and Okwudire CE. Intelligent feedrate optimization using a physics-based and data-driven digital twin. *CIRP Annals*, 2023.
- [4] Alkafafi L. Time and frequency optimal motion control of cnc machine tools. *International Journal of Machine Tools and Manufacture*, 46(10):1107–1123, 2006.
- [5] Duan M, Yoon D, and Okwudire CE. A limited-preview filtered b-spline approach to tracking control—with application to vibration-induced error compensation of a 3d printer. *Mechatronics*, 56:287–296, 2018.
- [6] Chatter marks during face milling. <https://www.planlauf.com/en/measurement/diagnosis-of-chatter-marks/>. Accessed 26 Apr 2023.
- [7] Lee J and Okwudire CE. Reduction of vibrations of passively-isolated ultra-precision manufacturing machines using mode coupling. *Precision Engineering*, 43:164–177, 2016.
- [8] Möhring HC, Wiederkehr P, Erkorkmaz K, and Kakinuma Y. Self-optimizing machining systems. *CIRP Annals*, 69(2):740–763, 2020.
- [9] Face milling machines. <https://www.mmsonline.com/articles/three-face-milling-myths-truths>. Accessed 26 Apr 2023.
- [10] Park nx-wafer operation process. <https://m.parksystems.com/products/industrial/park-wafer>. Accessed 10 May 2023.
- [11] Fff 3d printer. <https://3dprinting.com/news/a-closer-look-at-ai-build/>. Accessed 26 Apr 2023.
- [12] 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.

- [13] Erkorkmaz K, Chen QGC, Zhao MY, Beudaert X, , and Gao XS. Linear programming and windowing based feedrate optimization for spline toolpaths. *CIRP Annals*, 66(1):393–396, 2017.
- [14] 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.
- [15] Zhang Q and Li SR. Efficient computation of smooth minimum time trajectory for cnc machining. *Int J Adv Manuf Technol*, 68:683–692, 2013.
- [16] Sencer B, Altintas Y, and Croft E. Feed optimization for five-axis cnc machine tools with drive constraints. *International Journal of Machine Tools and Manufacture*, 48(7–8):733–745, 2008.
- [17] Beudaert X, Lavernhe S, and Tournier C. Feedrate interpolation with axis jerk constraints on 5-axis nurbs and g1 tool path. *International Journal of Machine Tools and Manufacture*, 57:73–82, 2012.
- [18] Zhang Q, Li S, Guo JX, and Gao XS. Time-optimal path tracking for robots under dynamics constraints based on convex optimization. *Robotica*, 34(9):2116–2139, 2016.
- [19] 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.
- [20] 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.
- [21] Dong J and Stori JA. Optimal feed-rate scheduling for high-speed contouring. *J. Manuf. Sci. Eng.*, 129(1):63–76, 2007.
- [22] 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.
- [23] 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.
- [24] 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.
- [25] 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.
- [26] 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.

- [27] 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.
- [28] 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.
- [29] Altintas Y, Kersting P, Biermann D, Budak E, Denkena B, and Lazoglu I. Virtual process systems for part machining operations. *CIRP Annals*, 63(2):585–605, 2014.
- [30] Lee HU and Cho DW. An intelligent feedrate scheduling based on virtual machining. *The International Journal of Advanced Manufacturing Technology*, 22:873–882, 2003.
- [31] Kim SJ, Lee HU, and Cho DW. International journal of machine tools and manufacture. *CIRP Annals*, 46(12–13):1589–1597, 2006.
- [32] Jerard RB and Hemmett JG Fussell BK, Ercan MT. Integration of geometric and mechanistic models of nc machining into an open-architecture machine tool controller. volume 26652, pages 675–682, 2000.
- [33] Sun Y, Ren F, Guo D, and Jia Z. Estimation and experimental validation of cutting forces in ball-end milling of sculptured surfaces. *International Journal of Machine Tools and Manufacture*, 59(15):1238–1244, 2009.
- [34] Larue A and Altintas Y. Simulation of flank milling processes. international journal of machine tools and manufacture. *International Journal of Machine Tools and Manufacture*, 45(4–5):549–559, 2005.
- [35] Zhang L. Process modeling and toolpath optimization for five-axis ball-end milling based on tool motion analysis. *The International Journal of Advanced Manufacturing Technology*, 57:905–916, 2011.
- [36] 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.
- [37] Altintas Y and Merdol SD. Virtual high performance milling. *CIRP annals*, 56(1):81–84, 2007.
- [38] 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.
- [39] Merdol SD and Altintas Y. Virtual simulation and optimization of milling applications—part ii: optimization and feedrate scheduling.
- [40] 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.

- [41] 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.
- [42] 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.
- [43] 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.
- [44] 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.
- [45] Xie J, Zhao P, Hu P, Yin Y, Zhou H, Chen J, and Yang J. Multi-objective feed rate optimization of three-axis rough milling based on artificial neural network. *The International Journal of Advanced Manufacturing Technology*, 114:1323–1339, 2021.
- [46] Erwinski K, Paprocki M, Wawrzak A, and Grzesiak LM. Neural network contour error predictor in cnc control systems. In *21st International Conference on Methods and Models in Automation and Robotics (MMAR)*, pages 537–542, 2016.
- [47] Ferry WB and Altintas Y. Virtual five-axis flank milling of jet engine impellers—part ii: feed rate optimization of five-axis flank milling. *Journal of manufacturing science and engineering*, 130(1), 2008.
- [48] 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.
- [49] Zuperl U, Cus F, and Reibenschuh M. Neural control strategy of constant cutting force system in end milling. *Robotics and Computer-Integrated Manufacturing*, 27(3):485–493, 2011.
- [50] 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.
- [51] Bharathi A and Dong J. Feedrate optimization for smooth minimum-time trajectory generation with higher order constraints. *CIRP Annals*, 82(5):1029–1040, 2016.
- [52] Erkorkmaz K and Heng M. A heuristic feedrate optimization strategy for nurbs toolpaths. *CIRP Annals*, 57(1):407–410, 2008.
- [53] Zhang K, Yuan CM, Gao XS, and Li H. A greedy algorithm for feedrate planning of cnc machines along curved tool paths with confined jerk. *Robotics and Computer-Integrated Manufacturing*, volume=28, number=4, pages=472–483, year=2012,.
- [54] Zhang Y, Ye P, Zhao M, and Zhang H. Dynamic feedrate optimization for parametric tool-path with data-based tracking error prediction. *Mechanical Systems and Signal Processing*, 120:221–223, 2019.

- [55] Erkorkmaz K, Zhao MY, Chen QGC, Beudaert X, , and Gao XS. Linear programming and windowing based feedrate optimization for spline toolpaths. *CIRP Annals*, 66(1):393–396, 2017.
- [56] Good MC, Lam D, Manzie C. Model predictive contouring control for biaxial systems. *IEEE Transactions on Control Systems Technology*, 2(1):552–559, 2012.
- [57] Gordon DJ and Erkorkmaz K. Accurate control of ball screw drives using pole-placement vibration damping and a novel trajectory prefilter. *Precision Engineering*, 37(2):308–322, 2013.
- [58] Guo J, Zhang Q, Gao XS, and Li H. Time optimal feedrate generation with confined tracking error based on linear programming. *Journal of Systems Science and Complexity*, 28(1):90–95, 2015.
- [59] Altintas Y and Khoshdarregi MR. Contour error control of cnc machine tools with vibration avoidance. *CIRP Annals*, 61(1):335–338, 2012.
- [60] Zhang K, Yuen A, and Altintas Y. Pre-compensation of contour errors in five-axis cnc machine tools. *Int J Mach Tool Manu*, 74:1–11, 2013.
- [61] 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.
- [62] Koren Y and Lo CC. Variable-gain cross-coupling controller for contouring. *CIRP Annals*, 40(1):371–374, 1991.
- [63] Chin JH and Lin TC. Cross-coupled precompensation method for the contouring accuracy of computer numerically controlled machine tools. *Int J Mach Tool Manu*, 37(7):947–967, 1997.
- [64] Ramani KS, Duan M, Okwudire CE, and Ulsoy AG. Tracking control of linear time-invariant nonminimum phase systems using filtered basis functions. *Journal of Dynamic Systems, Measurement, and Control*, 139(1):011001–1 to 011001–11, 2017.
- [65] Tomizuka M. Zero error tracking algorithm for digital control. *ASME J Dyn Sys Meas Control*, 109(1):65–68, 1987.
- [66] Okwudire C, Ramani K, and Duan M. Trajectory optimization method for improved tracking of motion commands using cnc machines that experience unwanted vibration. *CIRP Annals*, 65(1):373–376, 2016.
- [67] Luenberger DG. An introduction to observers. *IEEE Trans. On Automatic Control*, 16(6):596–602, 1971.
- [68] Okwudire C, Huggi S, Supe S, Huang C, and Zeng B. Low-level control of 3d printers from the cloud: A step toward 3d printer control as a service. *Inventions*, 3(3):56, 2018.
- [69] Nocedal J and Wright SJ. *Numerical Optimization*. Berlin, New York, 2006.

- [70] Camacho EF and Bordons C. *Model predictive control*. Springer Science & Business Media, 2013.
- [71] Gordon DJ and Erkorkmaz K. Accurate control of ball screw drives using pole-placement vibration damping and a novel trajectory prefilter. *Precision Engineering*, 37(2):308–322, 2013.
- [72] Suh SH, Kang SK, Chung DH, and Stroud IY. *Theory and design of CNC systems*. Springer-Verlag London Limited, 2008.
- [73] Okwudire C, Ramani K, and Duan M. A trajectory optimization method for improved tracking of motion commands using cnc machines that experience unwanted vibration. *CIRP Annals*, 65(1):373–376, 2016.
- [74] Duan M, Yoon D, and Okwudire CE. A limited-preview filtered b-spline approach to tracking control—with application to vibration-induced error compensation of a 3d printer. *Mechatronics*, 56:287–296, 2018.
- [75] 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.
- [76] Erkorkmaz K and Altintas Y. Quintic spline interpolation with minimal feed fluctuations. *Journal of Manufacturing Science and Engineering*, 127(2):339–349, 2005.
- [77] Okwudire C, Huggi S, Supe S, Huang C, and Zeng B. Low-level control of 3d printers from the cloud: A step toward 3d printer control as a service. *Inventions*, 3(3):56, 2018.
- [78] Butterfly plate. <https://www.thingiverse.com/thing:4658078>. Accessed 18 Nov 2020.
- [79] 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.
- [80] Polasek W and Krause A. Bayesian regression model with simple errors in variables structure. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 42(5):571–580, 1993.
- [81] Bishop CM and Nasrabadi NM. *Pattern recognition and machine learning*, volume 4. New York: springer, 2006.
- [82] MacKay DJ. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.
- [83] Rawlings JB. Model predictive control technology. volume 1, pages 662–676, 1999.
- [84] Liniger A, Varano L, Rupenyan A, and Lygeros J. Real-time predictive control for precision machining. pages 7746–7751, 2019.
- [85] Rupenyan A, Khosravi M, and Lygeros J. Performance-based trajectory optimization for path following control using bayesian optimization. pages 2116–2121, 2021.

- [86] 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.
- [87] Luenberger DG. An introduction to observers. *IEEE Trans. On Automatic Control*, 16(6):596–602, 1971.
- [88] 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.
- [89] Zhang Q and Li SR. Efficient computation of smooth minimum time trajectory for cnc machining. *Int J Adv Manuf Technol*, 68:683–692, 2013.
- [90] Garriga JL and Soroush M. Model predictive control tuning methods: A review. *industrial engineering chemistry research*. *Int J Adv Manuf Technol*, 49(8):3505–3515, 2010.
- [91] Gurobi optimization parameters. <https://www.gurobi.com/documentation/9.0/refman/parameters.html>. Accessed 11 Sep 2023.
- [92] Matlab fmincon parameters. <https://www.mathworks.com/help/optim/ug/fmincon.html#busp5fq-6>. Accessed 11 Sep 2023.
- [93] Trefethen LN and Bau D. *Numerical linear algebra*, volume 181. Siam, 2022.
- [94] Chou CH, Duan M, and Okwudire CE. A physics-guided data-driven feedforward tracking controller for systems with unmodeled dynamics—applied to 3d printing. *IEEE Access*, 11:14563–14574, 2023.
- [95] Berberich J, Kohler J, Muller MA, and Alligower F. Data-driven model predictive control with stability and robustness guarantees. *IEEE Transactions on Automatic Control*, 66(4):1702–1717, 2020.
- [96] Banerjee S, Carlin BP, and Gelfand AE. Hierarchical modeling and analysis for spatial data. *IEEE Transactions on Automatic Control*, 2003.