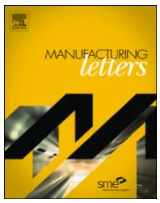
Available online at www.sciencedirect.com

Manufacturing Letters

Manufacturing Letters 00 (2025) 000–000



54th SME North American Manufacturing Research Conference (NAMRC 54, 2026)

An Interpretable Machine Learning based Predictive Control Method for Manufacturing Processes with Turning as A Case Study*

Mason Ma[a], Victor Wu[a], Jingang Yi[b], Hong Wang[c], Bradley Jared[d], Tony Schmitz[e], Tony Shi[a,d,*]

[a]*Manufacturing Intelligent Dynamics Laboratory, University of Tennessee Knoxville, Knoxville, TN 37996, USA*
[b]*Department of Mechanical and Aerospace Engineering, Rutgers University, Piscataway NJ, 08854, USA*
[c]*Energy Science and Technology Directorate, Oak Ridge National Laboratory, Oak Ridge, TN 37830, USA*
[d]*Department of Mechanical and Aerospace Engineering, University of Tennessee Knoxville, Knoxville, TN 37996, USA*
[e]*MSC Industrial Supply Co. Inc, Davidson, NC 28036, USA*

* Corresponding author. Tel.: 865-974-7654; *E-mail address:* tonyshi@utk.edu

Abstract

Modern manufacturing machines and sensing technologies can record accurate process signals through embedded internal controllers and instrumented external sensors, and that provides a good research opportunity for the development of data-driven modeling-based control techniques for manufacturing processes. This study develops an interpretable machine learning based predictive control method for modeling and control of manufacturing processes. A synthetic example of Lorenz 3 system with control is used to test and validate the capability of the proposed method on model discovery from noisy data, and performance of set point and trajectory controls. Numerical results show the proposed method outperforms the state-of-the-art data-driven modeling-based control methods. In addition, a case study on turning process based on experimentally validated time domain simulator is conducted. For turning process, the proposed method can discover the governing equations of turning dynamics from data and provide stable cutting process parameters with maximum material removal rate. The proposed method has the potential to be used for processing monitoring and control for machining and other manufacturing processes.

Peer-review under responsibility of the scientific committee of the NAMRI/SME.
Keywords: data-driven modeling-based control, interpretable machine learning, model predictive control, manufacturing processes, turning

1. Introduction

Through decades of technological advancements from evolution and revolution, modern manufacturing machines embedded with mature industrial controllers have become highly digital and automated complex systems, like computer numerically controlled (CNC) machine tools [1, 2], additive manufacturing machines [3, 4], and paper machines [5, 6]. Manufacturing processes occur when a machine or tool is used for processing materials. This is a complex physical process in a dynamic and stochastic manufacturing environment. These complex processes are governed by unknown thermo-mechanical dynamics, where the nonlinear dynamics determine the properties and quality of the manufactured parts. Modeling and control of manufacturing processes to obtain both desired material properties and machine productivity have been a challenging problem.

Built on the powerful capabilities of machine-embedded inner-loop controllers, the problem addressed in this paper falls in the field of data-driven modeling-based control for manufacturing processes. Particularly, this study aims to

*This manuscript has been authored in part by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The US government retains and the publisher, by accepting the article for publication, acknowledges that the US government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for US government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

Peer-review under responsibility of the scientific committee of the NAMRI/SME.

optimize the set points of the inner-loop controllers, i.e., process parameters selection at machine, by designing outer-loop controllers by utilizing interpretable machine learning and model predictive control. An emerging research opportunity lies in the digitalization and automation that enable the data recording of process control variables at machine and process state variables when the tool processes the materials. These in-process measurements can record the unknown physical information of manufacturing processes and lay a foundation for data-driven modeling-based control research. For example, given process parameters of axial depth of cut and spindle speed in CNC machining, internal high frequency controller data like spindle torque and power can be obtained from the modern industrial controller, external sensors like dynamometer and accelerometer can be instrumented to measure in-process cutting force and acceleration [7, 8]. Only a few prior studies have explored this research direction in either the manufacturing or control communities. Considering that process modeling is necessary and critical to understand the underlying physics in manufacturing processes, advanced data-driven model-free control methods [9-11] are outside the scope of this paper.

The manufacturing process modeling that can be used for control generally includes physics-based models and data-driven models [12]. The physics-based models are faced with limitations including boundary instability, high computational cost, and poor suitability for systems identification. In contrast, data-driven models have become increasingly relevant alongside advances in metrology, machine learning and artificial intelligence, and are more suitable for process monitoring and control. There are a wide variety of data-driven modeling techniques for control, including autoregressive models (NARX) [13, 14], subspace identification methods [15-17], dynamic mode decomposition [18-20], kernel-based Gaussian processes [21-24], and neural networks (NN) [13, 25]. These numerical approximation-based modeling techniques have been combined with effective control methods for dynamic systems of a variety of scales and scopes [22, 26-29]. However, pure numerical approximation cannot capture the underlying physics and therefore is not robust and reliable to manufacturing processes. Recent advances in data-driven modeling-based control by interpretable machine learning mainly adopt symbolic regression techniques with consideration of sparsity to obtain interpretable models. The representative method, SINDY-MPC [32], aims to learn ordinary differential equations (ODEs)-type models from very few samples in time, making it considerably more data efficient than NN, but with comparable or superior control performance, when the systems are symbolically representable [29]. The main procedures of SINDY-MPC are as follows: predefine a nonlinear set of state and control variables, obtain an interpretable system model from data via sparse linear regression with a threshold sequential least squares algorithm, and then use the obtained nonlinear dynamic model for model predictive control (MPC). However, the drawback of SINDY-MPC is that the used threshold sequential least squares algorithm is a heuristic. In specific, this means SINDY-MPC cannot accurately obtain system models under noisy conditions and therefore results in unstable control performance. As a summary, nonlinear system identification by existing data-driven modeling methods lacks interpretable physical meaning, often cannot generalize to new data, and are not robust to noisy measurements. These disadvantages lead to difficulties for model-based controller design, like optimal control [28, 30] and MPC [31], whose performances rely heavily on both model accuracy and anti-noise capability. Currently, there is no efficient data-driven method to deal with in-process noisy data for modeling-based control of manufacturing processes, so this study aims to fill this research gap.

More recently, discrete optimization related techniques as exact algorithms have been developed to solve sparse linear regression for data-driven modeling of nonlinear dynamic systems [32, 33]. Particularly, the discrete optimization-based interpretable machine learning method developed in [32] has shown effective model discovery capabilities under highly noisy data. This is because the discrete optimization-based exact method can solve the resulting sparse linear regression problem to optimality. This method was later extended as customized methods to model milling process [34] and a new additive manufacturing process, additive friction stir deposition [35]. However, these methods do not involve control capabilities for optimizing manufacturing processes.

This study develops an interpretable machine learning based predictive control (IMLPC) method for the purpose of modeling and control of manufacturing processes, by integrating the above discrete optimization-based interpretable machine learning into an MPC framework for investigating its control performance. This is done by utilizing current discrete optimization-based machine learning capabilities to acquire ODEs-type models that links process control variables and process state variables from highly noisy data. Because of the involved nonlinearities, the acquired models are integrated into an MPC framework for process control by optimizing the set points of the machine-embedded controllers. A toy example of the Lorenz 3 system with control is first presented to test and validate the proposed method. Then, the proposed method is validated with a case study using experimentally validated simulator of turning process. The primary contributions of this work are as follows.

- The developed IMLPC method can acquire accurate ODEs-type models and superior control performance for nonlinear dynamics under high-noise conditions. For dynamics model discovery, the anti-noise capacity of IMLPC is at least 10 times and 100 times better than SINDY-MPC for the Lorenz 3 system and turning process. Benefiting from the accurate identified models, the control system performance of IMLPC significantly outperforms SINDY-MPC and NN-based MPC (NARX-MPC). The fixed point and trajectory control results show IMLPC performs roughly 20 and 2 times better than SINDY-MPC.
- For the case study of turning process, the proposed method can efficiently discover the second order and time-delay governing equation of turning dynamics from the experimentally validated time domain simulator and provide stable cutting process parameters with maximized material removal rate. The proposed method has the

potential to be used for processing monitoring and control for machining and other manufacturing processes.

The rest of this paper is organized as follows. The problem statement is presented in Section 2. The proposed IMLPC method is described in Section 3. A toy example of Lorenz 3 system with control and a case study on turning process are presented in Sections 4 and 5, respectively. Section 6 concludes this paper.

2. Problem Definition: Data-Driven Modeling-Based Control for Manufacturing Processes

Nonlinear dynamics during manufacturing processes occur within a time window when the tool processes the materials. During this window, the underlying physics of the thermo-mechanical dynamics will be recorded by the process control and state variables for given process parameters at machine, which are defined as:

T: Process window – the time interval when the manufacturing processes occur.

$\mathbf{u}_{set}$: Process parameters/constant set points at machine as the inputs for its internal controllers to start the manufacturing process, where $\mathbf{u}_{set} \in \mathbb{R}^{1\times S}$.

$\mathbf{u}(t)$:Process control variables that can be recorded by machine internal controllers, where $\mathbf{u}(t) \in \mathbb{R}^{1\times S}, t \in T$. Note that $\mathbf{u}_{set}$ are the set points of process control variables $\mathbf{u}(t)$.

$\mathbf{x}(t)$: Process state variables that can be recorded by machine internal controllers and/or instrumented external sensors when the tool processes materials, where $\mathbf{x}(t) \in \mathbb{R}^{1\times J}, t \in T$.

Taking CNC machining as an example, the process parameters $\mathbf{u}_{set}$ are tool spindle speed Ω, axial depth of cut b, feed rate h_m, etc. These constant set points at machine inputs should be determined before the cutting process for the desired material properties. It should be noted that, for given set points $\mathbf{u}_{set}$, the machine-embedded internal controllers will regulate and record the instantaneous $\mathbf{u}(t)$ towards the given set points. Meanwhile, process state variables $\mathbf{x}(t)$ can also be recorded, including displacement x, velocity $\dot{x}$, acceleration $\ddot{x}$ of the tool tip dynamics, and the cutting force F_n that causes tooltip vibration (See Section 5 for more details).

Both process control variables $\mathbf{u}(t)$ and process state variables $\mathbf{x}(t)$ are recorded in the format of time series data for time points $t = 1, \dots, N$, where $t \in T$. The time series of $\mathbf{u}(t)$ and $\mathbf{x}(t)$ are denoted by $\mathbf{X} \in \mathbb{R}^{N\times J}$ and $\mathbf{U} \in \mathbb{R}^{N\times S}$, with noises injected due to the stochastic and dynamic manufacturing environment. Therefore, the data-driven modeling-based control problem addressed in this paper is described below.

The first principles model to describe the nonlinear dynamics with uncertainty in manufacturing processes is given as

$$\dot{\mathbf{x}} = \mathbf{f}\big(\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t)\big), \qquad t \in T \qquad (1a)$$

$$\mathbf{y} = \mathbf{g}\big(\mathbf{x}(t), \mathbf{u}(t), \mathbf{v}(t)\big), \qquad t \in T \qquad (1b)$$

where $\mathbf{y}(t) \in \mathbb{R}^{1\times M}$, $\mathbf{v}(t) \in \mathbb{R}^{1\times M}$, and $\mathbf{w}(t) \in \mathbb{R}^{1\times J}$ denote the system output vector, noise imposed on the output measurements, and noise imposed on the process state variables at time t. The governing equation Eq. (1a) is unknown and nonlinear, which needs to be explored. The system output Eq. (1b) is commonly known based on material property or machine productivity, for example, material removal rate in CNC machining. It is noted that when $t \notin T$, the nonlinear dynamics in Eqs. (1) do not exist since the manufacturing process does not happen. This is the rationale of data-driven modeling of manufacturing process utilizing in-process measurements of $\mathbf{x}(t)$ and $\mathbf{u}(t)$ to explore the analytical form of Eq. (1a).

Once the dynamic model Eq. (1a) is obtained, a nonlinear model-based outer-loop controller needs to be designed to optimize the manufacturing processes. The resulting optimal control problem can be generally formulated as

$$\mathbf{u}^*(\cdot) = \arg\max_{\mathbf{u}\in\mathcal{U}} L \coloneqq \int \ell\big(\mathbf{y}(t), \mathbf{r}(t)\big)\, dt\,, \qquad (2)$$

where $\mathcal{U}$ denotes the constraint sets that can include the unknown governing equations in Eq. (1a) and other physical constraints required by tool and materials in manufacturing process. For example, the chatter stability in CNC machining.

Within the time window T, the optimal process parameters $\mathbf{u}^*(\cdot)$ are a series of control inputs or constant set points, that can be used for the closed-loop and open-loop control scenarios in manufacturing processes; $\mathbf{r}(t) \in \mathbb{R}^{1\times M}$ is a reference trajectory for the measurable outputs to track. The objective function L can be defined as the integral of $\ell(\cdot)$ as the system output of interest like maximizing the material removal rate.

Thus, the research problem addressed in this study is: How can the analytical and nonlinear $\dot{\mathbf{x}} = \mathbf{f}\big(\mathbf{x}(t), \mathbf{u}(t)\big)$ be identified from noisy measurements and then design the optimal outer-loop controller $\mathbf{u}^*(\cdot)$ to improve the process parameters selection in manufacturing processes? The proposed method, case study, and research findings are presented in the following sections.

3. Proposed Learning based Control Method

This section presents the proposed IMLPC method, which consists of two components: interpretable machine learning for model discovery of dynamic system with control, and predictive control by acquired ordinary differential equations (ODEs). MPC is chosen as the control framework since it can be naturally applicable to nonlinear system models and integrate other physical constraints of manufacturing processes.

For the simplicity of notation, we use $\mathbf{x}$ and $\mathbf{u}$ for $\mathbf{x}(t)$ and $\mathbf{u}(t)$ in the following. Figure 1 shows the schematic illustration of IMLPC for data-driven modeling-based predictive control of the studied dynamic systems. The time series data $\mathbf{x}$ and $\mathbf{u}$ is fully utilized by the interpretable machine learning algorithm to obtain the model $\hat{\mathbf{f}}(\mathbf{x}, \mathbf{u})$. Then, the best control decision $\mathbf{u}^*_{p+1}$ in the next step will be acquired via model predictive control using the acquired model.

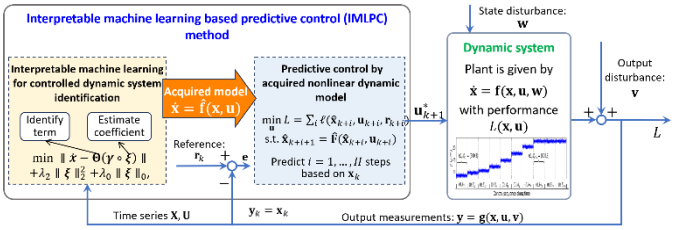


Figure 1. Schematic illustration of the proposed interpretable machine learning based predictive control (IMLPC) method.

3.1 Interpretable machine learning for model discovery of dynamic system with control

Here we extend the discrete optimization-based interpretable machine learning algorithm [32] to include both the state and control variables for robust system identification. The main idea behind system identification of unknown nonlinear dynamics $\mathbf{f}(\cdot)$ using discrete optimization is to identify the functional terms and the associated coefficients in a separate manner. The proposed algorithm learns the unknown dynamic system model from time series measurements, grounded in the fact that the right-hand-side of the governing equations of most dynamic systems include relatively few terms. Thus, discrete optimization-based interpretable machine learning can be effective in explicitly controlling the sparsity of terms in $\mathbf{f}(\mathbf{x},\mathbf{u})$, resulting in parsimonious models.

To do so, a dictionary of basis functions $\boldsymbol{\theta}(\mathbf{x},\mathbf{u})$ consisting of P candidate terms for $\mathbf{f}(\mathbf{x},\mathbf{u})$ is first constructed by basis expansion of $\mathbf{x}$ and $\mathbf{u}$, denoted by

$$\boldsymbol{\theta}(\mathbf{x},\mathbf{u}) = [\theta_1(\mathbf{x},\mathbf{u})\ \ \theta_2(\mathbf{x},\mathbf{u}) \cdots\ \theta_P(\mathbf{x},\mathbf{u})], \tag{3}$$

where $\theta_1(\mathbf{x},\mathbf{u}), \cdots, \theta_P(\mathbf{x},\mathbf{u})$ denote the linear or nonlinear operators of elements in $\mathbf{x}$ and/or $\mathbf{u}$. For brevity, we denote θ_p as the p-th term of $\boldsymbol{\theta}(\mathbf{x},\mathbf{u}) \in \mathbb{R}^{1\times P}$ in the following. For example, $\boldsymbol{\theta}(\mathbf{x},\mathbf{u})$ can be constructed to include the constant, polynomial, trigonometric terms, etc., of $\mathbf{x}$ and/or $\mathbf{u}$, that is

$$\boldsymbol{\theta}(\mathbf{x},\mathbf{u}) = [1\ \ \mathbf{x}\ \ \mathbf{u}\ \ (\mathbf{x}\otimes\mathbf{u})\ \ (\mathbf{x}\otimes\mathbf{x})\ \cdots \sin(\mathbf{x})\ \sin(\mathbf{u})\cdots], \tag{4}$$

where $\mathbf{x}\otimes\mathbf{u}$ denotes the vector of the element-wise products of $\mathbf{x}$ and $\mathbf{u}$. It is remarked that the construction of $\boldsymbol{\theta}(\mathbf{x},\mathbf{u})$ should incorporate physical knowledge of the studied system to ensure that all terms of $\mathbf{f}(\mathbf{x},\mathbf{u})$ are included in $\boldsymbol{\theta}(\mathbf{x},\mathbf{u})$. During this procedure, human interaction is critical in determining the initial pool of candidate terms in $\mathbf{f}(\mathbf{x},\mathbf{u})$. The initial pool of candidate terms usually consists of polynomials. By considering the underlying physical knowledge of the studied systems, the addition or removal of terms in $\boldsymbol{\theta}(\mathbf{x},\mathbf{u})$ is performed. Thus, the design of features is an iterative process with interactions between machine learning and humans. We assume that the unknown functions of $\mathbf{f}(\mathbf{x},\mathbf{u})$ live in the functional space spanned by of $\boldsymbol{\theta}(\mathbf{x},\mathbf{u})$, that is

$$\mathbf{f}(\mathbf{x},\mathbf{u}) = \boldsymbol{\theta}(\mathbf{x},\mathbf{u})\cdot\boldsymbol{\Xi}, \tag{5}$$

where $\boldsymbol{\Xi} = [\boldsymbol{\xi}_1\ \ \boldsymbol{\xi}_2\ \cdots\ \boldsymbol{\xi}_J] \in \mathbb{R}^{P\times J}$ is the matrix of coefficients, and each column $\boldsymbol{\xi}_j \in \mathbb{R}^{P\times 1}$ refers to those for function $f_j(\mathbf{x},\mathbf{u})$, $j \in \mathcal{J}$, where throughout this paper, we define set $\mathcal{J} = \{1,2,\cdots,J\}$.

To discover the unknown $\mathbf{f}(\mathbf{x},\mathbf{u})$ is to obtain the coefficients $\boldsymbol{\Xi}$. Note that the identification of the governing equation is independent for each $x_j(t)$. We denote the j-th column of $\dot{\mathbf{X}}$ as $\dot{\boldsymbol{x}}_j$. Instead of controlling sparsity through regularization on $\boldsymbol{\xi}_j$ for $\forall j \in \mathcal{J}$, we aim to directly control the number of active terms in $\boldsymbol{\xi}_j$, denoted by κ_j and $\kappa_j \in \mathcal{P} := \{1,\dots,P\}$, then identify the optimal combination of terms by discrete optimization. To begin with, we define an indicator matrix $\boldsymbol{\Gamma} \in \mathbb{B}^{P\times 1}$ to denote the existence of θ_p in $f_j(\mathbf{x},\mathbf{u})$ for $j \in \mathcal{J}$:

$$\boldsymbol{\Gamma} := [\boldsymbol{\gamma}_1\ \ \boldsymbol{\gamma}_2\ \cdots\ \ \boldsymbol{\gamma}_J], \tag{6}$$

where $\boldsymbol{\gamma}_j = (\gamma_{1j}\ \cdots \gamma_{Pj})^T \in \mathbb{B}^{P\times J}$ and $\mathbb{B}$ is the Boolean domain $\mathbb{B} = \{0,1\}$, that is, $\gamma_{pj} = 1$ if $f_j(\mathbf{x},\mathbf{u})$ includes θ_p, otherwise $\gamma_{pj} = 0$. Thus, for $j \in \mathcal{J}$, we formulate the problem as:

$$\min_{\langle\boldsymbol{\xi}_j,\boldsymbol{\gamma}_j\rangle} \| \dot{\boldsymbol{x}}_j - \boldsymbol{\Theta}(\mathbf{X},\mathbf{U})(\boldsymbol{\gamma}_j \circ \boldsymbol{\xi}_j) \|_2^2 + \lambda_2 \| \boldsymbol{\xi}_j \|_2^2 + \lambda_0 \| \boldsymbol{\xi}_j \|_0, \tag{7}$$

where $\boldsymbol{\Theta}(\mathbf{X},\mathbf{U}) \in \mathbb{R}^{N\times P}$ is the augmented matrix obtained by evaluating $\mathbf{X}$ and $\mathbf{U}$ using $\boldsymbol{\theta}(\mathbf{x},\mathbf{u})$, $\boldsymbol{\gamma}_j \circ \boldsymbol{\xi}_j$ is the element-wise product (Hadamard product) of $\boldsymbol{\gamma}_j$ and $\boldsymbol{\xi}_j$, $\| \boldsymbol{\xi}_j \|_0$ is the ℓ_0-norm (pseudo norm) that counts the number of nonzero entries of $\boldsymbol{\xi}_j$, $\| \boldsymbol{\xi}_j \|_2^2$ is the ℓ_2-norm for reducing the effects from noisy data, and λ_0 and λ_2 are the associated regularization strengths.

The goal of system identification for the dynamic system in Eq. (7) is to correctly identify $\boldsymbol{\gamma}_j$ and calculate the coefficients $\boldsymbol{\xi}_j$ from measurement data $\mathbf{X}$ and/or $\dot{\mathbf{X}}$. Existing works on sparse identification of dynamic systems, e.g., [36-39], perform term selection and promote sparsity by imposing penalties on the coefficients. Stated differently, these methods try to identify the product $\boldsymbol{\gamma}_j \circ \boldsymbol{\xi}_j$ as a whole. Notwithstanding the success of these methods, they are usually very sensitive to noisy measurements [32]. Instead, we identify $\boldsymbol{\gamma}_j$ and $\boldsymbol{\xi}_j$ separately. Eq. (7) is reformulated as the following discrete optimization problem to obtain $\boldsymbol{\gamma}_j$ and $\boldsymbol{\xi}_j$ for $f_j(\mathbf{x},\mathbf{u})$:

$$\langle \boldsymbol{\xi}_j^*, \boldsymbol{\gamma}_j^* \rangle = \operatorname{argmin}_{\langle\boldsymbol{\xi}_j,\boldsymbol{\gamma}_j\rangle\in\Delta} \| \dot{\boldsymbol{x}}_j - \boldsymbol{\Theta}(\mathbf{X},\mathbf{U})(\boldsymbol{\gamma}_j \circ \boldsymbol{\xi}_j) \|_2^2 + \lambda_2 \| \boldsymbol{\xi}_j \|_2^2 \tag{8a}$$

and the solution space Δ is defined as

$$\Delta := \{\langle \boldsymbol{\xi}_j, \boldsymbol{\gamma}_j \rangle \,|\, -M\boldsymbol{\gamma}_j \le \boldsymbol{\xi}_j \le M\boldsymbol{\gamma}_j, \ \boldsymbol{\gamma}_j^T\boldsymbol{e} = \kappa_j, \boldsymbol{\xi}_j \in \mathbb{R}^{P\times 1}, \boldsymbol{\gamma}_j \in \mathbb{B}^{1\times P}\}, \tag{8b}$$

where M is the upper bound of $\boldsymbol{\xi}_j$ and can be initially set as $M = \| \boldsymbol{\xi}_j \|_\infty$. Constraints $-M\boldsymbol{\gamma}_j \le \boldsymbol{\xi}_j \le M\boldsymbol{\gamma}_j$ indicate the element-wise relationship between $\boldsymbol{\xi}_j$ and $\boldsymbol{\gamma}_j$. That is, if $\gamma_{pj} = 0$, then $\xi_{pj} = 0$; otherwise ξ_{pj} is optimized within the interval $[-M, M]$. $\boldsymbol{e} \in \mathbb{R}^{P\times 1}$ is a vector with all entries set to 1, and κ_j refers to the number of non-zero terms in $f_j(\mathbf{x},\mathbf{u})$. As such, through the determination of the value of k_j, one can explicitly control the sparsity of the governing equations for a given dynamic system.

Following the two-stage optimization algorithm proposed in [32], we solve the problem in Eqs. (8) by first identifying the term for $\boldsymbol{\gamma}_j$ and then estimating the coefficient for $\boldsymbol{\xi}_j$. Once $\widehat{\boldsymbol{\gamma}}_j^*$ and $\widehat{\boldsymbol{\xi}}_j^*$ are determined for $j \in \mathcal{J}$, the acquired model is written as

$$\dot{\mathbf{x}} = \hat{\mathbf{f}}^*(\mathbf{x}, \mathbf{u}) = \boldsymbol{\theta}(\mathbf{x}, \mathbf{u})\left(\hat{\boldsymbol{\Gamma}}^* \circ \hat{\boldsymbol{\Xi}}^*\right). \quad (9)$$

Advanced parameter-tuning techniques for determining κ_j and λ_2 can also be integrated accordingly [32].

Remark 1. Extending the discrete optimization-based machine learning capability in [32] to nonlinear system identification of the controlled dynamic systems for manufacturing processes is straightforward. Including the control variables will result in larger problem size that increases the difficulty for the discrete optimization solver to obtain the optimal solution. However, due to the physical nature of manufacturing processes, the total number of process control and state variables recorded by machine internal controllers and external sensors is less than ten. Additionally, the polynomial orders are commonly less than two or three. Thus, a discrete optimization solver can quickly solve the resulting sparse linear regression problems to optimality. It should be noted that human knowledge for the nonlinear dynamics in manufacturing processes is important to acquire better nonlinear dynamic models. For example, machining dynamics are governed by the underlying second order and time-delay ordinary differential equations, where the inclusion of the time-delay terms as basis functions is crucial.

Remark 2. Discrete optimization can solve the resulting NP-hard ℓ_0 -norm linear regression problem to optimality for relatively large problem size. The heuristic sequential threshold least squares algorithm used by SINDY has been proved to approximate local minimizers of the ℓ_0-norm linear regression problem under noise-free condition [40]. For pure sparse linear regression without the context of dynamic systems, only a few theoretical guarantees like the restricted isometry property (RIP) in [41] have been provided for the sparse exact recovery under the strong independent and identically distributed (IID) assumption. However, the IID assumption cannot hold for dynamic systems with control data. The consistency result for the sparse linear regression under the context of dynamic systems remains an open question. Theoretical results on the linear in parameter approximator SINDY to approximate underlying dynamic systems have been provided, for more details please refer to [28].

3.2 Predictive control with acquired ordinary differential equations

Once the nonlinear ODEs are acquired, they are integrated into the MPC framework for outer-loop controller design to optimize the manufacturing processes. MPC employs a basic concept of iteratively updating model states with instantly measured system responses then solving the resulting optimal control problem over a receding horizon. The discrete-time dynamic system form $\mathbf{x}_{k+1} = \hat{\mathbf{F}}^*(\mathbf{x}_k, \mathbf{u}_k)$ is used. At time step k, current measurement $\mathbf{x}_k$ is obtained. Then, a control sequence for the prediction horizon of H time steps, $\mathbf{u}(\cdot\,|\mathbf{x}_k) = \{\mathbf{u}_{k+1}, \mathbf{u}_{k+2}, \cdots, \mathbf{u}_{k+H}\}$, is calculated by solving an open-loop optimal control problem considering future H steps using the acquired model. Next, the first value $\mathbf{u}_{k+1}$ of the computed control sequence is applied to the system as the system advances one step. An updated system state measurement $\mathbf{x}_{k+1}$ is then collected and used to reinitiate the model state.

For time step k where $k \in \mathcal{K} \coloneqq \{1, \dots, K\}$ and K is the maximum number of steps during MPC, we solve the below optimal control problem using a discrete-time dynamic system form to determine $\mathbf{u}_{k+1}$:

$$\begin{aligned}
&\min_{\mathbf{u}\in\boldsymbol{u}} \; L \coloneqq \sum\nolimits_{i=1}^{H} \ell(\hat{\mathbf{x}}_{k+i}, \mathbf{u}_{k+i}, \mathbf{r}_{k+i}) && (10a)\\
&s.t.\; \hat{\mathbf{x}}_{k+i+1} = \hat{\mathbf{F}}^*(\hat{\mathbf{x}}_{k+i}, \mathbf{u}_{k+i}), && \forall i = 0, \cdots, H-1, \; (10b)\\
&\quad \mathbf{h}_{\mathbf{x}}(\hat{\mathbf{x}}_{k+i}) \le \mathbf{0}, && \forall i = 0, \cdots, H-1, \; (10c)\\
&\quad \mathbf{h}_{\mathbf{u}}(\mathbf{u}_{k+i}) \le \mathbf{0}, && \forall i = 0, \cdots, H-1, \; (10d)\\
&\quad \mathbf{x}_k \in \mathcal{X}_{init}, && (10e)
\end{aligned}$$

where $\hat{\mathbf{x}}_{k+i}$ is the model prediction of the state, $\mathbf{x}_k$ is the given initial value for step k and $\hat{\mathbf{x}}_k = \mathbf{x}_k$. The objective of this formulation is to minimize the summation of cost $\ell(\hat{\mathbf{x}}_{k+i}, \mathbf{u}_{k+i}, \mathbf{r}_{k+i})$, at specific time epoch i over $i = 1, \cdots, H$. The dynamic system model in constraints Eq. (10b) represents the discrete-time or discretized version of the acquired model in Eq. (9). Inequalities (10c) and (10d) represent the constraints for the system variables and control variables, respectively. These constraints are particularly important for practical engineering systems where the expensive control expenditure is usually avoided. Specific system requirements, such as stability, accuracy, and robustness can also be integrated into these hard constraints. Constraints (10e) denote the initial conditions at step k for $\mathbf{x}_k$ belong to the feasible initial states $\mathcal{X}_{init}$.

Formulation in Eqs. (10) provides a general structure for MPC. The many possible designs of the objectives, controllers, and optimizers mean a broad spectrum of solution routines exist for solving problem in Eqs. (10). For example, one commonly used objective is of the form

$$L \coloneqq \sum_{i=1}^{H} \| \hat{\mathbf{x}}_{k+i} - \mathbf{r}_{k+i} \|_{\mathbf{Q}}^2 + \| \mathbf{u}_{k+i} \|_{\mathbf{R}}^2, \quad (11)$$

where $\hat{\mathbf{x}}_{k+i} - \mathbf{r}_{k+i}$ relates to the deviation of the state $\mathbf{x}_{k+i}$ to the commanded reference $\mathbf{r}_{k+i}$. Here we define $\| \mathbf{a} \|_{\mathbf{Q}}^2 = \mathbf{a}^T\mathbf{Q}\mathbf{a}$ as the weighted norm of vector $\mathbf{a} \in \mathbb{R}^{J\times 1}$, where $\mathbf{Q} \in \mathbb{R}^{J\times J}$ is a positive semi-definite weighting matrix for the state variables. In addition, the practical constraints of $\mathbf{u}_{k+i}$ may lead to the expenditure of expensive control inputs. Thus, to manage expenditure it is common practice to include a regularization term for the control inputs $\mathbf{u}_{k+i}$ in the second term, where $\mathbf{R} \in \mathbb{R}^{S\times S}$ is a positive semi-definite weighting matrix for the control inputs. Moreover, the inclusion of the regularization term may also be motivated by smoothness requirements for the control inputs.

Consequently, the outer-loop controller design for manufacturing process optimization depends on the acquired nonlinear dynamic models and the related objective function setting for system output. For synthetic example of chaotic Lorenz 3 system with control, classical MPC is employed to test and validate the data-driven modeling-based closed loop control performance (see Section 4). For the turning process case study, the specialty of turning dynamics stability results in different time steps of system and control. Hence, a new stability-based MPC formulation is developed for the cutting process parameters selection with maximum material removal

rate (see Section 5).

3.3 Algorithm procedure and metrics

We summarize the algorithm procedure of IMLPC in Algorithm 1, as shown in Table 1. The dynamic system model $\dot{\mathbf{x}} = \hat{\mathbf{f}}^*(\mathbf{x}, \mathbf{u})$ is first acquired via the discrete optimization-based interpretable machine learning algorithm, and model predictive control is applied to obtain the control sequence $\{\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_K\}$.

Table 1. Algorithm procedure of the proposed IMLPC method.

Algorithm 1. IMLPC method.

Input: Time series measurements $\mathbf{X}$, $\mathbf{U}$ and the computed or measured $\dot{\mathbf{X}}$;
Output: Set of equations $\dot{\mathbf{x}} = \hat{\mathbf{f}}^*(\mathbf{x}, \mathbf{u})$ and control sequence $[\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_K]$;
Function **IMLPC**$(\mathbf{X}, \mathbf{U}, \dot{\mathbf{X}})$
1. $\dot{\mathbf{x}} = \hat{\mathbf{f}}^*(\mathbf{x}, \mathbf{u}) \leftarrow$ Interpretable Machine Learning;
2. $[\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_K] \leftarrow$ Predictive Control with Acquired Model;
3. Return: Set of equations $\dot{\mathbf{x}} = \hat{\mathbf{f}}^*(\mathbf{x}, \mathbf{u})$ and control sequence $[\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_K]$;

Function **Interpretable Machine Learning** $(\mathbf{X}, \mathbf{U}, \dot{\mathbf{X}})$
Initialize: Construct $\boldsymbol{\theta}(\mathbf{x}, \mathbf{u})$ and $\boldsymbol{\Theta}(\mathbf{X}, \mathbf{U})$; $\hat{\boldsymbol{\Gamma}}^* \leftarrow \mathbf{0}$, $\widehat{\boldsymbol{\Xi}}^* \leftarrow \mathbf{0}$;
Compute $\widetilde{\boldsymbol{\Theta}}(\mathbf{X}, \mathbf{U})$, $\widetilde{\dot{\mathbf{X}}} \leftarrow$ Normalize $\boldsymbol{\Theta}(\mathbf{X}, \mathbf{U})$ and $\dot{\mathbf{X}}$;
for $j \in \mathcal{J}$ do
$\widehat{\boldsymbol{\gamma}}_j^* \leftarrow$ Solve the problem in Eqs. (8) with $\widetilde{\boldsymbol{\Theta}}(\mathbf{X}, \mathbf{U})$ and $\widetilde{\dot{\mathbf{x}}}_j$;
$\hat{\boldsymbol{\xi}}_j^* \leftarrow$ Estimate coefficients with $\boldsymbol{\Theta}(\mathbf{X}, \mathbf{U})$ and $\widehat{\boldsymbol{\gamma}}_j^*$;
Return: acquired models $\dot{\mathbf{x}} = \hat{\mathbf{f}}^*(\mathbf{x}, \mathbf{u}) = \boldsymbol{\theta}(\mathbf{x}, \mathbf{u})\left(\hat{\boldsymbol{\Gamma}}^* \circ \widehat{\boldsymbol{\Xi}}^*\right)$.

Function **Predictive Control with Acquired Model** $(\dot{\mathbf{x}} = \hat{\mathbf{f}}^*(\mathbf{x}, \mathbf{u}))$
Initialize: Construct the optimal control problem in Eqs. (10) using $\dot{\mathbf{x}} = \hat{\mathbf{f}}^*(\mathbf{x}, \mathbf{u})$.
For $k \in \mathcal{K}$ do
$\{\mathbf{u}_{k+1}, \mathbf{u}_{k+2}, \cdots, \mathbf{u}_{k+H}\} \leftarrow$ Solve the problem in Eqs. (10);
Select $\mathbf{u}_{k+1}$ to apply and obtain new measurement $\mathbf{x}_{k+1}$;
Return: control sequence $\{\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_K\}$.

The algorithm implementation and experiment settings are presented as follows. All the experiments are deployed on a mobile workstation with Intel XI(R) W-10885M CPU @ 2.40GHz, 128 GB memory, and the 64-bit Windows 10 Pro operating system for workstations. Detailed settings are presented below.

Noisy data. We generate data with noise to validate the robustness of system identification for the proposed IMLPC method. For all simulations, the time series data matrices $\mathbf{X}$, $\mathbf{U}$ and/or $\dot{\mathbf{X}}$ are generated from the numerical solutions of the governing equations. For the scenario where both the state variables $\mathbf{x}$ and their derivatives $\dot{\mathbf{x}}$ can be measured, the Gaussian noise is added to $\dot{\mathbf{x}}$, that is,

$$\dot{\boldsymbol{x}}_j^{\text{Noise}} = \dot{\boldsymbol{x}}_j + r\sigma_{\dot{\boldsymbol{x}}_j}\boldsymbol{\epsilon}, \qquad \forall j \in \mathcal{J}, \tag{12}$$

where $\dot{\boldsymbol{x}}_j \in \mathbb{R}^{N\times 1}$ is the j-th column of measured derivative matrix $\dot{\mathbf{X}}$, $\sigma_{\dot{\boldsymbol{x}}_j}$ is the standard deviation of column $\dot{\boldsymbol{x}}_j$, r is the ratio of noise level with respect to $\sigma_{\dot{\boldsymbol{x}}_j}$, and $\boldsymbol{\epsilon} \in \mathbb{R}^{N\times 1}$ is a vector with each entry following the standard normal distribution $\mathcal{N}(0,1)$. With such noise settings, a larger r indicates larger noise in the data.

Performance metrics. We evaluate the performance of the proposed IMLPC from both the system identification and model predictive control aspects. The comparison methods include those previously studied [37], including SINDY-MPC and a nonlinear autoregressive network with exogenous inputs (NARX) integrated with MPC (NARX-MPC).

For system identification, we examine the capability of the methods under comparison for exact identification of the target dynamic system from noisy data. This is defined by the number of exactly acquired equations as below:

$$A(\boldsymbol{\Gamma}) \coloneqq \sum_{j=1}^{J} \mathbf{1}_{\widehat{\boldsymbol{\gamma}}_j^* = \boldsymbol{\gamma}_j^\dagger} = \begin{cases} 1, & \text{if } \widehat{\boldsymbol{\gamma}}_j^* = \boldsymbol{\gamma}_j^\dagger, \\ 0, & \text{if } \widehat{\boldsymbol{\gamma}}_j^* \neq \boldsymbol{\gamma}_j^\dagger, \end{cases} \tag{13}$$

where $\widehat{\boldsymbol{\gamma}}_j^*$ is identified by the tested method and $\boldsymbol{\gamma}_j^\dagger$ is the ground truth of the target dynamic system. The exact identification of the dynamic system occurs when $A(\boldsymbol{\Gamma}) = J$. Upon exact identification, we further evaluate the accuracy of the estimated coefficients. Note the above $A(\boldsymbol{\Gamma})$ can only be used for comparison between IMLPC and SINDY-MPC as both can identify the analytical form of the governing equation. Whereas the resultant trajectories are used to evaluate the comparison between IMLPC and NARX-MPC.

For model predictive control, the primary performance metric is the steady state error. This error measures the difference between the steady state response and the desired control reference. Numerically, we define the e_{ss} as the sum of squared errors at the steady state:

$$e_{ss} \coloneqq \frac{1}{N_f J} \sum_{k=N_0+1}^{N_0+N_f} \| \mathbf{x}_k - \mathbf{r}_k \|_2^2, \tag{14}$$

Where N_0 is a positive time point after which the dynamic system model is at its steady state, and N_f is the number of measurements used for calculation. A smaller e_{ss} indicates a smaller average distance of the steady state output from the desired value. Thus, the smaller the steady state error the better control performance of the method.

Algorithm implementation. In Algorithm 1, the mixed integer linear programming (MILP) solver with branch and cut algorithm in CPLEX 20.1, is used to solve the problem in Eqs. (10) with the Python interface docplex. Unless otherwise noted, the parameters for IMLPC in the system identification procedure are as follows: $M = 1000, \lambda_2 = \sigma_{\dot{\boldsymbol{x}}_j}/\sqrt{N}$, and κ_j is set as the number of terms of the ground truth. After the model is acquired, it is implemented in MATLAB for MPC to harness the strength of the MATLAB embedded nonlinear optimization solver fmincon. Sequential quadratic programming (SQP) is chosen as the optimizer for solving problem in Eqs. (10). For SQP, we construct the objective of problem in Eqs. (10) to be twice continuously differentiable. For the benchmarks, we adopt PySINDy from Python and the MATLAB function narxnet for SINDY-MPC and NARX-MPC, respectively.

4. Toy Example: Lorenz 3 System with Control

The chaotic Lorenz 3 system with control is described by the below state space model with three state variables:

$$\dot{x}_1 = \alpha(x_2 - x_1) + u, \quad (15a)$$
$$\dot{x}_2 = x_1(\rho - x_3) - x_2, \quad (15b)$$
$$\dot{x}_3 = x_1 x_2 - \beta x_3. \quad (15c)$$

The following system parameter settings are selected to ensure chaotic behavior: $\alpha = 10, \beta = 8/3$, and $\rho = 28$. The Schroeder Phased Harmonic Signal (sphs) control signal is used for training data generation, where actuation is only added sto x_1. The dataset is generated using the initial condition $\mathbf{x}_0 = (-8, 8, 27)^{\mathrm{T}}$ and the time step $\Delta t = 0.001$ in $t \in [0,10]$ time units. As for the candidate terms construction, we set the maximum polynomial order to two and get a dictionary of 15 candidate terms in total. The hard threshold of coefficients in SINDY-MPC is set to 0.1. The model structure of NARX is set as [4, 10, 3] (four input nodes, one hidden layer with ten nodes, and three output nodes).

For MPC, both constant set point and trajectory control are conducted. For the constant set point regulation, the control goal is to stabilize the trajectory to one of the two fixed points, $\left(-\sqrt{72}, -\sqrt{72}, 27\right)^T$. The parameter settings are as follows: $H = 10$, $\mathbf{Q} = diag(1, 15, 15)$, $\mathbf{R} = 0.001$, and $[L_{\mathbf{u}}, U_{\mathbf{u}}] = [-50, 50]$. For the trajectory tracking control, the planned trajectory for x_1 is set as $\mathbf{r}(t) = 5\sin(5t)$, $H = 10$, $\mathbf{Q} = diag(1, 0, 0)$, $\mathbf{R} = 0.001$, and $[L_{\mathbf{u}}, U_{\mathbf{u}}] = [-400, 400]$. The Gaussian noise with noise ratio 50% is added to the derivatives for the system measurements before feedback to the MPC. It is noted here that different time steps are utilized for the control optimizer and the system evolution. In particular, the time step for the system evolution is $\Delta t_{sys} = 0.001$, and that for the MPC optimizer is $\Delta t_{MPC} = 0.01$. That is, at each control time step, the optimized control input will be applied as a constant control signal over a system evolution of 10 steps, with each system time step equal to 0.001.

Table 2. System identification performance by the number of exactly acquired equations $A(\boldsymbol{\Gamma})$ for Lorenz 3 system. The equations in the grey cells are used for control.

Noise ratio: r	SINDY-MPC	IMLPC
0%	3	3
10%	3	3
30%	2	3
50%	1	3
70%	1	3
100%	1	3
1000%	0	1
5000%	0	0

Table 2 shows the system identification results of IMLPC with SINDY-MPC methods. The IMLPC significantly outperforms the SINDY-MPC as evidenced by the largest noise ratio for accurate identification of IMLPC is 10 times larger than that of SINDY-MPC. For MPC analysis we choose the acquired models by SINDY-MPC and IMLPC at r = 100%. The results from MPC analysis for the models are presented in Table 3. It is observed that the identified coefficients of IMLPC deviate slightly from the ground truth, whereas the results from SINDY-MPC include many redundant terms.

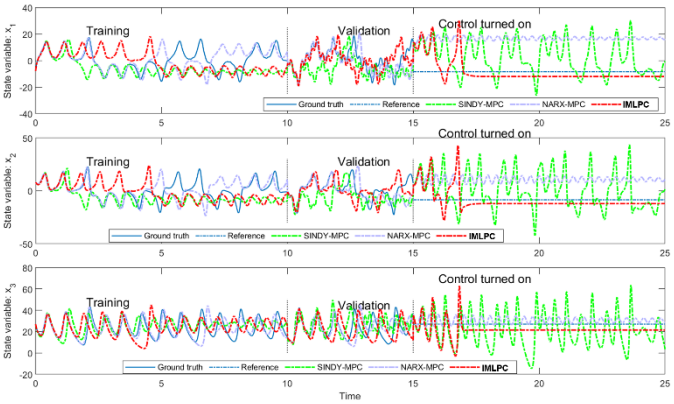


Figure 2. Simulation results comparison of constant set point control for Lorenz 3 system using the model acquired at noise ratio r = 100%. In the MPC stage, noise ratio r = 50% is used to inject noise to the measurements.

Table 3. Acquired equations under noise ratio r = 100% for the chaotic Lorenz 3 system

Ground truth	$\dot{x}_1 = 10x_2 - 10x_1 + u$ $\dot{x}_2 = 28x_1 - x_1x_3 - x_2$ $\dot{x}_3 = x_1x_2 - 8/3x_3$
SINDY-MPC	$\dot{x}_1 = 10.21x_2 - 10.25x_1 + 1.20u$ $\dot{x}_2 = 28.34x_1 - 1.61x_2 + 0.53u - 0.99x_1x_3$ $\dot{x}_3 = 1.25x_1 - 0.89x_2 - 2.8931x_3 + 0.52u$ $+1.05x_1x_2 - 0.23x_1u + 0.12x_2u$
IMLPC	$\dot{x}_1 = 10.21x_2 - 10.25x_1 + 1.20u$ $\dot{x}_2 = 28.89x_1 - 1.80x_2 - 1.00x_1x_3$ $\dot{x}_3 = 1.05x_1x_2 - 2.79x_3$

Figure 2 depicts the simulation results for constant set point control. For the training stage, we evolve the acquired models for 10 time units using the same controller as training data. A separate, new controller $u(t) = (5\sin^3(30t))$ is used to evolve another 5 time units for the model validation stage. Last, MPC is used to drive system evolution for another 10 time units. Results from all three methods fail to align closely with the ground truth trajectories for both training and validation stages, with an exception to the beginning of each simulation. The MPC stage demonstrates the strong performance of IMLPC for control. In stark contrast, IMLPC significantly outperforms others by tracking stably the planned trajectory under large uncertainties, while SINDY-MPC completely fails to track. As shown in Table 4, NARX-MPC has the largest steady state error followed by SINDY-MPC and the smallest error observed from IMLPC. Worthy of note, the steady state error of IMLPC is roughly 5% and 7% of that from NARX-MPC and SINDY-MPC, respectively.

Table 4. MPC performance comparison of constant set point control by steady state error e_{ss} for chaotic Lorenz 3 system.

Method	SINDY-MPC	NARX-MPC	IMLPC
e_{ss}	267.52	341.81	18.78

Table 5. MPC performance comparison of trajectory tracking control by steady state error e_{ss} for chaotic Lorenz 3 system

Method	SINDY-MPC	NARX-MPC	IMLPC
e_{ss}	23.32	15.68	14.33

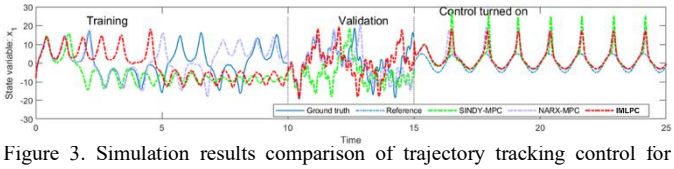


Figure 3. Simulation results comparison of trajectory tracking control for Lorenz 3 system using acquired model at noise ratio r=100%. In the MPC stage, noise ratio r=50% is used to inject noise to the measurements.

Figure 3 displays the trajectory tracking control results for Lorenz 3 system and Table 5 shows the corresponding results for e_{ss}. Note here e_{ss} is calculated by only considering x_1 since x_1 is the only variable to be tracked in the objective. Best control performance is obtained by IMLPC (Reference is the blue dashed line in the figure). This is followed by NARX-MPC and then SINDY-MPC, as shown in Table 5. Notably, the steady state error of IMLPC is recorded to be roughly 61% of error for SINDY-MPC.

To summarize, the proposed IMLPC exhibits 10 times better anti-noise capability than SINDY-MPC for accurate modeling. For control, IMLPC performs roughly 20 and 2 times better than SINDY-MPC for constant set point and trajectory tracking control, respectively. Altogether, IMLPC demonstrates greater performance for both data-driven modeling and control under high-noise conditions.

5. Case Study: Turning Process

In this section, an experimentally validated time domain simulator for turning process is used as a case study for the proposed IMLPC method. Machining dynamics stand for a typical class of dynamic systems that the proposed IMLPC method can be well suited. Evidence by the results in this section presents the first attempt of model-based control method for process parameters selection in machine tool research.

5.1 Experimentally validated time domain simulator of turning dynamics

In machine tool research, the equation of motion for the dynamics of a turning process can be modeled as

$$m\ddot{y} + c\dot{y} + ky = F_n = K_s b\big(h_m + y(t-\tau) - y(t)\big)\cos\beta\,, \quad (16)$$

where $y, \dot{y}$, and $\ddot{y}$ are the displacement, velocity, and acceleration, of the vibration caused by cutting force F_n (N), as shown in Figure 4. The mass, linear spring constant and viscous damping coefficient are denoted as m (kg), k (N/m) and c (N s/m). The left-hand-side of Eq. (16) describes the system dynamics while the right-hand-side shows a force model. The mean chip thickness h_m and time delay term $y(t-\tau) - y(t)$ indicate the varying instantaneous chip thickness and determine the instantaneous normal cutting force F_n with angle β (degree). $\tau = 1/\Omega$ is the time per rotation, $t - \tau$ indicates the time delay term, and K_s is a process dependent coefficient (N/mm2) depending on the workpiece materials and tool geometry. The two process parameters considered in this paper are chip width b (mm) and spindle rate Ω (rps).

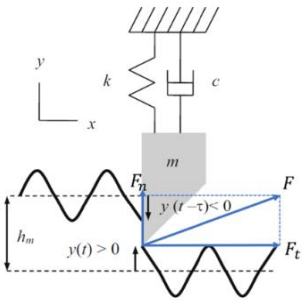


Figure 4. Schematic illustration of chatter vibration in turning process incurred by varying chip thickness and force.

The training data is generated using an experimentally validated time domain simulator. This simulator was developed by [42] based on the "regenerative force, dynamic deflection model" using forward Euler integration to solve Eq. (16). For doing this, the canonical form equation of motion in Eq. (16) is equivalently reformulated as its state-space form in Eq. (17).

$$\dot{y} = v_y, \quad (17a)$$

$$\dot{v}_y = \frac{K_s b\cos\beta}{m}\big(h_m + y(t-\tau) - y(t)\big) - \frac{c}{m}v_y - \frac{k}{m}y. \quad (17b)$$

The state variables are the displacement y and the velocity v_y, and forward Euler method is used for solving Eq. (17) in its discrete form directly. Note Eq. (17) is also the governing equations IMLPC needs to discover. Decades of research studies have validated the simulator's high accuracy and good agreement with the physical turning process for both turning and milling by including process damping, edge force, runout, and nonlinearities due to of loss of contact between the tool and the workpiece [42-45]. Thus, the time domain simulation of Eq. (16) is a high-fidelity simulator which can be regarded as a surrogate for experiments. Accordingly, a set of $[\Omega\ b]$ combinations are sampled for both stable and unstable high-speed cutting operations, where $\Omega \in \{400,600,800\}$ rps and $b \in \{2,4,6,8,10\}$ mm. In the simulation, variables $y, \dot{y}, \ddot{y}$ and F_n can all be obtained and contaminated with Gaussian noise. In practice, $y, \dot{y}, \ddot{y}$ and F_n can be measured using advanced metrology. For example, a capacitance probe, a laser vibrometer, a low mass accelerometer and a dynamometer can be used to measure $y, \dot{y}, \ddot{y}$ and F_n, respectively. For each combination of $\langle\Omega, b\rangle$, the first 2000 data points, which contain rich transient nonlinear dynamics, are used for training. The parameters used for simulation are as follows: m = 3.17 kg, k = 2×10E7 N/m, c = 795.77 N/m2, K_s = 1500 N/mm2, h_m = 0.1 mm, β = 68 deg and n_{rev} = 100 for number of revolutions. Accordingly, the natural frequency f_n = 400 Hz, the damping ratio ζ = 0.05 and the time step Δt = 1.25 × 10E−5 s.

5.2 Method validation using turning simulation

To apply the proposed IMLPC to turning process, machining-specific state variables, control variables, objective, and constraints need to be made. First, we treat the process parameters spindle speed and axial depth of cut as the set point of process control variables $\mathbf{u}$, that is, $\mathbf{u}_{set} = [\Omega\ b]$. As $\mathbf{u}_{set}$ are selected before process and usually cannot be changed during machining, the process parameters selection is an open-loop control problem for machine tools. The objective of

optimal control here is to select the process parameters $[\Omega^* \ b^*]$ to maximize the material removal rate (MRR) as defined in Eq. (18), subject to the acquired governing equations in the forms of Eq. (17).

$$L \coloneqq MRR = \Omega b h_m. \tag{18}$$

Besides, one critical consideration for process parameter selection is the machining stability, which is a hard constraint that must be satisfied for practical machine tool operations. Indeed, machining stability or chatter research is a widely studied topic with many significant achievements [46]. In this study, a common stability metric is used and calculated as follows. Since the cycle time per revolution is $1/\Omega_l$ (s), the time step for the dynamic system model evolution can be calculated by $\Delta t = 1/(\Omega_l N_\tau)$, where N_τ is the number of steps per revolution. As stability is determined by varying chip thickness between current and previous revolutions, we define the time delay error as

$$e_i^l = \hat{y}_{iN_\tau}^l - \hat{y}_{(i-1)N_\tau}^l, \tag{19}$$

where $l = k+1, \dots, k+H, i = N_0+1, \dots, N_0+N_f$, and the displacement $\hat{y}_{iN_\tau}^l$ is obtained by once-per-revolution sampling. That is, starting from revolution N_0+1 where N_0 is large enough to reach the steady state of the system evolution, N displacements $\hat{y}_{iN_\tau}^l$ are periodically sampled. Based on the time delay term, the first stability constraint is defined as in Eq. (20):

$$\frac{1}{N_f}\sum_{i=N_0+1}^{N_0+N_f}\left(e_i^l\right)^2 - M_0 \le 0. \tag{20}$$

The sum of squared errors between successively sampled points is averaged and compared with a defined positive threshold, M_0. For stable cuts, the sampled points are close to each other, so $M \le M_0$ with M ideally equalling zero. For unstable cuts, however, $M > M_0$.

Thus, the optimal control problem to determine the best process parameters in turning dynamics can be formulated as

$$\max_{\langle \boldsymbol{\Omega}, \boldsymbol{b} \rangle} \ L \coloneqq \sum_{l=k+1}^{k+H} MRR_l^2 = \sum_{l=k+1}^{k+H} h_m^2 \Omega_l^2 b_l^2 \tag{21a}$$

$$s.t. \ \hat{y}_{j+1}^l = \hat{y}_j^l + \frac{1}{\Omega_l N_\tau}\hat{z}_j^l, \qquad \forall l \in [H]_k, \tag{21b}$$

$$\hat{z}_{j+1}^l = \hat{z}_j^l + \frac{1}{\Omega_l N_\tau}\begin{pmatrix} -\frac{c}{m}\hat{z}_j^l - \frac{k}{m}\hat{y}_j^l + \\ \frac{K_s b_l}{m}\left(h_m + \hat{y}_{j-N_\tau}^l - \hat{y}_j^l\right) \end{pmatrix}, \quad \forall l \in [H]_k, \tag{21c}$$

$$\frac{1}{N_f}\sum_{i=N_0+1}^{N_0+N_f}\left(e_i^l\right)^2 - M_0 \le 0, \qquad \forall l \in [H]_k, \tag{21d}$$

$$\Omega_{l+1} - \Omega_l \in [L_{\Delta\Omega}, U_{\Delta\Omega}], \qquad \forall l \in [H]_k, \tag{21e}$$

$$b_{l+1} - b_l \in [L_{\Delta b}, U_{\Delta b}], \qquad \forall l \in [H]_k, \tag{21f}$$

$$\Omega_l \in [L_\Omega, U_\Omega], \qquad \forall l \in [H]_k, \tag{21g}$$

$$b_l \in [L_b, U_b], \qquad \forall l \in [H]_k, \tag{21h}$$

$$\langle y_0^l, z_0^l \rangle \in \mathcal{Y}_{init}^l, \qquad \forall l \in [H]_k, \tag{21j}$$

where $[H]_k = [k+1, \dots, k+H]$, $\hat{y}_j^l$ and $\hat{z}_j^l$ are the predictions for displacement and velocity, of the tool vibration at discretization step j under parameters $[\Omega_l \ b_l]$. Note here the objective (21a) is set as the sum of squares of the $MRR_l = h_m \Omega_l b_l$ for parameters $[\Omega_l \ b_l]$ and is equivalent to maximizing only MRR. This is for better utilizing the SQP algorithm which requires the objective to be twice continuously differentiable. Constraints (21b) and (21c) form the discrete state-space model for Eq. (17). In Eqs. (21e) - (21f), the increase rates of the process parameters between two consecutive cuts are restricted because severe adjustments of cutting conditions also need to be avoided to protect the machine tools.

Lastly, the experimental settings of IMLPC for turning process are as follows. For system identification, different dictionaries of physical terms are used for discovering Eq. (17a) and Eq. (17b). For discovering Eq. (17a), the second order polynomials of $[y \ v_y]$ is constructed to include 6 candidate terms (including constant). Based on these 6 terms, we include an additional time delay term $y(t) - y(t-\tau)$ for discovering Eq. (17b), where $\tau = 1/\Omega$ is determined based on physical knowledge. To identify b, we stack all datasets from the same Ω but different b as a single dataset for training. Other experimental settings remain the same as those in Section 3.3. For optimal control formulation in Eq. (21), $H = 1, N_0 = 200$, $N_f = 50$, and $N_\tau = 20$. The threshold for the stability metric is set as M_0 =1E-12, through some preliminary tests. The lower and upper bounds for the simulation are $[L_\Omega, U_\Omega] = [300,800]$ rps, $[L_b, U_b] = [2,8]$ mm, $[L_{\Delta\Omega}, U_{\Delta\Omega}] = [-50, 50]$ rps, $[L_{\Delta b}, U_{\Delta b}] = [-1, 1]$ mm. The initial condition $\langle y_0^{l+1}, z_0^{l+1} \rangle$ is set as the last values of displacement and velocity under $[\Omega_l^* \ b_l^*]$. Noise ratio of $r = 0.1$ is used for injecting noises into the MPC procedure. The stopping criteria include both the maximum number of iterations $K = 10$ for MPC, and the change ratio of the objective such that $\Delta L \le 0.01\%$, whichever is satisfied at first. The threshold of SINDY-MPC is tuned to 0.01 and 0.005 for the system dynamics and the force model, respectively. We do not report the results for NARX-MPC since the acquired model is a black-box model and cannot be used for subsequent stability analysis.

5.3 Results and Analysis

We first present the results for system identification under different noises by the number of correctly acquired equations, $A(\boldsymbol{\Gamma})$, followed by the frequency domain analysis of the acquired model. Then, we show the acquired model can be used for MPC to identify the best process parameters.

Table 6 shows the values of $A(\boldsymbol{\Gamma})$ for turning dynamics at 6 different Ω under noise levels ranging from 0.01% to 500%. The bold 2 indicates that the equation of motion in the state-space form of Eq. (17) is correctly acquired. IMLPC significantly outperforms SINDY-MPC, as evidenced by the failure of SINDY-MPC to identify the equations from data with any noise ratio greater than 1%, whereas IMLPC can identify the equations from data generated using a noise ratio up to 100%. Also, lower values of Ω lead to better performance. That is, we can utilize lower value of Ω to collect highly effective in-process data. This can be used to guide experimental data collection for practical implementation of IMLPC.

Table 6. Number of exactly acquired equations ($A(\boldsymbol{\Gamma})$) under different $[\Omega \ b]$ combinations.

Noise ratio: r	Process parameter Ω (rps)		
	400	600	800

	SINDY-MPC	IMLPC	SINDY-MPC	IMLPC	SINDY-MPC	IMLPC
0%	**2**	**2**	**2**	**2**	**2**	**2**
0.01%	**2**	**2**	0	**2**	0	**2**
0.1%	**2**	**2**	0	**2**	0	**2**
1%	**2**	**2**	0	**2**	0	**2**
10%	0	**2**	0	**2**	0	**2**
50%	0	**2**	0	0	0	0
100%	0	**2**	0	0	0	0
500%	0	0	0	0	0	0

Table 2. IMLPC acquired equation of motion under Ω = 400. The equation in the grey cell is chosen for the subsequent MPC.

Noise ratio: r	Acquired equations of motion
0%	$3.17\ddot{y} + 795.77\dot{y} + 20000000.00y = 0.06b + 561.91b\big(y(t) - y(t - 1/\Omega)\big)$
0.01%	$3.17\ddot{y} + 795.79\dot{y} + 19999998.22y = 0.06b + 561.91b\big(y(t) - y(t - 1/\Omega)\big)$
0.1%	$3.17\ddot{y} + 795.96\dot{y} + 19999981.68y = 0.06b + 561.88b\big(y(t) - y(t - 1/\Omega)\big)$
1%	$3.17\ddot{y} + 797.59\dot{y} + 19999768.94y = 0.06b + 561.58b\big(y(t) - y(t - 1/\Omega)\big)$
10%	$3.15\ddot{y} + 811.27\dot{y} + 19992911.91y = 0.06b + 552.06b\big(y(t) - y(t - 1/\Omega)\big)$
50%	$2.87\ddot{y} + 816.16\dot{y} + 19862105.73y = 0.06b + 412.92b\big(y(t) - y(t - 1/\Omega)\big)$
100%	$2.28\ddot{y} + 727.80\dot{y} + 19503105.65y = 0.06b + 226.55b\big(y(t) - y(t - 1/\Omega)\big)$

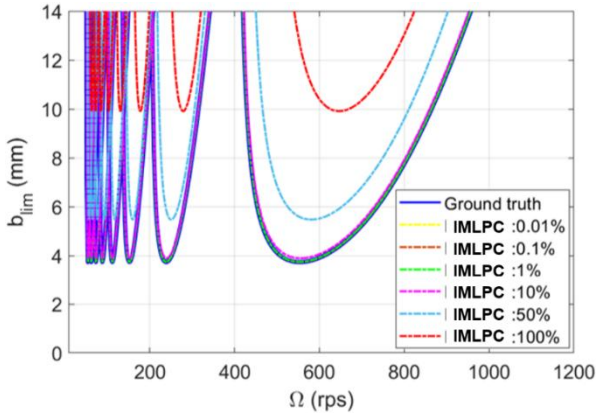


Figure 5. The stability lobe diagrams models in Table 7. The stability lobes under noises 0.01%, 0.1% and 1% are almost perfectly overlapped with those of noise-free 0.00% (ground truth).

Table 7 shows the exactly discovered equation of motion for parameters $\Omega = 400$ under different noisess (grey cells in Table 6). Note that in Table 7, the two discovered equations in the state-space form in Eq. (17) are combined into the canonical form as in Eq. (16) for presentation. We perform frequency analysis to obtain the stability lobe diagrams, see Figure 5. As the noise level increases the coefficients of the acquired equation of motion gradually deviate from the ground truth (0.00% noise). This deviation results in an increased damping estimate, and decreased mass and stiffness estimates. Accordingly, the resulting stability lobes in Figure 5 shift upward and to the right. Figure 5 is an illustration of the good interpretability of the generated IMLPC models in two ways. First, the coefficients of the terms $\ddot{y}$, $\dot{y}$, and y represent specific physical meanings for mass, linear spring dynamics and damping, each of which inherently describes the mechanical properties of the turning machine. Second, based on the linear form of the discovered system dynamics and time-delayed force model, frequency analysis can be conducted for machining stability. In other words, white-box models with good interpretability allow for the potential for subsequence analysis; however, black-box models are unable to provide similar levels of interpretation, thereby limiting follow-up analyses. Due to the inferior performance of SINDY-MPC for identifying interpretable models, we do not include those results in this section.

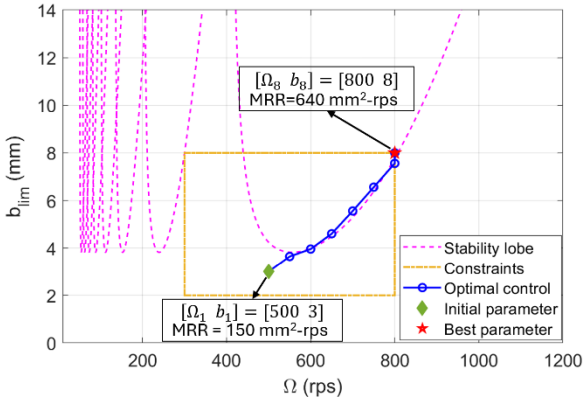


Figure 6. The MPC result for best process parameter selection of turning dynamics using model acquired at noise ratio r=10%. In the MPC stage, noise ratio r=10% is used to inject noise to the measurements.

Lastly, we report the MPC results from the model acquired on data with 10% noise (grey cell in Table 7) for MPC. Figure 6 shows the results of starting from an initial cut $= [\Omega_1 \;\; b_1] = [500 \;\; 3]$ (green diamond) and ending at the best cut $[\Omega^* \;\; b^*] = [800 \;\; 8]$ (red asterik). The optimal control formulation (24) is solved for ten iterations within 15 seconds, and the best cut is found in the 8-th iteration of MPC, see Figure 7. Note that the stability lobes (virtual magenta curves) are plotted as a reference of the stability boundary. They are not used during the solution procedure for the optimal control problem. Figure 7 depicts the steady state trajectory of displacement and velocity under the identified parameters during the MPC process. The small magnitude of the state variables over time validates the stability for the identified process parameters. The initialized process parameter with MRR of 150 mm^2-rps is improved to the best process parameter setting with MRR of 640 mm^2-rps. This change is evidence for the effectiveness and efficiency of the optimal control formulation in Eqs. (22) in identifying the best process parameters.

In summary, the anti-noise capability of IMLPC is at least 100 times better than SINDY-MPC for system identification of turning process using experimentally verified simulator data. The proposed IMLPC can be used for pre-process selection of

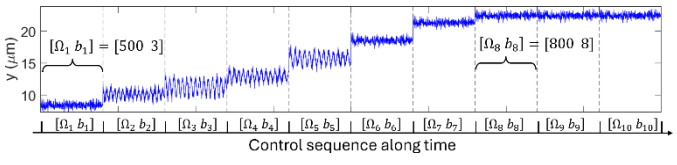


Figure 7. The steady-state trajectories of displacement during the MPC process. The best process parameters $[\Omega^* \; b^*] = [800 \; 8]$ is found in the 8-th iteration of MPC

stable process parameters with maximized material removal rate within 15 seconds. This shows that the IMLPC method has the potential to sbe used for both system identification and pre-process process parameter selection in real time of practical turning machine tool operations.

6. Conclusions

This paper presents an interpretable machine learning based predictive control method, denoted as IMLPC, for manufacturing processes. IMLPC can acquire ODEs-type models from highly noisy data and then integrate the acquired models into a model predictive control framework for manufacturing process optimization. IMLPC is developed by fully utilizing the dynamic data of process control and state variables that can be recorded by the machine internal controllers and instrumented external sensors. The adoption of interpretable machine learning capabilities for IMLPC yields powerful performance for model discovery from noisy data. This benefits the subsequent model predictive control procedure to exhibit superior control performance. Results on the synthetic example of chaotic Lorenz 3 system with control shows that IMLPC outperforms the state-of-the-art data-driven modeling-based control methods SINDY-MPC and NARX-MPC in both model accuracy and control performance under high noise conditions. Furthermore, a case study on turning process shows that IMLPC can robustly identify the exact second order, time-delay governing equations from experimentally validated time domain simulator with high noise injection. Additionally, stable cutting parameters as set points with a maximum material removal rate can be provided for turning machine tool operations. Future work will focus on developing the proposed method as an add-on digital tool for model-based process monitoring and control for turning processing and extending it for other manufacturing processes like milling and additive manufacturing.

Acknowledgements

The authors acknowledge support from the NSF Engineering Research Center for Hybrid Autonomous Manufacturing Moving from Evolution to Revolution (ERC-HAMMER) under Award Number EEC-2133630. This work was partially supported by the DOE Office of Energy Efficiency and Renewable Energy (EERE), under contract DE-AC05 00OR22725. This study was prepared under contract with the, Southeastern Advanced Machine Tools Network (SEAMTN), with financial support from the Office of Local Defense Community Cooperation, Department of Defense. The content reflects the SEAMTN and does not necessarily reflect the views of the Office of Local Defense Community Cooperation. The authors also gratefully acknowledge the AI Tennessee Initiative to partially support this research.

References

[1] Karandikar, J., K. Saleeby, T. Feldhausen, T. Kurfess, T. Schmitz and S. Smith, 2022. Evaluation of automated stability testing in machining through closed-loop control and Bayesian machine learning. *Mechanical Systems and Signal Processing*. 181: p. 109531.

[2] Schmitz, T.L. and R. Donalson, 2000. Predicting high-speed machining dynamics by substructure analysis. *CIRP Annals*. 49(1): p. 303-308.

[3] Shi, T., J. Wu, M. Ma, E. Charles and T. Schmitz, 2023. AFSD-Nets: A Physics-informed machine learning model for predicting the temperature evolution during additive friction stir deposition. *Available at SSRN 4627557*.

[4] Kincaid, J., R. Zameroski, E. Charles, T. No, J. Bohling, B. Compton and T. Schmitz, 2023. Hybrid manufacturing by additive friction stir deposition, metrology, CNC machining, and microstructure analysis. *Manufacturing Letters*. 35: p. 549-556.

[5] Wang, H., H. Baki and P. Kabore, 2001. Control of bounded dynamic stochastic distributions using square root models: an applicability study in papermaking systems. *Transactions of the Institute of Measurement and Control*. 23(1): p. 51-68.

[6] Li, M., P. Zhou, H. Wang and T. Chai, 2017. Nonlinear multiobjective MPC-based optimal operation of a high consistency refining system in papermaking. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. 50(3): p. 1208-1215.

[7] Rubeo, M., R. Copenhaver, S. Landge, T. Schmitz and N. Charlotte, 2016. Experimental platform for in-process metrology during orthogonal turning. in *American Society for Precision Engineering Annual Meeting, October*.

[8] No, T., M. Gomez and T. Schmitz, 2021. Contributions of scanning metrology uncertainty to milling force prediction. *Procedia Manufacturing*. 53: p. 213-222.

[9] Bertsekas, D., 2019. *Reinforcement learning and optimal control*. Vol. 1. Athena Scientific.

[10] Parunandi, K.S., A. Sharma, S. Chakravorty and D. Kalathil, 2020. D2C 2.0: Decoupled Data-Based Approach for Learning to Control Stochastic Nonlinear Systems via Model-Free ILQR. *arXiv preprint arXiv:2002.07368*.

[11] Fliess, M. and C. Join, 2013. Model-free control. *International journal of control*. 86(12): p. 2228-2252.

[12] Arrazola, P.J., T. Özel, D. Umbrello, M. Davies and I.S. Jawahir, 2013. Recent advances in modelling of metal machining processes. *CIRP Annals*. 62(2): p. 695-718.

[13] Noriega, J.R. and H. Wang, 1998. A direct adaptive neural-network control for unknown nonlinear systems and its application. *IEEE Transactions on Neural Networks*. 9(1): p. 27-34.

[14] Akaike, H., 1969. *Fitting autoregreesive models for prediction*, in *Selected Papers of Hirotugu Akaike*. Springer. p. 131-135.

[15] Ho, B. and R.E. Kálmán, 1966. Effective construction of linear state-variable models from input/output functions: Die Konstruktion von linearen Modeilen in der Darstellung durch Zustandsvariable aus den Beziehungen für Ein-und Ausgangsgrößen. *at-Automatisierungstechnik*. 14(1-12): p. 545-548.

[16] Juang, J.-N. and R.S. Pappa, 1986. Effects of noise on modal parameters identified by the eigensystem realization algorithm. *Journal of Guidance, Control, and Dynamics*. 9(3): p. 294-303.

[17] Akaike, H., 1974. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*. 19(6): p. 716-723.
[18] Budišić, M., R. Mohr and I. Mezić, 2012. Applied koopmanism. *Chaos: An Interdisciplinary Journal of Nonlinear Science*. 22(4).
[19] Mezić, I., 2005. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*. 41: p. 309-325.
[20] Rowley, C.W., I. Mezić, S. Bagheri, P. Schlatter and D.S. Henningson, 2009. Spectral analysis of nonlinear flows. *Journal of Fluid Mechanics*. 641: p. 115-127.
[21] Kocijan, J., R. Murray-Smith, C.E. Rasmussen and A. Girard, 2004. Gaussian process model based predictive control. in *Proceedings of the 2004 American Control Conference*.
[22] Asadi, F., A. Olleak, J. Yi and Y. Guo, 2021. Gaussian process (gp)-based learning control of selective laser melting process. in *Proceedings of the 2021 American Control Conference*.
[23] Han, F. and J. Yi, 2021. Stable learning-based tracking control of underactuated balance robots. *IEEE Robotics and Automation Letters*. 6(2): p. 1543-1550.
[24] Chen, K., J. Yi and D. Song, 2022. Gaussian-Process-Based Control of Underactuated Balance Robots With Guaranteed Performance. *IEEE Transactions on Robotics*. 39(1): p. 572-589.
[25] Yi, J., Y. Sheng and C.S. Xu, 2003. Neural network based uniformity profile control of linear chemical-mechanical planarization. *IEEE Transactions on Semiconductor Manufacturing*. 16(4): p. 609-620.
[26] Yin, Z., T. Liu, C. Wang, H. Wang and Z.-P. Jiang, 2023. Reducing urban traffic congestion using deep learning and model predictive control. *IEEE Transactions on Neural Networks and Learning Systems*. p. 4131-4145.
[27] Soudbakhsh, D., A.M. Annaswamy, Y. Wang, S.L. Brunton, J. Gaudio, H. Hussain, D. Vrabie, J. Drgona and D. Filev, 2023. Data-Driven Control: Theory and Applications. in *Proceedings of 2023 American Control Conference*.
[28] Sharma, A. and S. Chakravorty, 2023. On Data-Driven Surrogate Modeling for Nonlinear Optimal Control. *arXiv preprint arXiv:2310.13147*.
[29] Drgoňa, J., K. Kiš, A. Tuor, D. Vrabie and M. Klaučo, 2022. Differentiable predictive control: Deep learning alternative to explicit model predictive control for unknown nonlinear systems. *Journal of Process Control*. 116: p. 80-92.
[30] Bryson, A.E. and Y.-C. Ho, 1975. *Applied optimal control: optimization, estimation and control*. CRC Press.
[31] Garcia, C.E., D.M. Prett and M. Morari, 1989. Model predictive control: Theory and practice—A survey. *Automatica*. 25(3): p. 335-348.
[32] Shi, T., M. Ma, H. Tran and G. Zhang, 2022. Compressive-sensing-assisted mixed integer optimization for dynamical system discovery with highly noisy data. *Numerical Methods for Partial Differential Equations*. 41(1): p. e23164.
[33] Bertsimas, D. and W. Gurnee, 2023. Learning sparse nonlinear dynamics via mixed-integer optimization. *Nonlinear Dynamics*. 111(7): p. 6585-6604.
[34] Ren, A., M. Ma, J. Wu, J. Karandikar, C. Tyler, T. Shi and T. Schmitz, 2025. A Cutting Mechanics-based Machine Learning Modeling Method to Discover Governing Equations of Machining Dynamics. *Manufacturing Letters*. 44: p. 759-769.
[35] Shi, T., M. Ma, J. Wu, C. Post, E. Charles and T. Schmitz, 2024. AFSD-Physics: Exploring the governing equations of temperature evolution during additive friction stir deposition by a human-AI teaming approach. *Manufacturing Letters*. 41: p. 1004-1015.
[36] Brunton, S.L., J.L. Proctor and J.N. Kutz, 2016. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*. 113(15): p. 3932-3937.
[37] Kaiser, E., J.N. Kutz and S.L. Brunton, 2018. Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. *Proceedings of the Royal Society A*. 474(2219): p. 20180335.
[38] Mangan, N.M., T. Askham, S.L. Brunton, J.N. Kutz and J.L. Proctor, 2019. Model selection for hybrid dynamical systems via sparse regression. *Proceedings of the Royal Society A*. 475(2223): p. 20180534.
[39] Wang, W.-X., R. Yang, Y.-C. Lai, V. Kovanis and C. Grebogi, 2011. Predicting catastrophes in nonlinear dynamical systems by compressive sensing. *Physical Review Letters*. 106(15): p. 154101.
[40] Zhang, L. and H. Schaeffer, 2019. On the convergence of the SINDy algorithm. *Multiscale Modeling & Simulation*. 17(3): p. 948-972.
[41] Candes, E.J. and T. Tao, 2005. Decoding by linear programming. *IEEE transactions on information theory*. 51(12): p. 4203-4215.
[42] Smith, S. and J. Tlusty, 1991. An overview of modeling and simulation of the milling process.
[43] Tyler, C.T., J.R. Troutman and T.L. Schmitz, 2016. A coupled dynamics, multiple degree of freedom process damping model, Part 1: Turning. *Precision Engineering*. 46: p. 65-72.
[44] Salehi, M., T. Schmitz, R. Copenhaver, R. Haas and J. Ovtcharova, 2019. Probabilistic sequential prediction of cutting force using Kienzle model in orthogonal turning process. *Journal of Manufacturing Science and Engineering*. 141(1): p. 011009.
[45] Schmitz, T.L. and K.S. Smith, 2009. Machining dynamics. *Springer*. p. 303.
[46] Altintas, Y., G. Stepan, E. Budak, T. Schmitz and Z.M. Kilic, 2020. Chatter stability of machining operations. *Journal of Manufacturing Science and Engineering*. 142(11): p. 110801.