NAPI-MPC: Neural Accelerated Physics-Informed MPC for Nonlinear PDE Systems

Peilun Li PEILUN.LI@VANDERBILT.EDU
Kaiyuan Tan KAIYUAN.TAN@VANDERBILT.EDU
Thomas Beckers THOMAS.BECKERS@VANDERBILT.EDU
Department of Computer Science, Vanderbilt University, Nashville, TN 37212

Editors: N. Ozay, L. Balzano, D. Panagou, A. Abate

Abstract

The control of systems governed by *nonlinear* partial differential equations (PDEs) can present substantial challenges for model predictive control (MPC) approaches. Data-driven MPC has emerged as a solution for dealing with unknown nonlinear dynamics, but issues regarding safety guarantees, accuracy, and data efficiency remain a concern. We propose NAPI-MPC: a novel physics-informed scenario-based MPC approach for control of nonlinear PDE systems with partially unknown dynamics. Unlike other physics-informed learning methods that require extensive knowledge on the governing equations, our NAPI-MPC leverages distributed Port-Hamiltonian systems as a generalized, energy-based representation of the PDE dynamics, in which the Hamiltonian is modeled and learned by a Gaussian process. The Bayesian nature of the Gaussian process enables the drawing of scenario samples that are used in scenario-based predictive control to determine the optimal control action for the PDE system. To ensure applicability in time-sensitive contexts, we leverage a neural network as proxy for the MPC controller, trained offline on states and optimal control actions to enable fast inference for real-time operation.

Keywords: Physics-informed Learning, Gaussian Process, Model Predictive Control, Partial Differential Equations

1. Introduction

Nonlinear partial differential equation systems (PDEs) are common in various engineering fields, such as soft robotics (Gillespie et al., 2018) and underwater pipelines (Jovanović, 2021). Nonlinear behavior can arise from nonlinear material properties, such as interactions across multiple flow scales in turbulent fluid dynamics. Despite their widespread presence, controlling these systems remains challenging due to their inherent complexity (Krstic et al., 1995). Model Predictive Control (MPC) (Hirt et al., 2024) is an optimization-based control methodology well-suited for managing nonlinear ODE systems. MPC can directly incorporate both input and state constraints into the controller design, providing robust control under challenging conditions. Due to these advantages, MPC is widely adopted in industrial applications that require precise control and constraint handling, such as autonomous driving (Zanon et al., 2014), robot trajectory tracking (Carron et al., 2019), and medical therapy (Wolff et al., 2022). However, conventional MPC assumes a deterministic model, with known parameters, which is often unrealistic in practice. To address this, Scenario Model Predictive Control (SCMPC) (Calafiore and Fagiano, 2012, 2013) considers model uncertainties by optimizing across multiple candidate scenarios to ensure robust control. Furthermore, explicit quantification and propagation of uncertainty within the predictive model are critical for both safety and performance. By treating uncertainty as random variables, SCMPC can provide

probabilistic guarantees by avoiding the compromised performance inherent to approaches that assume the most extreme outcome. In particular, sampling from the disturbance distribution or an appropriately chosen surrogate enables the controller to anticipate likely realizations of model error, hence improve the performance. Moreover, by adjusting the number of scenarios, one can directly trade off between sensitivity to risk and aggressiveness of control, making SCMPC a flexible framework for systems where reliability under uncertainty is of primary importance. Despite this, the effectiveness of SCMPC still depends heavily on the accuracy of the model used to predict system dynamics. To overcome this limitation, various data-driven methods have been proposed for modeling partial differential equations. For instance, Stephany and Earls (2024); Bhan et al. (2024) employs data-driven method to learn PDEs under noisy conditions, while Liu et al. (2024) uses alternative activation functions to enhance the interpretability of the learned PDEs. Similarly, Kaptanoglu et al. (2022) utilizes different basis functions to approximate the target PDEs. However, such basis functions may not always be practically available. For example, researchers try to model the spatiotemporal growth of a tumor using PDEs, see Matzavinos et al. (2004); Rahman et al. (2017) for details. However, due to insufficient prior knowledge of the underlying physics, tissue properties, and non-smooth tumor behavior, they fail to align with typical smooth basis functions. The absence of inherent physics often compromises model performance under noisy measurements, and most importantly, these methods lack consideration of the uncertainties inherent in the learning process. Therefore, a robust, physics-constrained, function-free approach with uncertainty quantification is needed. By referring to function-free, we are addressing that only a basic physical structure is needed but not the exact governing equation with potentially unknown parameters.

Considering the problems mentioned above, several physics-informed MPCs have been proposed recently, e.g. (Nicodemus et al., 2022; Alhajeri et al., 2022). These controllers leverage Physics-Informed Neural Networks (PINNs) (Raissi et al., 2019), which integrate physics into the learning process through the loss function, ensuring that the learned models respect the governing physics. Despite advances, two critical issues remain. First, exact knowledge of the underlying physical system is not always available, which complicates the effect of PINNs. Secondly, even with physics-based priors, there remains a need to assess model accuracy comprehensively for dependable decision-making. This highlights the need for uncertainty quantification in MPC applications.

To address the challenge of limited knowledge about the underlying system dynamics, Port-Hamiltonian Systems (PHS) (Nageshrao et al., 2015; Duong and Atanasov, 2021) have gained increasing attention due to their capability to represent nonlinear systems based on energy. PHS provides a structured approach for modeling the complete system, abstracting from the detailed equations to focus on energy interactions and conservation principles. Recent advancements have leveraged this framework to encode energy-based representations for both ordinary differential equations (ODEs) and partial differential equations (PDEs) (Tan et al., 2024), enabling the extraction of latent energy representations for consistent system predictions without explicit knowledge of the governing equations. Despite these strides in structured machine learning, there are several data-driven methods for predictive control for ODEs (Beckers, 2023) and PDEs (Arbabi et al., 2018). However, safe and reliable control of nonlinear PDEs with unknown dynamics is still an open challenge.

Contribution: We propose a novel physics-constrained scenario-based model predictive control algorithm named *NAPI-MPC*: Neural Accelerated Physics-informed MPC for nonlinear PDEs. Our approach exploits a Gaussian Process to capture system uncertainty, which is then used to generate multiple scenarios for SCMPC to optimize reliable control by incorporating uncertainty quantification. NAPI-MPC addresses the challenges of predictive control in PDE systems with nonlinear

dynamics where the underlying function is challenging to obtain. Unlike existing purely data-driven methods, NAPI-MPC leverages physics-based constraints, enhancing its robustness under various scenarios using the provided uncertainty quantification. By integrating neural learning with physics-informed MPC, our method significantly accelerates the computation, making it ideal for time-sensitive applications. Ultimately, NAPI-MPC provides a robust control framework for nonlinear PDEs with partially unknown nonlinear dynamics, ensuring both reliability and efficiency.

2. Preliminaries and Problem Setting

In this section, we provide a concise review of Gaussian Process distributed Port-Hamiltonian systems in 2.1, scenario-based model predictive control in 2.2, and introduce the problem setting.

2.1. Gaussian Process distributed Port-Hamiltonian System (GP-dPHS)

In this section, we briefly review GP-dPHS, which was recently proposed as physics-informed Bayesian learning technique for PDE systems, see (Tan et al., 2024) for more details. The main idea is to formularize the nonlinear PDE as a distributed port-Hamiltonian system Σ[1]

$$\Sigma(J, R, \mathcal{H}, G_d) = \begin{cases} \frac{\partial}{\partial t}\boldsymbol{x}(t, \boldsymbol{z}) = (J - R)\delta_{\boldsymbol{x}}\mathcal{H} + G_d\boldsymbol{u} \\ \boldsymbol{y} = G_d^*\delta_{\boldsymbol{x}}\mathcal{H} \\ w = B_{\mathcal{Z}}(\delta_{\boldsymbol{x}}\mathcal{H}, \boldsymbol{u}), \end{cases} \tag{1}$$

where the spatial-temporal state is denoted as $\boldsymbol{x}(t, \boldsymbol{z}) \in \mathbb{R}^n$ for the corresponding time t and spatial point $\boldsymbol{z}$. The interconnection matrix J is a skew-adjoint constant differential operator, R denotes a constant differential operator defining the dissipation of energy and G_d is a constant differential operator for the input $\boldsymbol{u}$, and G_d^* denotes the adjoint operator of G_d, defined such that for appropriate test functions f and g, the identity $\langle G_d f, g\rangle = \langle f, G_d^* g\rangle$ holds under the inner product in the Hilbert space of square-integrable functions. The functional derivative operator $\delta_{\boldsymbol{x}}$ is comparable to the partial derivative operator, but acts on the Hamiltonian functional $\mathcal{H}$. Finally, the boundary conditions are defined by $w = B_{\mathcal{Z}}(\delta_{\boldsymbol{x}}\mathcal{H}, \boldsymbol{u})$, where $B_{\mathcal{Z}}$ is the boundary operator. The boundary operator B_Z extracts the boundary port variables from the Hamiltonian derivative and inputs, and maps from the space of functional derivatives to that of boundary evaluations at the endpoints of spatial domain. That is, B_Z enforces physically meaningful constraints, such as Dirichlet or Neumann-type boundary conditions, on the PDE system by evaluating traces of $\delta_x\mathcal{H}$ at the spatial boundaries ∂Z.

However, the Hamiltonian functional $\mathcal{H}$ is often challenging to obtain due to the unstructured nonlinearities in the system. Therefore, we proposed in (Tan et al., 2024) to use a Gaussian process (GP) to learn the structured representations given by dPHS (1). A GP is fully defined by mean function m_{GP} and the covariance function k_{GP}, expressed as $\mathcal{GP}(m_{GP}(\mathbf{x}), k_{GP}(\mathbf{x}, \mathbf{x}'))$. The kernel $k_{GP}(\mathbf{x}, \mathbf{x}')$ determines the covariance between any two inputs, making GPs popular for non-parametric Bayesian regression. Given the input dataset $X = [\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}]$ with outputs $Y = [y^{(1)}, \ldots, y^{(N)}]$ as training set $\mathcal{D}$ sampled from an unknown function $y = f(\mathbf{x})$, a GP prior on

1. Vectors $\boldsymbol{a}$ and vector-valued functions $\boldsymbol{f}(\cdot)$ are denoted with bold characters. Matrices are described with capital letters. I_n is the n-dimensional identity matrix and 0_n the zero matrix. The expression $A_{:,i}$ denotes the i-th column of A. For a positive semidefinite matrix Λ, $\|x - y\|_\Lambda^2 = (x - y)^\top \Lambda (x - y)$. $\mathbb{R}_{>0}$ denotes the set of positive real numbers, while $\mathbb{R}_{\geq 0}$ is the set of non-negative real numbers. $\mathcal{C}^i$ denotes the class of i-th times differentiable functions. The operator $\nabla_{\boldsymbol{x}}$ with $\boldsymbol{x} \in \mathbb{R}^n$ denotes $[\frac{\partial}{\partial x_1}, \ldots, \frac{\partial}{\partial x_n}]^\top$.

f allows using Bayesian theorem to infer the posterior for a new input $\mathbf{x}^*$, updating the mean and covariance of the GP as following equations

$$\begin{aligned}\mu(y^*|\mathbf{x}^*) &= k_{GP}(\mathbf{x}^*, X)K_{GP}(X, X)^{-1}Y \\ \Sigma(y^*|\mathbf{x}^*) &= k_{GP}(\mathbf{x}^*, \mathbf{x}^*) - k_{GP}(\mathbf{x}^*, X)K_{GP}(X, X)^{-1}k_{GP}(\mathbf{x}^*, X),\end{aligned} \tag{2}$$

where $k_{GP}(\mathbf{x}^*, X)$ represents the covariance between the new point $\mathbf{x}^*$ and the training set X, and $k_{GP}(\mathbf{x}^*, \mathbf{x}^*)$ is the prior variance at the new point. Hence, the desired mean value and uncertainty measurement can be acquired by the learned predictive distribution $p_{GP}(\mathbf{y}^*|\mathbf{x}^*, X, Y)$. See (Rasmussen and Williams, 2006) for more details on GPs.

After defining the nonlinear PDE in the dPHS structure, GP-dPHS aims to learn the unknown Hamiltonian with a non-parametric GP based on observations of the PDE system. In addition, this approach allows us to estimate unknown parameters of the matrices J, R, and G of the dPHS by treating them as hyperparameters. There are several benefits for such an approach: First, the GP-dPHSs do not require prior knowledge about the parametric structure of the Hamiltonian functional, making it suitable to learn complex nonlinear system dynamics. Furthermore, the GP as a Bayesian learning framework encapsulates all possible dPHS configurations under the Bayesian prior from a finite set of observations, resulting in predictions with confidence intervals. This is critical for model identification and is exploited to design a robust model-based controller for the PDE system.

2.2. Scenario-based Model Predictive Control (SCMPC)

In this section, we introduce discrete-time scenario-based MPC, which is also known as a receding-horizon optimization problem. Traditional MPC schemes rely on precise models and deterministic predictions, which may not be adequate in the presence of uncertainty. By contrast, scenario-based MPC addresses this limitation by explicitly considering uncertainty through multiple scenarios, enhancing robustness and performance under stochastic conditions.

Let us consider a discrete-time system whose dynamics is described by $\boldsymbol{x}(t_{i+1}) = \boldsymbol{f}\left(\boldsymbol{x}(t_i), \boldsymbol{u}(t_i)\right)$, where $\boldsymbol{f}$ represents a vector-valued function, $\boldsymbol{x}(t_i) \in \mathbb{R}^n$ is the state and $\boldsymbol{u}(t_i) \in \mathbb{R}^m$ denotes the control input at time $t_i \in \{t_0, \cdots, t_{N_H-1}\}$, where N_H is the total horizon for the control problem with equidistant time steps. A model of the dynamics $\hat{\boldsymbol{f}} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is a function that maps current state and control input to the next state $\boldsymbol{x}(t_{i+1})$.

For discrete-time scenario-based MPC, we are provided with $N_s \in \mathbb{R}$ number of scenarios of the dynamics $\{\hat{\boldsymbol{f}}^{(k)}(\cdot,\cdot)\}_{k=1}^{N_s}$, each of which is a possible candidate of the actual system over the prediction horizon N_p. The optimization problem to scenario-based MPC is formulated as

$$\begin{aligned}&\min_{\boldsymbol{u}_t^*} \mathcal{J}(\hat{\boldsymbol{x}}_t^{(k)}, \boldsymbol{u}_t^*) = \sum_{k=1}^{N_s} \mathcal{J}_k(\hat{\boldsymbol{x}}_t^{(k)}, \boldsymbol{u}_t^*) \quad \text{s.t. } \forall k \in \{1, \cdots, N_s\},\ \forall j \in \{0, \cdots, N_p - 1\} \\ &\hat{\boldsymbol{x}}_{t_{j+1}}^{(k)} = \hat{\boldsymbol{f}}^{(k)}(\hat{\boldsymbol{x}}_{t_j}^{(k)}, \boldsymbol{u}_{t_j}^*),\ \hat{\boldsymbol{x}}_{t_0}^{(k)} = \boldsymbol{x}(t_0),\ \hat{\boldsymbol{x}}_{t_j}^{(k)} \in \mathcal{X},\ \boldsymbol{u}_{t_j}^* \in \mathcal{U}\end{aligned} \tag{3}$$

where at time t, $\boldsymbol{u}_t^* = \boldsymbol{u}_{t_{0,\ldots,N_p-1}}^*$ is the optimal control input sequence, and $\hat{\boldsymbol{x}}_{t_j}^{(k)} \in \mathbb{R}^n$ denotes the estimated state from model $\hat{\boldsymbol{f}}^{(k)}$. We further constrain the estimated states to be in some set $\mathcal{X}$, and the control inputs in set $\mathcal{U}$. The loss function $\mathcal{J}(\cdot,\cdot)$ measures the mismatch between the desired and the current state, defining the desired control goal for the MPC setting. The solution to equation (3) is denoted by $\boldsymbol{u}_t^*$, and the first element $\boldsymbol{u}_{t_0}^*$ is applied to the controlled system. Hence, the MPC minimization problem is to be solved iteratively until reaching the end of the total horizon.

2.3. Problem setting

Given a PDE system in distributed Port-Hamiltonian, the evolution of its state $\boldsymbol{x}(t,z) \in \mathbb{R}^n$ over time $t \in \mathbb{R}_{\geq 0}$ and spatial domain $\boldsymbol{z} \in \mathcal{Z}$ should be defined by the equation (1) with initial state $\boldsymbol{x}(0,\boldsymbol{z})$. The Hamiltonian functional $\mathcal{H} \in \mathcal{C}^\infty$ is assumed to be *completely unknown* due to the complexity of the system and the unstructured uncertainties such as non-linear stress / strain curve in soft materials, physical coupling effects, or highly non-linear electrical and magnetic fields. We assume that we can measure the state $\boldsymbol{x}(t,z)$ over both temporal and spatial domain. We aim to find a robust control input $\boldsymbol{u}^*$ that achieves a feasible control goal while considering model uncertainties. To address the problem, we introduce the following assumptions.

Assumption 1

(i) The structure of the system matrices J, R, G is known, except for a finite set of parameters $\Theta \subset \mathbb{R}^{n_\Theta}$, where n_Θ is the number of unknowns.

(ii) Observations of the PDE state (1) are available at equidistant temporal points $\{t_i\}_{i=0}^{N_H-1}$ and spatial points $\{z_j\}_{j=0}^{N_z-1}$, i.e., $\{\boldsymbol{x}(t_i,z_j) \mid i=0,\dots,N_H-1;\ j=0,\dots,N_z-1\}$.

(iii) There exists a unique, smooth solution $\boldsymbol{x}(t,z)$ of the PDE under the boundary conditions $B_\mathcal{Z}$ in (1) for all z.

(iv) The PDE (1) is controllable: for a set of initial state $\boldsymbol{x}(0,z) \in \mathcal{X}_0$ and desired final state $\boldsymbol{x}_f(z) \in \mathcal{X}_f$ there exist inputs $\boldsymbol{u}_t$ on $[0,T]$ such that $\boldsymbol{x}(T,z) = \boldsymbol{x}_f(z)$.

Item (i) is only lightly restrictive because J and G are typically simple matrix differential operators, given by prior knowledge of the PDE system to be learned. While knowing the dissipation matrix R is implied, there are unknown parameters such that only the general structure needs to be known, e.g., the friction model but not the parameters. Item (ii) ensures that we can collect data from the PDE system, requiring observability of the state. In addition, item (iii) guarantees the existence of a solution so that the derivation of a model is meaningful. Finally, item (iv) highlights the controllability and general feasibility of finding an adequate control input.

2.4. Physics-informed Scenario-based Model Predictive Control

In this section, we propose NAPI-MPC, a physics-informed scenario-based MPC framework leveraging GP-dPHS for robust control. The physics-informed feature arises by encoding the distributed port-Hamiltonian structure into the GP. Initially, NAPI-MPC learns the dPHS model from data, generating physically informed dynamic samples that satisfy energy conservation and passivity constraints. Given the probabilistic nature of GP-dPHS, multiple samples representing possible system dynamics are generated and passed to SCMPC. For real-time applicability, a neural network accelerator (introduced in Section 2.4.4) learns the mapping from initial states to optimal controls provided by the physics-informed SCMPC. By exploiting energy representations, this neural accelerator efficiently generalizes across varying initial states, rapidly delivering near-optimal controls.

2.4.1. Data Collection

First, we aim to collect data of the PDE system with unknown dynamics to train our GP-dPHS model before we leverage this model for MPC. As mentioned in the setting part 2.3, we define our observation of the system as $\mathcal{D} = \{t_i, \boldsymbol{z}_j, \boldsymbol{x}(t_i,\boldsymbol{z}_j), \boldsymbol{u}(t_i)\}_{i=0,j=0}^{i=N_t-1,j=N_z-1}$, where $\boldsymbol{x}(t_i,\boldsymbol{z}_j)$ denotes the observed state at time t_i and spatial point $\boldsymbol{z}_j$, corresponding to N_t time steps and N_z spatial points. These observations align with an input sequence $\{\boldsymbol{u}(t_1), \boldsymbol{u}(t_2), \dots, \boldsymbol{u}(t_{N_t})\}$. While exact

level of excitation is not required for such input sequences, truly arbitrary or insufficiently varied inputs can lead to overly conservative controllers. Considering practice, the number of observations is usually limited across the spatial domain but we also require state derivatives for training of the GP-dPHS model, see Tan et al. (2024). Hence, we deploy a smoothed kernel k, such as the squared exponential kernel, to reconstruct the state observations with GP models for upsampling data. The derivatives of the GP function provide estimates for state derivative $\frac{\partial}{\partial t}\boldsymbol{x}$ and allows the dataset to be augmented with N_e additional spatial points, where $N_e \gg N_z$. Consequently, we construct a new dataset $\mathcal{E}$, which comprises the states $X = [\tilde{\boldsymbol{x}}(t_0), \dots, \tilde{\boldsymbol{x}}(t_{N_t-1})]$ and the state derivatives $\dot{X} = \left[\frac{\partial \tilde{\boldsymbol{x}}(t_0)}{\partial t}, \dots, \frac{\partial \tilde{\boldsymbol{x}}(t_{N_t-1})}{\partial t}\right]$, where $\tilde{\boldsymbol{x}}(t_i) = [\boldsymbol{x}(t_i, \boldsymbol{z}_0)^\top, \dots, \boldsymbol{x}(t_i, \boldsymbol{z}_{N_e-1})^\top]^\top$ represents the spatially stacked state at time t_i. In this way, the dataset is defined as $\mathcal{E} = [X, \dot{X}]$.

2.4.2. TRAINING OF THE GAUSSIAN PROCESS DISTRIBUTED PORT-HAMILTONIAN MODEL

With the exact underlying dynamics unknown, we leverage the dataset $\mathcal{E}$ to learn a distributed Port-Hamiltonian model of the PDE system. This approach uses a Gaussian process to approximate the unknown Hamiltonian function while treating the parametric uncertainties in the matrices J, R, and G as hyperparameters. In simpler terms, learning the Hamiltonian derivatives is equivalent to learning the energy representations of the PDE system. This is made possible because GPs are invariant under linear transformations, see (Beckers, 2021), allowing us to incorporate the Hamiltonian functional derivatives into the GP framework. This integration of dPHS dynamics into the GP is mathematically represented as $\frac{\partial \boldsymbol{x}}{\partial t} \sim \mathcal{GP}(\hat{G}_\Theta \boldsymbol{u}, k_{dphs}(\boldsymbol{x}, \boldsymbol{x}'))$, where the system dynamics are encapsulated within the probabilistic model. Consequently, we can rewrite a new physics-informed kernel function k_{dphs} under the dPHS formulism, which is formally described as $k_{dphs}(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2 (\hat{JR}_\Theta) \delta_{\boldsymbol{x}} \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|^2}{2\varphi_l^2}\right) \delta_{\boldsymbol{x}'}^\top (\hat{JR}_\Theta)^\top$, see (Tan et al., 2024) for more details, where $\hat{JR}_\Theta = \hat{J}_\Theta - \hat{R}_\Theta$ and $\delta_{\boldsymbol{x}'}^\top$ is the adjoint of the functional derivative operator. This kernel is based on the squared exponential function to represent the smooth Hamiltonian functional. The model is trained as a standard GP using the dataset $\mathcal{E}$. The matrices J, R, and G of the dPHS system are inferred through their estimates $\hat{J}_\Theta$, $\hat{R}_\Theta$, and $\hat{G}_\Theta$, where the unknown parameters, denoted by Θ, are treated as hyperparameters. The hyperparameters Θ, φ_l, and σ_f are optimized by minimizing the negative log marginal likelihood. This completes the training of the proposed GP-dPHS model. Utilizing the invariance of GPs under linear transformation, we can rewrite Section 2.4.2 as

$$\frac{\partial}{\partial t}\boldsymbol{x}(t, \boldsymbol{z}) = (\hat{J}_\Theta - \hat{R}_\Theta)\delta_{\boldsymbol{x}} \hat{\mathcal{H}} + \hat{G}_\Theta \boldsymbol{u}, \tag{4}$$

where the Hamiltonian $\hat{\mathcal{H}}$ follows the posterior distribution of the GP. As consequence, we achieve a stochastic PDE as a model for the unknown PDE system based on the training data set. Next, we will use the GP-dPHS model in a scenario-based MPC apporach.

2.4.3. SCENARIO-BASED MPC WITH GP-DPHS MODEL

Now that we achieve the posterior distribution of GP, we can leverage the joint distribution to extract predictions for the flow field at test state $\boldsymbol{x}^* \in \mathbb{R}^n$, which facilitates the procurement of a vector field $\boldsymbol{f}$ that represents the right-hand side of (4). From this distribution, we could sample a $\boldsymbol{\xi}$, i.e., a sample of the learned dPHS system, and solve the resulting PDE with a numerical solver. However, the demands of numerical integration require the ability to access this vector field at various

points, which is computationally expensive (Beckers and Hirche, 2021). Instead, the approach advocates sampling the estimated Hamiltonian functional $\hat{\mathcal{H}}$, which leads to (4). With a deterministic sample of Hamiltonian $\hat{\mathcal{H}}^{(i)}$ from the GP distribution, we can then use a numerical PDE solver to compute a solution for (4). Since a GP sample is the deterministic function and the dPHS is designed to follow 1, it is fully compliant with the Stokes-Dirac structure. This procedure guarantees that the solution of the learned dynamics satisfies the dPHS form, thereby rendering physically correct solutions in terms of the energy evolution. The construction of GP-dPHS provides uncertainty quantification stemming from the posterior distribution of GP. Specifically, a set of N_s deterministic samples are drawn from the posterior of Hamiltonian $\hat{\mathcal{H}}$, each of which being a possible interpretation of the training dataset $\mathcal{E}$. Thus, selecting N_s requires balancing the fidelity of uncertainty sampling against computational cost based on a priori knowledge of the system: larger N_s values improve coverage of model variability, particularly for high-dimensional systems or wide noise bounds, while smaller N_s suffice for low-dimensional dynamics with tightly bounded uncertainties to meet real-time constraints. Let function $h(t, \boldsymbol{z})$ denotes the dPHS dynamics $\frac{d\boldsymbol{x}}{dt}$ for notation simplicity. The scenario with i-th sample from GP of Hamiltonian $\hat{\mathcal{H}}$ can be formally described as $h^{(i)}(\boldsymbol{x}, \boldsymbol{u}) = (\hat{J}_\Theta - \hat{R}_\Theta)\delta_{\boldsymbol{x}}\hat{\mathcal{H}}^{(i)} + \hat{G}_\Theta \boldsymbol{u}$. In other words, from GP-dPHS we are able to derive a set of N_s scenarios $\{h^{(i)}\}_{i=1}^{i=N_s}$, each of which is physically correct. This approach enables the MPC to explicitly account for uncertainty, balance performance and robustness, and maintain system stability under a wide range of operating conditions. For efficient sampling of the scenarios from the GP model, we leverage Matheron's rule to decompose the posterior into a prior and an update, as proposed in Wilson et al. (2020). Then, the prior is approximated using random Fourier features and the update via basis functions, enabling linear-time sampling while maintaining posterior accuracy.

After exploiting the sampling method, we draw samples from the probabilistic Hamiltonian and then we are provided with a N_s number of dynamic scenarios associated with the drawn Hamiltonian $\{h^{(i)}(\cdot,\cdot)\}_{i=1}^{N_s}$, each of which is a possible candidate of the actual system over the prediction horizon N_p. For each sample $h^{(i)}(\cdot,\cdot)$, we are able to derive $\bar{h}^{(i)}(\cdot,\cdot)$ through spatial discretization with N_z points and time discretization, the state dimensionality consistent with the original observation. Thus, we translate the scenarios of the GP-dPHS model into a set of ordinary difference equations. Then, the optimization problem is formulated as

$$\begin{aligned} &\min_{\boldsymbol{u}_t^*} \mathcal{J}(\hat{\boldsymbol{x}}_t^{(k)}, \boldsymbol{u}_t^*) = \sum_{k=1}^{N_s} \mathcal{J}_k(\hat{\boldsymbol{x}}_t^{(k)}, \boldsymbol{u}_t^*) \;\text{ s.t. } \hat{\boldsymbol{x}}_{t_{j+1}}^{(k)} = \bar{h}^{(i)}(\hat{\boldsymbol{x}}_{t_j}^{(i)}, \boldsymbol{u}_{t_j}^*),\; \hat{\boldsymbol{x}}_{t_0}^{(k)} = \boldsymbol{x}(t_0) \\ &w = B_{\mathcal{Z}}(\delta_{\boldsymbol{x}}\hat{\mathcal{H}}^{(i)}, \boldsymbol{u}_{t_j}^*), \hat{\boldsymbol{x}}_{t_j}^{(k)} \in \mathcal{X},\; \boldsymbol{u}_{t_j}^* \in \mathcal{U},\; \forall k \in \{1, \dots, N_s\},\; \forall j \in \{0, \dots, N_p - 1\} \end{aligned} \tag{5}$$

2.4.4. Neural Network Accelerator

Due to the computational complexity associated with GP and high-dimensional state variables, solving the MPC optimization problem can become computationally demanding, so that the optimization problem can hardly be solved with real-time constraints. To address this, particularly for time-sensitive systems, we propose training a deep neural network (DNN) F to approximate the mapping from states $\boldsymbol{x}$ to the optimal control reference $\boldsymbol{u}^*$ generated by the physics-constrained SCMPC. Using neural networks to accelerate the MPC is common, see: (Zhang et al., 2020; Salzmann et al., 2023). This approach allows for rapid inference of control actions in real-time, retaining the benefits of optimal control without online optimization. Analogously to Section 2.4.1, a dataset $\mathcal{D}_{mpc}$ is constructed for a set of (inital) states $X_{mpc} = [\boldsymbol{x}_0, \dots, \boldsymbol{x}_{N_{mpc}-1}]$, each associated with

the optimal control input $\boldsymbol{u}_i^*, i = 0, \ldots, N_{mpc} - 1$ derived via the physics-consistent SCMPC (5). Note that since we do not assume the actual model of the system, the states X_{mpc} are simulated via discretized version of the GP-dPHS model. In general, we see two approaches for the generation of this dataset. First, a random set of states is generated and for each state $\boldsymbol{x}_i \in X_{mpc}$ the SCMPC (5) is solved, where $\boldsymbol{x}_{t_0}^{(k)}$ is set to $\boldsymbol{x}_i$. Second, we use the physics-consistent SCMPC as we normally would do, but replace the actual system to control with a proxy. This proxy can be, for example, the mean prediction of the trained GP-dPHS system. In this way, we might generate more expressive dataset for the DNN as the data consists of trajectories instead of randomly distributed states. After creating the dataset, the DNN loss function $\mathcal{J}_{\mathcal{DNN}}(\boldsymbol{u}_i^*, F(\boldsymbol{x}_i))$ is defined to measure the squared Euclidean norm between predicted and SCMPC control inputs, ensuring the DNN approximates optimal control. Minimizing this loss enables learning the mapping from states to controls. The energy-based representation of the physics-informed model further enhances DNN generalization across various initial states, ensuring robust control in nonlinear environments.

3. Numerical Example

As an abstraction of controlling flexible objects with highly nonlinear behavior, e.g. soft robots, we aim here to control the dynamics of a lateral vibration of a string with a nonlinear stress-strain curve. This system is succinctly represented by the ensuing PDE

$$\frac{\partial^2 x}{\partial t^2} = s\left(\frac{\partial x}{\partial z}\right)\frac{\partial^2 x}{\partial z^2} - c\frac{\partial x}{\partial t}, \tag{6}$$

where s denotes the stress/strain curve that is, for soft materials, a highly nonlinear function of the stress $\frac{\partial x}{\partial z}$. Further, we assume that s follows a sigmoid function $s(x) = \frac{1}{1+e^{-x}}$ and a damping factor of $c = 0.05$. We refer to (6) as the ground truth. In the following, s is assumed to be *unknown*, as it would be the case in the real world (for simplicity we consider the constant c to be known).

First, we need to translate the PDE system into a dPHS structure. Recalling equation (1), it is postulated that the constant matrix differential operators $\mathcal{J}$, $\mathcal{R}$, and $\mathcal{G}$ are structurally known differential operators. Then, the desired system can be modeled via a dPHS in the following form

$$\frac{\partial}{\partial t}\begin{bmatrix} p(t,z) \\ q(t,z) \end{bmatrix} = \underbrace{\begin{bmatrix} -c & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial z} & 0 \end{bmatrix}}_{J-R} \delta_{\boldsymbol{x}}\mathcal{H}, \tag{7}$$

where $p = \frac{\partial x}{\partial t}$ and $q = \frac{\partial x}{\partial z}$. This transformation is necessary to ensure that J is an adjoint operator. The Hamiltonian is given by $\mathcal{H} = \int_{\mathcal{Z}} \int s(q)q(t,z)dq + p(t,z)^2 dz$ with, for us unknown, stress/strain function s. Following the problem statement, we assume that the structure of (7) is known, but the Hamiltonian is unknown to us. This example demonstrates that this is a reasonable assumption as the nonlinearity of the system is encapsulated in the Hamiltonian.

In our setting, the string spans the spatial domain $z \in [0, 10]$. We are examining a string system with left end fixed $x(t, 0) = 0$ and right end controllable $x(t, 0) = u(t)$. For the sake of simplicity, we assume no input and imply a non-trivial initial condition of the system to generate the dataset $\mathcal{E}$. At time $t = 0$, we set the initial condition to $x_0 = \exp(-(z - 5)^2)$. We observe the ground truth system for 20 seconds with a time step of $\Delta t = 0.01$ seconds. The spatial domain is observed at 8 points, which are upsampled to $N_e = 400$ points by using a GP regression model. To accelerate the

training, we flatten all the observations, apply a downsample rate = 50 to all the data, and train the proposed GP-dPHS model using the steps mentioned above in section 2.4.2.

We further assume the number of scenarios $N_s = 3$ and prediction horizon $N_p = 3$. We randomly generate 10 initial conditions, half of which following the form of $x_0 = \exp(-(\lambda z - 5)^2)$ and other half following $x_0 = \alpha \sin(\frac{\lambda\pi}{10} z)$, where $\alpha, \lambda \in \mathbb{R}$ are randomly generated with $\alpha \in [-2, 2]$ and $\lambda \in \{-3, -2, -1, 1, 2, 3\}$ to ensure the right end at 0 initially. All of the newly generated initial conditions are not seen in the training phase of GP-dPHS. The SCMPC complementary to GP-dPHS ran offline to provide control sequences that stabilize the string. For computational efficiency of SCMPC, we reduce the spatial discretizations from $N_e = 400$ to 100 points at each time step for the newly generated samples. The state of the spring system consists, then, of the partial derivatives across both temporal and spatial domains, hence is of dimension 200.

We trained DNN accordingly, which consists of three fully connected layers with 512, 256, and 1 neurons, respectively, transforming an input of size 200 into a single output. The activation function $\tanh$ introduces nonlinearity after the first two layers, enabling the model to capture complex relationships in the input data. DNN learns the mapping from state pairs to the control action, whose effect can be visualized in Figure 2. Once properly trained, we leverage the DNN as proxy for our SCMPC in the closed control loop. For testing the proposed control approach, we use $x_0 = \exp(-(\lambda z - 5))^2$ with $\lambda \in [-3.5, -2.5]$ as initial condition, and we aim to stabilize the spring system to $x = 0$. Note that this initial condition was neither used for the training of the GP-dPHS model nor for the training of the DNN. In fact, the newly generated initial condition is greater in magnitude, making the dynamics more challenging than training datasets. One of the advantages for NAPI-MPC is its efficiency. With the total prediction horizon $N_h = 100$ seconds and $\Delta t = 0.01$ seconds, the inference time of the NAPI-MPC is in average 0.00075 seconds. Compared to running with SCMPC, whose inference time is in average 65 seconds, the NAPI-MPC behaves in a time-sensitive manner, which is applicable to real world. The results are visualized in Figure 1, where the NAPI-MPC stabilizes the system quickly, whereas the uncontrolled system converges significantly slower. Control inputs from SCMPC are usually conservative, which is reflected in the results produced by NAPI-MPC. We further visualize the control performance of NAPI-MPC over both temporal and spatial domain, shown in Figure 2. In the control problem, we assume the left end of the string is fix while the right end is controllable, which is illustrated in Figure 2. The

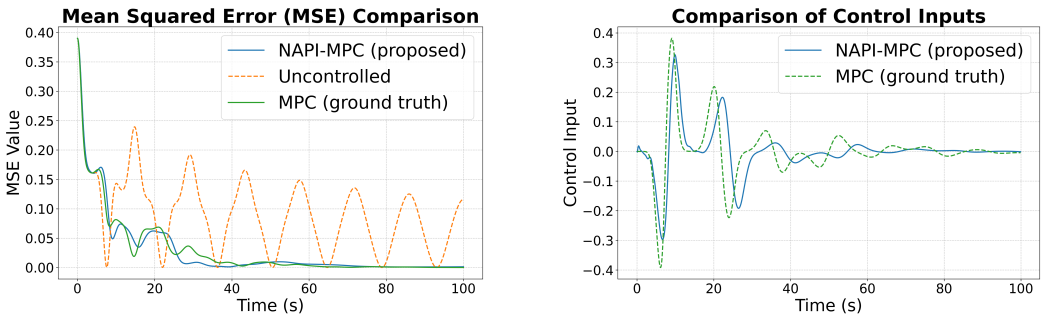

Figure 1: (a) Comparison between uncontrolled (dashed, orange), NAPI-MPC (blue), and MPC using ground truth model (green). NAPI-MPC stabilizes the string at horizon, with performance comparable to the MPC. (b) Comparison between control inputs from NAPI-MPC and MPC using ground truth model. NAPI-MPC is more conservative due to the uncertainties in the model.

results further show that the NAPI-MPC is able to stabilize the string at horizon, demonstrating the successful learning of control from SCMPC.

Other PDE learning approaches, such as Sparse Identification of Nonlinear Dynamical Systems (SINDy; Kaptanoglu et al., 2022), directly fit the wave equation in the form of (6) by selecting a sparse representation. Using the exact same dataset as GP-dPHS, we train SINDy to express the second partial derivative $\frac{\partial^2 x}{\partial t^2}$ from the feature library $\{\frac{\partial^2 x}{\partial z^2}, \frac{\partial x}{\partial t}\}$, then integrate its learned equation to reconstruct states. As shown in Figure 2, this reconstruction captures the overall string motion but gradually overestimates the oscillation frequency and amplitude. In contrast, GP-dPHS embeds the distributed port-Hamiltonian structure directly into its Gaussian-process prior, guaranteeing energy conservation by construction. At test time, GP-dPHS produces more accurate results, demonstrating its effectiveness even in sparse data and nonlinear dynamics without assuming prior knowledge.

Discussion The evaluation shows the effectiveness of the proposed pipeline with promising results. The GP-dPHS model generalizes well, even with small datasets. However, the quality of DNN proxy depends on the volume of SCMPC-generated data, which is time-consuming. Further research should focus on enhancing the efficiency and scalability of data generation, enabling NAPI-MPC applications to higher-dimensional spatial domains.

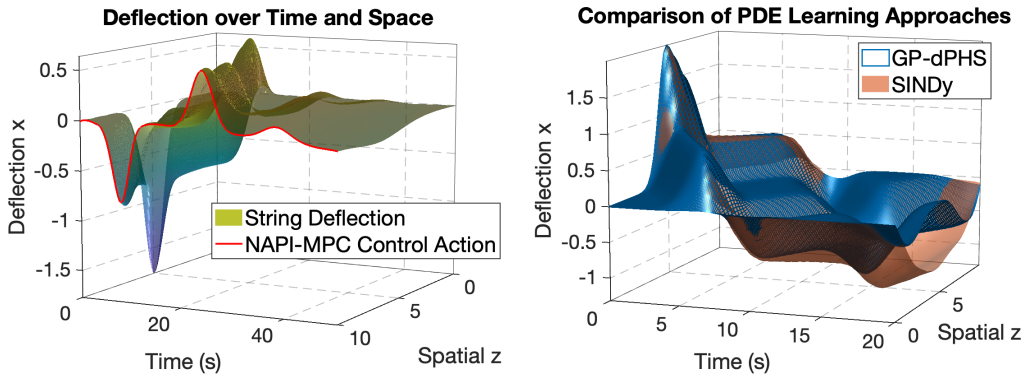

Figure 2: Left figure demonstrates NAPI-MPC controlled string with an unseen initial condition. The string is fixed at the rear end ($x(t, 0) = 0$) with control action at the front end of the string ($x(t, 10) = u$). NAPI-MPC controls the string to converge rapidly to zero. Right figure compares the prediction given by GP-dPHS and SINDy. The GP-dPHS model (MSE=0.0053) performs better in approximating the deflection in comparison to the ground truth than SINDy (MSE=0.078).

4. Conclusion

In this paper, we propose a physics-enhanced MPC method for control of nonlinear PDE systems with partially unknown dynamics. We leverage GP-dPHS to learn a probabilistic model of the system and draw samples of the posterior distribution for a scenario-based MPC. Finally, a deep neural network is utilized to approximate the optimal control input which gives a significant speedup. In the numerical evaluation, we demonstrate on a nonlinear wave equation, where the proposed method is able to quickly stabilize the system. In the future, we will test on more complex control problems (higher-dimensional systems, nonlinearities, etc.), as well as explore stability guarantees for the proposed controller.

References

Mohammed S Alhajeri, Fahim Abdullah, Zhe Wu, and Panagiotis D Christofides. Physics-informed machine learning modeling for predictive control using noisy data. *Chemical Engineering Research and Design*, 186:34–49, 2022.

Hassan Arbabi, Milan Korda, and Igor Mezić. A data-driven koopman model predictive control framework for nonlinear partial differential equations. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 6409–6414. IEEE, 2018.

Thomas Beckers. An introduction to Gaussian process models. *arXiv preprint arXiv:2102.05497*, 2021.

Thomas Beckers. Data-driven bayesian control of port-Hamiltonian systems. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pages 8708–8713. IEEE, 2023.

Thomas Beckers and Sandra Hirche. Prediction with approximated Gaussian process dynamical models. *IEEE Transactions on Automatic Control*, pages 1–1, 2021.

Luke Bhan, Yuexin Bian, Miroslav Krstic, and Yuanyuan Shi. PDE control gym: A benchmark for data-driven boundary control of partial differential equations. In *6th Annual Learning for Dynamics & Control Conference*, pages 1083–1095. PMLR, 2024.

Giuseppe C Calafiore and Lorenzo Fagiano. Robust model predictive control via scenario optimization. *IEEE Transactions on Automatic Control*, 58(1):219–224, 2012.

Giuseppe C Calafiore and Lorenzo Fagiano. Stochastic model predictive control of lpv systems via scenario optimization. *Automatica*, 49(6):1861–1866, 2013.

Andrea Carron, Elena Arcari, Martin Wermelinger, Lukas Hewing, Marco Hutter, and Melanie N Zeilinger. Data-driven model predictive control for trajectory tracking with a robotic arm. *IEEE Robotics and Automation Letters*, 4(4):3758–3765, 2019.

Thai Duong and Nikolay Atanasov. Hamiltonian-based neural ode networks on the SE(3) manifold for dynamics learning and control. In *Robotics: Science and Systems (RSS)*, 2021.

Morgan T Gillespie, Charles M Best, Eric C Townsend, David Wingate, and Marc D Killpack. Learning nonlinear dynamic models of soft robots for model predictive control with neural networks. In *2018 IEEE International Conference on Soft Robotics (RoboSoft)*, pages 39–45. IEEE, 2018.

Sebastian Hirt, Maik Pfefferkorn, and Rolf Findeisen. Safe and stable closed-loop learning for neural-network-supported model predictive control. In *2024 IEEE 63rd Conference on Decision and Control (CDC)*, pages 4764–4770, 2024.

Mihailo R Jovanović. From bypass transition to flow control and data-driven turbulence modeling: an input–output viewpoint. *Annual Review of Fluid Mechanics*, 53(1):311–345, 2021.

Alan A Kaptanoglu, Brian M de Silva, Urban Fasel, Kadierdan Kaheman, Andy J Goldschmidt, Jared Callaham, Charles B Delahunt, Zachary G Nicolaou, Kathleen Champion, Jean-Christophe

Loiseau, et al. Pysindy: A comprehensive python package for robust sparse system identification. *Journal of Open Source Software*, 7(69):3994, 2022.

Miroslav Krstic, Petar V Kokotovic, and Ioannis Kanellakopoulos. *Nonlinear and adaptive control design*. John Wiley & Sons, Inc., 1995.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y Hou, and Max Tegmark. Kan: Kolmogorov-arnold networks. *arXiv preprint arXiv:2404.19756*, 2024.

Anastasios Matzavinos, Mark AJ Chaplain, and Vladimir A Kuznetsov. Mathematical modelling of the spatio-temporal response of cytotoxic t-lymphocytes to a solid tumour. *Mathematical Medicine and Biology*, 21(1):1–34, 2004.

Subramanya P Nageshrao, Gabriel AD Lopes, Dimitri Jeltsema, and Robert Babuška. Port-Hamiltonian systems in adaptive and learning control: A survey. *IEEE Transactions on Automatic Control*, 61(5):1223–1238, 2015.

Jonas Nicodemus, Jonas Kneifl, Jörg Fehr, and Benjamin Unger. Physics-informed neural networks-based model predictive control for multi-link manipulators. *IFAC-PapersOnLine*, 55(20):331–336, 2022.

Mohammad Mamunur Rahman, Yusheng Feng, Thomas E Yankeelov, and J Tinsley Oden. A fully coupled space–time multiscale modeling framework for predicting tumor growth. *Computer methods in applied mechanics and engineering*, 320:261–286, 2017.

Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

Carl Edward Rasmussen and Christopher KI Williams. *Gaussian processes for machine learning*. MIT press Cambridge, 2006.

Tim Salzmann, Elia Kaufmann, Jon Arrizabalaga, Marco Pavone, Davide Scaramuzza, and Markus Ryll. Real-time neural mpc: Deep learning model predictive control for quadrotors and agile robotic platforms. *IEEE Robotics and Automation Letters*, 8(4):2397–2404, 2023.

Robert Stephany and Christopher Earls. PDE-learn: Using deep learning to discover partial differential equations from noisy, limited data. *Neural Networks*, 174:106242, 2024.

Kaiyuan Tan, Peilun Li, and Thomas Beckers. Physics-constrained learning of PDE systems with uncertainty quantified port-Hamiltonian models. In *Proceedings of the 6th Annual Learning for Dynamics and Control Conference*, volume 242 of *PMLR*, pages 1753–1764. PMLR, 15–17 Jul 2024.

James Wilson, Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, and Marc Deisenroth. Efficiently sampling functions from Gaussian process posteriors. In *International Conference on Machine Learning*, pages 10292–10302. PMLR, 2020.

Tobias M Wolff, Johannes W Dietrich, and Matthias A Müller. Optimal hormone replacement therapy in hypothyroidism-a model predictive control approach. *Frontiers in Endocrinology*, 13: 884018, 2022.

Mario Zanon, Janick V Frasch, Milan Vukov, Sebastian Sager, and Moritz Diehl. Model predictive control of autonomous vehicles. In *Optimization and optimal control in automotive systems*, pages 41–57. Springer, 2014.

Xiaojing Zhang, Monimoy Bujarbaruah, and Francesco Borrelli. Near-optimal rapid mpc using neural networks: A primal-dual policy learning framework. *IEEE Transactions on Control Systems Technology*, 29(5):2102–2114, 2020.