

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

Peilun Li

Kaiyuan Tan

Thomas Beckers

PEILUN.LI@VANDERBILT.EDU

KAIYUAN.TAN@VANDERBILT.EDU

THOMAS.BECKERS@VANDERBILT.EDU

*Department of Computer Science, Vanderbilt University, Nashville, TN 37212*

## Abstract

The control of systems governed by *nonlinear* partial differential equations (PDEs) can present substantial challenges for traditional linear 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 in soft robotics or nonlinear interactions across multiple flow scales in turbulent fluid dynamics. Despite their widespread presence, controlling these systems remains challenging due to their inherent complexity. 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 (Rasmussen and Williams, 2006), 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 environmental uncertainties by optimizing across multiple scenarios to ensure robust control. Despite this, SCMPC’s effectiveness 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) employs

deep neural networks 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., 2021) utilizes different basis functions to approximate the target PDEs. However, such basis functions may not always be practically available. Moreover, the absence of inherent physical constraints often compromises model performance under noisy measurements, and most importantly, these methods lack consideration of the uncertainties inherent in the learning process. Therefore, a relatively robust, physics-constrained, function-free approach with uncertainty quantification is needed.

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 physical laws into the learning process through the loss function, ensuring that the learned models respect the governing physics. Despite these advances, two critical issues remain. First, exact knowledge of the underlying physical system is not always available, which complicates the effective training of PINNs. Secondly, even with physics-based priors, there remains a need to assess model accuracy comprehensively to ensure 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) (Nagesh Rao et al., 2015; Duong and Atanasov, 2021) have gained increasing attention due to their capability to represent nonlinear systems in an energy-based way. PHS provides a systematic and structured approach for modeling the complete system, abstracting away from the detailed internal 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) (Beckers et al., 2022; Li et al., 2024) and partial differential equations (PDEs) (Tan et al., 2024a), 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:** In this paper, we propose a novel physics-constrained scenario-based model predictive control algorithm named *NAPI-MPC*: Neural Accelerated Physics-informed MPC for nonlinear PDEs, designed to achieve reliable control by incorporating uncertainty quantification. Our approach 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 generalization capabilities across unseen data and initial conditions. By integrating neural learning with physics-informed MPC, our method significantly accelerates the computation, making it suitable 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., 2024a) for more details. The main idea is to formularize the nonlinear PDE as a distributed port-Hamiltonian system  $\Sigma^1$

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

where the spatial-temporal state is denoted as  $\mathbf{x}(t, \mathbf{z})$  for the corresponding time  $t$  and spatial point  $\mathbf{z}$ . 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  $\mathbf{u}$ . The functional derivative operator  $\delta_{\mathbf{x}}$  is comparable to the partial derivative operator but acts on the Hamiltonian functional  $\mathcal{H}$ . Finally, the boundary conditions are defined by  $w = B_{\mathcal{Z}}(\delta_{\mathbf{x}}\mathcal{H}, \mathbf{u})$ , where  $B_{\mathcal{Z}}$  is the boundary operator.

After defining the nonlinear PDE in the dPHS structure, GP-dPHS aims to learn the unknown Hamiltonian with a non-parametric Gaussian process (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 dPHS by treating them as hyperparameters. There are several benefits for such approach: First, GP-dPHS does not require prior knowledge about parametric structure of the Hamiltonian functional, making it suitable to learn complex nonlinear system dynamics. Furthermore, the GP as Bayesian learning framework encapsulates all possible dPHS configurations under the Bayesian prior from a finite set of observations, resulting in predictions with confidence intervals. Not only is this feature critical for model identification, but we will exploit it 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

$$\mathbf{x}(t_{i+1}) = \mathbf{f}(\mathbf{x}(t_i), \mathbf{u}(t_i)), \quad (2)$$

where  $\mathbf{f}$  represents a vector-valued function,  $\mathbf{x}(t_i) \in \mathbb{R}^n$  is the state and  $\mathbf{u}(t_i) \in \mathbb{R}^m$  denotes the control input at time  $t_i \in \{t_0, \dots, t_{N_H-1}\}$ , where  $N_H$  is the total horizon for the control problem with equidistant time steps. A model of the dynamics  $\hat{\mathbf{f}} : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$  is a function that maps current state and control input to the next state  $\mathbf{x}(t_{i+1})$ .

1. Vectors  $\mathbf{a}$  and vector-valued functions  $\mathbf{f}(\cdot)$  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  $\Lambda$ ,  $\|x - y\|_{\Lambda}^2 = (x - y)^{\top} \Lambda (x - y)$ .  $\mathbb{R}_{>0}$  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  $\nabla_{\mathbf{x}}$  with  $\mathbf{x} \in \mathbb{R}^n$  denotes  $[\frac{\partial}{\partial x_1}, \dots, \frac{\partial}{\partial x_n}]^{\top}$ .

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

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

where at time  $t$ ,  $\mathbf{u}_t^* = \mathbf{u}_{t_0, \dots, N_p-1}^*$  is the optimal control input sequence, and  $\hat{\mathbf{x}}_{t_j}^{(k)} \in \mathbb{R}^n$  denotes the estimated state from model  $\hat{\mathbf{f}}^{(k)}$ . We further constrain the estimated states to be in some set  $\mathcal{X}$ , and the control inputs in set  $\mathcal{U}$ . The loss function  $\mathcal{J}(\cdot, \cdot)$  typically measures the mismatch between the desired state and the current state, defining the desired control goal for the MPC setting. The solution to equation (3) is denoted by  $\mathbf{u}_t^*$ , and the first element  $\mathbf{u}_{t_0}^*$  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  $\mathbf{x}(t, z) \in \mathbb{R}^n$  over time  $t \in \mathbb{R}_{\geq 0}$  and spatial domain  $z \in \mathcal{Z}$  should be defined by the equation (1) with initial state  $\mathbf{x}(0, z)$ . 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  $\mathbf{x}(t, z)$  over both temporal and spatial domain. We aim to find a robust control input  $\mathbf{u}^*$  that achieves a feasible control goal while considering model uncertainties. To address the problem, we introduce the following assumptions.

**Assumption 1** *The structure of the system matrices  $J$ ,  $R$ , and  $G$  is assumed to be known, except for a finite set of parameters  $\Theta \subset \mathbb{R}^{n_\Theta}$ , where  $n_\Theta$  is the total number of unknown parameters in these matrices.*

**Assumption 2** *Observations of the state of the PDE system (1) are available at equidistant temporal points  $\{t_i\}_{i=0}^{N_H-1}$  and spatial points  $\{z_j\}_{j=0}^{N_z-1}$  such that  $\{\mathbf{x}(t_i, z_j) \mid i = 0, \dots, N_H - 1; j = 0, \dots, N_z - 1\}$ .*

**Assumption 3** *There exists a unique and smooth solution  $\mathbf{x}(t, z)$  for the PDE system under the given boundary conditions  $B_{\mathcal{Z}}$  (1).*

**Assumption 4** *The PDE system (1) is assumed to be controllable via control inputs  $\{\mathbf{u}_t\}_{t=1}^{N_t}$ . Specifically, for a set of initial state  $\mathbf{x}(0, z) \in \mathcal{X}_0$  and desired final state  $\mathbf{x}_f(z) \in \mathcal{X}_f$ , there exists control inputs  $\mathbf{u}_t$  defined on time horizon  $[0, T]$  such that  $\mathbf{x}(T, z) = \mathbf{x}_f(z)$ .*

Assumption 1 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. Assumption 2 ensures that we can collect data from the PDE system, requiring observability of the state. In addition, Assumption 3 guarantees the existence of a solution so that the derivation of a model is meaningful. Finally, Assumption 4 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, our physics-informed scenario-based MPC approach based on GP-dPHS for robust control. The first phase of NAPI-MPC involves learning of the dPHS model of the PDE system from data. Once trained, by the probabilistic nature of GP-dPHS, we are able to draw samples of the model dynamics, each of which is a possible representation of the actual dynamics. These samples are then propagated to the SCMPC. To address the real-time applications, we eventually employ a deep neural network accelerator in Section 2.4.4 to learn the mapping from the initial state to the optimal control provided by the physics-informed SCMPC. By leveraging the physics-informed properties through energy representation learning, this neural accelerator efficiently generalizes across systems with diverse initial states, providing near-optimal control rapidly.

### 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  $\mathcal{D} = \{t_i, \mathbf{z}_j, \mathbf{x}(t_i, \mathbf{z}_j), \mathbf{u}(t_i)\}_{i=0, j=0}^{i=N_t-1, j=N_z-1}$ , where  $\mathbf{x}(t_i, \mathbf{z}_j)$  denotes the observed state at time  $t_i$  and spatial point  $\mathbf{z}_j$ , corresponding to  $N_t$  time steps and  $N_z$  spatial points. These observations align with an input sequence  $\{\mathbf{u}(t_1), \mathbf{u}(t_2), \dots, \mathbf{u}(t_{N_t})\}$  that can be arbitrary but should excite the system well enough to gather an expressive dataset. 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. (2024b). 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  $\frac{\partial}{\partial t} \mathbf{x}$  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  $X = [\tilde{\mathbf{x}}(t_0), \dots, \tilde{\mathbf{x}}(t_{N_t-1})]$  and the state derivatives  $\dot{X} = \left[ \frac{\partial \tilde{\mathbf{x}}(t_0)}{\partial t}, \dots, \frac{\partial \tilde{\mathbf{x}}(t_{N_t-1})}{\partial t} \right]$ , where  $\tilde{\mathbf{x}}(t_i) = [\mathbf{x}(t_i, \mathbf{z}_0)^\top, \dots, \mathbf{x}(t_i, \mathbf{z}_{N_e-1})^\top]^\top$  represents the spatially stacked state at time  $t_i$ . In this way, the dataset is defined as  $\mathcal{E} = [\dot{X}, X]$ .

### 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

$$\frac{\partial \mathbf{x}}{\partial t} \sim \mathcal{GP}(\hat{G}_\Theta \mathbf{u}, k_{dphs}(\mathbf{x}, \mathbf{x}')), \quad (4)$$

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

$$k_{dphs}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 (\hat{J}R_\Theta) \delta_{\mathbf{x}} \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\varphi_l^2}\right) \delta_{\mathbf{x}'}^\top (\hat{J}R_\Theta)^\top, \quad (5)$$

see (Tan et al., 2024a) for more details, where  $\hat{J}R_\Theta = \hat{J}_\Theta - \hat{R}_\Theta$ . 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  $\Theta$ , are treated as hyperparameters. The hyperparameters  $\Theta$ ,  $\varphi_l$ , and  $\sigma_f$  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 (4) as

$$\frac{\partial}{\partial t} \mathbf{x}(t, \mathbf{z}) = (\hat{J}_\Theta - \hat{R}_\Theta) \delta_{\mathbf{x}} \hat{\mathcal{H}} + \hat{G}_\Theta \mathbf{u}, \quad (6)$$

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 approach.

#### 2.4.3. SCENARIO-BASED MPC WITH GP-DPHS MODEL

Now that we achieved the posterior distribution of GP, we can leverage the joint distribution to extract predictions for the flow field at test state  $\mathbf{x}^* \in \mathbb{R}^n$

$$\begin{bmatrix} \dot{\mathbf{X}} \\ \boldsymbol{\xi}(\mathbf{x}^*) \end{bmatrix} = \mathcal{N}\left(\hat{G}_\Theta \mathbf{u}, \begin{bmatrix} K_{dphs}(X, X) & K_{dphs}(X, \mathbf{x}^*) \\ K_{dphs}(X, \mathbf{x}^*)^T & K_{dphs}(\mathbf{x}^*, \mathbf{x}^*) \end{bmatrix}\right), \quad (7)$$

which facilitates the procurement of a vector field  $\mathbf{f}$  that represents the right-hand side of (6). 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 (6). 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 (6). 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  $\mathcal{E}$ . Let function  $h(t, \mathbf{z})$  denotes the dPHS dynamics  $\frac{d\mathbf{x}}{dt}$  for notation simplicity. The scenario with  $i$ -th sample from GP of Hamiltonian  $\hat{\mathcal{H}}$  can be formally described as

$$h^{(i)}(\mathbf{x}, \mathbf{u}) = (\hat{J}_\Theta - \hat{R}_\Theta)\delta_{\mathbf{x}}\hat{\mathcal{H}}^{(i)} + \hat{G}_\Theta\mathbf{u}. \quad (8)$$

In other words, from GP-dPHS we are able to derive a set of  $N_s$  scenarios  $\{h^{(i)}\}_{i=1}^{N_s}$ , each of which is physically correct. In light of scenario-based MPC, such a set of models can be leveraged to account for uncertainties and variations in the system's dynamics. By incorporating the scenarios in the MPC framework, the controller can optimize its decisions not for a single deterministic model, but for the entire ensemble, ensuring robust performance across all plausible scenarios. 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 are provided with  $N_s$  number of scenarios of the dPHS dynamics  $\{h^{(i)}(\cdot, \cdot)\}_{i=1}^{N_s}$ , each of which is a possible realization 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

$$\begin{aligned} \min_{\mathbf{u}_t^*} \mathcal{J}(\hat{\mathbf{x}}_t^{(k)}, \mathbf{u}_t^*) &= \sum_{k=1}^{N_s} \mathcal{J}_k(\hat{\mathbf{x}}_t^{(k)}, \mathbf{u}_t^*) \\ \text{s.t. } \forall k \in \{1, \dots, N_s\}, \forall j \in \{0, \dots, N_p - 1\} \\ \hat{\mathbf{x}}_{t_{j+1}}^{(k)} &= \bar{h}^{(i)}(\hat{\mathbf{x}}_{t_j}^{(i)}, \mathbf{u}_{t_j}^*), \hat{\mathbf{x}}_{t_0}^{(k)} = \mathbf{x}(t_0) \\ w &= B_{\mathcal{Z}}(\delta_{\mathbf{x}}\hat{\mathcal{H}}^{(i)}, \mathbf{u}_{t_j}^*), \hat{\mathbf{x}}_{t_j}^{(k)} \in \mathcal{X}, \mathbf{u}_{t_j}^* \in \mathcal{U} \end{aligned} \quad (9)$$

#### 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 computational demanding such that the optimization problem can not be solved in time. To address this, particularly for time-sensitive systems, we propose training a deep neural network (DNN)  $F$  to approximate the mapping from states  $\mathbf{x}$  to the optimal control reference  $\mathbf{u}^*$  generated by the physics-constrained SCMPC. 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 (initial) states  $X_{mpc} = [\mathbf{x}_0, \dots, \mathbf{x}_{N_{mpc}-1}]$ , each associated with the optimal control input  $\mathbf{u}_i^*, i = 0, \dots, N_{mpc} - 1$  derived via the physics-consistent SCMPC (9). Note, that this state belongs to the 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  $\mathbf{x}_i \in X_{mpc}$  the SCMPC (9) is solved, where  $\mathbf{x}_{t_0}^{(k)}$  is set to  $\mathbf{x}_i$ . 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  $\mathcal{J}_{\mathcal{DNN}}(\mathbf{u}_i^*, F(\mathbf{x}_i))$  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. The complete algorithm of the proposed method is presented in algorithm 1.

### 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

$$\frac{\partial^2 x}{\partial t^2} = s\left(\frac{\partial x}{\partial z}\right) \frac{\partial^2 x}{\partial z^2} - c \frac{\partial x}{\partial t}, \quad (10)$$

where  $s$  denotes the stress/strain curve that is, for soft materials, a highly nonlinear function of the stress  $\frac{\partial x}{\partial z}$ . In this example, 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 (10) as the ground truth. In the following,  $s$  is assumed to be unknown, as it would be the case in the real world (for the sake of 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

$$\frac{\partial}{\partial t} \begin{bmatrix} p(t, z) \\ q(t, z) \end{bmatrix} = \underbrace{\begin{bmatrix} -c & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial z} & 0 \end{bmatrix}}_{J-R} \delta_{\mathbf{x}} \mathcal{H}, \quad (11)$$

---

#### Algorithm 1 NAPI-MPC: Neural Accelerated Physics-informed MPC for nonlinear PDEs

---

**Input:** {Initial state training dataset  $\mathcal{D}$ , Number of scenarios  $N_s$ }  
**Output:** {Accelerate control output controller  $F$ }  
 Use dataset  $\mathcal{D}$  to train a GP-dPHS model (4) of the PDE system (1)  
 Draw  $N_s$  samples (8) from the learned Hamiltonian functional  $\mathcal{H}$   
 Discretize the samples to get  $\bar{h}^{(i)}, i = 1, \dots, N_s$   
**while**  $i$  in  $N_{mpc}$  **do**  
   Randomly generate initial condition  $\mathbf{x}(t_0)$   
   Solving SCMPC (9) with the  $N_s$  samples and apply the optimal control action  $\mathbf{u}_{t_0}^*$  to (8), where  $\hat{\mathcal{H}}$  is the posterior mean  
   Collect the state  $\mathbf{x}_i$  and optimal control action  $\mathbf{u}_i^*$   
**end while**  
 Split data into testing and training set  
 Train a DNN by minimize the risk function  $\mathcal{J}_{\mathcal{DNN}}(\mathbf{u}_i^*, F(\mathbf{x}_i))$   
 Output: DNN  $F$ , which is than used as controller for the acutal PDE system

---



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 (11) 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  $x(t, 10) = u(t)$ . 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.

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 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. 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

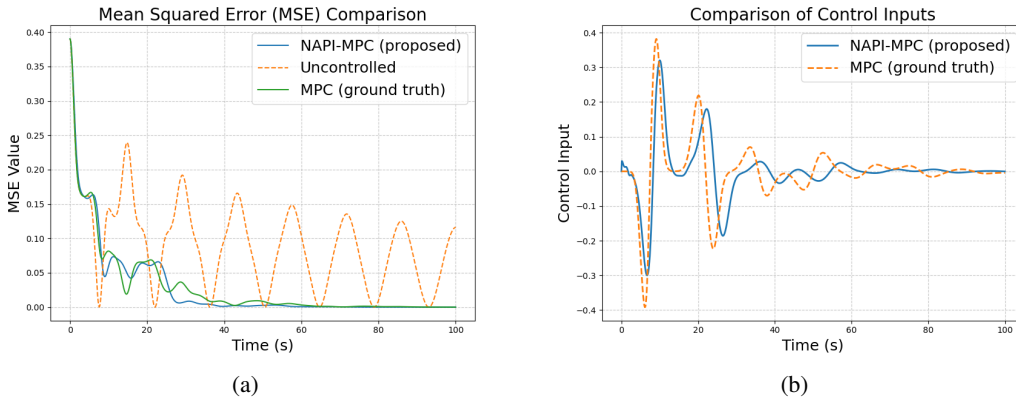


Figure 1: (a) Comparison between uncontrolled (dashed, orange), NAPI-MPC (blue), and MPC using ground truth model (green). NAPI-MPC stabilizes the string at horizon, with performance comparable to the MPC. (b) Comparison between control inputs from NAPI-MPC and MPC using ground truth model. NAPI-MPC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

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

**Discussion** The evaluation shows the proposed pipeline’s effectiveness, with promising results. The GP-dPHS model generalizes well, even with small datasets. However, the DNN proxy’s quality 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.

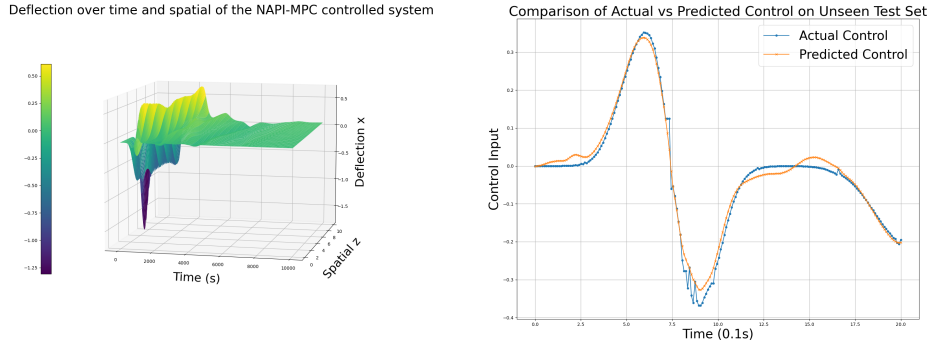


Figure 2: Left figure demonstrates NAPI-MPC controlled string over the unseen initial condition. The string is fixed at the left end ( $x(t, 0) = 0$ ) with control input at the right end of the string ( $x(t, 10) = u$ ). NAPI-MPC controls the string to converge rapidly to zero. Right figure compairs the prediction given by DNN (orange) over unseen initial conditions to the reference control given by SCMPC (blue). It shows excellent generalizability.

## 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 DNN is utilized to approximate the optimal control input which gives a significant speedup. In the numerical evaluation, we demonstrate the proposed method on a nonlinear wave equation, where the proposed MPC is able to quickly stabilize the system. In the future, we will test on more complex control problems.

## 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 (to appear)*, pages 1–1, 2021. doi: 10.1109/TAC.2021.3131988.
- Thomas Beckers, Jacob Seidman, Paris Perdikaris, and George J Pappas. Gaussian process port-hamiltonian systems: Bayesian learning with physics prior. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 1447–1453. IEEE, 2022.
- 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 ltv 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. *arXiv preprint arXiv:2106.12782*, 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. *arXiv preprint arXiv:2409.10171*, 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 L Callaham, Charles B Delahunt, Zachary G Nicolaou, Kathleen Champion, Jean-Christophe Loiseau, et al. Pysindy: A comprehensive python package for robust sparse system identification. *arXiv preprint arXiv:2111.08481*, 2021.
- Peilun Li, Kaiyuan Tan, and Thomas Beckers. Pygpphs: A python package for bayesian modeling of port-hamiltonian systems. *IFAC-PapersOnLine*, 58(6):54–59, 2024.
- 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.
- Subramanya P Nagesh Rao, 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.
- 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.
- 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 2024a.
- Kaiyuan Tan, Peilun Li, and Thomas Beckers. Physics-constrained learning for pde systems with uncertainty quantified port-hamiltonian models. *arXiv preprint arXiv:2406.11809*, 2024b.
- 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.