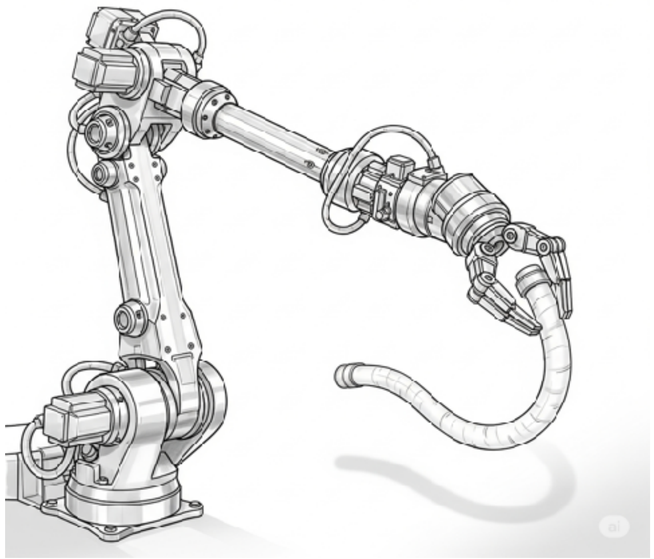
VANDERBILT UNIVERSITY

Structure- and Stability-Preserved Learning with Port-Hamiltonian Systems

Dr.-Ing. Thomas Beckers
Vanderbilt University

Tutorial Session on Safe Physics-Informed Machine Learning for Dynamics and Control
July 8th, 2025

Modeling of complex systems



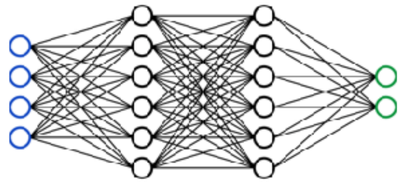
Physics-based

Robot $\dot{x} = f(x, u) = \dots$

Tube $0 = f(\frac{\partial x}{\partial t}, \dots, \frac{\partial x}{\partial z}, \dots)$

- ✓ Physical consistent
- ✓ Generalization
- ✗ Model selection (nonlinear)
- ✗ Time-consuming

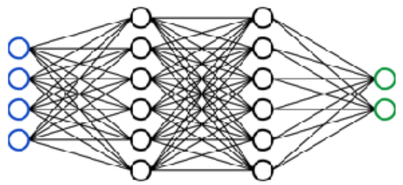
Learning-based



- ✓ Expressive models
- ✓ Minimal expert knowledge
- ✗ Physical correctness (structure)
- ✗ Stability, passivity?

Combining the best of both worlds

Learning-based

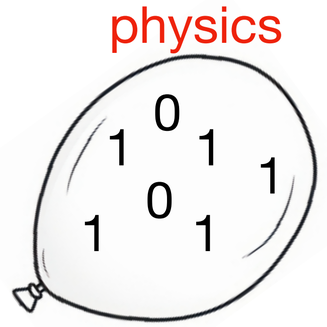


+

Structure / Physics-based

$$\dot{x} = f(x, u) = \dots$$

$$0 = f(\frac{\partial x}{\partial t}, \dots, \frac{\partial x}{\partial z}, \dots)$$



- Structure to constrain/inform the model output to be physically consistent
- Data-driven methods to fill the physics structure

“Best” Learning technique?

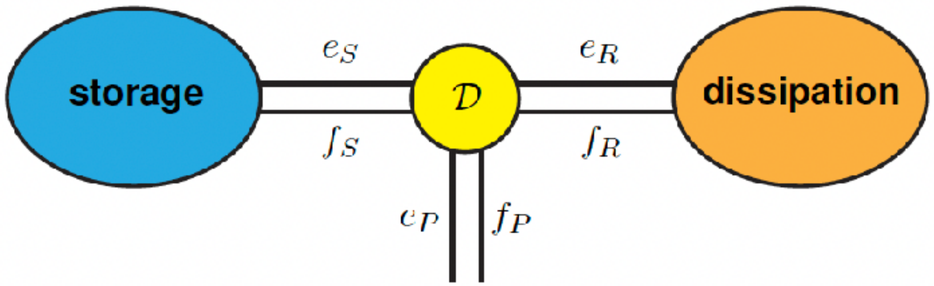
- Neural Networks
- GNNs, RNNs, LSTMs
- Gaussian Processes
- Operator Learning

“Best” Structure?

- Parametric equation
- Hamiltonian Mechanics
- Lagrangian Mechanics
- Port-Hamiltonian Systems

Choice depends on the use-case!

Physics model



[A. Van Der Schaft and D. Jeltsema]

Port-Hamiltonian system

Interconnection Dissipation Hamiltonian

$$\dot{x} = [J(x) - R(x)]\frac{\partial H}{\partial x} + G(x)u$$

$$y = G(x)^\top \frac{\partial H}{\partial x}$$

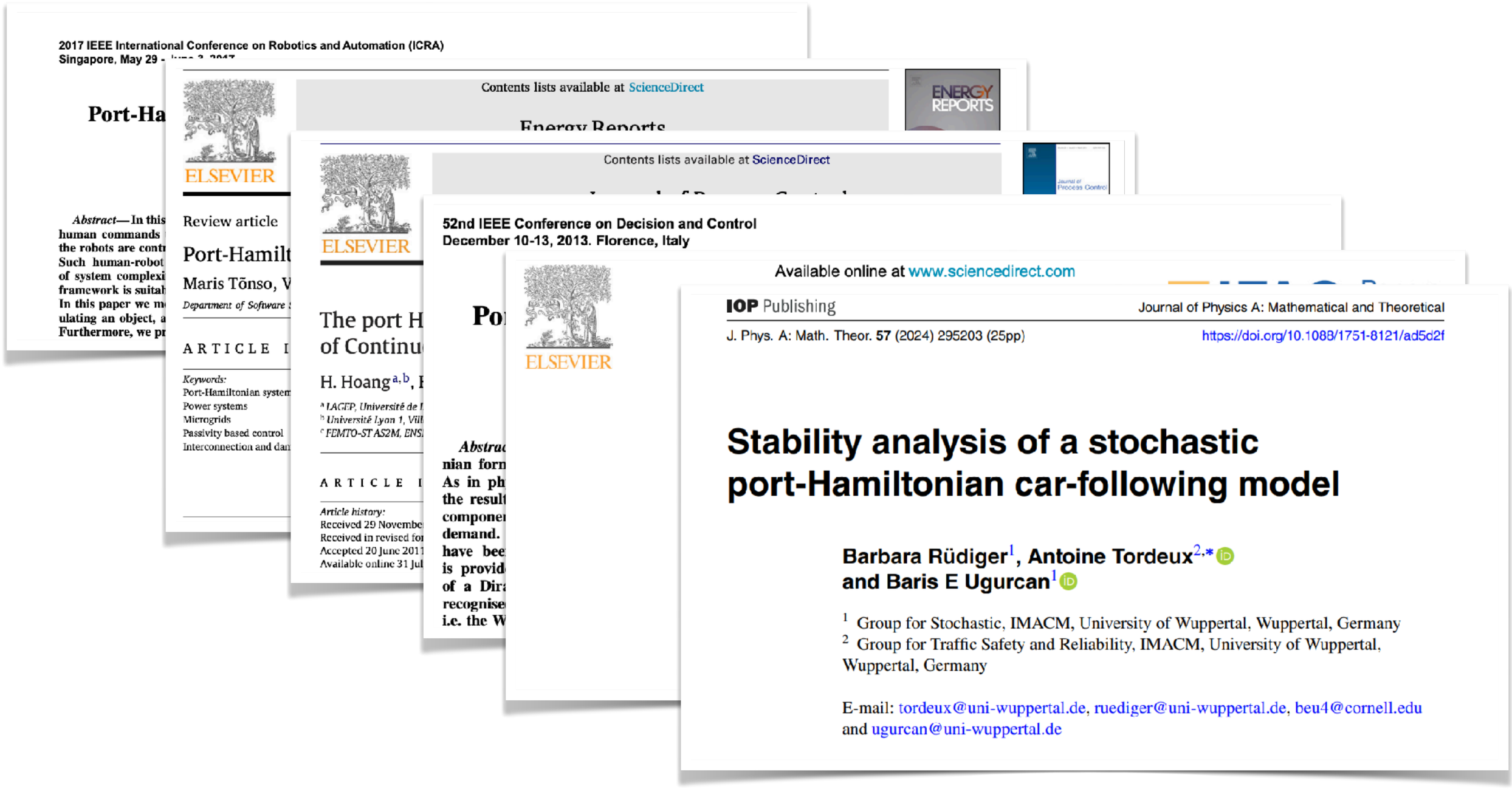
Input

Output

- Skew-symmetric J ensures energy conservation
- Input and output defines a passive systems
- Connection with another PHS preserves the structure
- Suitable for multi-domain applications via energy flows

A. Van Der Schaft and D. Jeltsema. "Port-Hamiltonian systems theory: An introductory overview." Foundations and Trends in Systems and Control. 2014

Port-Hamiltonian use-cases



Structured-learning approach

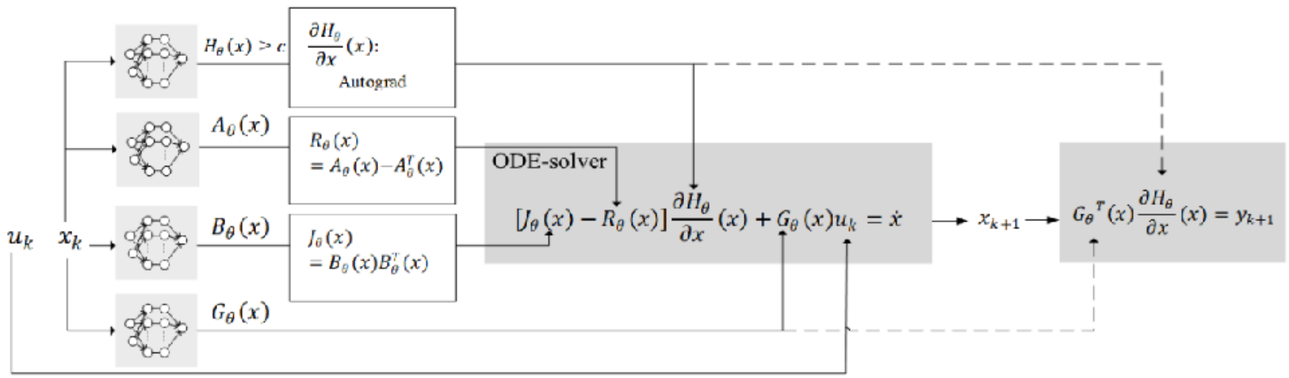
Port-Hamiltonian system

$$\dot{x} = [J(x) - R(x)]\frac{\partial H}{\partial x} + G(x)u$$

$$y = G(x)^\top \frac{\partial H}{\partial x}$$

Port-Hamiltonian Neural Networks

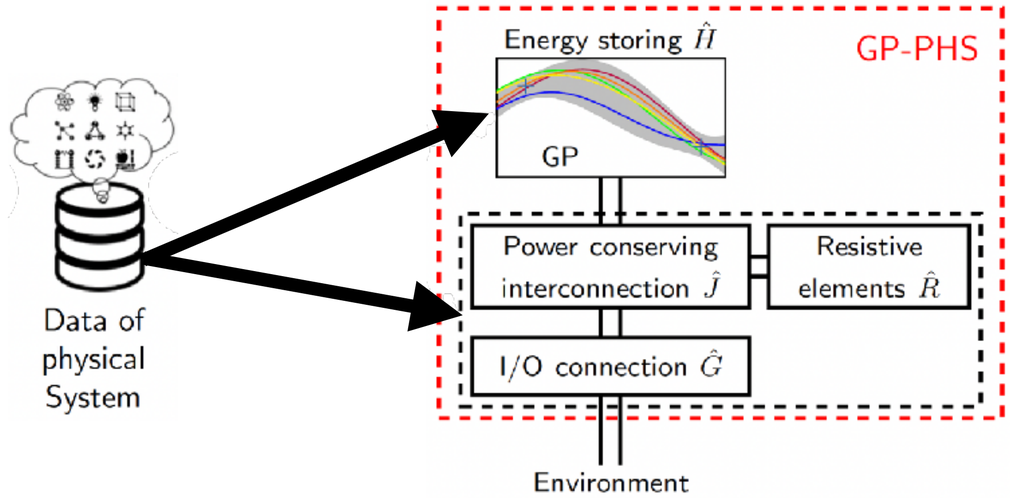
- Elements are approximated by a NN



[S. Moradi, et al.]

Gaussian Process Port-Hamiltonian systems

- Hamiltonian approximated by a GP



S. Moradi, et al. "Port-Hamiltonian Neural Networks with Output Error Noise Models." arXiv preprint arXiv:2502.14432, 2025
T. Beckers, et al. Gaussian Process Port-Hamiltonian Systems: Bayesian Learning with Physics Prior. CDC, 2022

Gaussian process

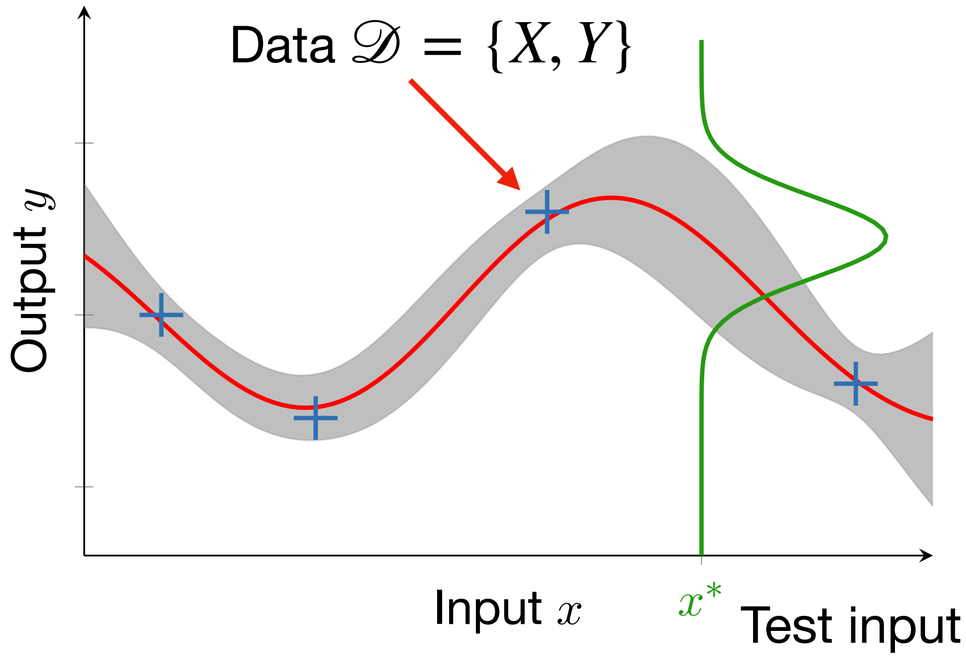
Prior: Gaussian distribution over function space

$$f(x) \sim GP(m(x), k(x, x'))$$

Mean function

Covariance

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}}$$



Posterior mean

Data points

Covariance

$$\mu(y \mid x, \mathscr{D}) = m(x) + \sum_{i}^{N} w_i k(x, X_i)$$

Prior mean function

Weighting factor includes X, Y

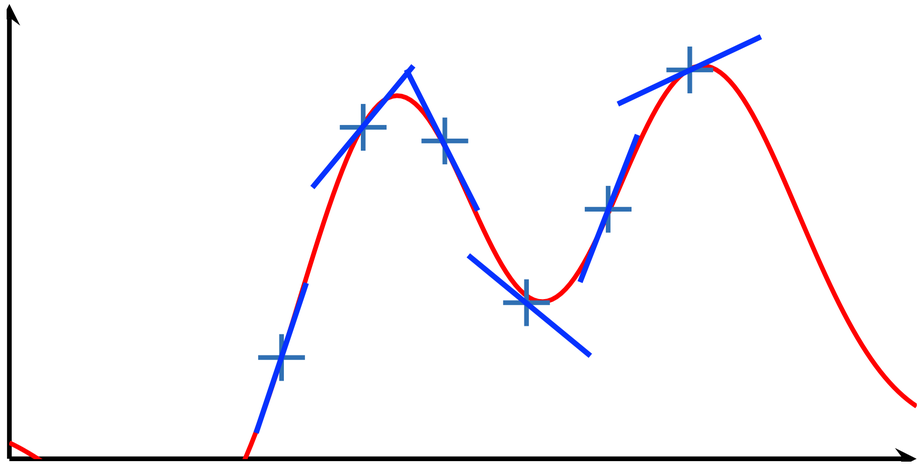
Linear operators

GPs are closed under linear operators

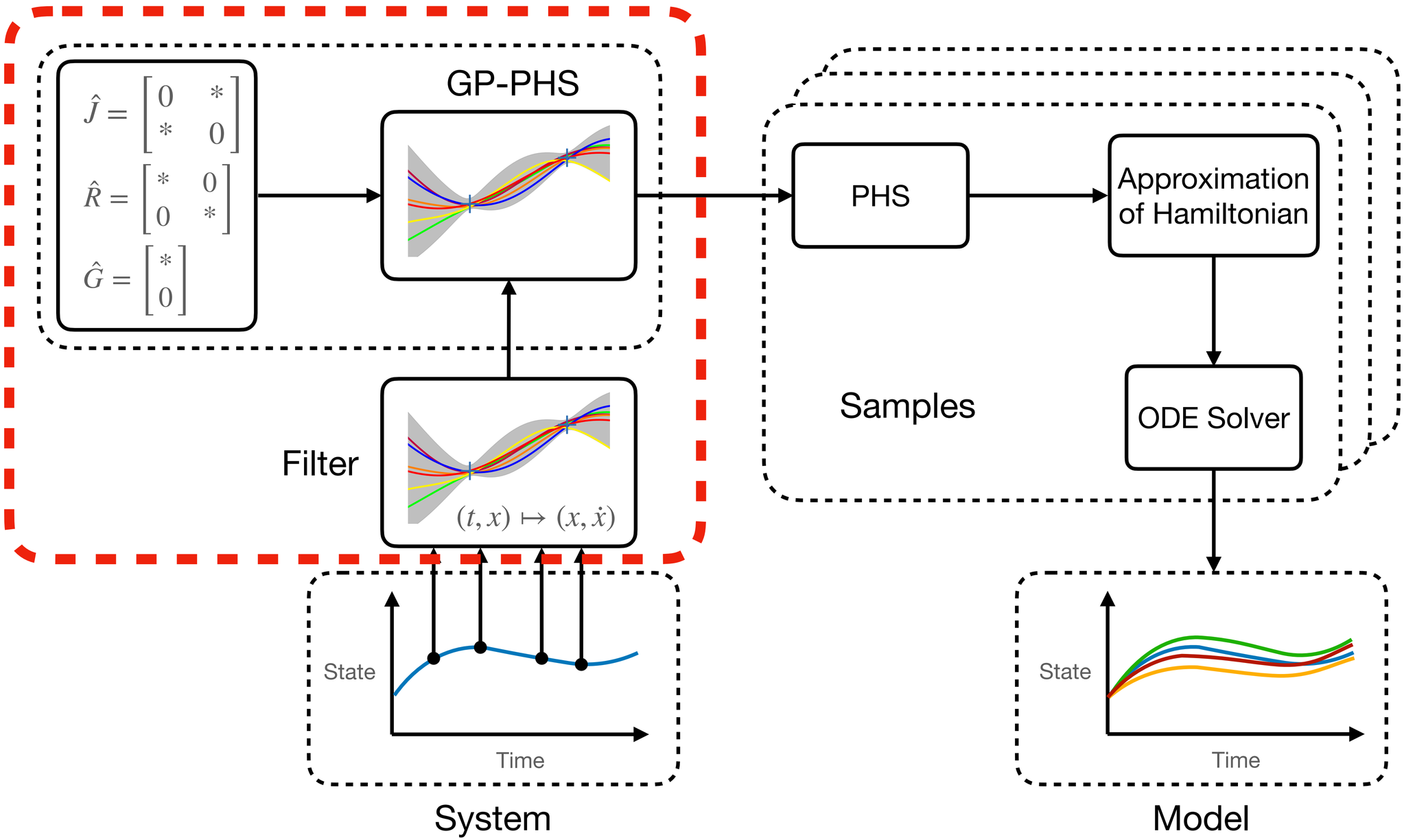
$$f(x) = \mathscr{L}_x g(x) \sim GP(\mathscr{L}_x \mu, \mathscr{L}_x k \mathscr{L}_x^\top)$$

Example: Differential operator $\mathscr{L}_x = \dfrac{\partial}{\partial x}$

$$\begin{bmatrix} y \\ Y' \end{bmatrix} = \mathcal{N}\left(0, \begin{bmatrix} k(x,x) & \frac{\partial}{\partial x} k(x,X) \\ \frac{\partial}{\partial x} k(x,X)^\top & \frac{\partial}{\partial x}^\top k(X,X) \frac{\partial}{\partial x} \end{bmatrix}\right)$$



GP-PHS



GP-PHS training

GP prior on Hamiltonian

$$\dot{x} = [J(x) - R(x)]\frac{\partial H}{\partial x} + G(x)u \quad \rightarrow \quad \dot{x} \sim GP(\hat{G}(x)u, k_{phs}(x, x'))$$

Linear operator

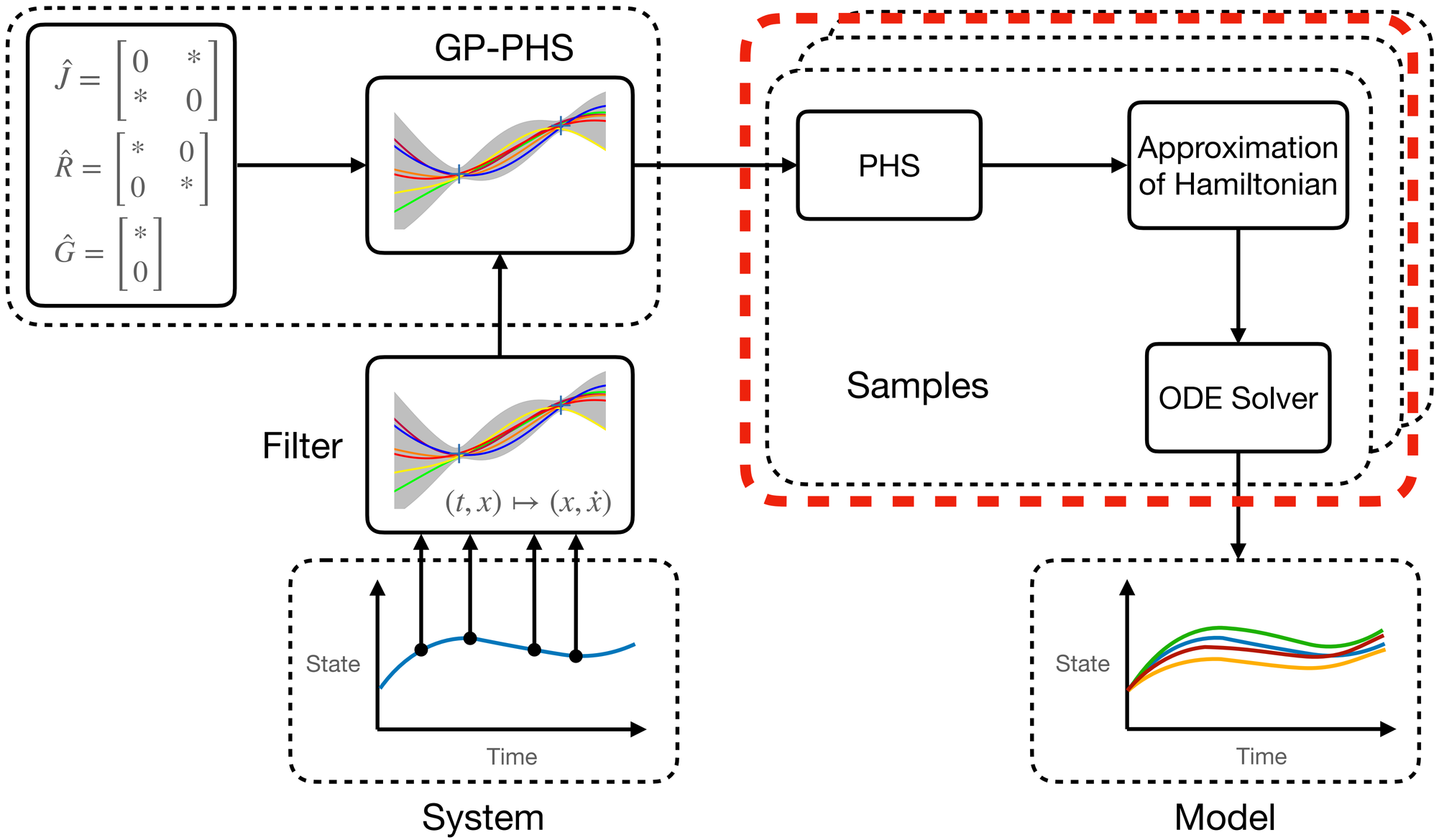
Port-Hamiltonian Kernel

$$k_{phs}(x, x') = \sigma_f^2 [\hat{J}(x) - \hat{R}(x)] \left[\frac{\partial}{\partial x \partial x'} \underbrace{\exp(-\|x - x'\|_\Lambda^2)}_{\text{Squared exp. kernel}} \right] \overbrace{[\hat{J}(x') - \hat{R}(x')]^\top}^{\text{Parametrized estimates}}$$

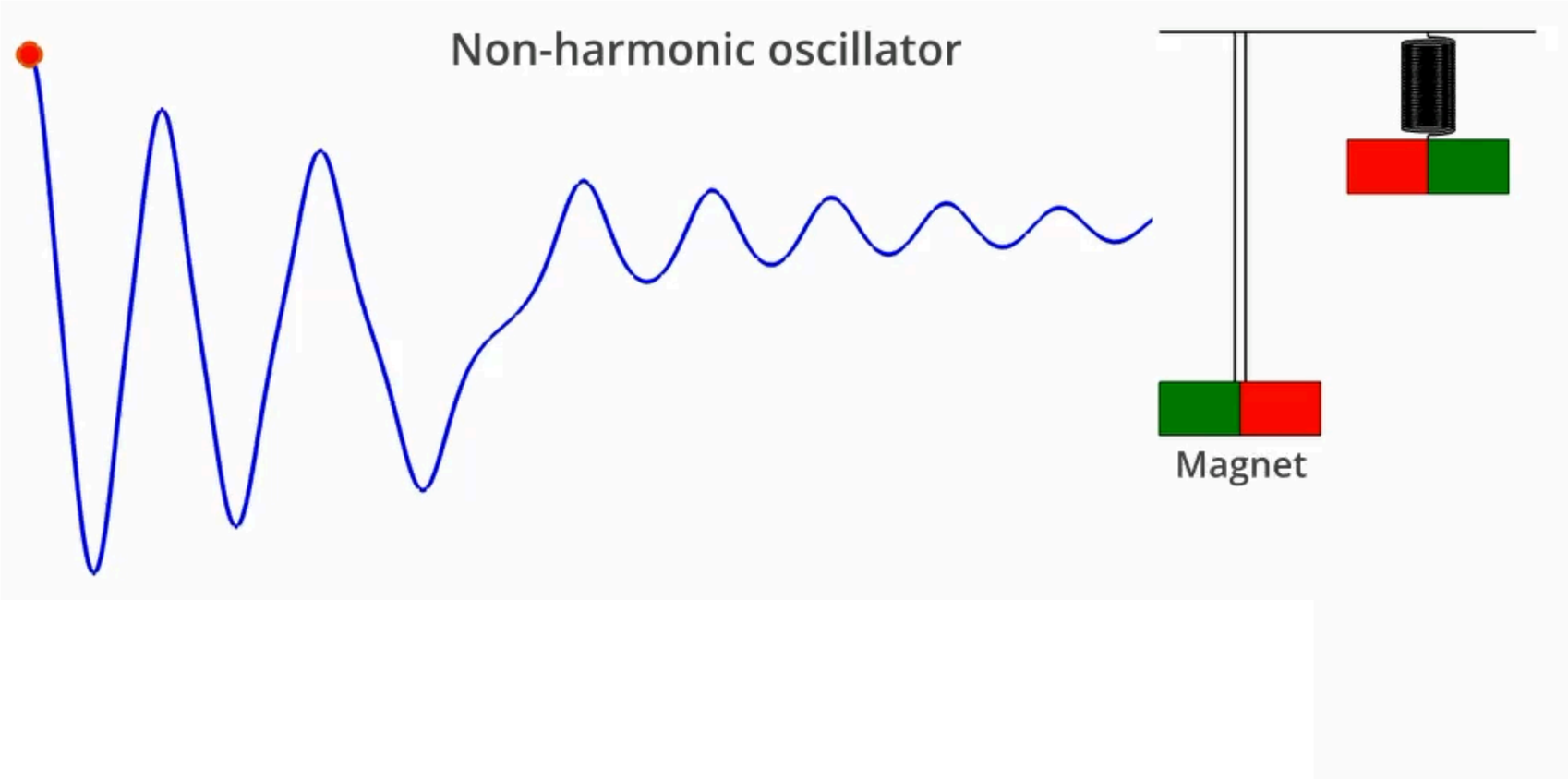
Hyperparameters $\varphi =$ Kernel parameters + unknown PHS parameters

Minimizing NLML $-\log p(\dot{X} \,|\, \varphi, X) \sim \underbrace{\dot{X}^\top [K_{phs} + \Delta]^{-1} \dot{X} + \log |\, K_{phs} + \Delta \,|}_{\text{Uncertainty of filter}}$

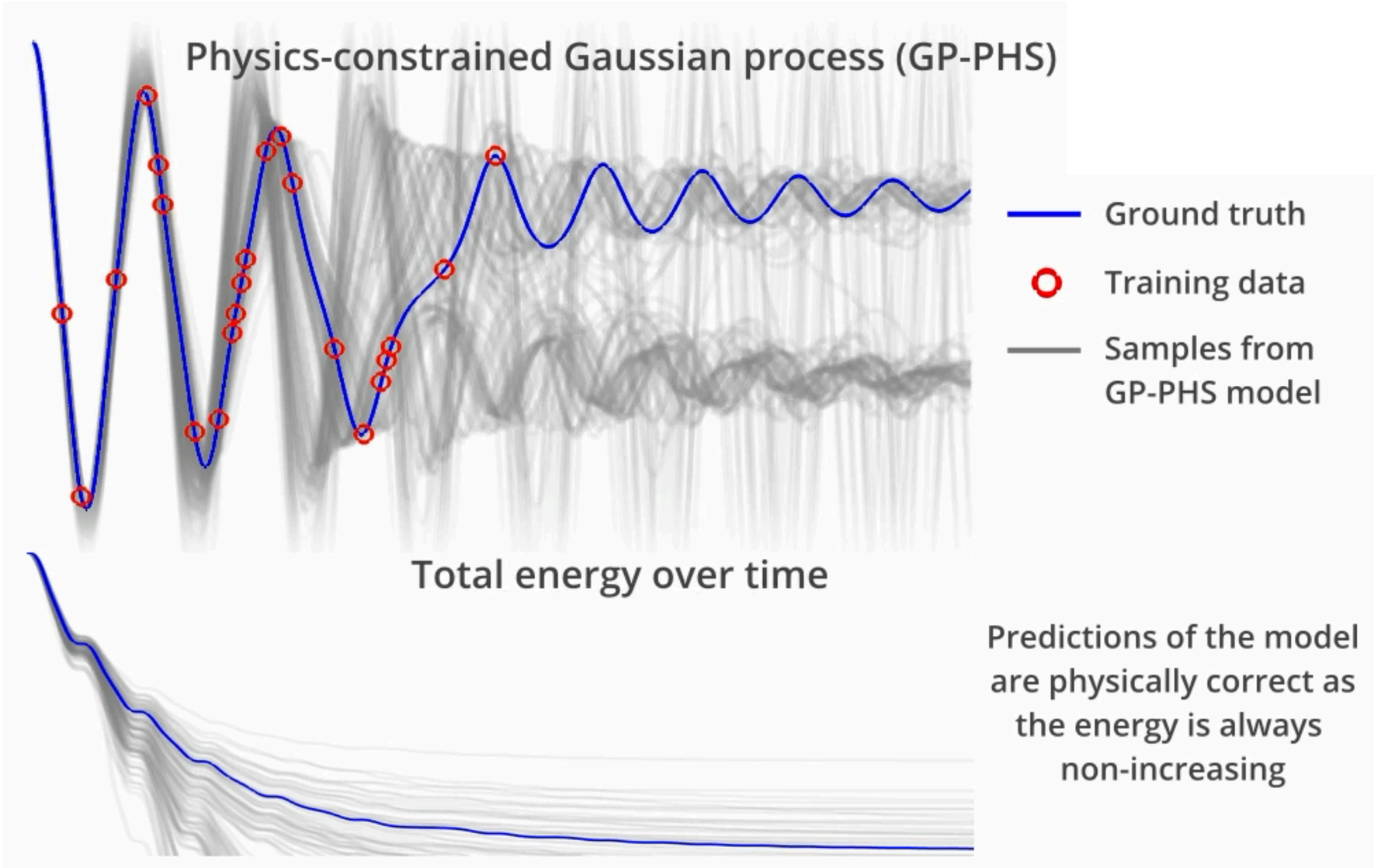
GP-PHS



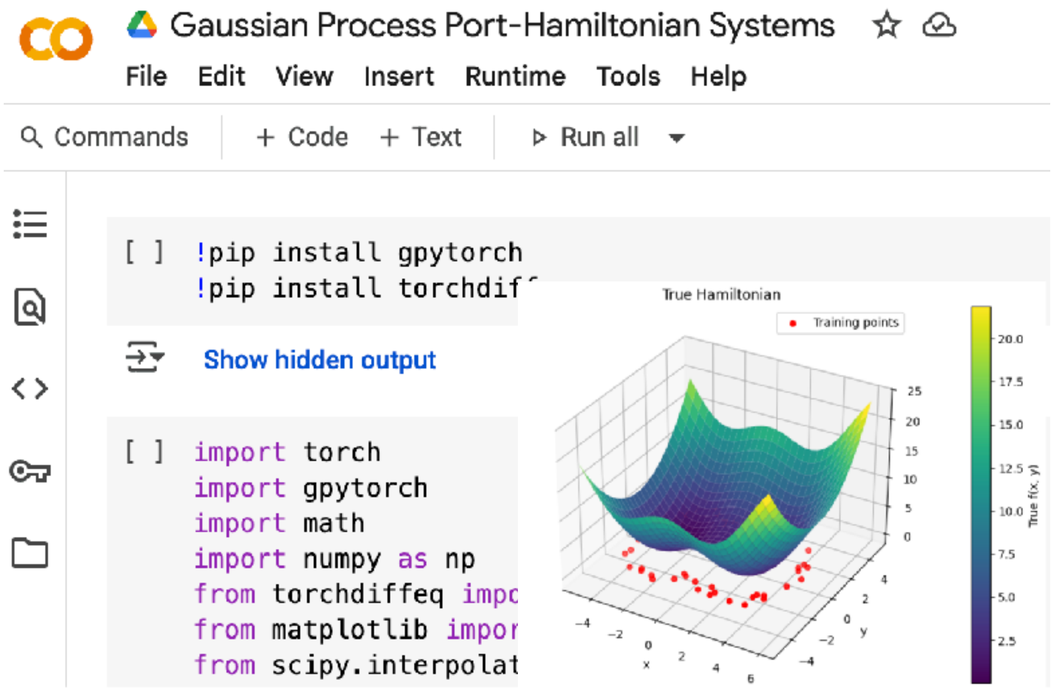
Example



Example


Physics-constrained Gaussian process (GP-PHS)
Ground truth
Training data
Samples from GP-PHS model
Total energy over time
Predictions of the model are physically correct as the energy is always non-increasing

Python Implementation



https://www.tbeckers.com/research

Implementation based on Torch and GPyTorch

- Use of Torch features such as GPU learning
- Tools from GPyTorch (speed up computation, etc.)
- PHS-Kernel implemented based on

RBFKernelGradGrad

class gpytorch.kernels.RBFKernelGradGrad(*ard_num_dims=None, batch_shape=None, active_dims=None, lengthscale_prior=None, lengthscale_constraint=None, eps=1e-06, **kwargs*) [source]

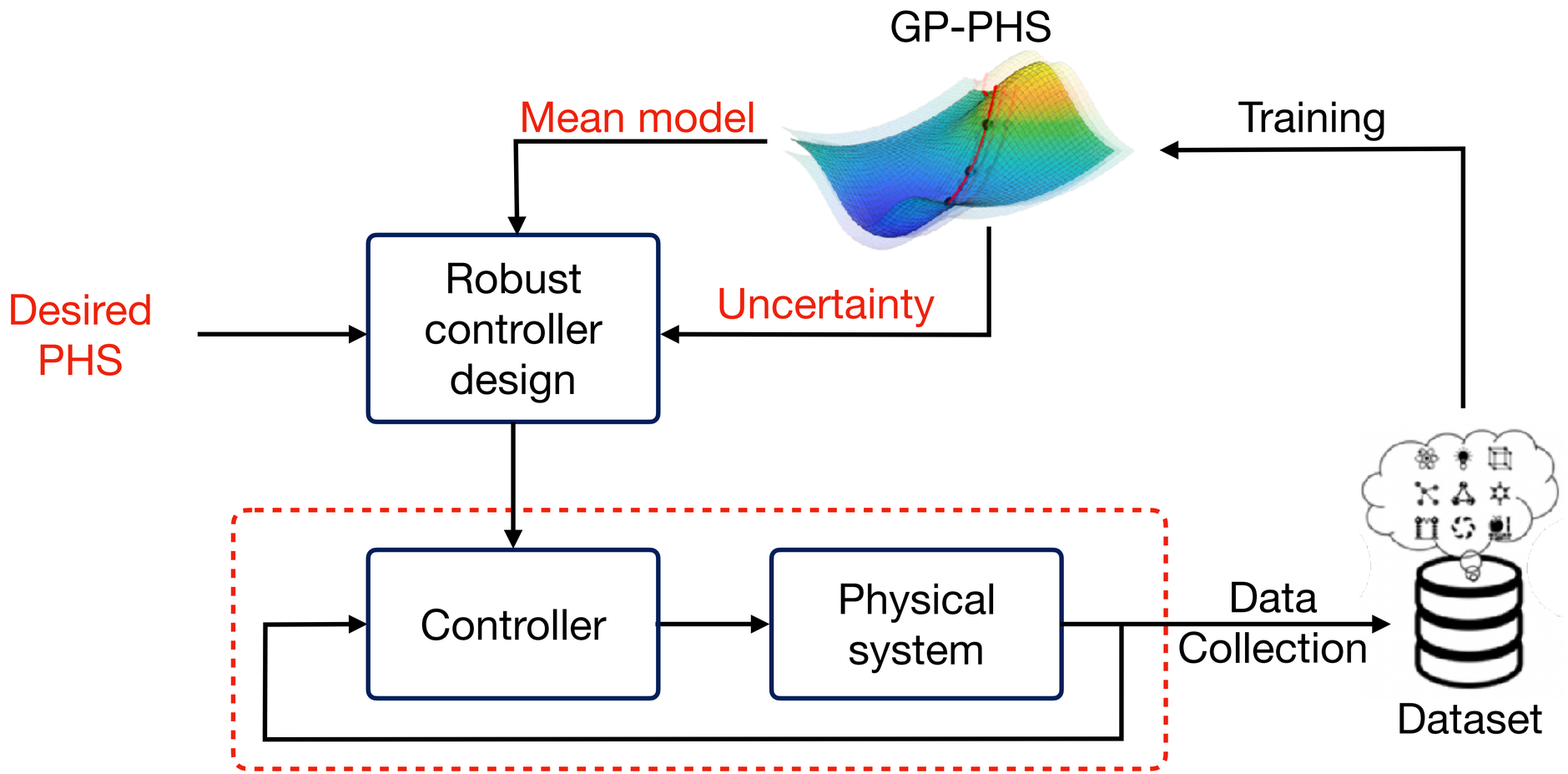
Computes a covariance matrix of the RBF kernel that models the covariance between the values and first and second (non-mixed) partial derivatives for inputs $\mathbf{x}_1$ and $\mathbf{x}_2$.

See `gpytorch.kernels.Kernel` for descriptions of the lengthscale options.

$$\begin{bmatrix} k(x,x) & \frac{\partial}{\partial x}k(x,X) \\ \frac{\partial}{\partial x}k(x,X)^\top & \frac{\partial}{\partial x}^\top k(X,X)\frac{\partial}{\partial x} \end{bmatrix}$$

Control

Design of a stabilizing controller for the physical system with unknown dynamics



Controller design

Closed-loop dynamics = Desired PHS

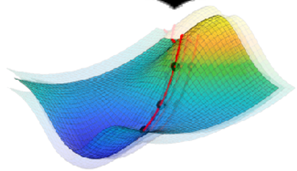
$$\dot{x} = [J_d(x) - R_d(x)]\nabla H_d$$

If satisfied

IDA-PBC PDE

Physical system | Desired PHS

$$G^{\perp}(x)[J(x) - R(x)]\nabla H = G^{\perp}(x)[J_d(x) - R_d(x)]\nabla H_d$$



GP-PHS model

Mean $\mu(x)$

Uncertainty bound $\eta(x)$

Stabilization criterion

$$G^{\perp}(x)\mu(x) = G^{\perp}(x)[J_d(x) - R_d(x)]\nabla H_d$$

Robustness criterion

$$[\nabla H_d]^{\top}\eta(x) \leq [\nabla H_d]^{\top}R_d(x)\nabla H_d$$

Closed-loop system with desired equilibrium —more data, less uncertainty→ Closed-loop system with desired PHS dynamics

GP-PHS for PDEs

Goal: Learning the dynamics of a vibrating string with unknown stress-strain curve

$$\frac{\partial^2 x}{\partial t^2} - f\left(\frac{\partial x}{\partial z}\right)\frac{\partial^2 x}{\partial z^2} = 0,$$

In PHS form

Prior knowledge

$$\frac{\partial}{\partial t}\begin{bmatrix} p(t,z) \\ q(t,z) \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial z} & 0 \end{bmatrix}}_{J-R} \delta_{pq}\mathscr{H}(p,q)$$

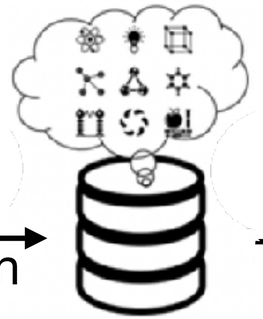
Training data

16 spatial points every 4ms

Control of PDEs

Design of an MPC controller for a physical PDE system with unknown dynamics

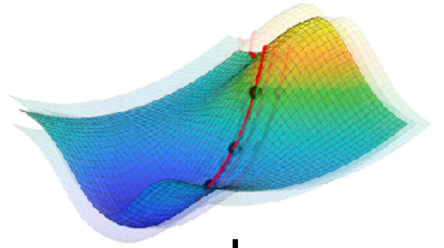
Data Collection



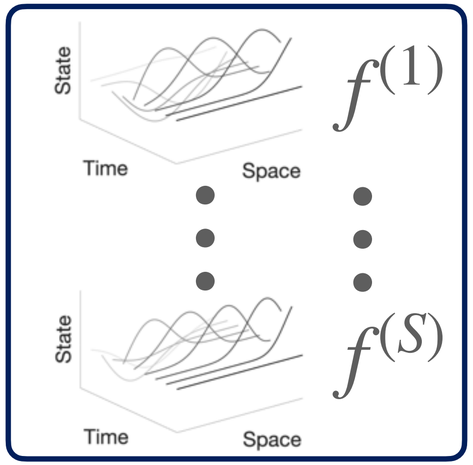
Dataset

Training

GP-dPHS



Samples



Discreti-zation

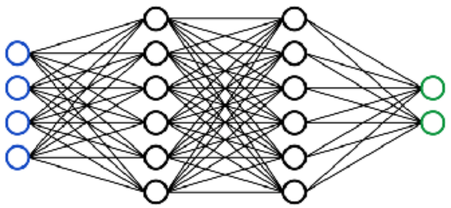
Scenario-based MPC

$$\min_{\{u_{k|t}\}_{k=0}^{N-1}} \sum_{i=1}^{S} \sum_{k=0}^{N-1} \ell\left(x_{k|t}^{(i)}, u_{k|t}\right)$$

$$\text{s.t. } x_{k+1|t}^{(i)} = f^{(i)}\left(x_{k|t}^{(i)}, u_{k|t}\right)$$

Imitation

NN controller



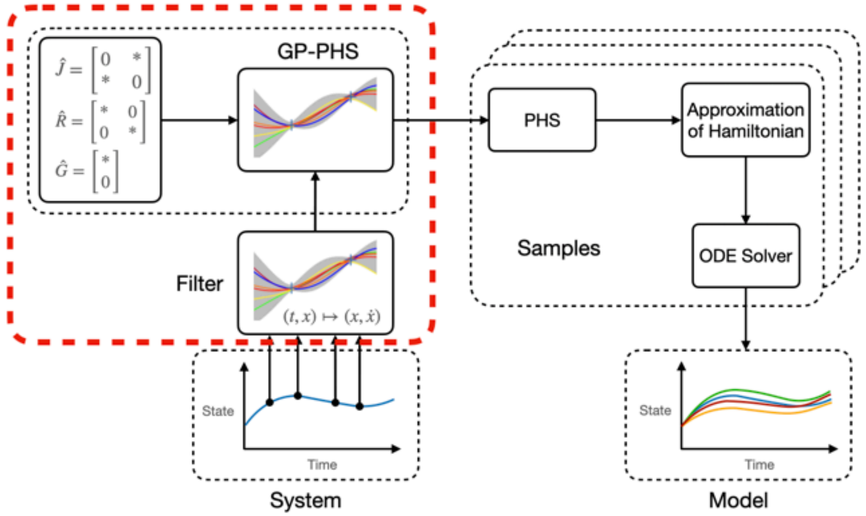
Boundary control

Conclusion

Structure-preserved learning

- Using data to “fill” a physics structure such as PHS

- GP-PHS model is physically consistent and enables UQ
- Preserve the structure and passivity characteristic
- Robust control of physical system based on GP-PHS



www.tbeckers.com **thomas.beckers@vanderbilt.edu**