VANDERBILT UNIVERSITY

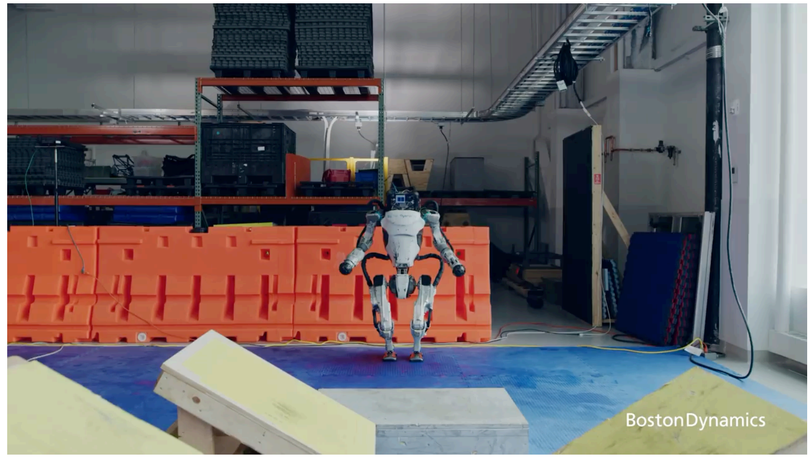
PyGpPHs: A Python Package for Bayesian Modeling of Port-Hamiltonian Systems

Peilun (Tommy) Li, Kaiyuan Tan, Thomas Beckers

Department of Computer Science, Vanderbilt University, USA

16th World Congress on Computational Mechanics and 4th Pan American Congress on Computational Mechanics, July 24, 2024

Motivation



[Boston Dynamics]

[Festo]

Modeling and control of electromechanical systems become more challenging

Need for expressive, reliable and well-generalizing modeling tools

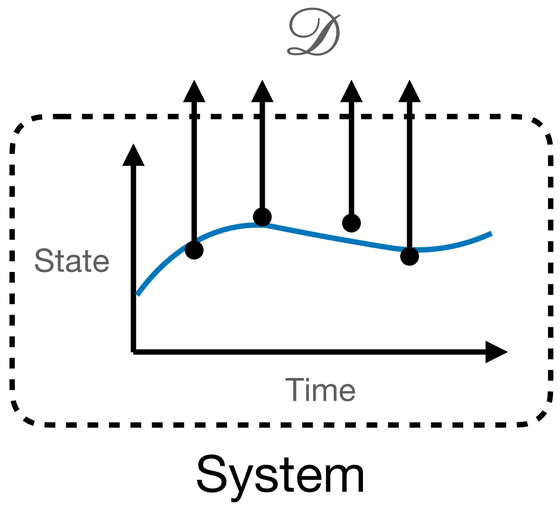
Problem statement

unknown

(Electro-)mechanical system $\dot{x}(t) = f(x) + G(x)u(t)$

Noisy measurements $\tilde{x}_i = x_i + \nu, \quad \nu \sim \mathcal{N}(0, \sigma^2 I)$

Dataset $\mathscr{D} = \{(t_1, \tilde{x}_1, u_1), \ldots, (t_N, \tilde{x}_N, u_N)\}$



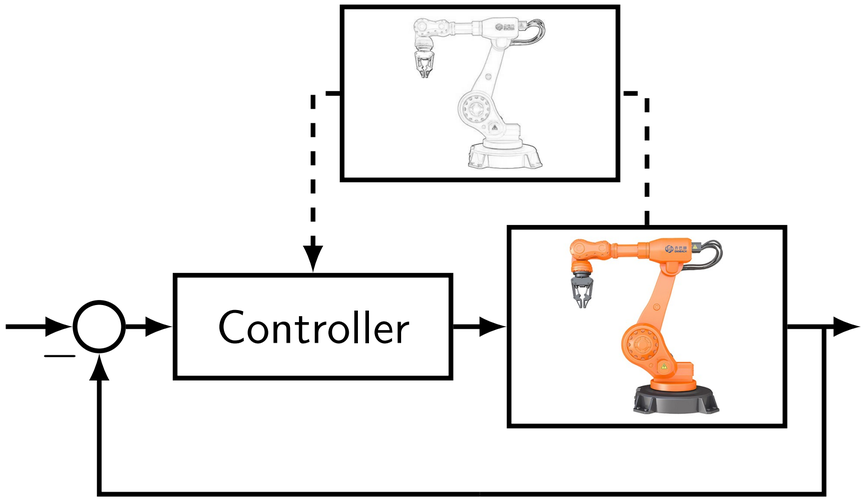
Goal

Given the dataset $\mathscr{D}$, we aim to find a model toolbox that

- represents a large class of systems -> Expressive
- allows uncertainty quantification -> Reliable
- consistent with physics -> Generalizes well

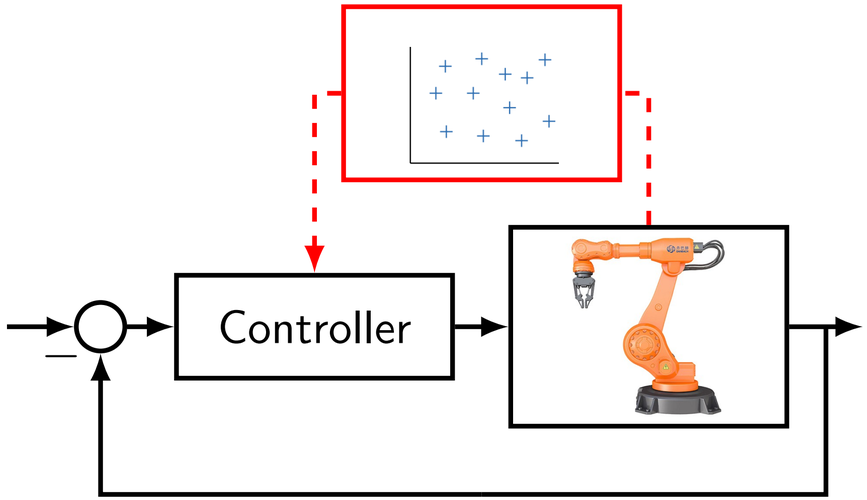
Modeling and control of physical systems

Physics-based



- ✓ Generalization
- ✓ Physical correctness
- ✗ Model selection
- ✗ Time-consuming

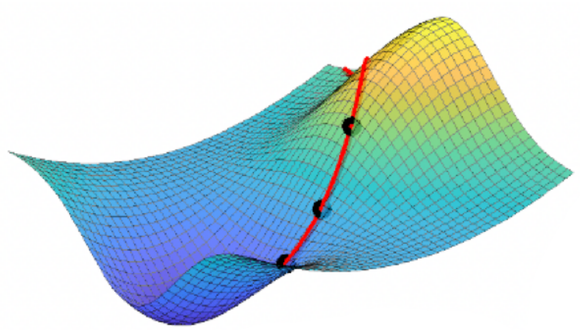
Learning-based



- ✓ Highly flexible
- ✓ Minimal expert knowledge
- ✗ Generalization
- ✗ Physical correctness

Port-Hamiltonian systems + Gaussian Processes

Physics model



Port-Hamiltonian systems

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

Gaussian process

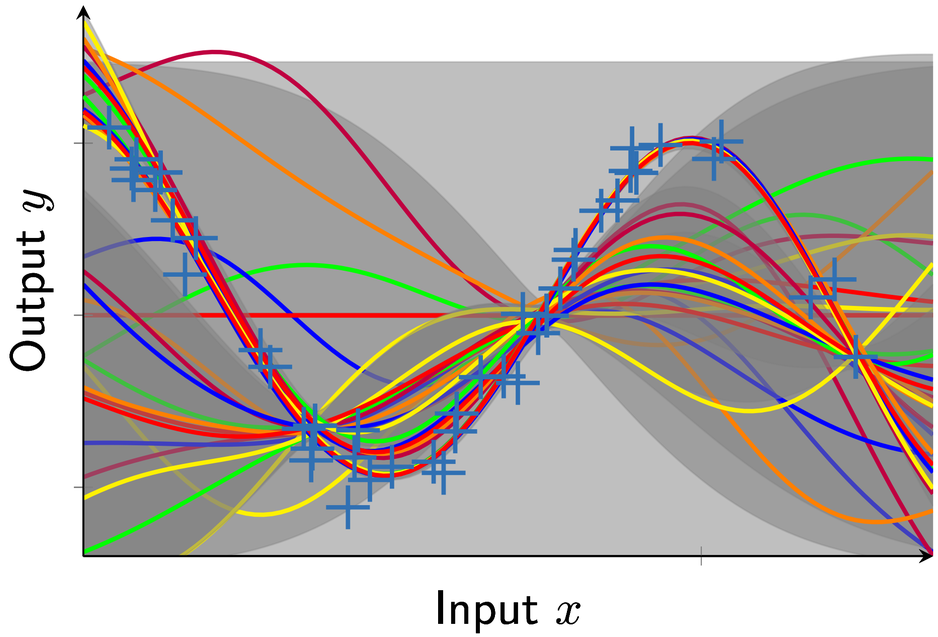
Prior: Gaussian distribution over function space

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

Mean function

Covariance

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



Gaussian process

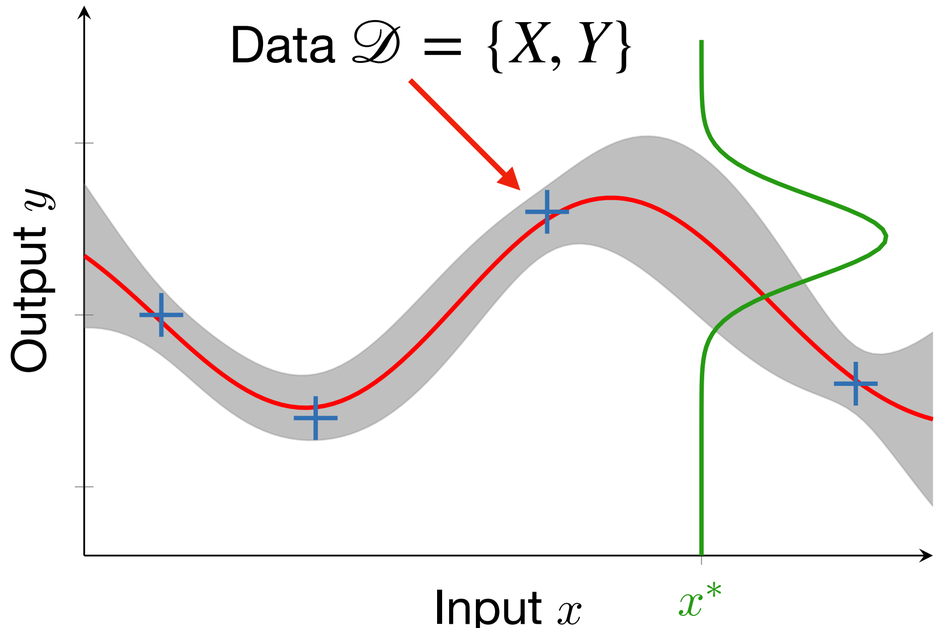
Prior: Gaussian distribution over function space

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

Mean function

Covariance

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



Posterior

$$\mu(y^* \,|\, x^*, \mathscr{D}) = k(x^*, X)[K(X, X) + \sigma^2 I]^{-1} Y$$

$$\Sigma(y^* \,|\, x^*, \mathscr{D}) = k(x^*, x^*) - k(x^*, X)[K(X, X) + \sigma^2 I]^{-1} k(x^*, X)^\top$$

C.E. Rasmussen. Gaussian Processes for Machine Learning. 2006

Gaussian process

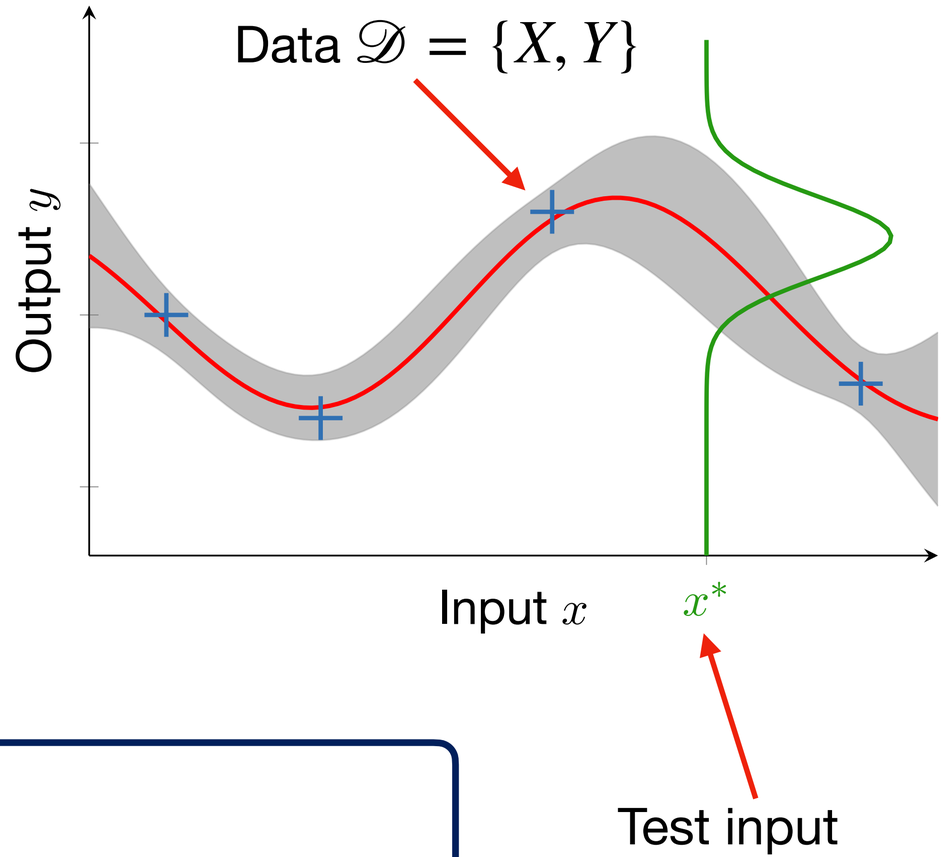
Prior: Gaussian distribution over function space

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

Mean function

Covariance

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

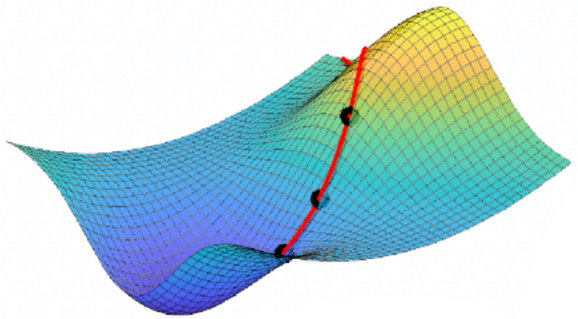


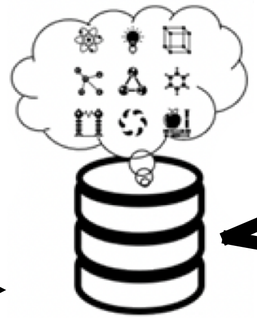
Closed under linear operators

$$f(x) = \mathscr{G}_x g(x) \sim GP(\mathscr{G}_x \mu_g, \mathscr{G}_x k_g \mathscr{G}_x^\top)$$

C.E. Rasmussen. Gaussian Processes for Machine Learning. 2006

GP-PHS

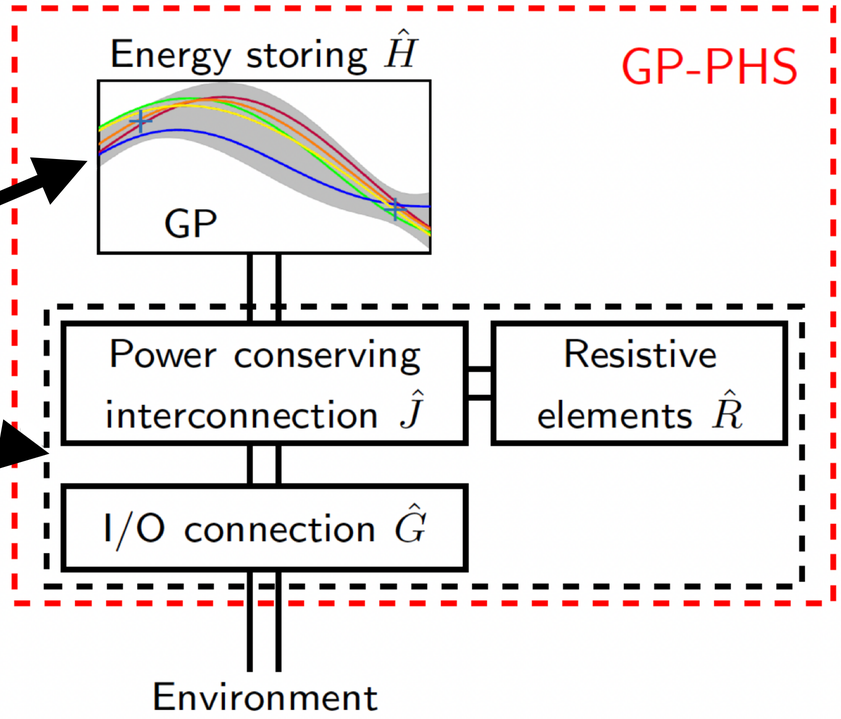




Data of physical System

Nonparametric modeling

Parameter estimation



- Gaussian Process model to learn the Hamiltonian
- GP-PHS includes all possible PHS under the GP prior on the Hamiltonian

PyGpPHs Toolbox

Python Gaussian Process port-Hamiltonian System

User-friendly interface

Object-oriented programming

Analogous to popular ML toolboxes

Fast & accurate computation

Acceleration of expensive operations

C++ Eigen library

Efficient GP algorithm

GPyTorch

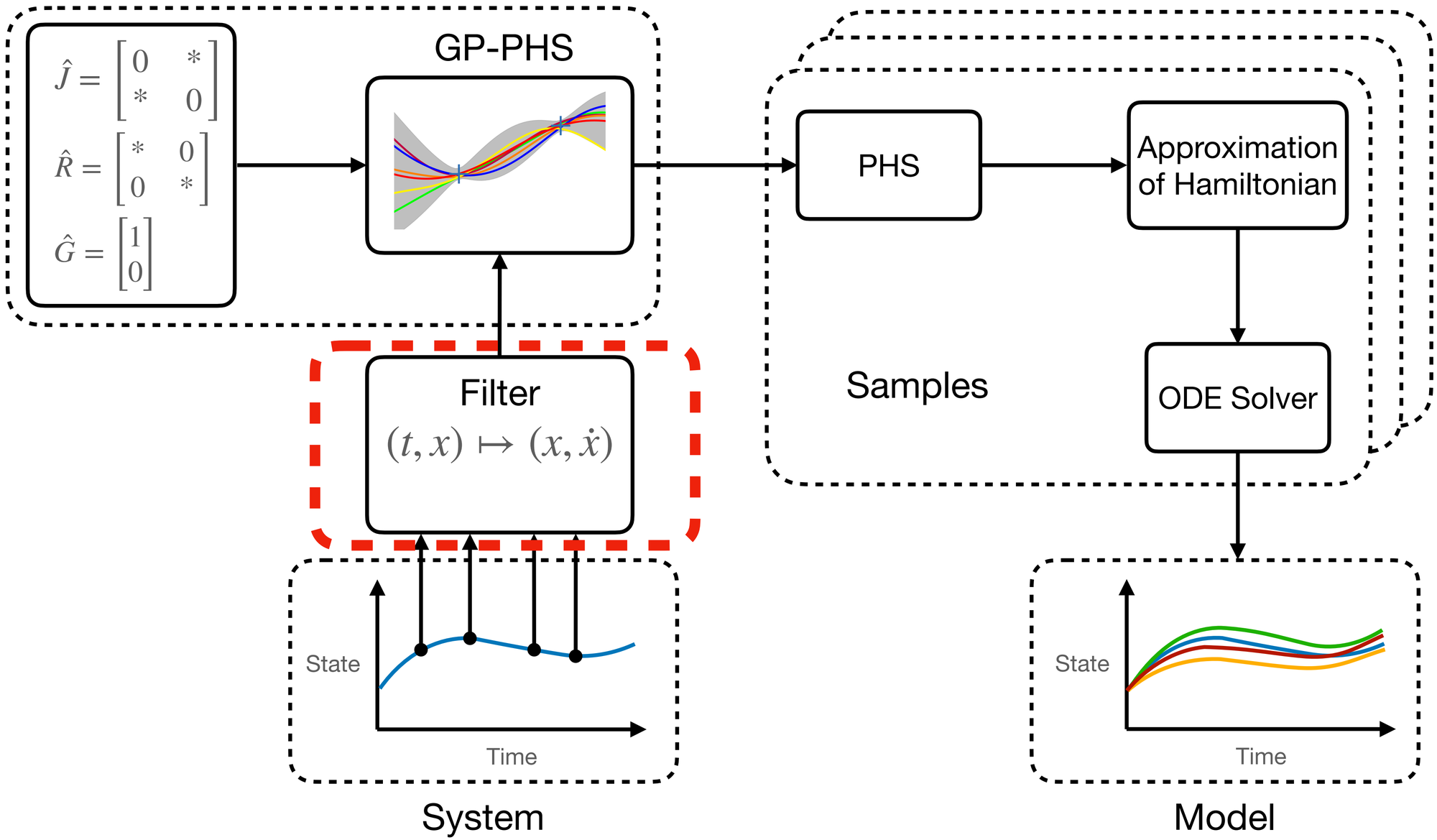
CPU & GPU support

Support

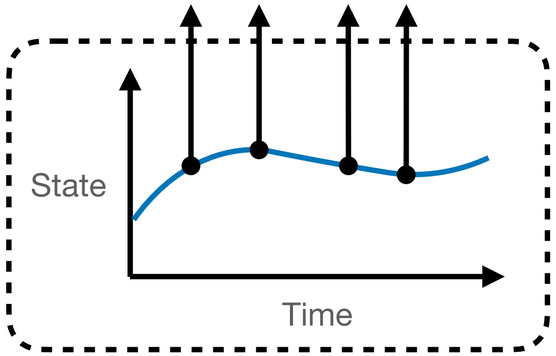
Comprehensive documentation

maintenance, new features, …

GP-PHS


$\hat{J} = \begin{bmatrix} 0 & * \\ * & 0 \end{bmatrix}$
$\hat{R} = \begin{bmatrix} * & 0 \\ 0 & * \end{bmatrix}$
$\hat{G} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$
GP-PHS
Filter
$(t, x) \mapsto (x, \dot{x})$
State
Time
System
PHS
Approximation of Hamiltonian
ODE Solver
Samples
State
Time
Model

1. Initialization

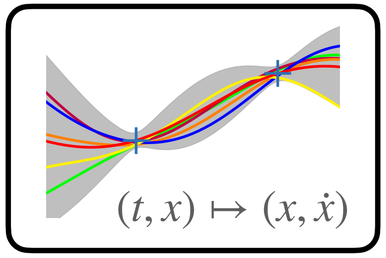


Time stamps $T = [t_1, \ldots, t_N]$

State observations $X = [\tilde{x}_1, \ldots, \tilde{x}_N]$

```
model = Model(T, X, dX=None, G=G, u=u)
```

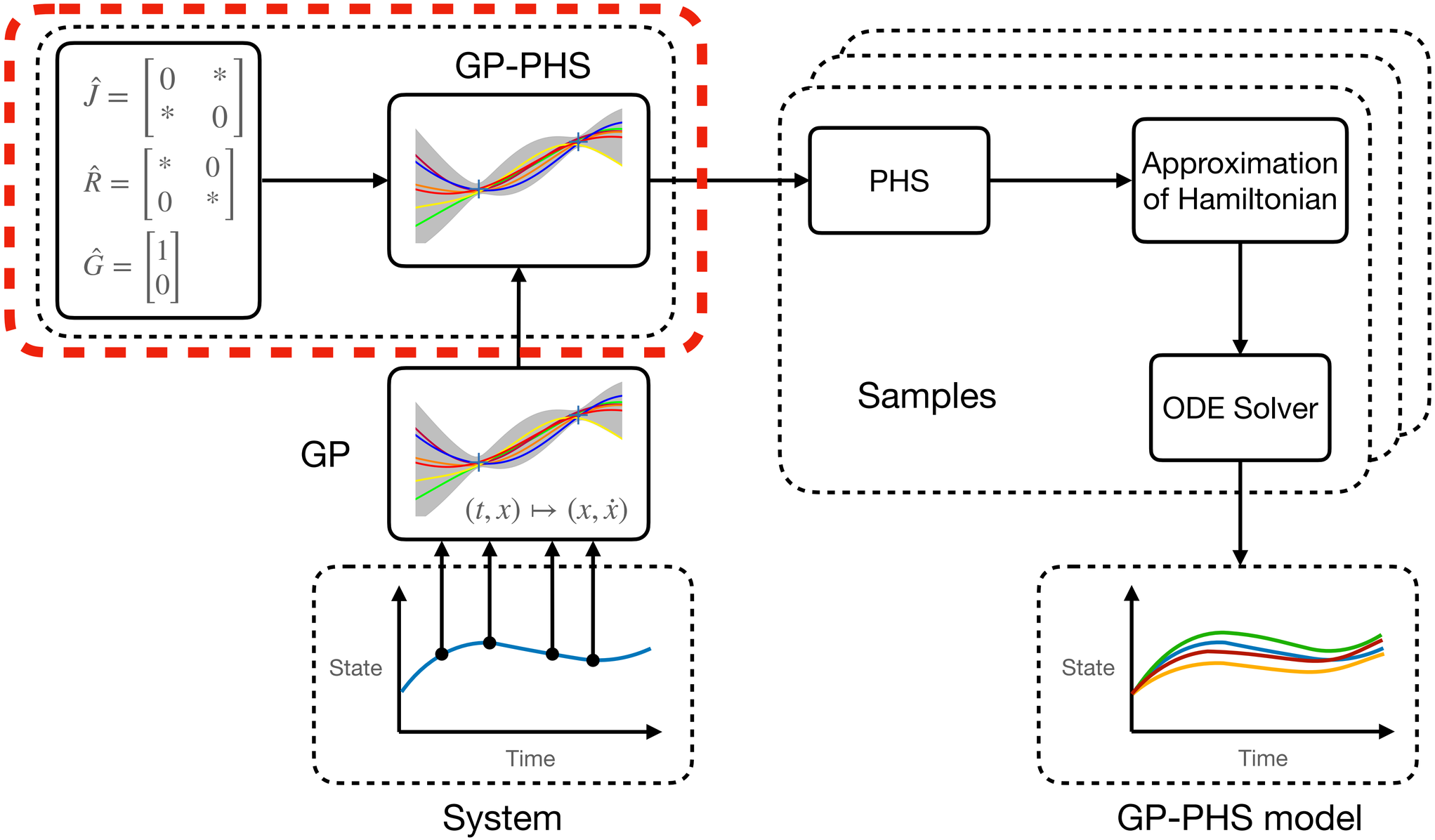
GP



$$\mu(x \mid t, \mathscr{D}) = k(t, T)K(T, T)^{-1}Y \qquad \mu(\dot{x} \mid t, \mathscr{D}) = \frac{\partial}{\partial t}k(t, T)K(T, T)^{-1}Y$$

$$\mathscr{D}_x = \{x(t_i), \dot{x}(t_i)\}_{i=1,\ldots,N}$$

GP-PHS


$\hat{J} = \begin{bmatrix} 0 & * \\ * & 0 \end{bmatrix}$
$\hat{R} = \begin{bmatrix} * & 0 \\ 0 & * \end{bmatrix}$
$\hat{G} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$
GP-PHS
GP
$(t, x) \mapsto (x, \dot{x})$
State
Time
System
PHS
Approximation of Hamiltonian
Samples
ODE Solver
State
Time
GP-PHS model

2. 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

Parametrized estimates

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

Optimizing parameters by means of likelihood function $p(\dot{x} \,|\, x, \sigma_f, \Lambda, \hat{J}, \hat{R}, \hat{G})$

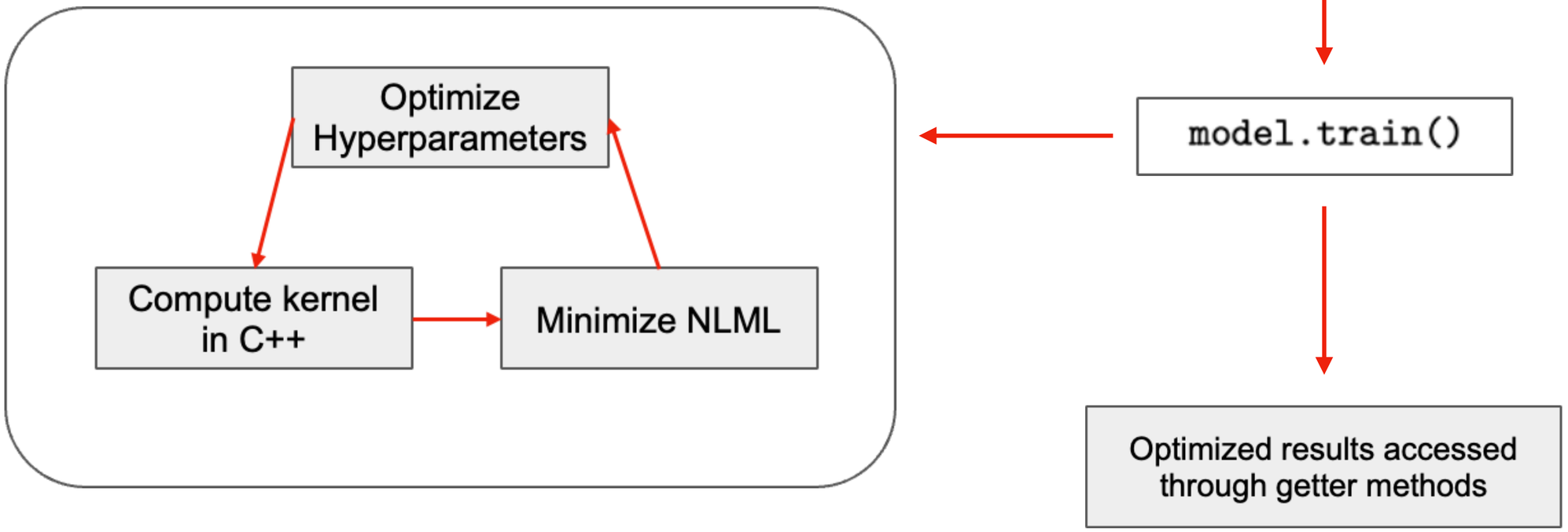
PHS kernel creates distribution over PHS

2. Training

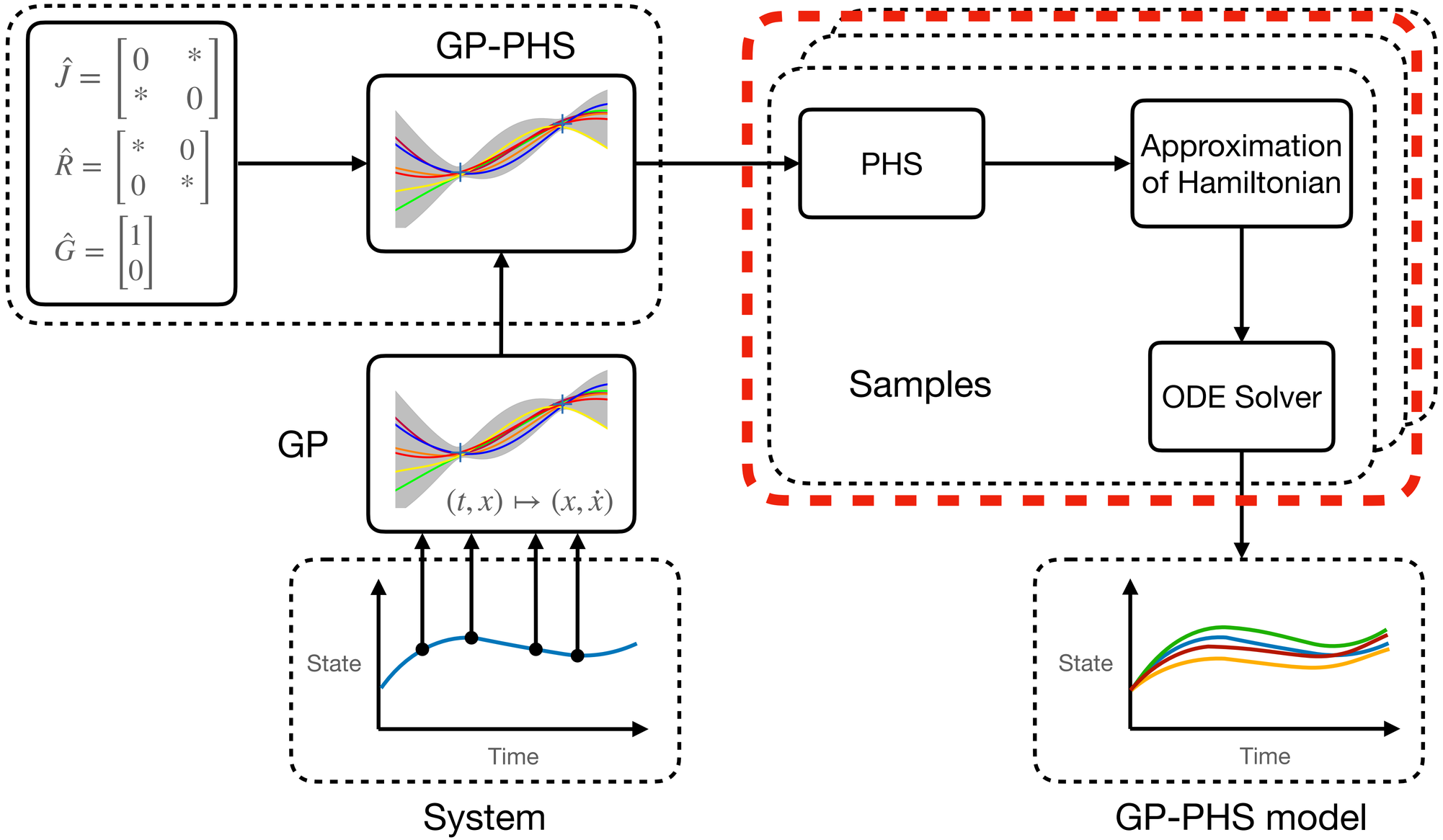
Estimated parameters

$$\hat{J} - \hat{R} = \begin{bmatrix} jr_{11} & jr_{12} \\ -jr_{12} & jr_{22} \end{bmatrix}$$

```
hpr=Hyp(JR=[[jr11, jr12, jr22],to_optimize])
model.set_Hyp(hpr)
```



GP-PHS


$\hat{J} = \begin{bmatrix} 0 & * \\ * & 0 \end{bmatrix}$
$\hat{R} = \begin{bmatrix} * & 0 \\ 0 & * \end{bmatrix}$
$\hat{G} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$
GP-PHS
GP
$(t, x) \mapsto (x, \dot{x})$
State
Time
System
PHS
Approximation of Hamiltonian
Samples
ODE Solver
State
Time
GP-PHS model

3. Prediction

PHS

$$\begin{bmatrix} \dot{X} \\ \hat{f}(x^*) \end{bmatrix} = \mathcal{N}\left(0, \begin{bmatrix} K_{phs} & k_{phs}(X, x^*) \\ k_{phs}(X, x^*)^\top & k_{phs}(x^*, x^*) \end{bmatrix}\right)$$

$$\dot{x} = \hat{f}(x, \omega)$$

Not callable
(computational expensive)

Approximation of Hamiltonian

$$\begin{bmatrix} \dot{X} \\ \hat{H}(x^*) \end{bmatrix} = \mathcal{N}\left(0, \begin{bmatrix} K_{phs} & k_{\dot{x}H}(X, x^*) \\ k_{\dot{x}H}(X, x^*)^\top & k_{HH}(x^*, x^*) \end{bmatrix}\right)$$

$$\hat{H}(x^*, \omega) \xrightarrow{\text{Approx.}} \hat{H}^*(x^*)$$

ODE Solver

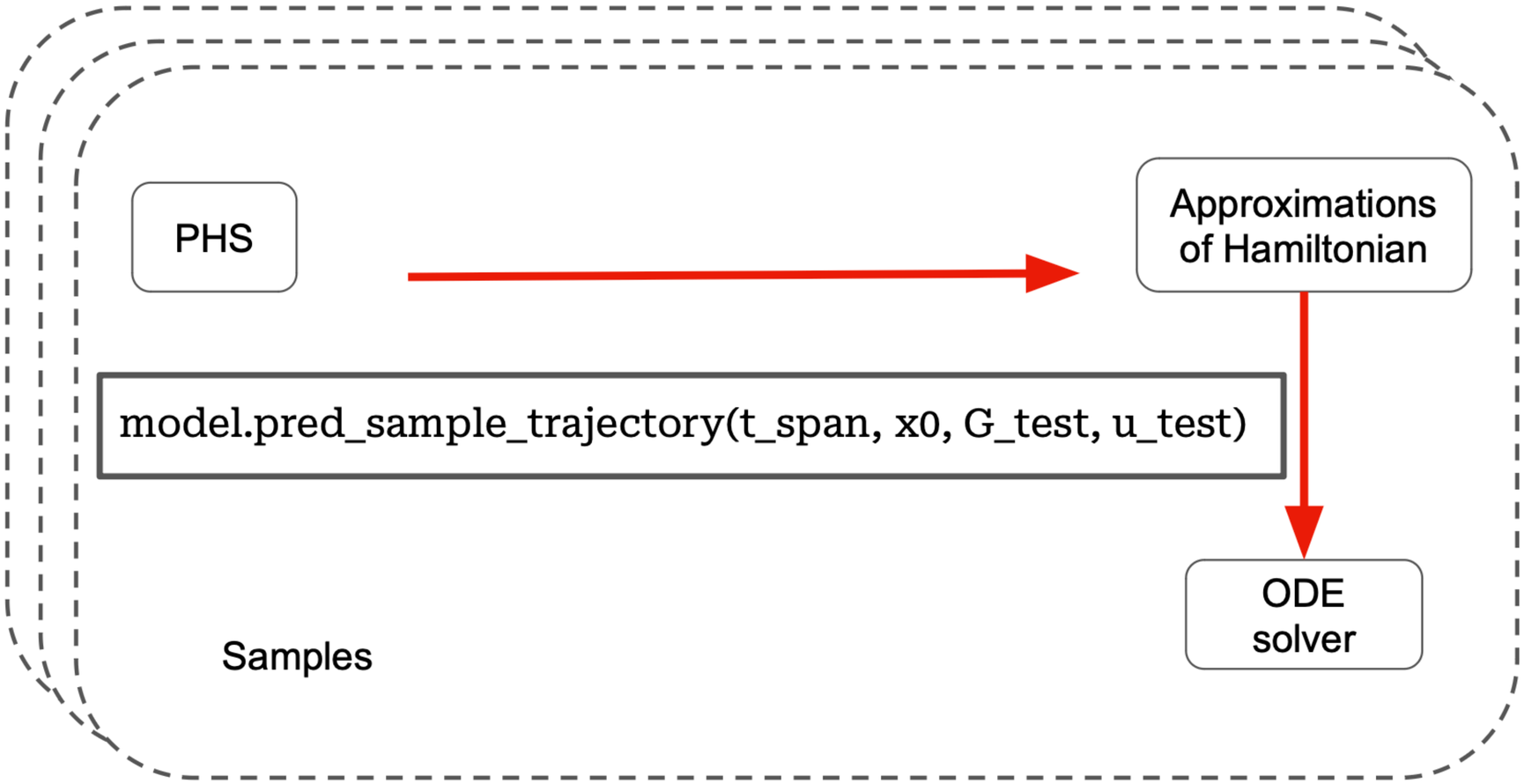
$$\dot{x} = [\hat{J}(x) - \hat{R}(x)]\frac{\partial \hat{H}^*}{\partial x} + \hat{G}(x)u \xrightarrow{\text{Solver}} x(t)$$

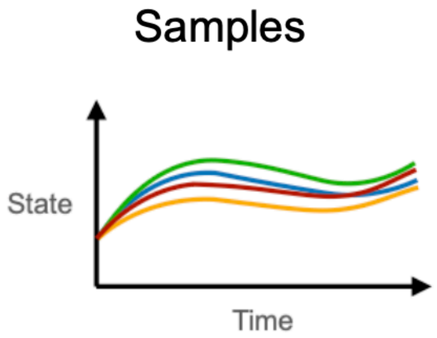
Approximation of H: Callable and structure preserving

3. Prediction

Time stamps $T_{span} = [t_1, \ldots, t_N]$

Initial value x_0, test input $u_{test}(t)$

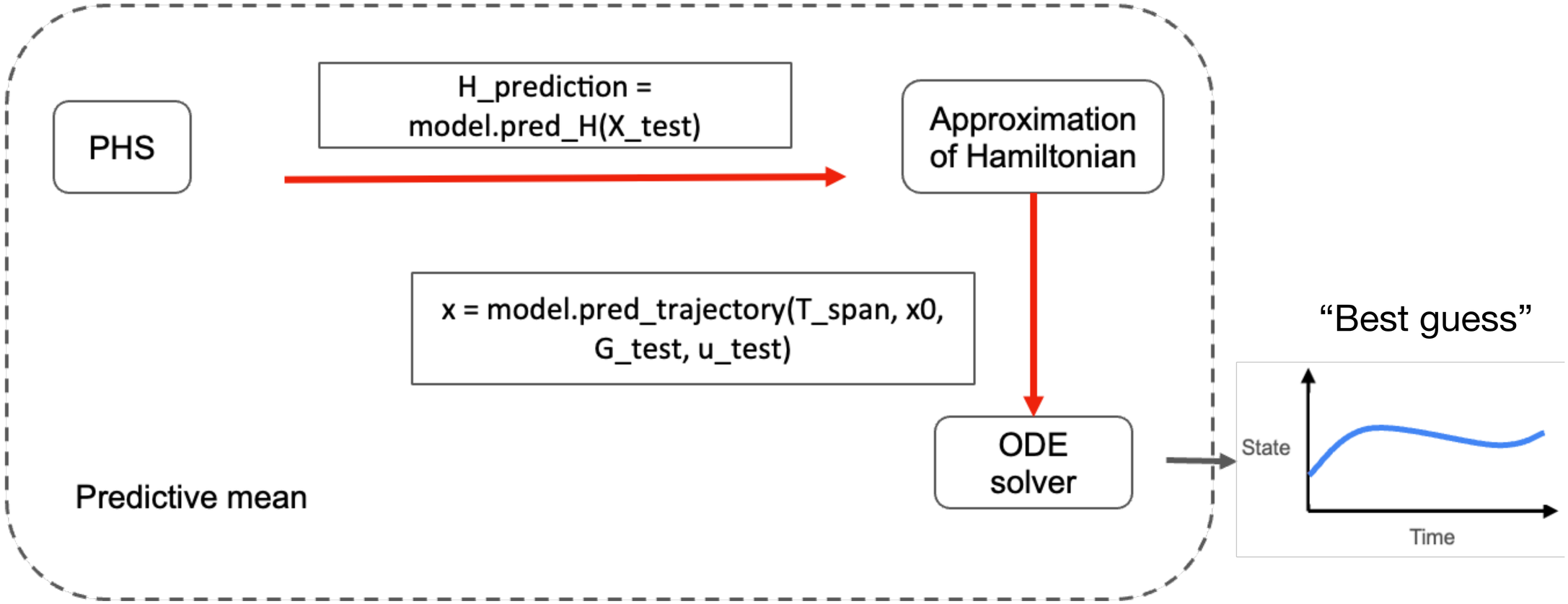




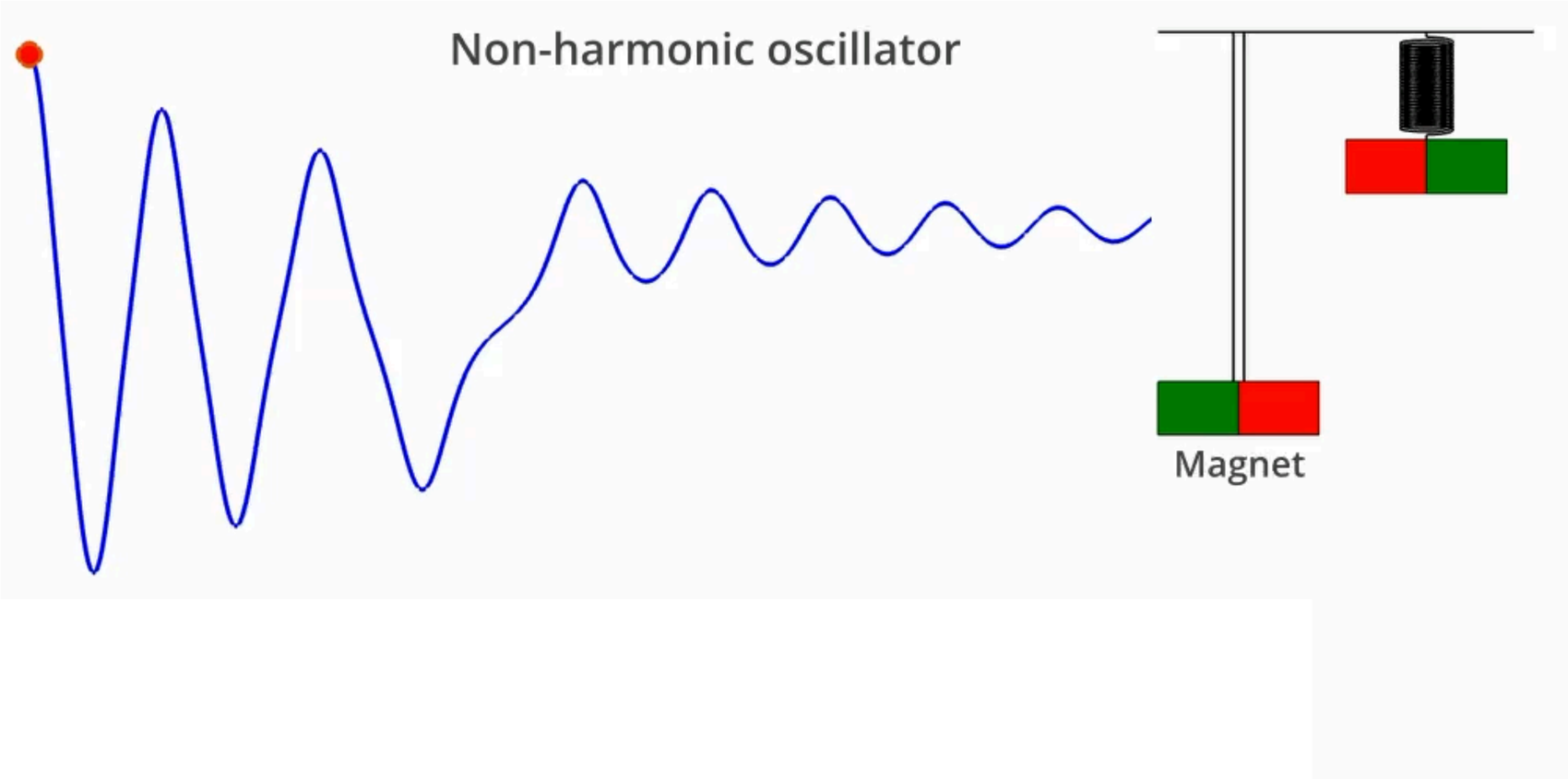
3. Prediction

Time stamps $T_{span} = [t_1, \ldots, t_N]$

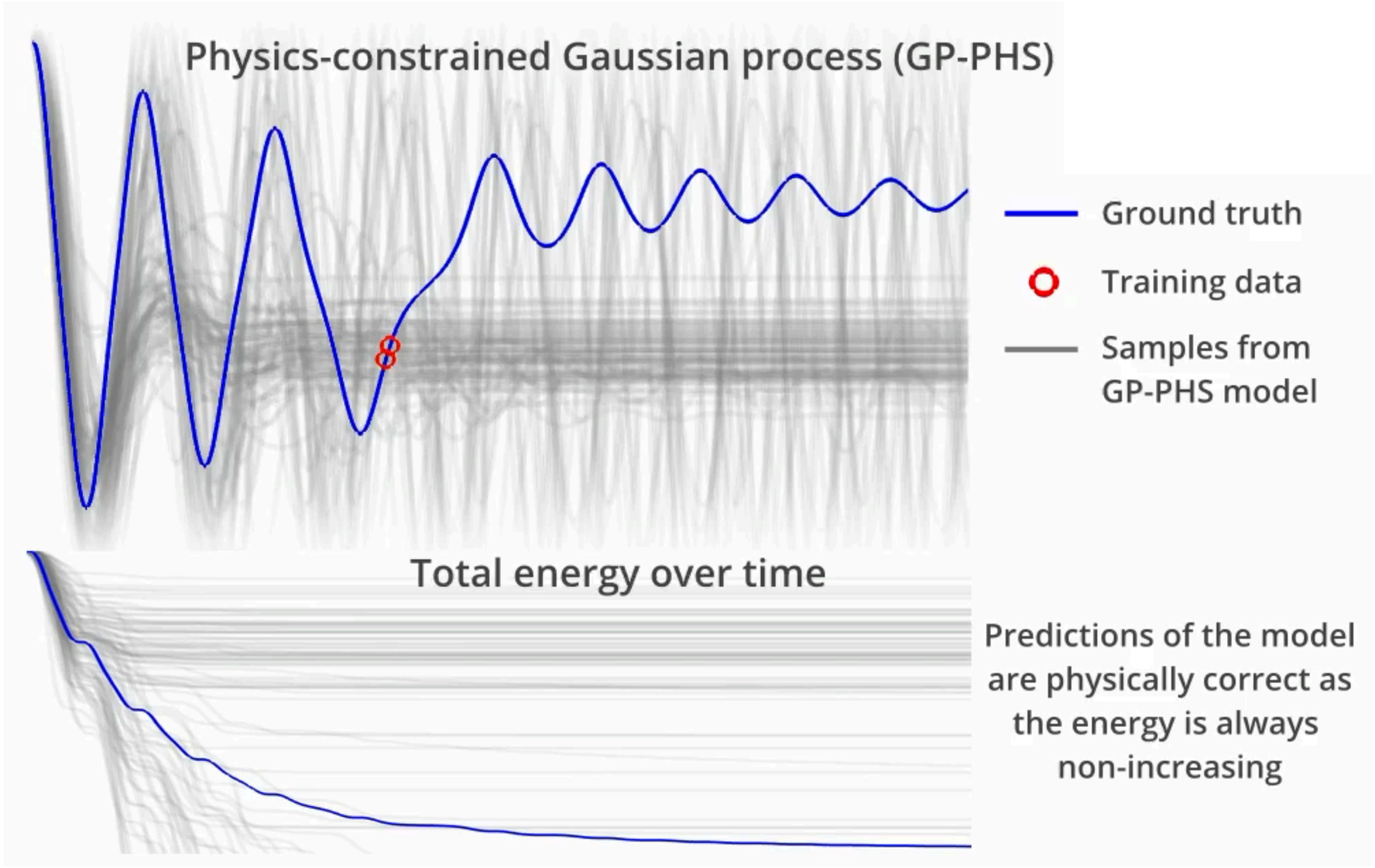
Initial value x_0, test input $u_{test}(t)$



Example

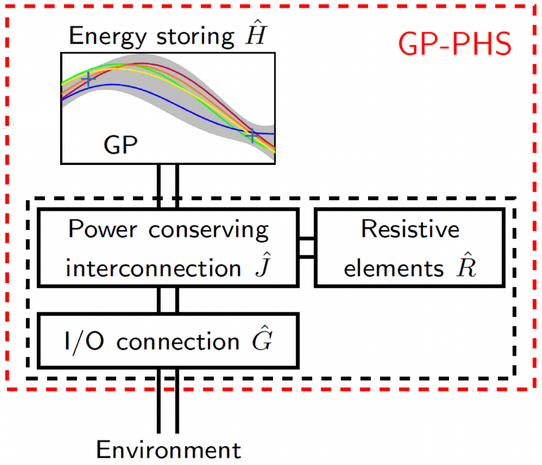


Example


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

Conclusion

Python Gaussian Process port-Hamiltonian System Toolbox



- User-friendly toolbox for learning of PHS based on observed data
- Physically correct data-driven model with uncertainty quantification
- Preserve the interconnection property and passivity characteristic

Tommy Li website:

Phs dae by gp, Peter z

www.tbeckers.com

https://peilun-tommy-li.github.io

T. Beckers. “Data-driven Bayesian Control of Port-Hamiltonian Systems”.In: *Proceedings of the Conference on Decision and Control*. 2023

T. Beckers, T. Z. Jiahao, and G. J. Pappas. “Learning Switching Port-Hamiltonian Systems with Uncertainty Quantification”. In: *Proceedings of the IFAC World Congress*. 2023.

T. Beckers, J. H. Seidman, P. Perdikaris, and G. J. Pappas. “Gaussian Process Port-Hamiltonian Systems: Bayesian Learning with Physics Prior”. In: Proceedings of the Conference on Decision and Control. 2022.