Bayesian Physics-informed Models for Soft Robotics

Peilun Li, Thomas Beckers

Institute for Software Integrated Systems | Vanderbilt University, Nashville, TN, USA

Problem Setup and Background

Problem Setup

Soft robotics is a growing field which relies on mimicking mechanisms of flexible bodies in nature to allow for soft interactions with their environment. Soft robots have a wide range of potential applications across various industries such as medical and healthcare, search and rescue, assistive robotics, and agriculture.

However, the modeling and control of soft robots is challenging due to

- Highly non-linear dynamics due to the soft materials (elastomers, non-linear elasticity, damping, etc.)
- Complex geometric and deformable structure that is difficult to represent

Example of soft robot

The problem can be summarized to a given dataset

$$D = \{\{t_0, \widetilde{x_0}, u_0\}, \ldots, \{t_n, \widetilde{x_n}, u_n\}\},$$

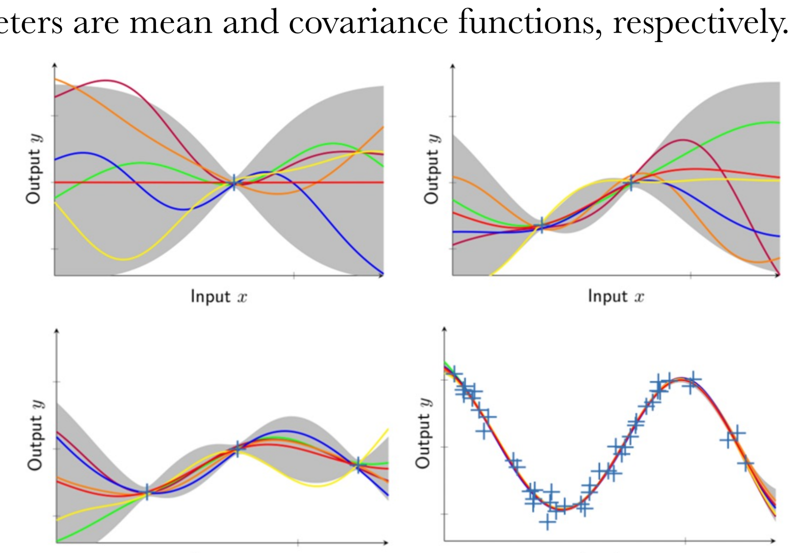
where t_i is the time variable, $\widetilde{x_i} = x_i + \eta$, $\eta \sim N(0, \sigma I)$ is the measured state variable (position, momentum) corrupted by Gaussian noise, and u_i is the input to the system, we aim to find a dynamical model that is:

- Expressive → able to represents large class of systems
- Reliable → allows uncertainty quantification
- Generalizable → follows physics laws

Background

Gaussian Process (GP)

- GPs represent a distribution over a function space and can be used as nonparametric regression method with uncertainty quantification
- Let the Gaussian prior be Gaussian distribution over function space F, therefore any $f(x) \in F$, we have $f(x) \sim GP(m(x), k(x, x'))$, where the parameters are mean and covariance functions, respectively.



- Instead of producing one fitted function given a dataset, the GP models generate a distribution of all possible functions.

Port-Hamiltonian System (PHS)

- Due to the derivation from physics, PHS is used because it is physically correct, highly interpretable, and structured.

Interconnection Dissipation Hamiltonian Input matrix

$$\dot{x} = [J(x) - R(x)]\nabla_x H(x) + G(x)u$$
$$y = G(x)^T \nabla_x H(x)$$

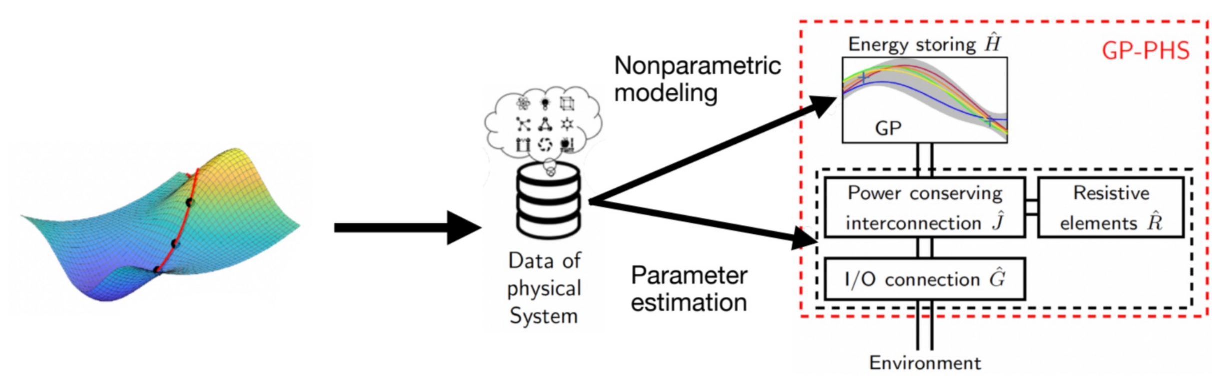
- Input and outputs define a passive system, and that the compose of PHS forms a PHS, which preserve modularity and extensibility.
- The time-derivative $\dot{x}$ can be used by ordinary differential equation solvers to compute the state variable x.

Methods and Results

Physics-informed learning for GP-PHS

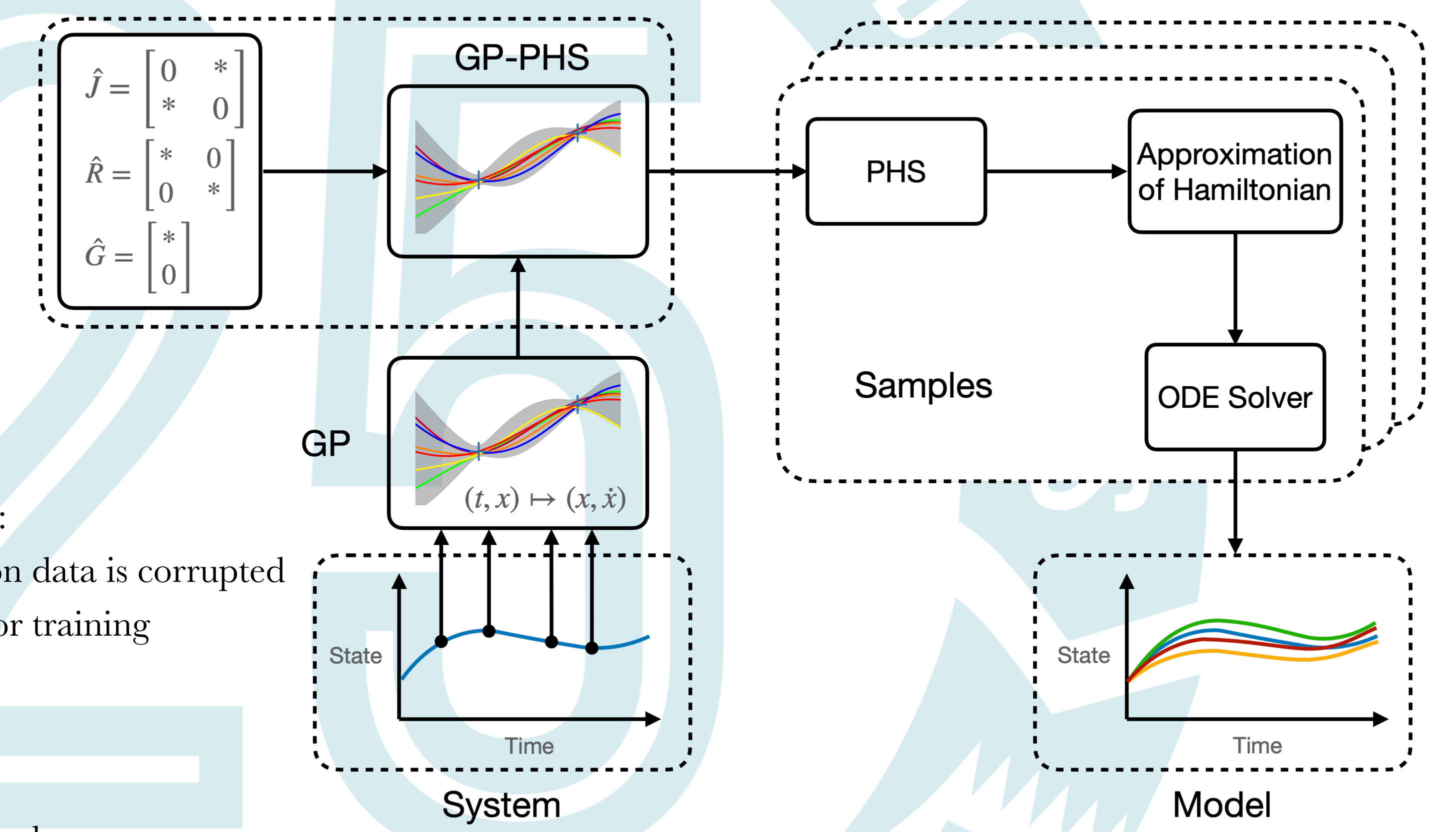
Depending on the robotic systems, the parameters in the PHS framework varies and are difficult to derive. Therefore, we aim to utilize a data-driven approach as a generalizable way for the model to learn the parameters. Specifically, we use GP to approximate the Hamiltonian functions and parameters in the PHS.

- The unknown variables are H(x), the Hamiltonian function, and matrices such as J(x), R(x), G(x).
- We rely on no prior knowledge on H(x) about the parametric structure, and thus utilize non-parametric learning.
- Because parameters such as J(x), R(x), G(x) follows physics properties, we know their structure and thus use parametric learning.



$$J(x) = \begin{bmatrix} j_{11} & & \mathcal{J} \\ & \ddots & \\ -\mathcal{J} & & j_{nn} \end{bmatrix}, R(x) = \begin{bmatrix} \star & & 0 \\ & \ddots & \\ 0 & & \star \end{bmatrix}, G(x) = \begin{bmatrix} \star \\ \\ 0 \end{bmatrix}$$

- Training data is obtained through modification of measurements. Define $T = [t_1, \ldots, t_N] \in R^{1 \times N}$ and $\tilde{X} = [\tilde{x}(t_1), \ldots, \tilde{x}(t_N)]^T \in R^{N \times n}$. Use square exponential kernel k, the distribution for every state variables can be calculated. The new training data is used.



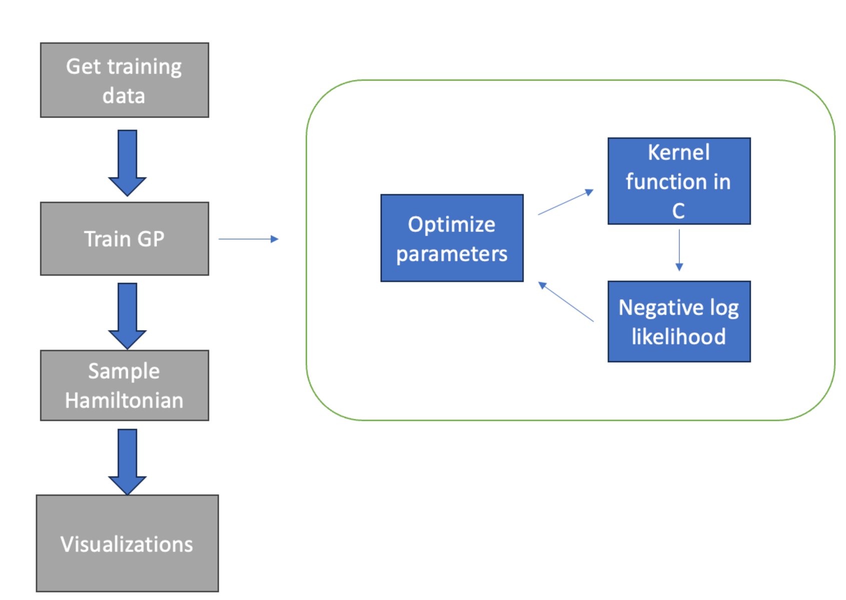
- GP-PHS models the state variables in the following steps:

1. Collecting data from the physical system, with assumption data is corrupted
2. Using a GP to generate a new dataset that will be used for training
3. Learning the Hamiltonian and hyperparameters
4. Sampling from GP, the posterior of the distribution
5. Approximating the Hamiltonian function
6. Using ODE solver, with methods like Runge-Kutta method…

- The parameters are learned through minimization of negative log marginal likelihood (NLML).
- After obtaining parameters such as Hamiltonian function and matrices, the ODE solver is applied to compute numerical integration for x, the state variable.

Implementation

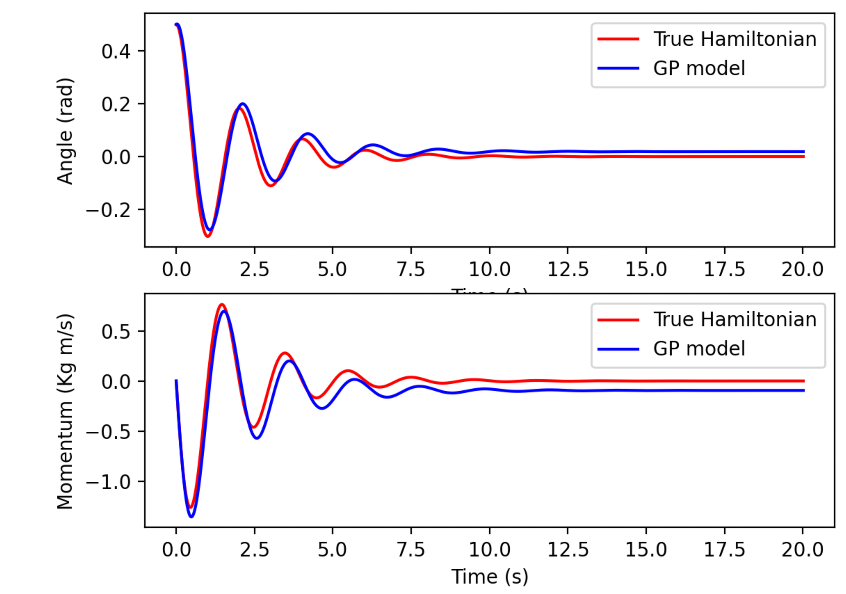
- The model is implemented in Python using libraries such as SciPy, with written C modules to increase computation efficiency.
- For example, mathematical computations such as the kernel function and Cholesky decomposition that are called multiple times and implemented in C compatible with Python.
- After sampling a Hamiltonian function and knowing matrices, we can use them in the PHS for numerical integration in solving for the state variable.

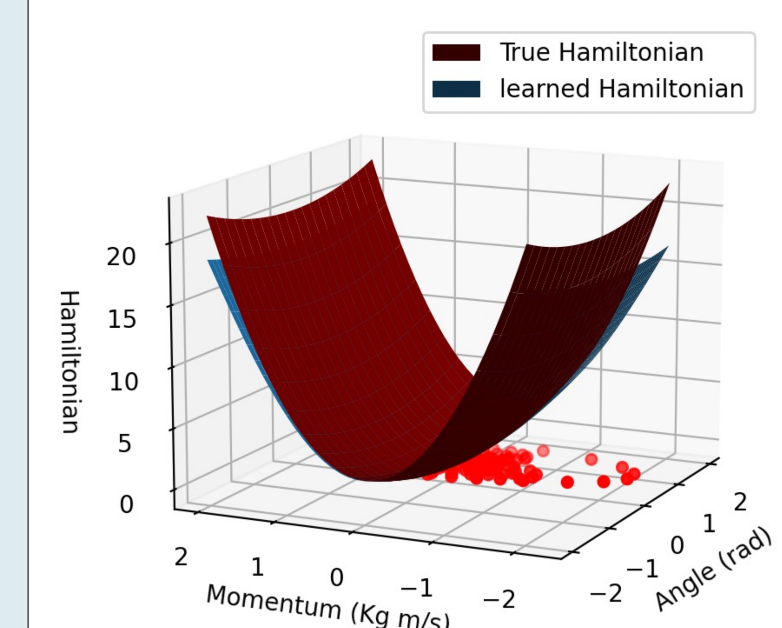


Results

As first benchmark, we have tested the proposed approach on a mass-spring-damper systems. The learned Hamiltonian is compared with ground truth.

The mass-spring damper system shows oscillation after the point mass being released from an initial angle. At the beginning, the momentum of the ball is zero. The state variable changes as energy of the system changes.





As the state variable being 2-dimentional, and Hamiltonian mapping to a scaler value, the graph forms a 3-dimentional projection. The red dots are data points, representing state variables at a given time, from which the Hamiltonian function is interpolated.

- This model is physically correct using the backbone of PHS, and the results show high degree of accuracy, with quantified uncertainty.
- The GP-PHS model generates all possible realizations of a learned PHS under the GP prior distribution.
- The GP-PHS preserves the interconnection and passivity properties, which are important for robotic control.

Future Work

After testing the concepts and codes, next steps are:

- Test with approximated soft robot model using datasets from simulation
- Use data measured from real soft robot
- Design model-based controller

Acknowledgement and References

Thomas Beckers, Jacob H. Seidman, Paris Perdikaris, and George J. Pappas. "Gaussian Process Port-Hamiltonian Systems: Bayesian Learning with Physics Prior".

Special thanks to Dr. Beckers for his patient guidance and mentorship, without of which I can hardly continue.

Thanks to ISIS and VUSE for supporting for this summer program.