Enforcing Physical Phenomena in System Identification using Bayesian Inference and Stochastic Models

Nicholas Galioto (ngalioto@umich.edu) and Alex Arkady Gorodetsky

Department of Aerospace Engineering, University of Michigan – Ann Arbor

Mechanistic Machine Learning and Digital Twins for Computational Science, Engineering, and Technology 2021

September 29, 2021

Outline

- Motivation
- Existing approaches
- Methodology
  - Problem formulation
  - Bayesian inference
- Results
  - Henon-Heiles
  - Reaction-diffusion PDE
- Conclusions

ngalioto@umich.edu

Motivation

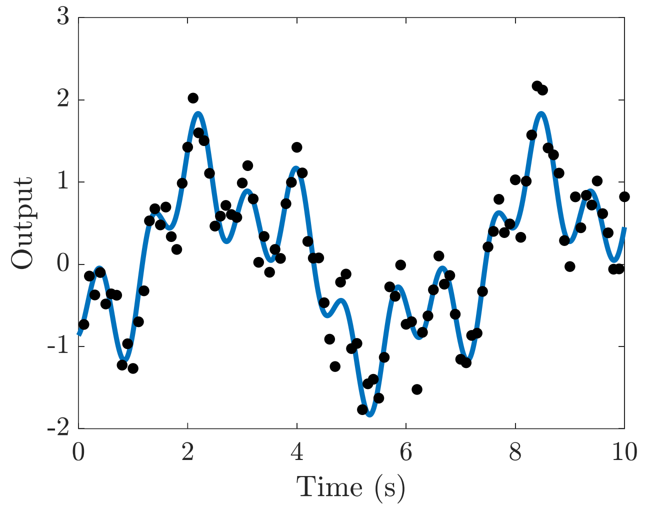
Objective: learn a model of a dynamical system from data

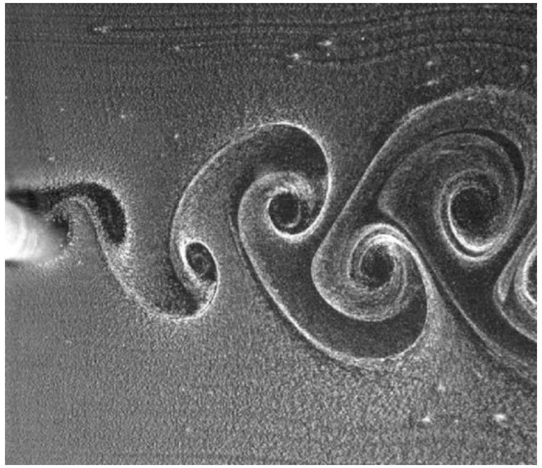
Two primary design choices in system identification:

- Model structure
  - Neural networks
  - Universal approximators
- Objective function
  - Least squared error
  - Regularization

A good algorithm will:

- Handle sparse and noisy data
- Scale well with dimension
- Trade off bias and variance optimally



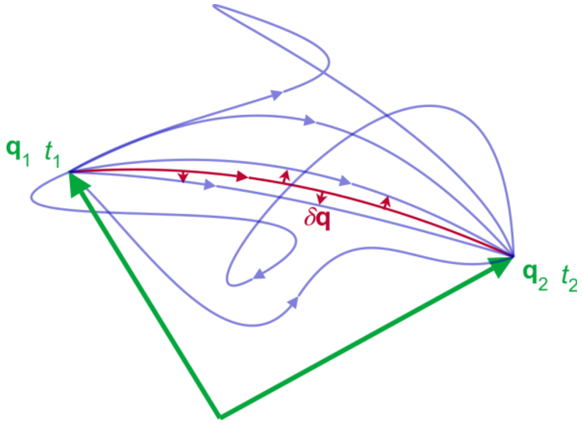


Ljung, L. (1999). System identification. *Wiley encyclopedia of electrical and electronics engineering*, 1-19.
Schoukens, J., & Ljung, L. (2019). Nonlinear system identification: A user-oriented road map. *IEEE Control Systems Magazine*, *39*(6), 28-99.

ngalioto@umich.edu

Motivation

- Incorporate all available information into our learning setup
  - Data collected from the system
  - Knowledge from physics
- We have a breadth of knowledge on physical systems from physics
  - Conservation of energy
  - Principle of least action
  - Stability
- In this work, we seek to enforce physical phenomena to learn Hamiltonian systems
  - Conservation
  - Reversibility
  - Symplecticness



$$\mathcal{H}(q,p) = T(q,p) + U(q,p)$$

$$\dot{q} = \frac{\partial \mathcal{H}}{\partial p} \quad \dot{p} = -\frac{\partial \mathcal{H}}{\partial q}$$

ngalioto@umich.edu

UNIVERSITY OF MICHIGAN

Existing Approaches

Least squares-based objective functions

(a) Assumes perfect model

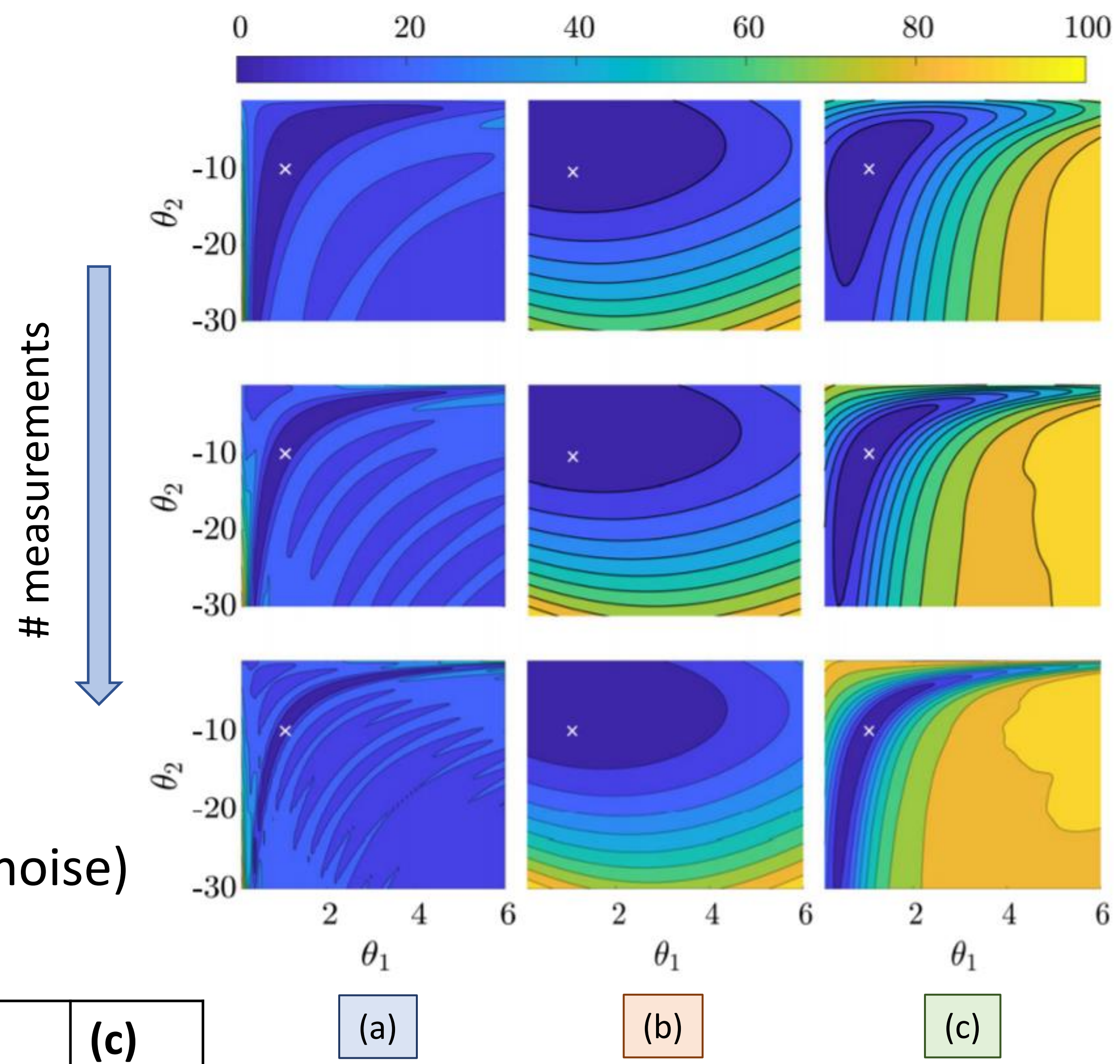
$$J(\theta) = \sum_{k=1}^{n} \|y_k - h(x(t_k), \theta)\|_2^2 \quad \text{s.t.} \quad \frac{\mathrm{d}x}{\mathrm{d}t} = f(t, x; \theta)$$

(b) Assumes noiseless measurements

$$J(\theta) = \sum_{k=1}^{n} \|y_k - \Psi(y_{k-1}; \theta)\|_2^2$$

(c) Noisy measurements + model error (process noise)

- Optimal combination of (a) and (b)



	(a)	(b)	(c)
Steep optimization surfaces without plateaus	✓	✗	✓
Suppresses local minima	✗	✓	✓
Increased confidence with data	✓	✗	✓

ngalioto@umich.edu

Existing Approaches

- Hamiltonian neural network (HNN) (Greydanus et al., 2019)
    - Parameterize the Hamiltonian
    - Minimize the objective

$$J(\theta) = \sum_{i=1}^{n} \left\| q_i - \int_{t_{i-1}}^{t_i} \frac{\partial \mathcal{H}_\theta}{\partial q} dt - q_{i-1} \right\|^2 + \left\| p_i + \int_{t_{i-1}}^{t_i} \frac{\partial \mathcal{H}_\theta}{\partial p} dt - p_{i-1} \right\|^2$$

- Originally forward Euler integration was used
- Leapfrog integration compared to forward Euler (Toth et al., 2019; Chen et al., 2019)
    - Leapfrog conserves the Hamiltonian
    - Leapfrog 2nd order accurate; forward Euler only 1st order accurate

S. Greydanus, M. Dzamba, and J. Yosinski, "Hamiltonian neural networks," in *Advances in Neural Information Processing Systems*, 2019, pp. 15 353–15 363.
P. Toth, D. J. Rezende, A. Jaegle, S. Racaniere, A. Botev, ` and I. Higgins, "Hamiltonian generative networks," *arXiv preprint arXiv:1909.13789*, 2019.
Z. Chen, J. Zhang, M. Arjovsky, and L. Bottou, "Symplectic recurrent neural networks," *arXiv preprint arXiv:1909.13334*, 2019.
N. Galioto and A. A. Gorodetsky, "Bayesian system id: optimal management of parameter, model, and measurement uncertainty," *Nonlinear Dynamics*, vol 102, no 1, pp. 241-267, 2020.

ngalioto@umich.edu

Probabilistic Formulation

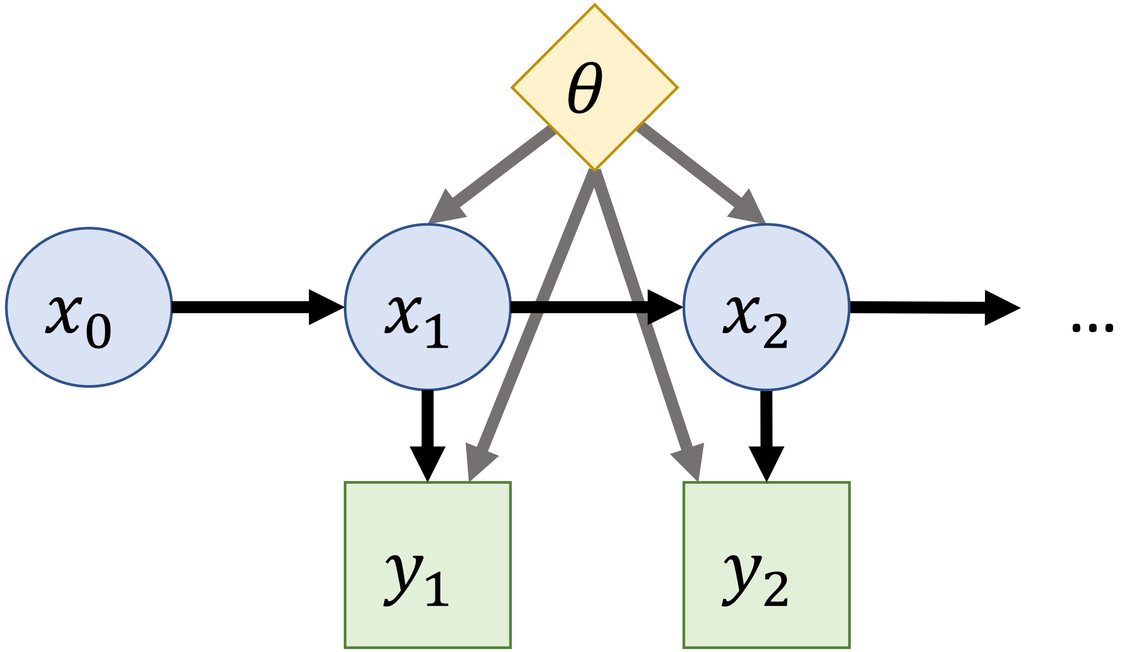
Joint parameter-state estimation with stochastic dynamics

$$X_k \in \mathbb{R}^{d_x}, \qquad Y_k \in \mathbb{R}^{d_y}, \qquad \theta = (\theta_\Psi, \theta_h, \theta_\Sigma, \theta_\Gamma) \in \mathbb{R}^{d_\theta}$$

$$X_k = \Psi(X_{k-1}, \theta_\Psi) + \xi_k; \quad \xi_k \sim \mathcal{N}(0, \Sigma(\theta_\Sigma))$$

$$Y_k = h(X_k, \theta_h) + \eta_k; \quad \eta_k \sim \mathcal{N}(0, \Gamma(\theta_\Gamma))$$

The process noise term ξ_k accounts for model error
- Parameter error
- Integration error
- Insufficient model expressiveness



1. Parameter Uncertainty
2. Model Uncertainty
3. Measurement Uncertainty

ngalioto@umich.edu

Posterior Flow Chart

Log Joint Likelihood

$$\log \mathcal{L}(\theta;\, \mathcal{X}_n, \mathcal{Y}_n) \propto -\frac{1}{2}\sum_{k=1}^{n} \|y_k - h(x_k, \theta_h)\|^2_{\Gamma(\theta_\Gamma)} - \frac{1}{2}\sum_{k=1}^{n} \|x_k - \Psi(x_{k-1}, \theta_\Psi)\|^2_{\Sigma(\theta_\Sigma)}$$

Deterministic dynamics:

$$x_k = \Psi(x_{k-1})$$

$$\log \mathcal{L}(\theta;\, \mathcal{Y}_n) \propto -\frac{1}{2}\sum_{k=1}^{n} \left\| x_k - h\left(\Psi^k(x_0, \theta_\Psi), \theta_h\right) \right\|^2$$

- Ayed et al., 2019
- Long et al., 2018
- Zhong et al., 2019

Identity observations:

$$y_k = x_k$$

$$\log \mathcal{L}(\theta;\, \mathcal{Y}_n) \propto -\frac{1}{2}\sum_{k=2}^{n} \|y_k - \Psi(y_{k-1}, \theta_\Psi)\|^2$$

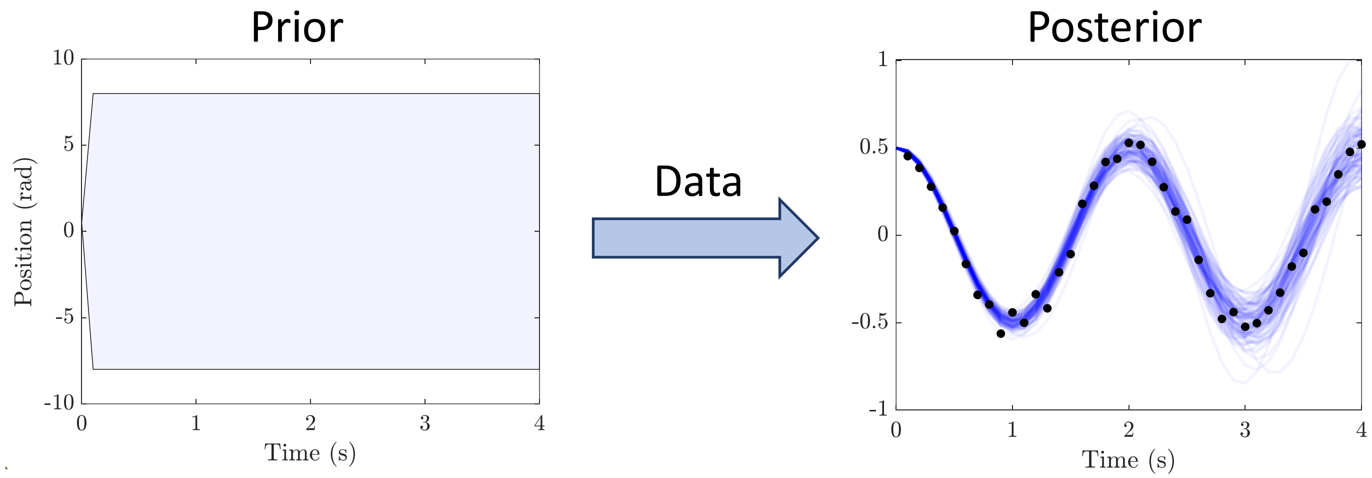
- Hamiltonian neural network
- Hills et al., 2015
- Qin et al., 2019
- Raissi, 2018

Ayed, I., de Bézenac, E., Pajot, A., Brajard, J., & Gallinari, P. (2019). Learning dynamical systems from partial observations. *arXiv preprint arXiv:1902.11136*.
Long, Z., Lu, Y., Ma, X., & Dong, B. (2018, July). Pde-net: Learning pdes from data. In *International Conference on Machine Learning* (pp. 3208-3216).
Zhong, Y. D., Dey, B., & Chakraborty, A. (2019). Symplectic ode-net: Learning hamiltonian dynamics with control. *arXiv preprint arXiv:1909.12077*.
Hills, D. J., Grütter, A. M., & Hudson, J. J. (2015). An algorithm for discovering Lagrangians automatically from data. *PeerJ Computer Science*, *1*, e31.
Qin, T., Wu, K., & Xiu, D. (2019). Data driven governing equations approximation using deep neural networks. *Journal of Computational Physics*, *395*, 620-635.
Raissi, M. (2018). Deep hidden physics models: Deep learning of nonlinear partial differential equations. *The Journal of Machine Learning Research*, *19*(1), 932-955.

ngalioto@umich.edu

Bayesian Inference

- Goal: compute $p(\theta|\mathcal{Y}_n)$ where $\mathcal{Y}_n = (y_1, y_2, \dots, y_n)$
- Bayes' rule: $p(\theta|\mathcal{Y}_n) = \frac{\mathcal{L}(\theta; \mathcal{Y}_n)p(\theta)}{p(\mathcal{Y}_n)}$



- Due to uncertainty in the states, we can only access the joint likelihood: $\mathcal{L}(\theta;\ \mathcal{X}_n, \mathcal{Y}_n)$
- To get the marginal likelihood, we must evaluate the integral

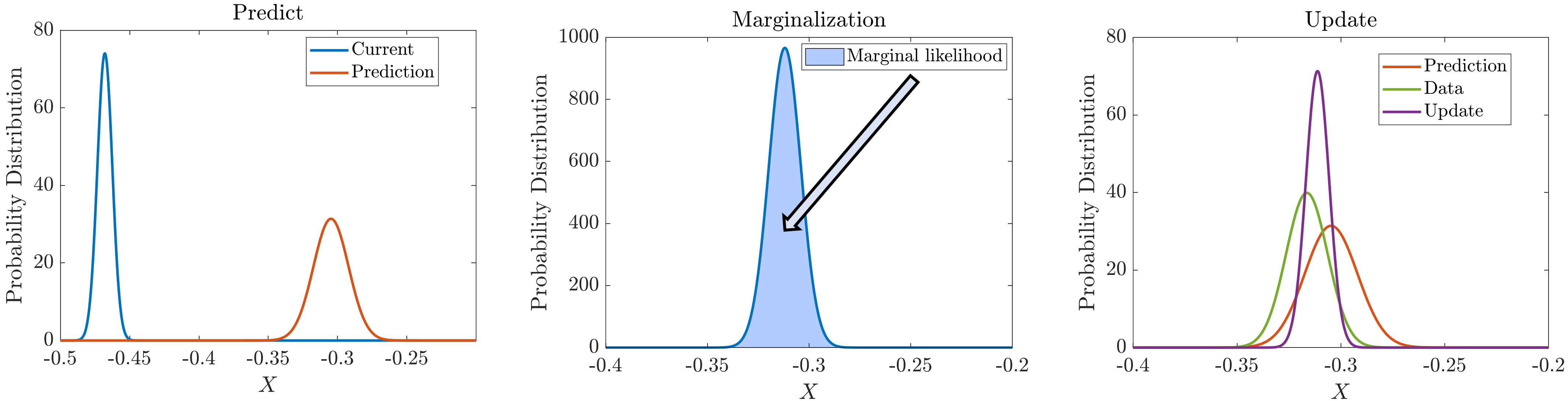
$$\mathcal{L}(\theta;\ \mathcal{Y}_n) = \int \mathcal{L}(\theta;\ \mathcal{X}_n, \mathcal{Y}_n) d\mathcal{X}_n$$

ngalioto@umich.edu

Approximate Marginal Posterior (Särkkä, 2013)

1. **for** k $= 1, \dots, n$
2. Predict: $p(X_{k+1} | \mathcal{Y}_k, \theta) = \int p(X_{k+1} | X_k, \theta) p(X_k | \mathcal{Y}_k, \theta) dX_k$
3. Marginalize: $\mathcal{L}_k(\theta; \mathcal{Y}_{k+1}) = \int p(y_{k+1} | X_{k+1}, \theta) p(X_{k+1} | \mathcal{Y}_k, \theta) dX_{k+1}$
4. Update: $p(X_{k+1} | \mathcal{Y}_{k+1}, \theta) = \frac{p(y_{k+1} | X_{k+1}, \theta) p(X_{k+1} | \mathcal{Y}_k, \theta)}{p(y_{k+1} | \mathcal{Y}_k, \theta)}$
5. **end for**

Unscented Kalman Filter



S. Särkkä, *Bayesian Filtering and Smoothing*, ser. Institute of Mathematical Statistics Textbooks. Cambridge University Press, 2013.

ngalioto@umich.edu

Dynamical Model Parameterization

Ensures the learned system is Hamiltonian

$$\mathcal{H}(q, p, \theta_\Psi) = \frac{1}{2} p^T p + U(q, \theta_\Psi)$$

Differentiation

$$\dot{q} = p, \qquad \dot{p} = -\frac{\partial U(q, \theta_\Psi)}{\partial q}$$

Conserves Hamiltonian and preserves symplectic structure throughout evaluation

Leapfrog Method

$$\Psi(q_k, p_k; \theta_\Psi) = \begin{bmatrix} q_k + \Delta t p_k - \frac{\Delta t^2}{2} \frac{\partial U(q, \theta_\Psi)}{\partial q}\Big|_{q_k} \\ p_k - \frac{\Delta t}{2}\left(\frac{\partial U(q, \theta_\Psi)}{\partial q}\Big|_{q_k} + \frac{\partial U(q, \theta_\Psi)}{\partial q}\Big|_{q_{k+1}} \right) \end{bmatrix}$$

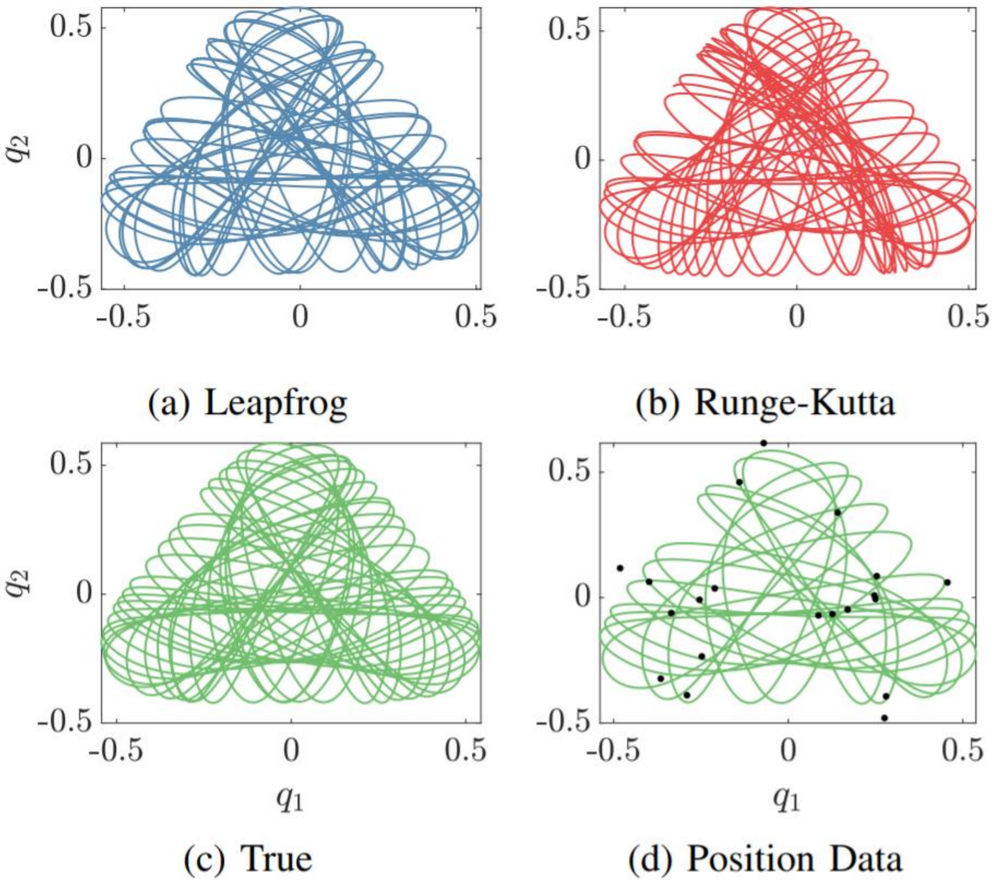
ngalioto@umich.edu

Results: Hénon-Heiles

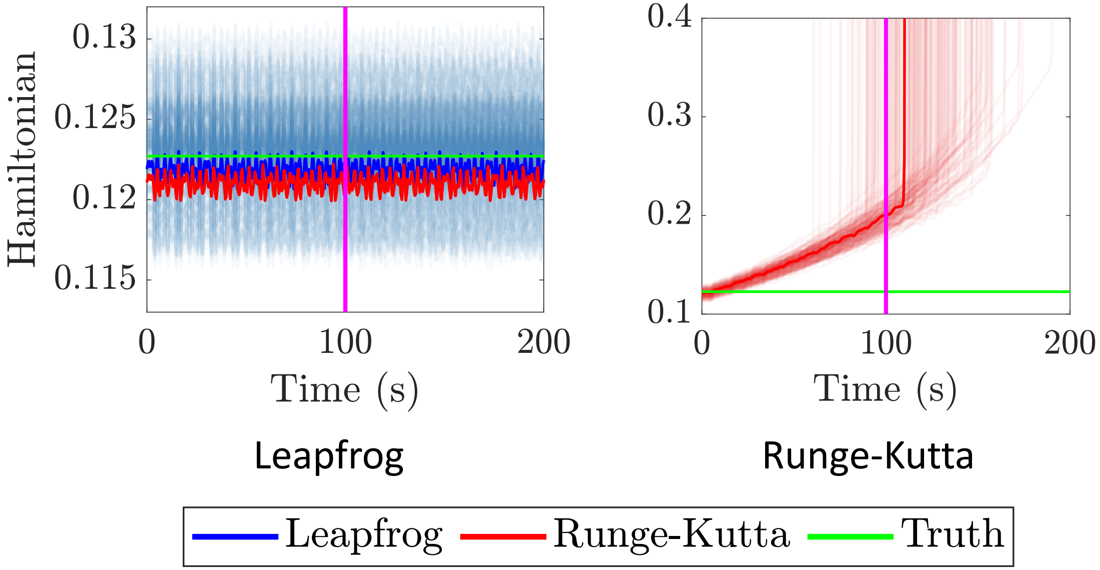
The symplectic approach learns a more accurate Hamiltonian

Truth: $U(q_1, q_2) = \frac{1}{2}q_1^2 + \frac{1}{2}q_2^2 + q_1^2 q_2 - \frac{1}{3}q_2^3$

Phase plots



Hamiltonian over time



The method equipped with RK must learn a smaller Hamiltonian to compensate for being non-conservative

Relative mean error:
Leapfrog: 0.7%
Runge-Kutta: 1.3%

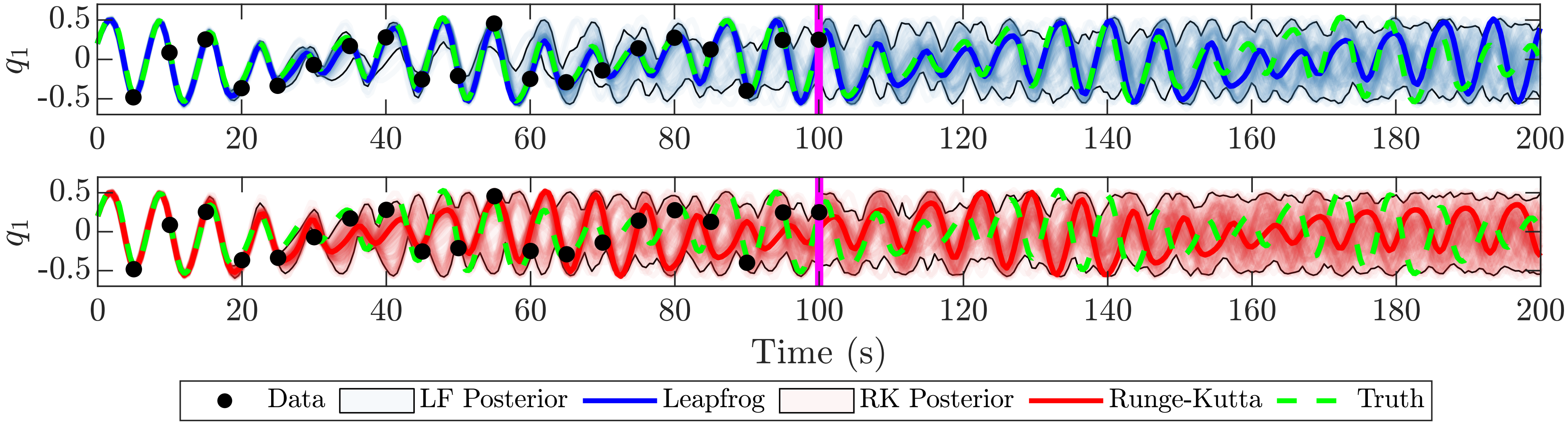
Data Generation:
$n = 20, \quad \Delta t = 5, \quad \sigma = 0.05$

ngalioto@umich.edu

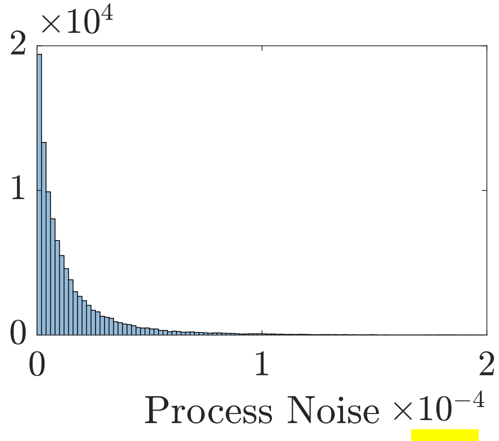
Results: Hénon-Heiles

The symplectic approach yields greater certainty

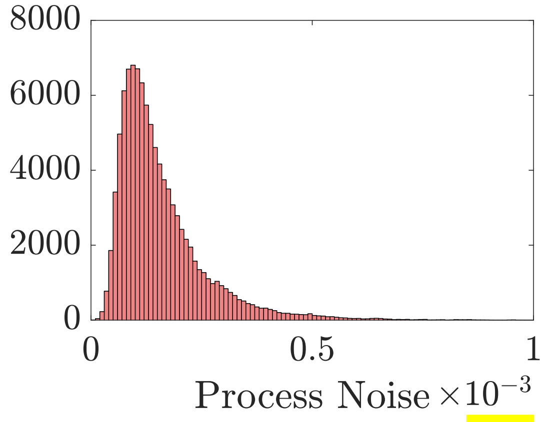
Posterior estimates of q_1 trajectory



Process noise marginal posteriors



Leapfrog



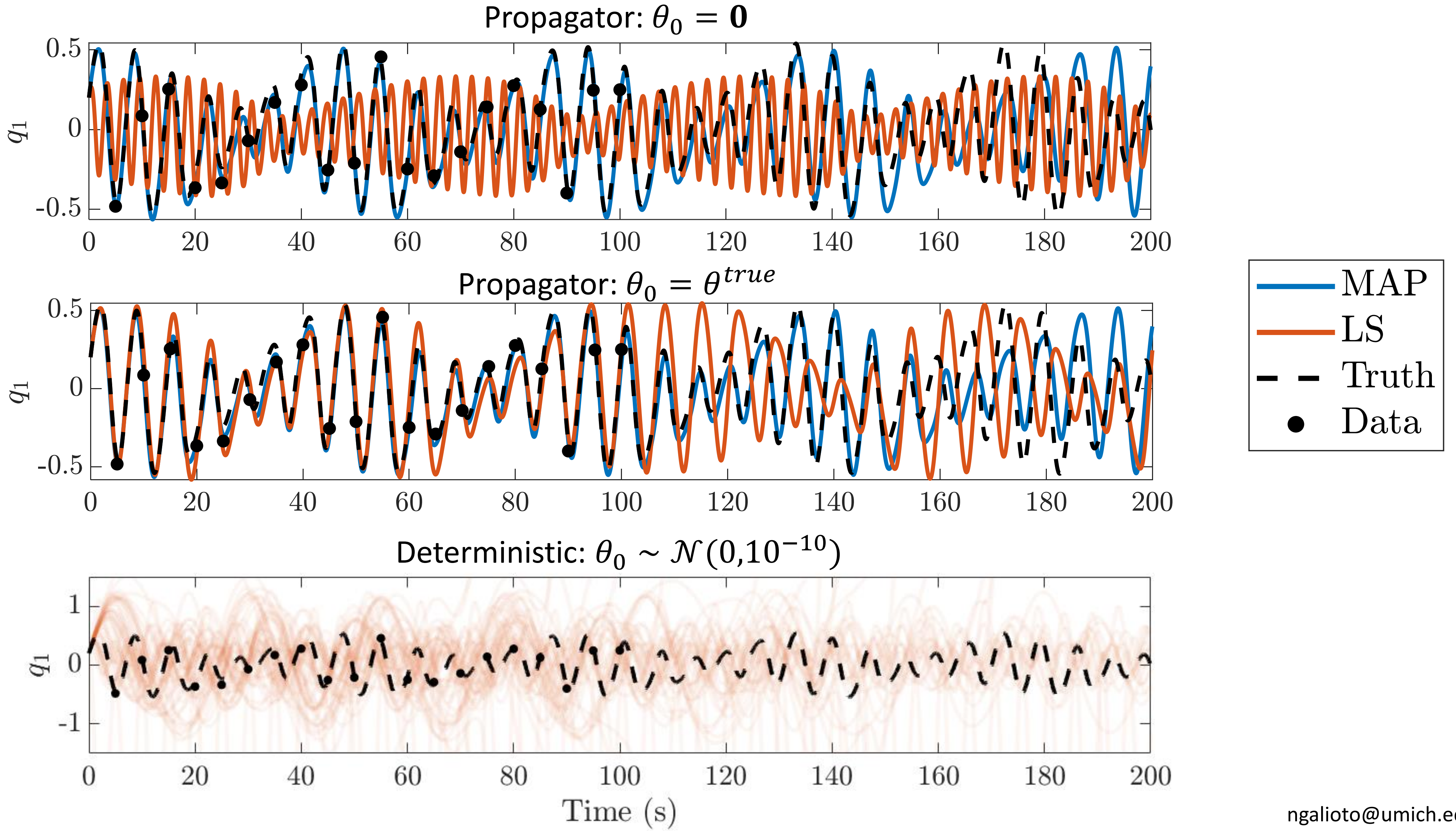
Runge-Kutta

Symplectic approach learns a model with an order of magnitude greater certainty

ngalioto@umich.edu

Results: Hénon-Heiles

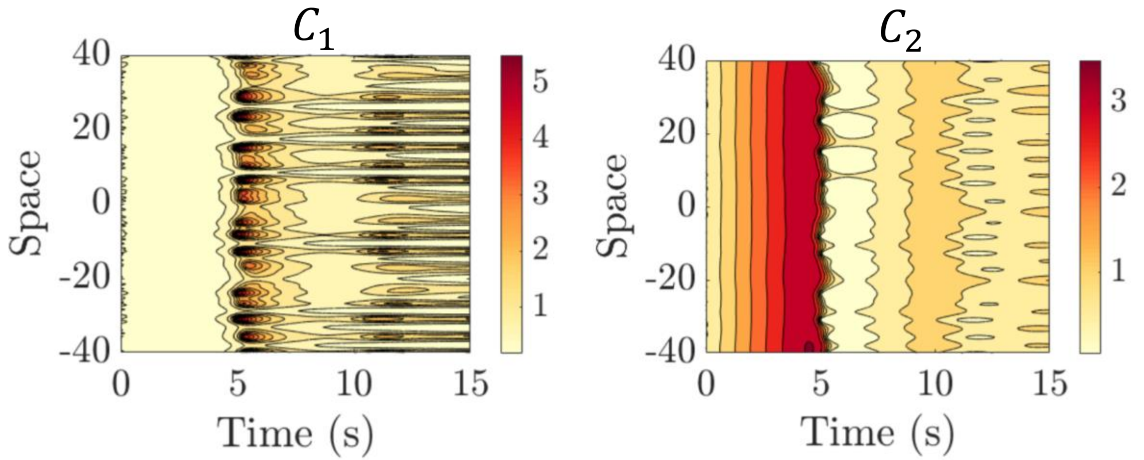
MAP estimate outperforms least squares approaches



ngalioto@umich.edu

Results: Reaction-Diffusion PDE

$$\frac{\partial C_1}{\partial t} = \theta_1 \frac{\partial^2 C_1}{\partial x^2} + 0.1 - C_1 + \theta_3 C_1^2 C_2$$
$$\frac{\partial C_2}{\partial t} = \theta_2 \frac{\partial^2 C_2}{\partial x^2} C_2 + 0.9 - C_1^2 C_2$$



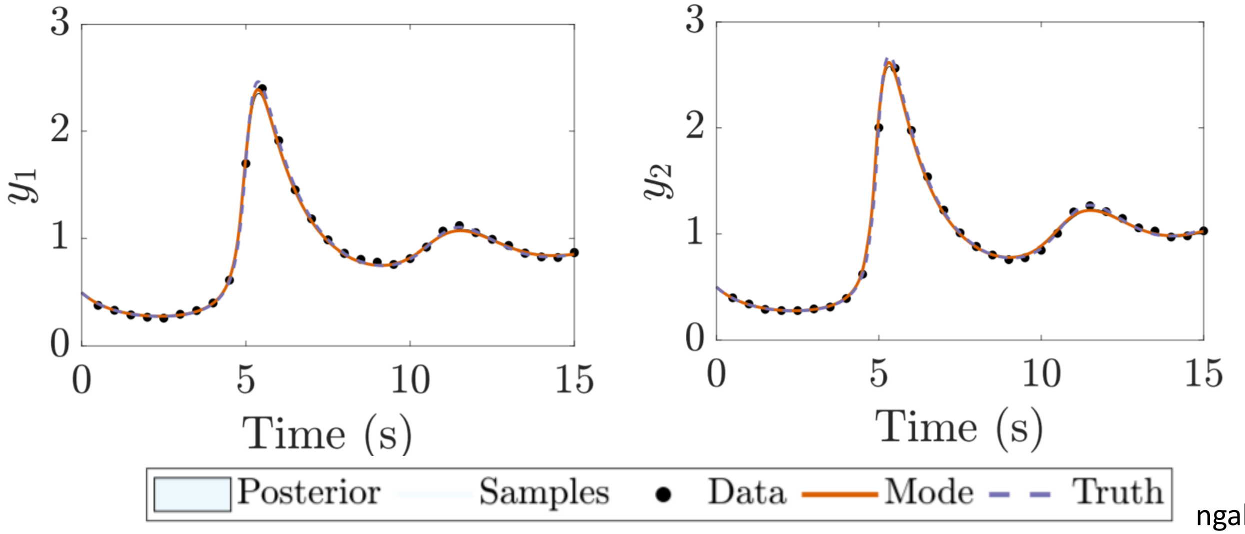
Neumann Boundary Conditions

$$\frac{\partial C_1}{\partial x} = \frac{\partial C_2}{\partial x} = 0$$

$(C_i)_j \sim \mathcal{U}(0.4, 0.6)$ for $t = 0; i = 1,2; j = 1,2, \ldots, 201.$

$$y_1(t) = \int_{-40}^{40} C_1(t)\,\mathrm{d}x$$
$$y_2(t) = \int_{-40}^{40} C_1^2(t)\,\mathrm{d}x$$

$n = 30, \sigma = 10^{-2}$



ngalioto@umich.edu

Main Takeaway

- Optimally accounting for different types of uncertainty can lead to robustness even when data are few and/or noisy[1]
- Embedding the learning process with a symplectic integrator yields two main benefits[2]
  - Greater accuracy
  - Greater certainty

Funding

- DARPA Physics of AI Program
  - “Physics Inspired Learning and Learning the Order and Structure of Physics.”
- AFOSR Program in Computational Mathematics

1. Galioto, N., & Gorodetsky, A. A. (2020). Bayesian system ID: optimal management of parameter, model, and measurement uncertainty. *Nonlinear Dynamics*, *102*(1), 241-267.
2. Galioto, N., & Gorodetsky, A. A. (2020, December). Bayesian Identification of Hamiltonian Dynamics from Symplectic Data. In *2020 59th IEEE Conference on Decision and Control (CDC)* (pp. 1190-1195). IEEE.

ngalioto@umich.edu

Thank You

ngalioto@umich.edu

Marginal Likelihood

Regularization derived from first principles

Let the state be distributed normally as $X_k \sim \mathcal{N}(m_k, P_k)$

The negative log-likelihood is equivalent to a time-varying generalized least-squares objective with regularization

$$\mathcal{L}(\theta; \mathcal{Y}_n) \propto \sum_{k=1}^{n} \|y_k - H(\theta) m_k^-(\theta)\|^2_{S_k^{-1}(\theta)} + \log|2\pi S_k(\theta)|$$

Where

$$P_k^-(\theta) = A(\theta) P_{k-1}^+(\theta) A^T(\theta) + Q(\theta)$$
$$S_k(\theta) = H(\theta) P_k^-(\theta) H^T(\theta) + R(\theta)$$

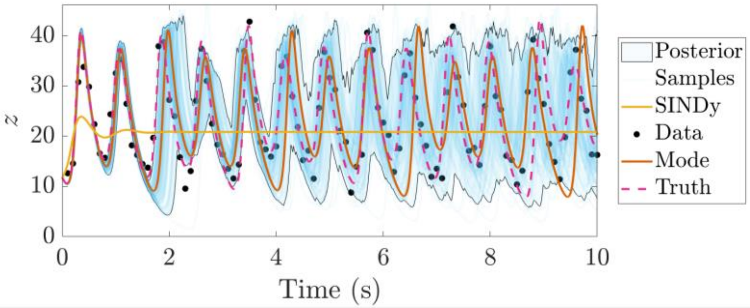
This objective prioritizes:

- low bias: $\|y_k - H(\theta) m_k^-(\theta)\|^2_{S_k^{-1}(\theta)}$
- low variance: $\log|2\pi S_k(\theta)|$

ngalioto@umich.edu

Results: Lorenz ’63

Accounting for model error enhances robustness

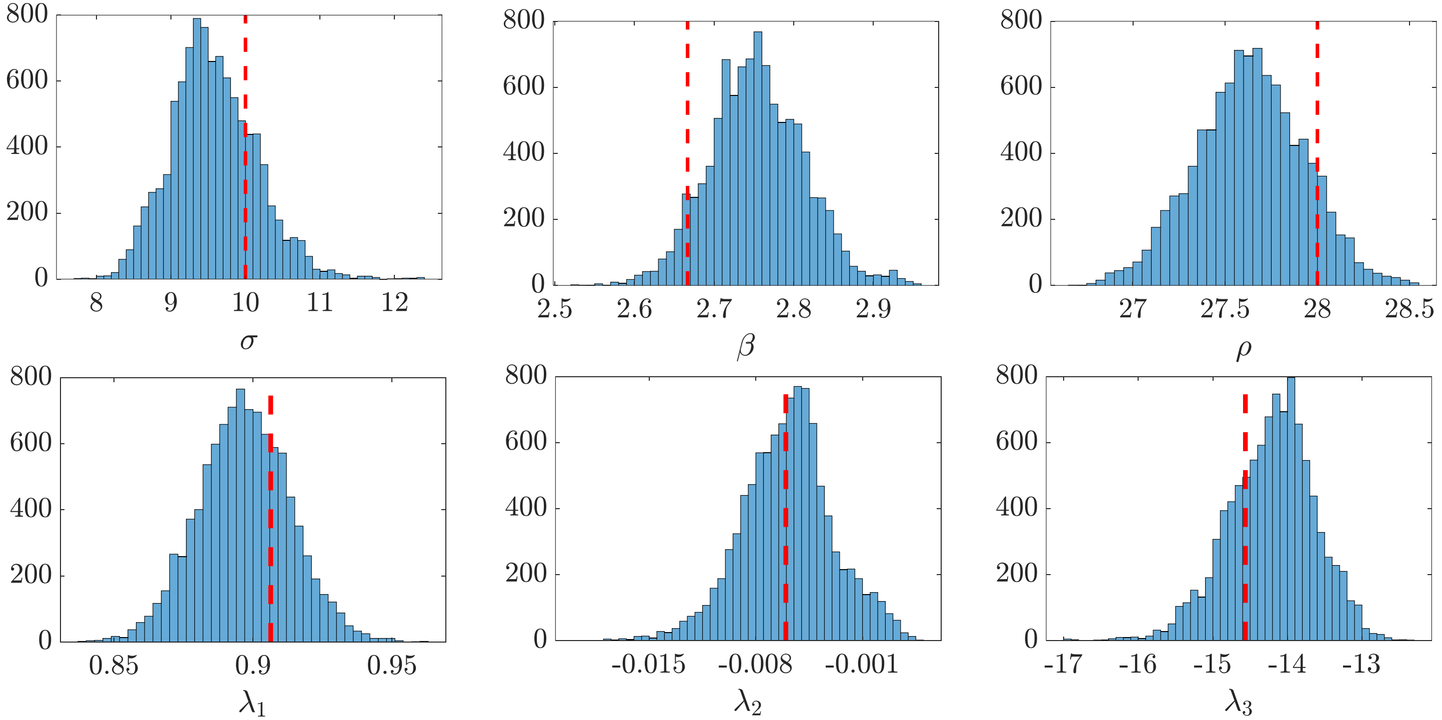


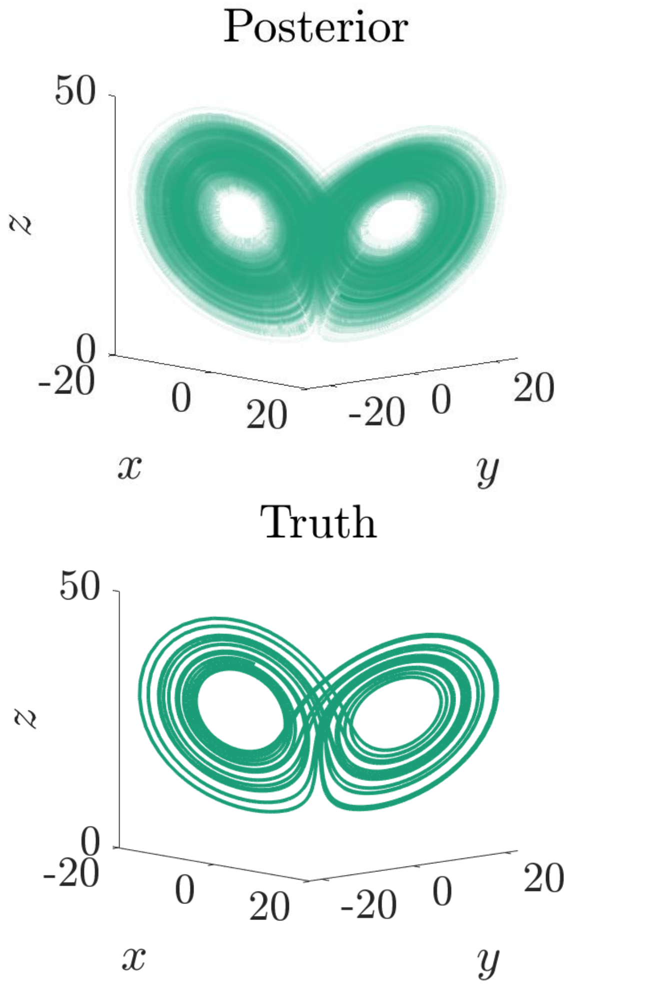
Most positive Lyapunov exponent: $\lambda_1 = 0.906$

$$\dot{x} = \sigma(y - x)$$
$$\dot{y} = x(\rho - z) - y$$
$$\dot{z} = xy - \beta z$$

Recent works[1,2,3] commonly use:
$n = 300$
$\Delta t = 0.01s$
$\sigma_R = 0.0$

Here we use:
$n = 100$
$\Delta t = 0.10s$
$\sigma_R = 2.0$





1. Lazzús, J. A., Rivera, M., & López-Caraballo, C. H. (2016). Parameter estimation of Lorenz chaotic system using a hybrid swarm intelligence algorithm. *Physics Letters A*, *380*(11-12), 1164-1171.
2. Xu, S., Wang, Y., & Liu, X. (2018). Parameter estimation for chaotic systems via a hybrid flower pollination algorithm. *Neural Computing and Applications*, *30*(8), 2607-2623.
3. Zhuang, L., Cao, L., Wu, Y., Zhong, Y., Zhangzhong, L., Zheng, W., & Wang, L. (2020). Parameter Estimation of Lorenz Chaotic System Based on a Hybrid Jaya-Powell Algorithm. *IEEE Access*, *8*, 20

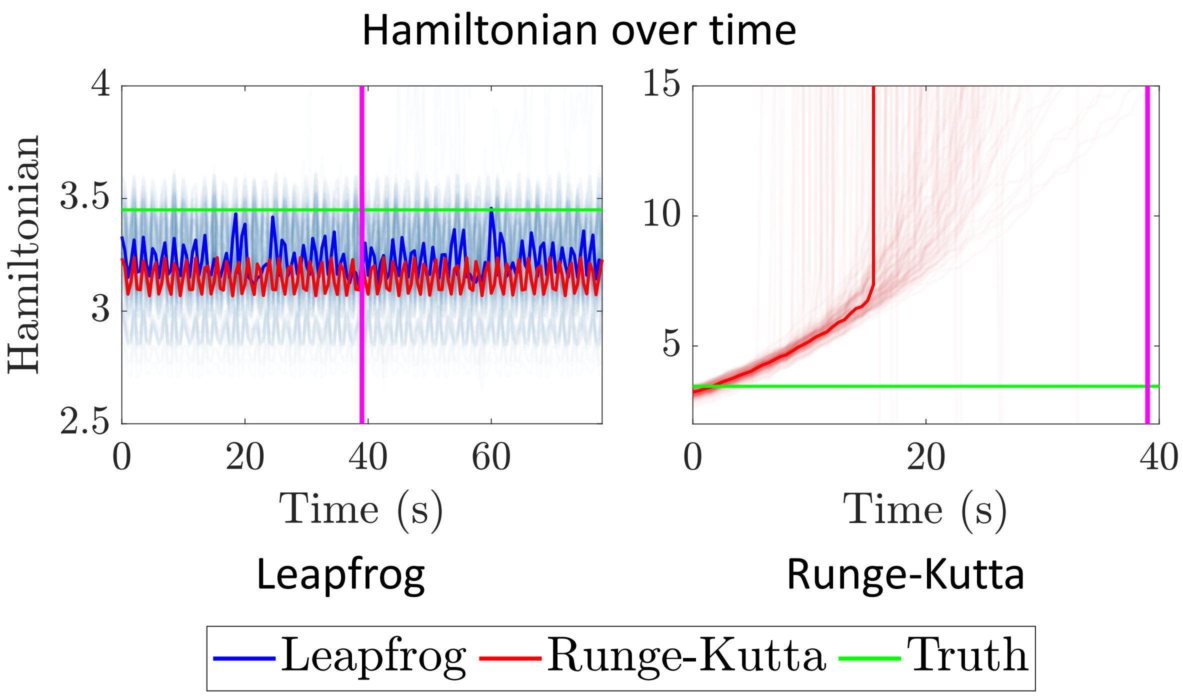
ngalioto@umich.edu

Numerical Experiments: FPU Chain

The symplectic approach learns a more accurate Hamiltonian

$$U(q) = \sum_{i=1}^{N} \frac{(q_{i+1} - q_i)^2}{2} + \frac{\beta (q_{i+1} - q_i)^4}{4}$$

- We choose $N = 2, \beta = 0.1$
- Parameterize $U(q, \theta_\Psi)$ with polynomials up to total order 4 (14 terms)



Relative mean error:
Leapfrog: 4.3%
Runge-Kutta: 6.7%

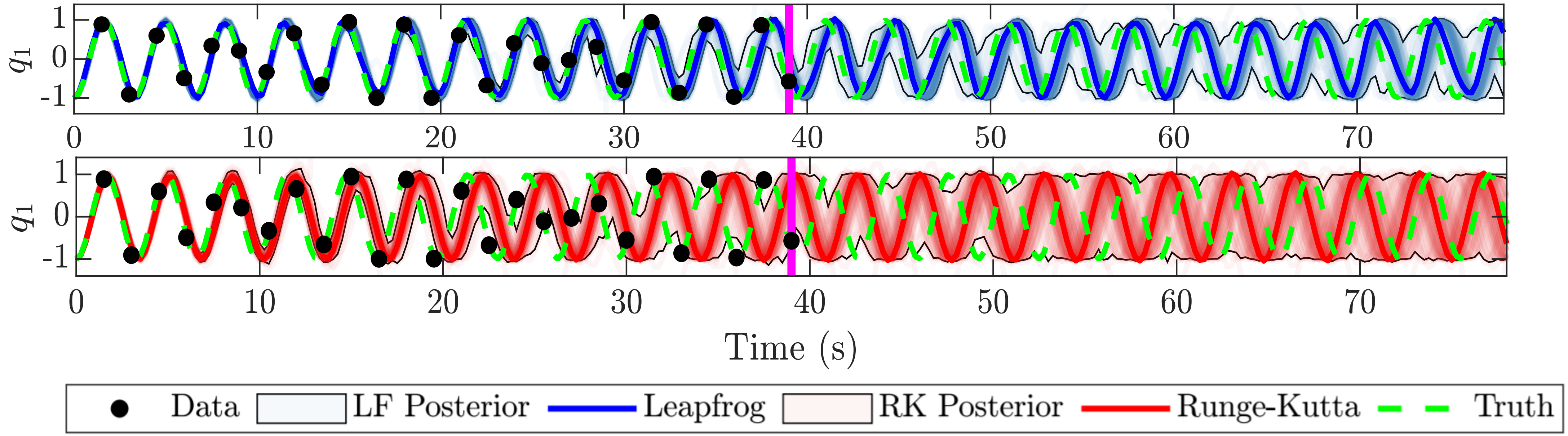
Data Generation:
- $n = 26$
- $\Delta t = 1.5$
- $\sigma_q = 0.01$
- $\sigma_p = 0.02$

ngalioto@umich.edu

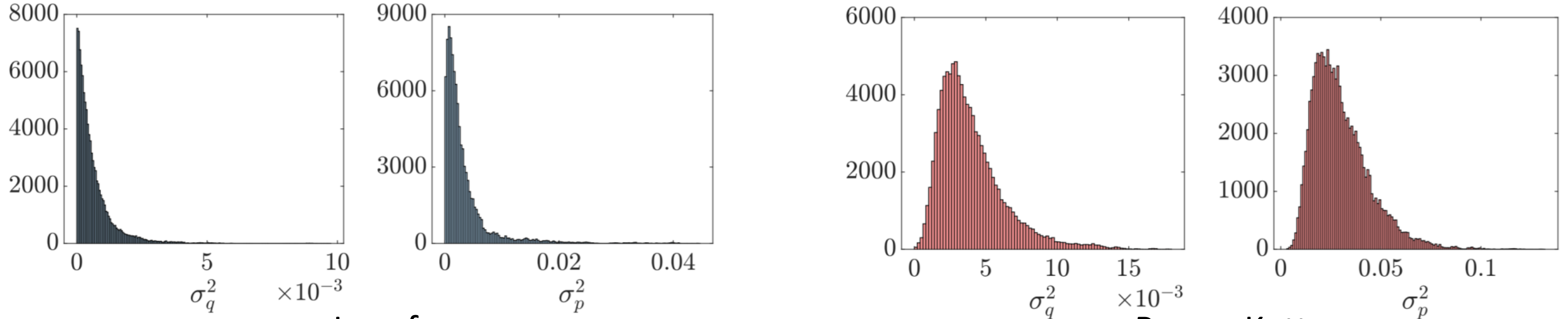
Numerical Experiments: FPU Chain

The symplectic approach yields greater certainty

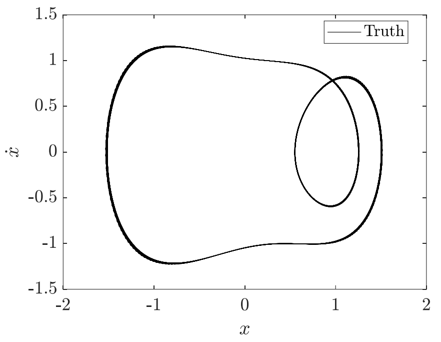
Posterior estimates of q_1 trajectory



Process noise marginal posteriors



ngalioto@umich.edu



Duffing Oscillator with Forcing

$$\begin{bmatrix}\dot{x}\\ \ddot{x}\end{bmatrix} = \begin{bmatrix}0 & 1\\ \alpha & \delta\end{bmatrix}\begin{bmatrix}x\\ \dot{x}\end{bmatrix} + \beta\begin{bmatrix}0\\ x^3\end{bmatrix} + \begin{bmatrix}0\\ 1\end{bmatrix}\gamma\cos(\omega t), \qquad y_k = x_k \qquad T = 400, \Delta t = 0.25,\ \sigma_\Gamma = 10^{-8}$$

$\alpha = 1, \delta = -0.3, \beta = -1, \gamma = 0.65, \omega = 1.2$

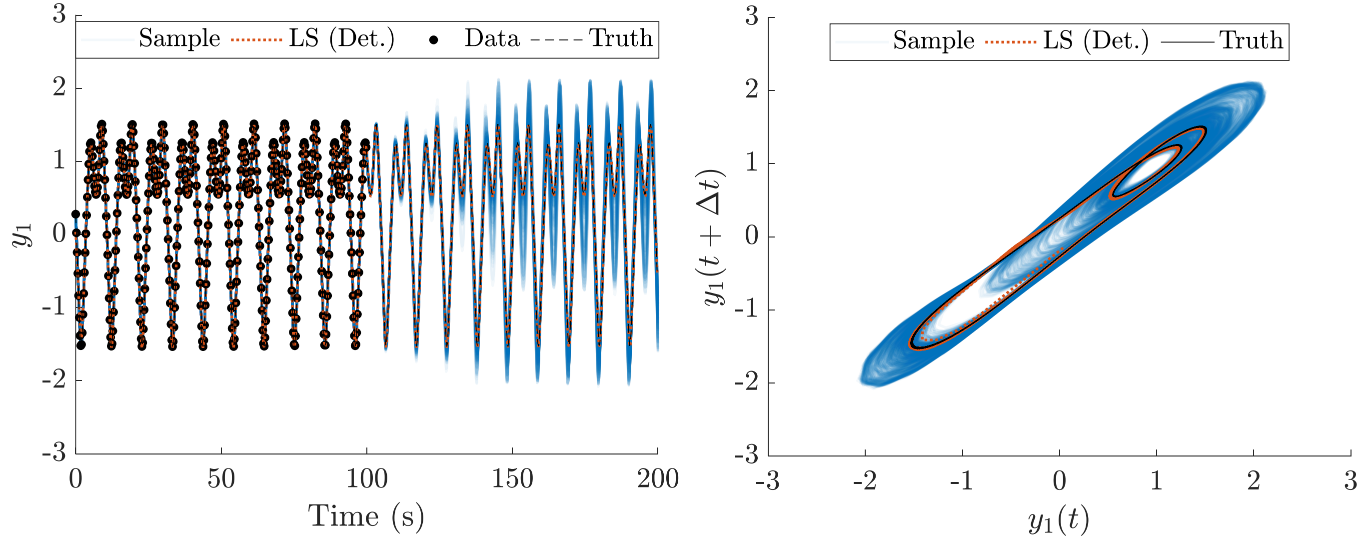
Period-2 solution[1]

Model formulation:

$x_0 = x_0(\theta),\ d_x = 2$

$$x_{k+1} = f(x_k, u_k; \theta) + \xi_k, \qquad \xi_k \sim \mathcal{N}\big(0, \Sigma(\theta)\big)$$

$$y_k = [1 \quad 0]x_k + \eta_k, \qquad \eta_k \sim \mathcal{N}(0, \Gamma)$$



Priors:

$\theta_\Psi \sim \mathcal{N}(0{,}5)$

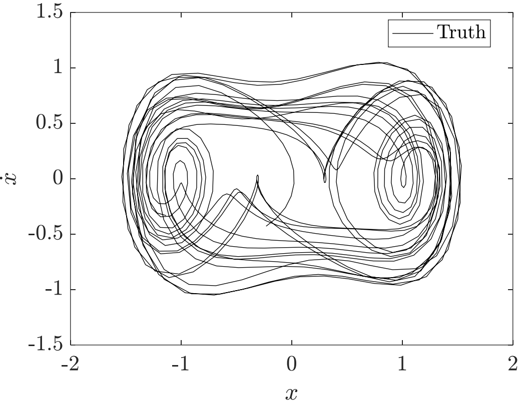
$\theta_\Sigma \sim$ half-$\mathcal{N}(0{,}10^{-5})$

Neural network architecture[2]

$$f(x, u; \theta) = A_1(\theta)\tanh\left(A_2(\theta)\begin{bmatrix}x\\ u\end{bmatrix} + b_2(\theta)\right) + A_3(\theta)\begin{bmatrix}x\\ u\end{bmatrix} + b_3(\theta)$$

1. Jordan, D., & Smith, P. (2007). *Nonlinear ordinary differential equations: an introduction for scientists and engineers* (Vol. 10). Oxford University Press on Demand.
2. Beintema, G., Toth, R., & Schoukens, M. (2021, May). Nonlinear state-space identification using deep encoder networks. In *Learning for Dynamics and Control* (pp. 241-250). PMLR.

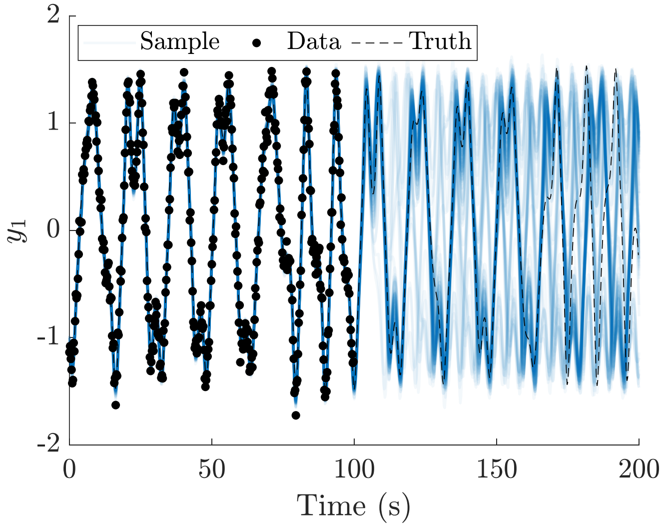
Duffing Oscillator with Forcing

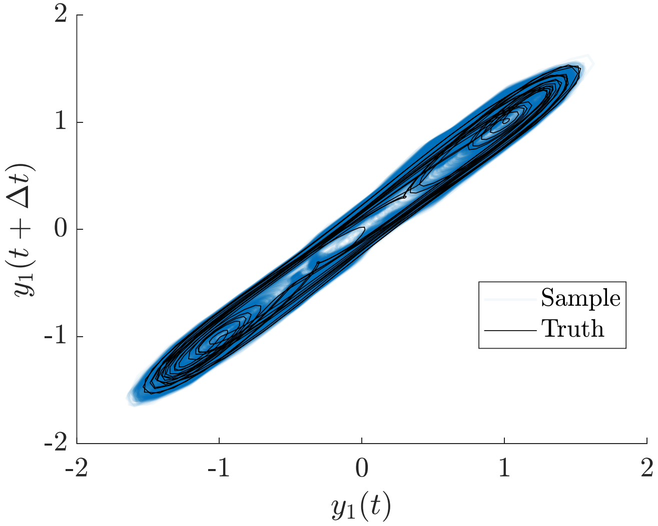


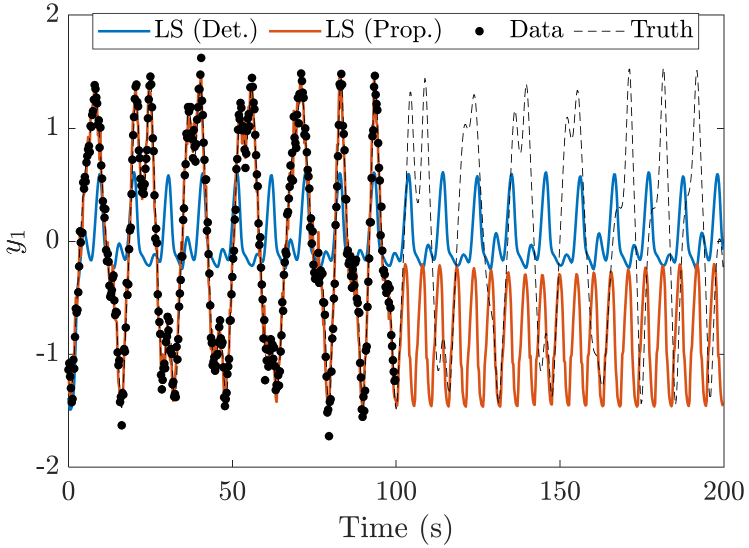
$$\begin{bmatrix}\dot{x}\\ \ddot{x}\end{bmatrix} = \begin{bmatrix}0 & 1\\ \alpha & \delta\end{bmatrix}\begin{bmatrix}x\\ \dot{x}\end{bmatrix} + \beta\begin{bmatrix}0\\ x^3\end{bmatrix} + \begin{bmatrix}0\\ 1\end{bmatrix}\gamma\cos(\omega t), \qquad y_k = x_k$$

$\alpha = 1, \delta = -0.3, \beta = -1, \gamma = 0.5, \omega = 1.2$

Chaotic solution[1]







$T = 400$

$\Delta t = 0.25$

$\sigma_\Gamma = 0.1$

1. Jordan, D., & Smith, P. (2007). *Nonlinear ordinary differential equations: an introduction for scientists and engineers* (Vol. 10). Oxford University Press on Demand.