# Robust Bayesian Inference by Accounting for Model Error: with Applications to Hamiltonian Systems

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

Aerospace Engineering Department, University of Michigan – Ann Arbor

SIAM Conference on Computational Science and Engineering 2021

March 4, 2021

# Motivation

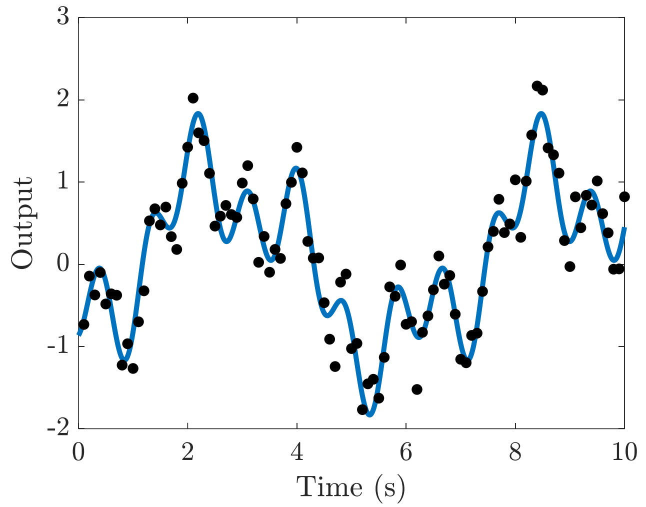
Objective: learn a model of a dynamical system from data

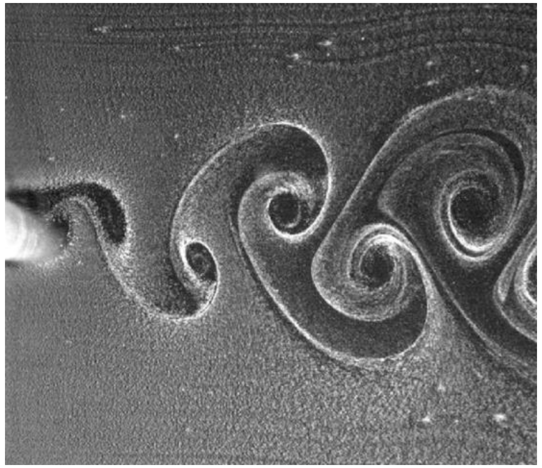
Two primary design choices in system identification:

- 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.

# Outline

1. Existing approaches
2. Probabilistic formulation
3. Bayesian inference
4. Algorithm
5. Results
6. Takeaways

# Existing Approaches

Least squares-based objective functions

(a) Assumes perfect model $J(\theta) = \sum_{i=1}^{n} \|y_i - x(t_i)\|_2^2 \quad \text{s.t.} \quad \frac{\mathrm{d}x}{\mathrm{d}t} = f(t, x; \theta)$

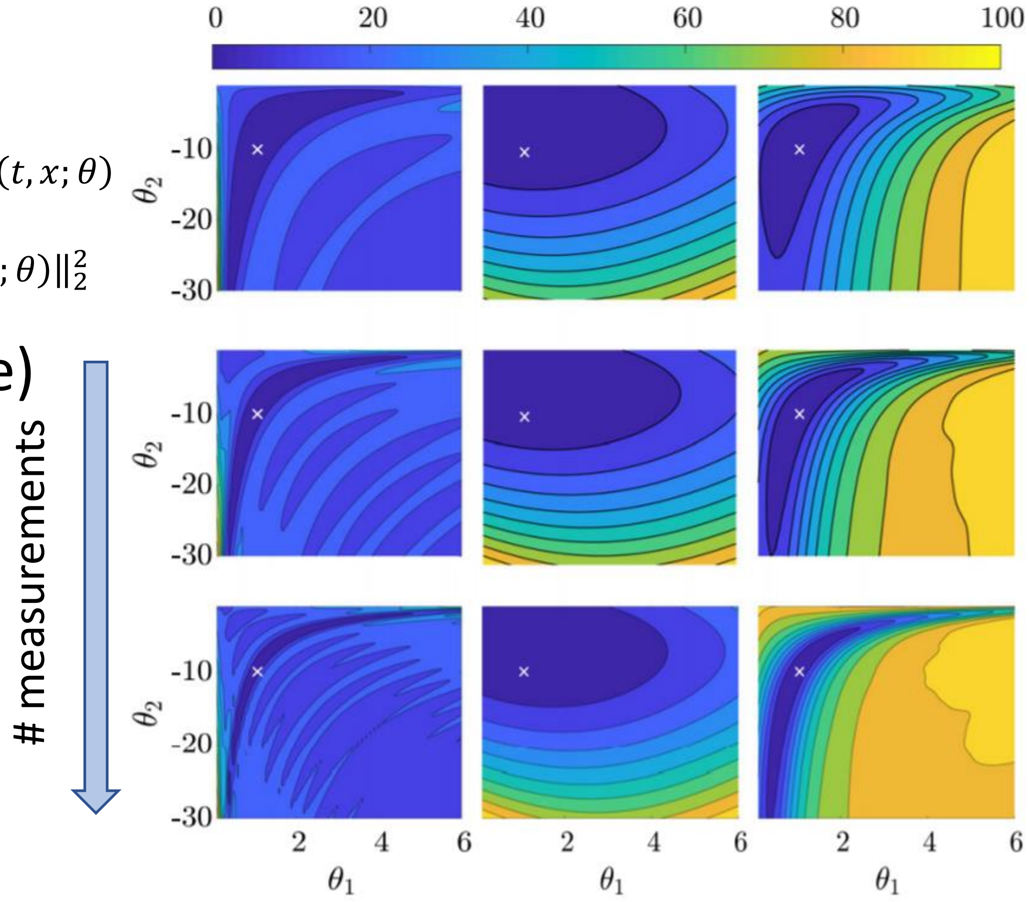
(b) Assumes noiseless measurements $J(\theta) = \sum_{i=1}^{n} \|y_i - \Psi(y_{i-1}; \theta)\|_2^2$

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

- Optimal combination of (a) and (b)

Regularization

- Sparse regularization
  - Lasso[1]
  - Ridge regression/Tikhonov regularization[2]
- Kernel-based
  - Stable spline/tuned-correlated[3]




(a) (b) (c)

ngalioto@umich.edu


1. Lin, W., Tong, T., Gao, Q., Guo, D., Du, X., Yang, Y., ... & Alzheimer's Disease Neuroimaging Initiative. (2018). Convolutional neural networks-based MRI image analysis for the Alzheimer's disease prediction from mild cognitive impairment. *Frontiers in neuroscience*, *12*, 777.
2. Liu, H., Miao, E. M., Wei, X. Y., & Zhuang, X. D. (2017). Robust modeling method for thermal error of CNC machine tools based on ridge regression algorithm. *International journal of machine tools and manufacture*, *113*, 35-48.
3. Pillonetto, G., Chen, T., Chiuso, A., De Nicolao, G., & Ljung, L. (2016). Regularized linear system identification using atomic, nuclear and kernel-based norms: The role of the stability constraint. *Automatica*, *69*, 137-149.

# Probabilistic Formulation

## Joint parameter-state estimation with stochastic dynamics

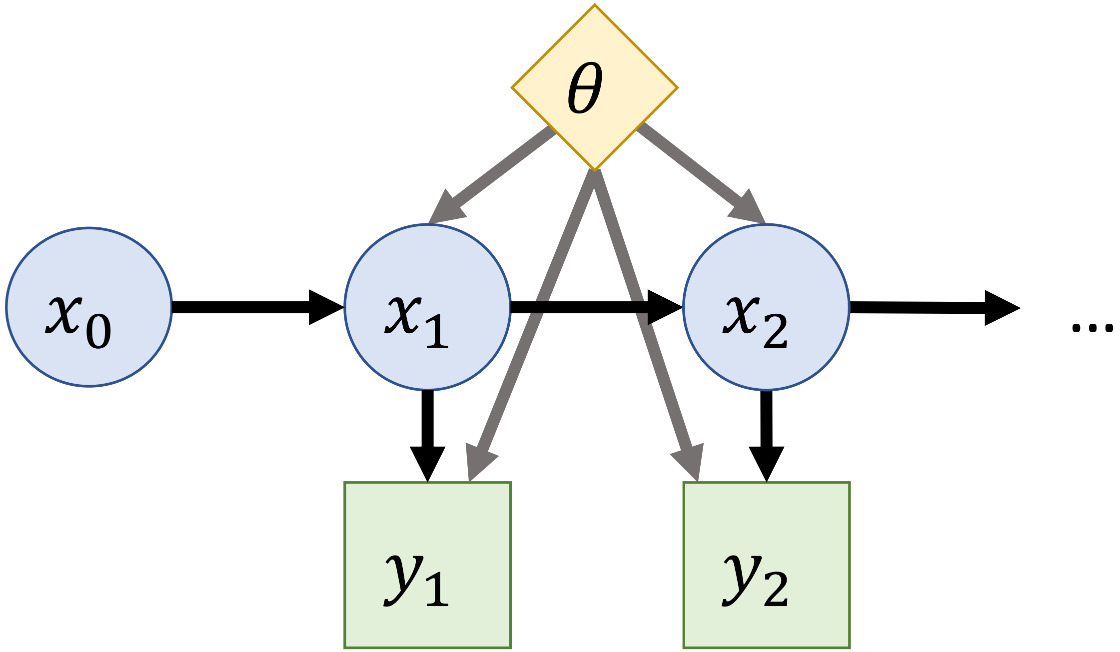
$$X_k \in \mathbb{R}^{d_x}, \qquad Y_k \in \mathbb{R}^{d_y}, \qquad \theta = (\theta_\Psi, \theta_h, \theta_\Sigma, \theta_\Gamma) \in \mathbb{R}^{d_\theta}$$

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

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

The process noise term $\xi_k$ accounts for model error

- Parameter error
- Integration error
- Insufficient model expressiveness




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

# 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:

$$y_k = x_k$$

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

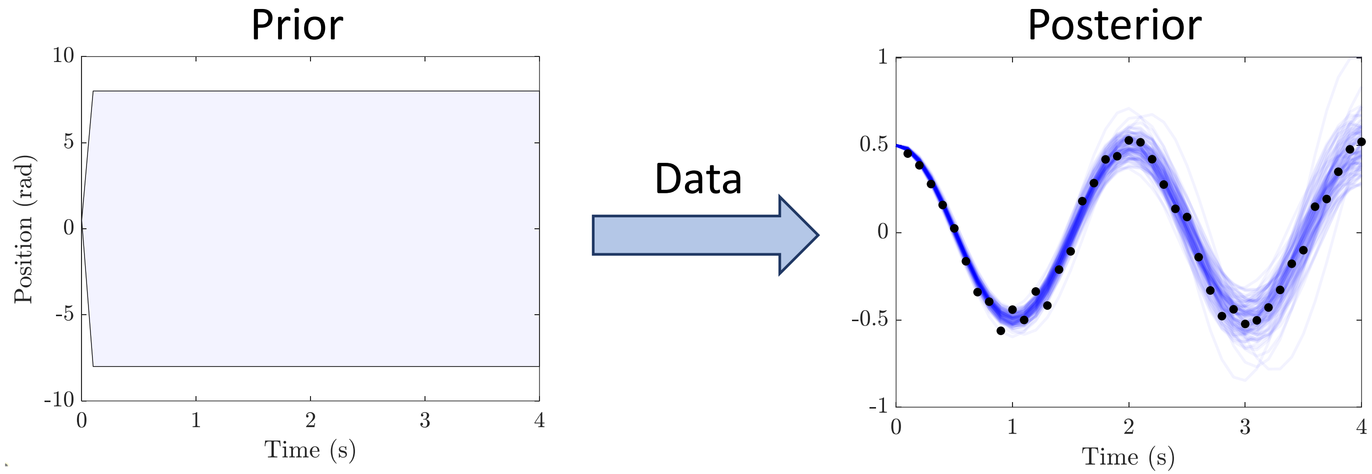
- Hills et al., 2015
- Qin et al., 2019
- Raissi, 2018

ngalioto@umich.edu

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).
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.

# Bayesian Inference

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




- 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

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

# Approximate Marginal Markov Chain Monte Carlo (MCMC) Algorithm (Särkkä, 2013)

1. **for** $i = 1, \dots, N$
2. Propose sample $\theta^*$

   <u>Evaluate posterior:</u>
3. for $\mathrm{k} = 1, \dots, n$
4. Predict: $p(X_{k+1}|\mathcal{Y}_k, \theta) = \int p(X_{k+1}|X_k, \theta) p(X_k|\mathcal{Y}_k, \theta) dX_k$
5. Update: $p(X_{k+1}|\mathcal{Y}_{k+1}, \theta) = \dfrac{p(y_{k+1}|X_{k+1}, \theta) p(X_{k+1}|\mathcal{Y}_k, \theta)}{p(y_{k+1}|\mathcal{Y}_k, \theta)}$
6. Marginalize: $\mathcal{L}(\theta; \mathcal{Y}_{k+1}) = \int p(y_{k+1}|X_{k+1}, \theta) p(X_{k+1}|\mathcal{Y}_k, \theta) dX_{k+1}$
7. **end for**
8. Accept $\theta^*$ with Metropolis-Hastings probability; otherwise reject
9. **end for**

Unscented Kalman Filter (steps 3–7)

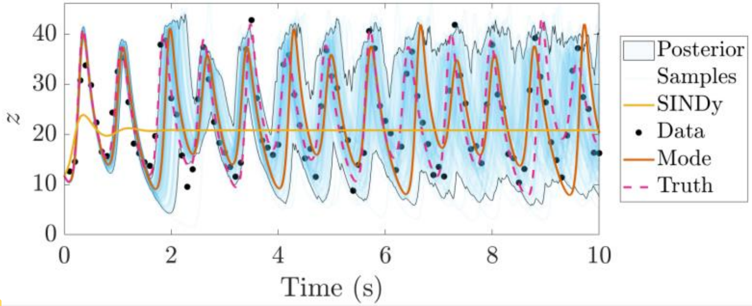
MCMC (steps 1–9)



Särkkä, S. (2013). *Bayesian filtering and smoothing* (No. 3). Cambridge University Press.

# Results: Lorenz '63

## Accounting for model error enhances robustness

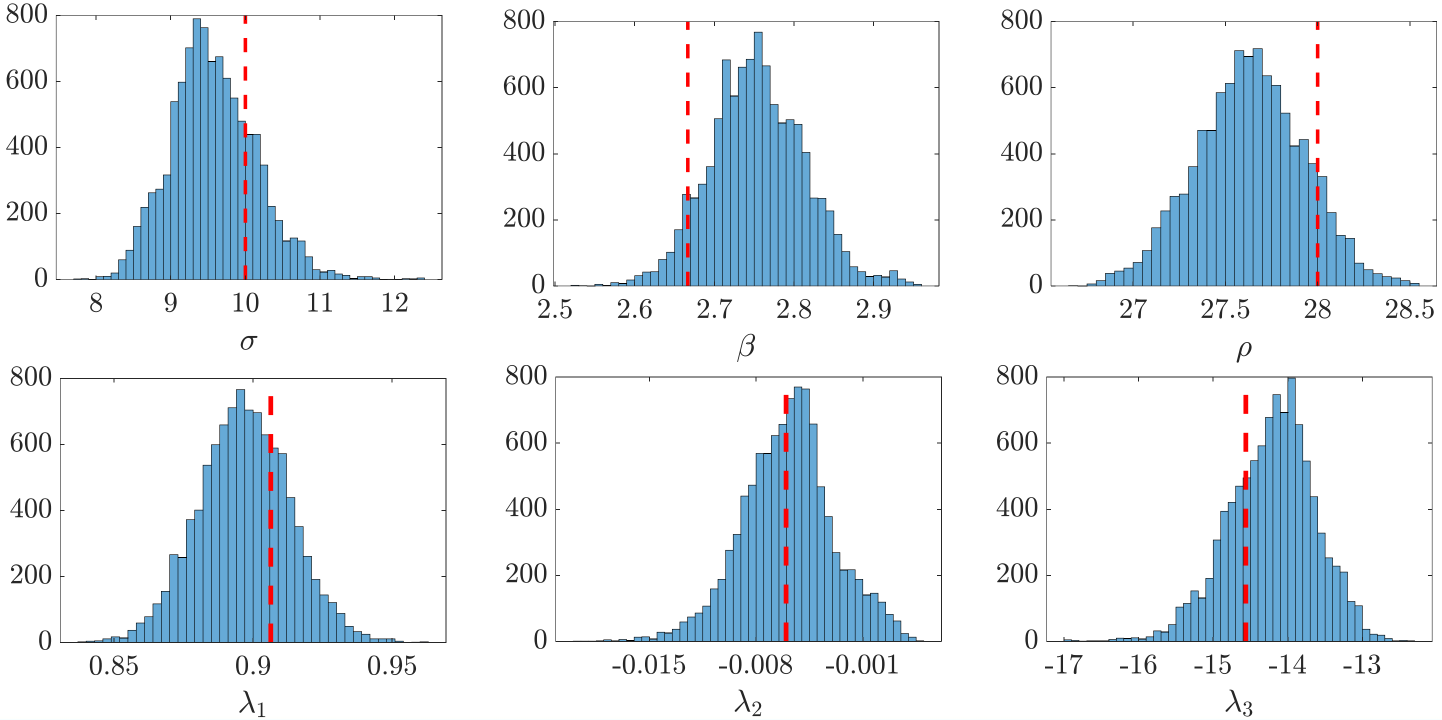


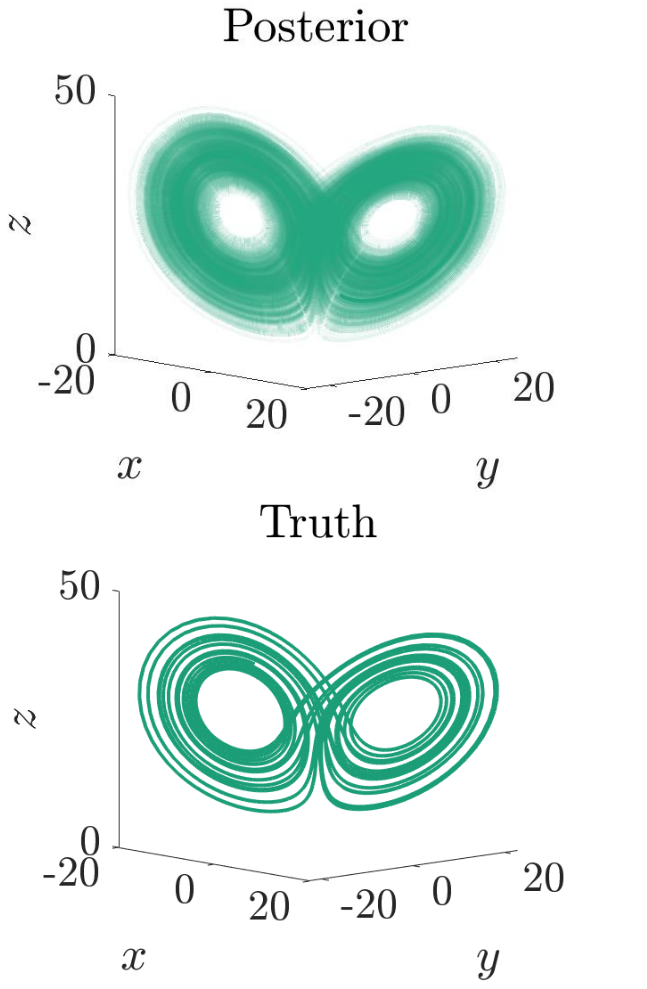

Most positive Lyapunov exponent: $\lambda_1 = 0.906$

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

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

$n = 100$
$\Delta t = 0.10s$
$\sigma_R = 2.0$



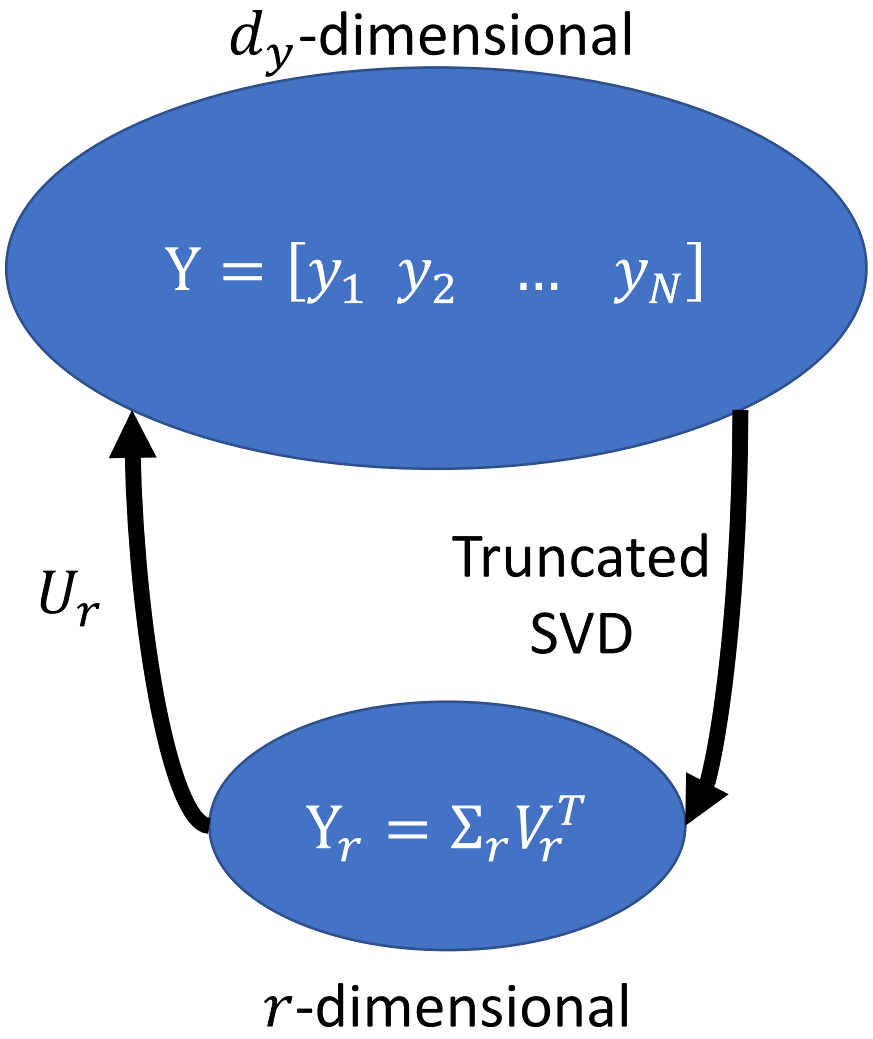

ngalioto@umich.edu


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*, 20514-20522.

# High-Dimensional Systems

Main idea: project data onto low-dimensional subspace with SVD

1. Collect all data into one matrix Y
$$Y = [y_1 \;\; y_2 \quad \dots \quad y_N] \in \mathbb{R}^{d_y \times N}$$
2. Take the SVD
$$Y = U\Sigma V^T$$
3. Truncate the SVD to rank $r \ll d_y$, and let the transformed data $Y_r$ be defined as
$$Y_r = \Sigma_r V_r^T \in \mathbb{R}^{r \times N}$$
4. Perform inference as usual with the transformed data
5. Map predictions back to high-dimensional space
$$\hat{y} = U_r \hat{y}_r$$



# Results: Kuramoto-Sivashinsky Equation

## Our approach can be applied to high-dimensional systems

$$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} - \frac{\partial^2 u}{\partial x^2} - \frac{\partial^4 u}{\partial x^4}$$

Most positive Lyapunov exponent[1]: $\lambda_1 = 0.088$

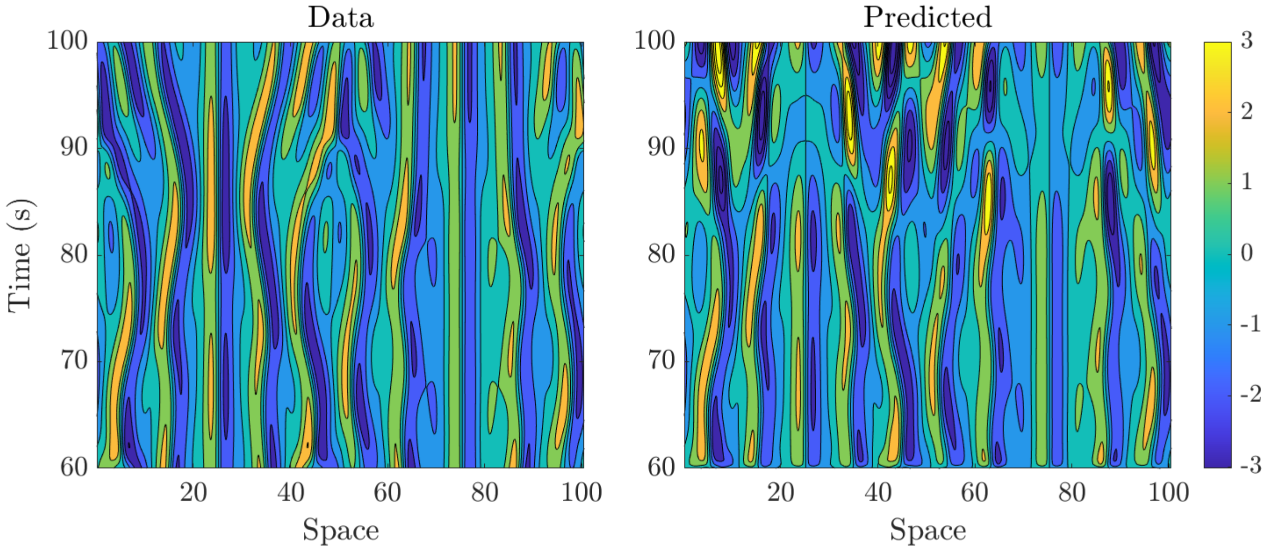
$d_x = 8$
$d_y = 1{,}024$

$$Q(\theta) = \theta_1 \frac{(x - x^T)^2}{\theta_2}$$

Training data: $n = 50$ from $60s$ to $79.6s$

$$NRMSE = \frac{RMSE}{y_{max} - y_{min}} \times 100\%$$

Training $NRMSE$: 2.63%




ngalioto@umich.edu

1. Edson, R. A., Bunder, J. E., Mattner, T. W., & Roberts, A. J. (2019). Lyapunov exponents of the Kuramoto-Sivashinsky PDE. *arXiv preprint arXiv:1902.09651*.
Data provided by:
Raissi, M. (2018). Deep hidden physics models: Deep learning of nonlinear partial differential equations. *The Journal of Machine Learning Research*, *19*(1), 932-955.

# Hamiltonian Systems

In mechanical systems, the Hamiltonian $\mathcal{H}$ is the sum of potential energy $U$ and kinetic energy $T$

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

$q$ generalized position
$p$ generalized momentum

Equations of motion are derived from the Hamiltonian

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

Hamiltonian systems have a number of physical properties

- Conservation
- Reversibility
- Symplecticness

# 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

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

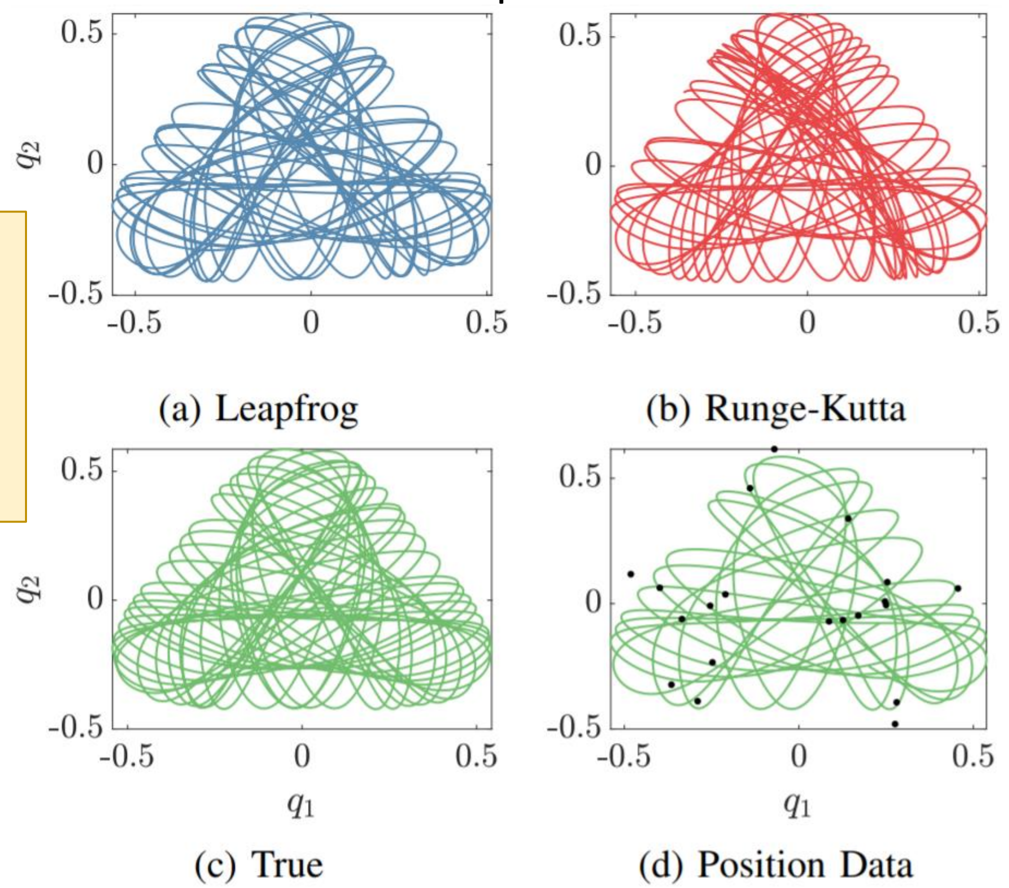
# Results: Hénon-Heiles

## The symplectic approach learns a more accurate Hamiltonian

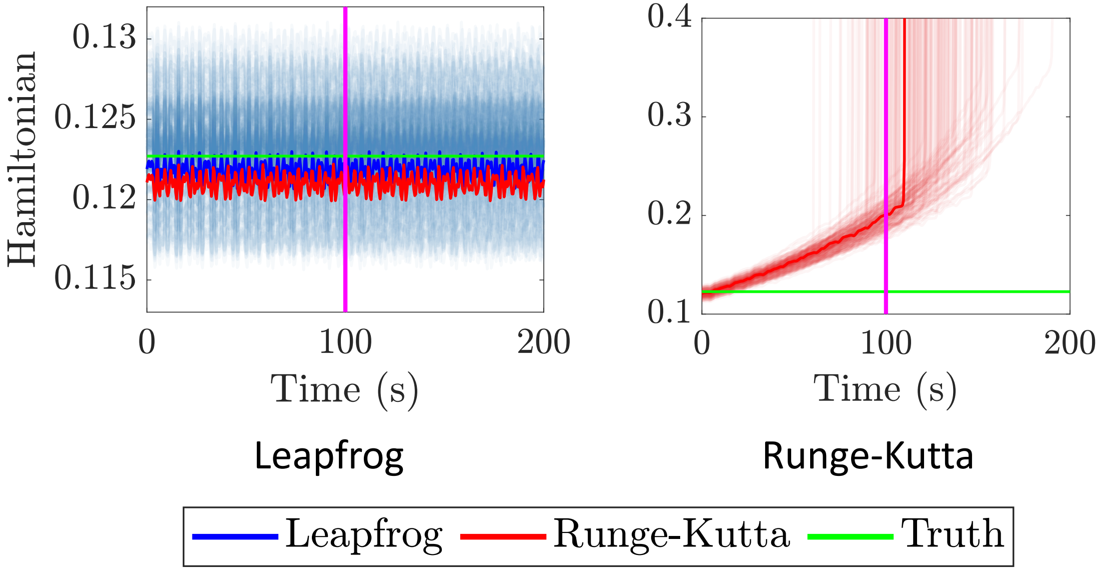
**Truth:** $U(q_1, q_2) = \frac{1}{2}q_1^2 + \frac{1}{2}q_2^2 + q_1^2 q_2 - \frac{1}{3}q_2^3$

**Data Generation:**
- $n = 20$
- $\Delta t = 5$
- $\sigma = 0.05$

Phase plots




Hamiltonian over time




Leapfrog    Runge-Kutta

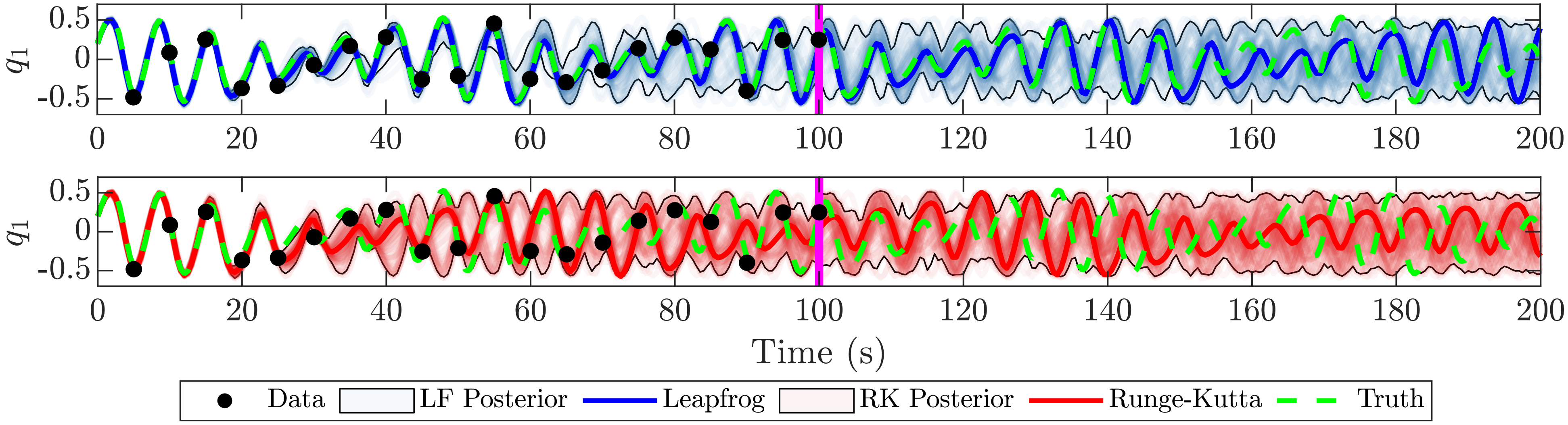
The method equipped with RK must learn a smaller Hamiltonian to compensate for being non-conservative

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

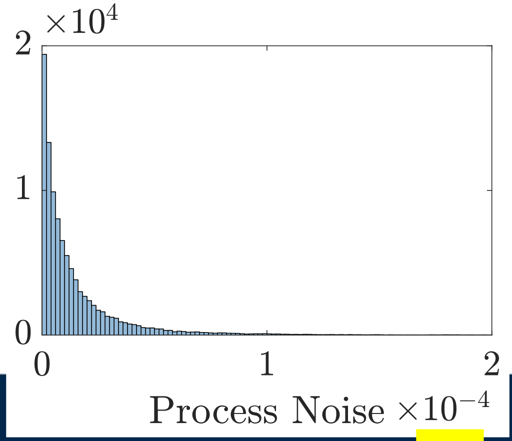
# Results: Hénon-Heiles

## The symplectic approach yields greater certainty

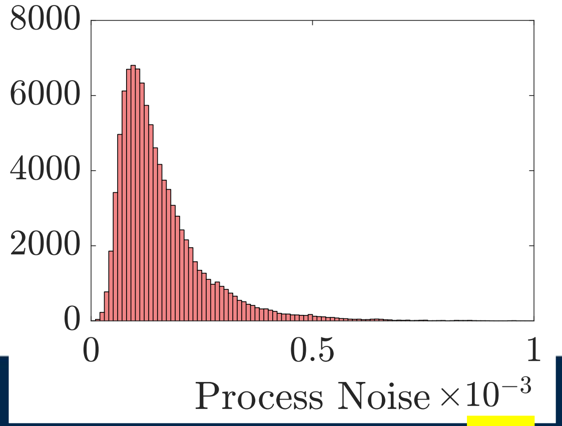
Posterior estimates of $q_1$ trajectory




Process noise marginal posteriors




Leapfrog




Runge-Kutta

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

# 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 of a Hamiltonian system 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


ngalioto@umich.edu


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.