Accounting for Model Errors in Probabilistic Linear Identification of Nonlinear PDE Systems

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

Department of Aerospace Engineering, University of Michigan – Ann Arbor

16th United States National Congress on Computational Mechanics

July 27, 2021

Motivation

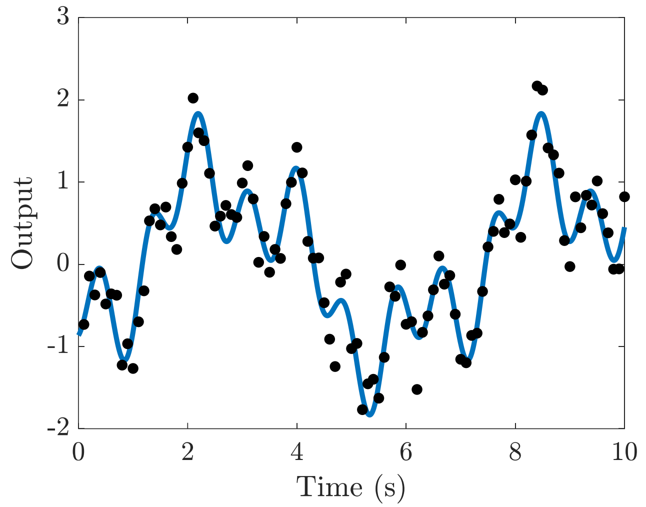
Objective: learn a model of a dynamical system from data

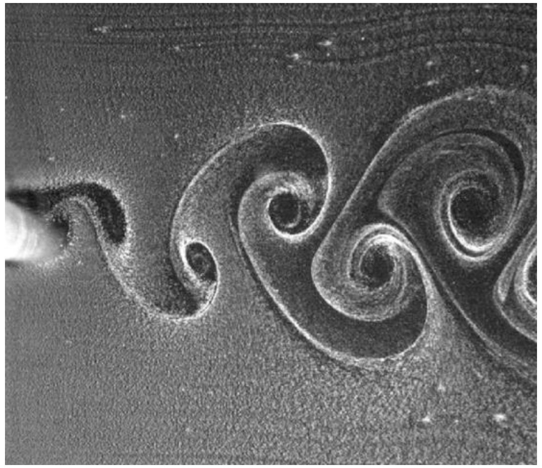
Two primary design choices in system identification:

- 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

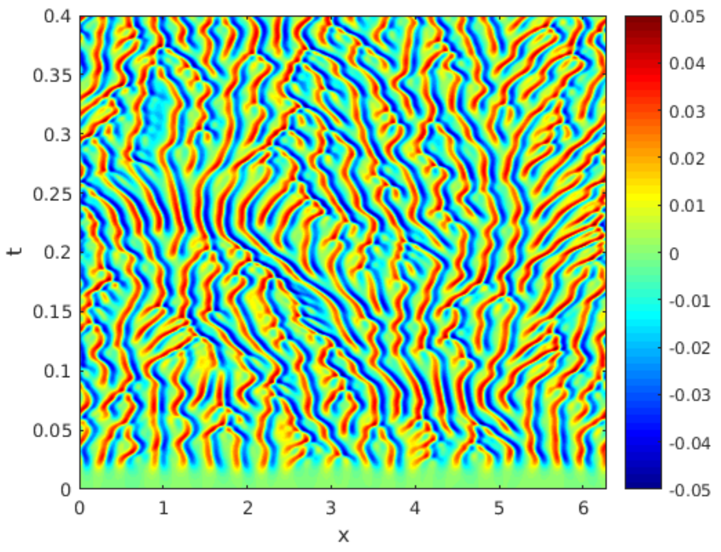




Ljung, L. (1999). System identification. *Wiley encyclopedia of electrical and electronics engineering*, 1-19.
Schoukens, J., & Ljung, L. (2019). Nonlinear system identification: A user-oriented road map. *IEEE Control Systems Magazine*, *39*(6), 28-99.

Motivation

- Linear approximations are commonly used to reduce the computational burden of evaluating a model
  - Estimation of high-dimensional models can be intractable
  - Real-time prediction/control requires inexpensive models
- Estimating a model that does not match the underlying system introduces uncertainty into the problem
- Common methods struggle when this model uncertainty is accompanied with measurement noise
- Good estimation requires proper management of (1) model, (2) measurement, and (3) parameter uncertainty



Gousami, et. al

Gouasmi, A., Parish, E., & Duraisamy, K. (2016). A Priori Estimation Of Memory Effects In Coarse-Grained Nonlinear Systems Using The Mori-Zwanzig Formalism. *arXiv preprint arXiv:1611.06277*.

Outline

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

Existing Approaches

Least squares-based objective functions

(a) Assumes perfect model

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

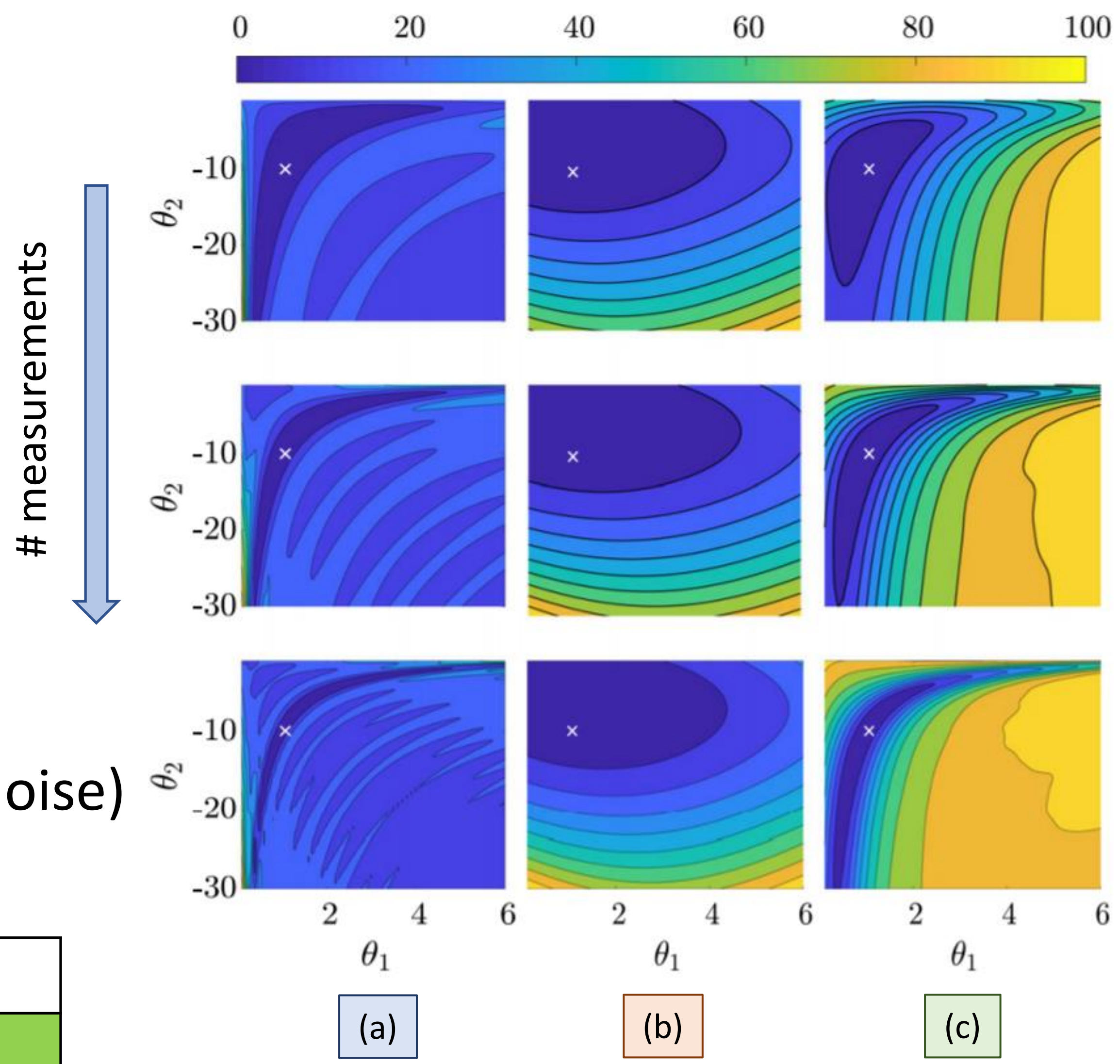
(b) Assumes noiseless measurements

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

(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	✓	×	✓



Probabilistic Formulation

Joint parameter-state estimation with stochastic dynamics

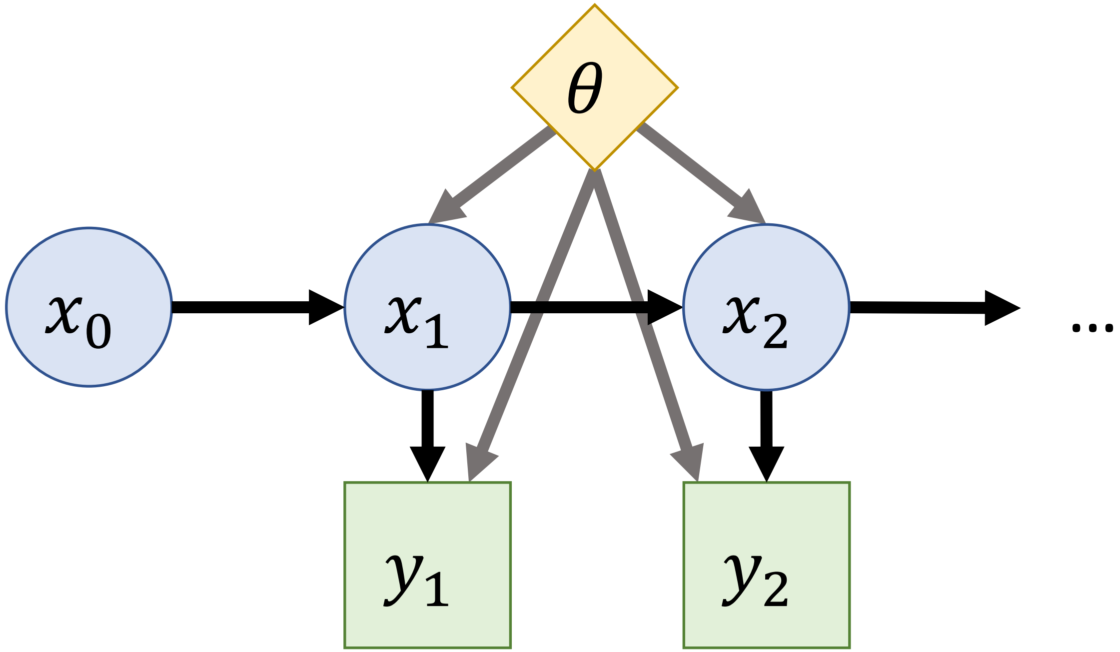
$$X_k \in \mathbb{R}^{d_x}, \qquad Y_k \in \mathbb{R}^{d_y}, \qquad \theta = (\theta_\Psi, \theta_h, \theta_\Sigma, \theta_\Gamma) \in \mathbb{R}^{d_\theta}$$

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

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

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

Probabilistic Formulation (Linear)

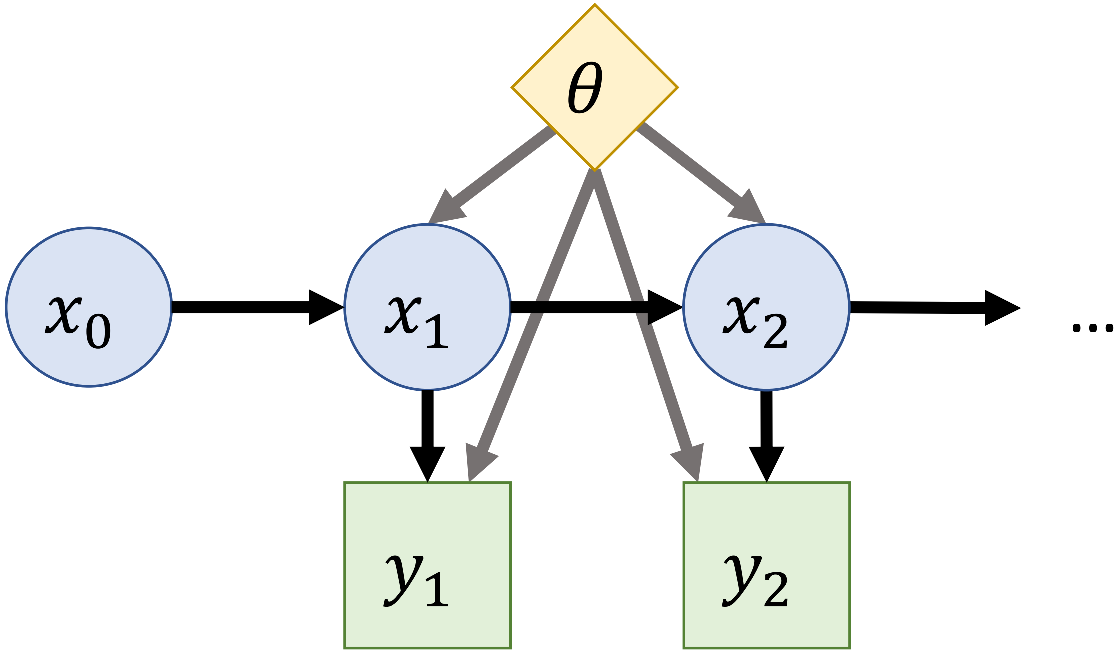
Joint parameter-state estimation with stochastic dynamics

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

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

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

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

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:

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

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

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

Identity observations:

$$y_k = x_k$$

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

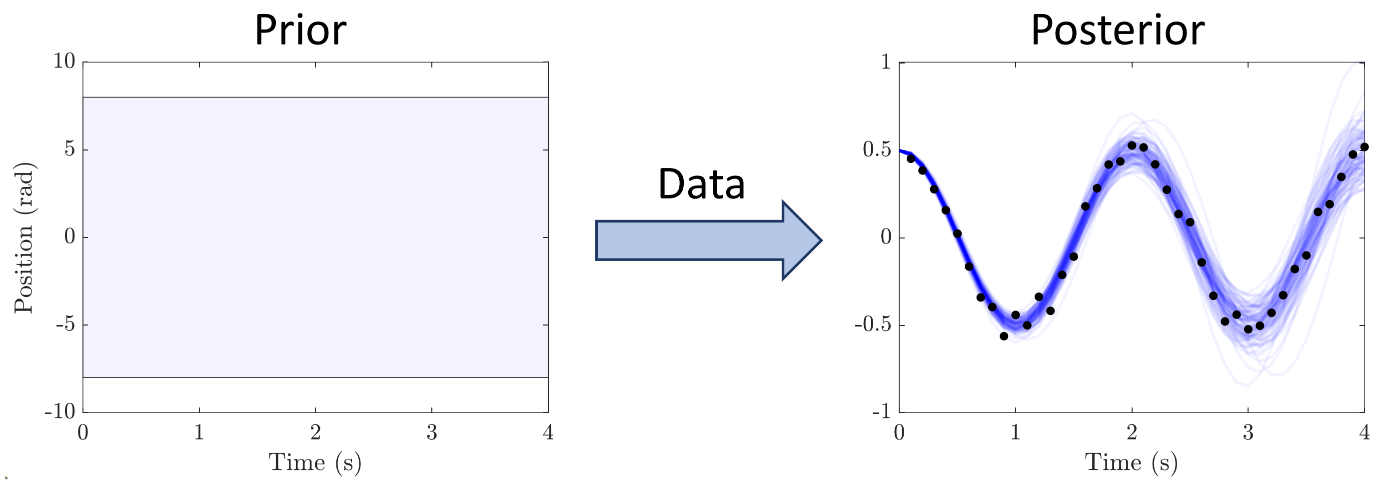
- Hills et al., 2015
- Qin et al., 2019
- Raissi, 2018

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).
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.

Bayesian Inference

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



- 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

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

ngalioto@umich.edu

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

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

   <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: $p(X_{k+1}|\mathcal{Y}_{k+1}, \theta) = \frac{p(y_{k+1}|X_{k+1}, \theta) p(X_{k+1}|\mathcal{Y}_k, \theta)}{p(y_{k+1}|\mathcal{Y}_k, \theta)}$
6. Marginalize: $\mathcal{L}_{k+1}(\theta; \mathcal{Y}_{k+1}) = \int p(y_{k+1}|X_{k+1}, \theta) p(X_{k+1}|\mathcal{Y}_k, \theta) dX_{k+1}$
7. **end for**
8. Accept θ^* with Metropolis-Hastings probability; otherwise reject
9. **end for**

Kalman Filter (steps 3–7)

MCMC (steps 1–9)

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

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

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

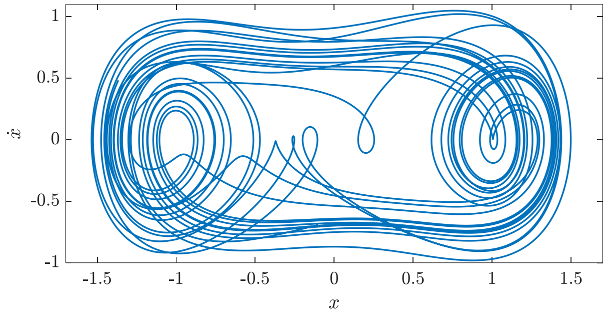
This objective prioritizes:

- 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

Duffing Oscillator with Forcing

Our method resists over-fitting



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

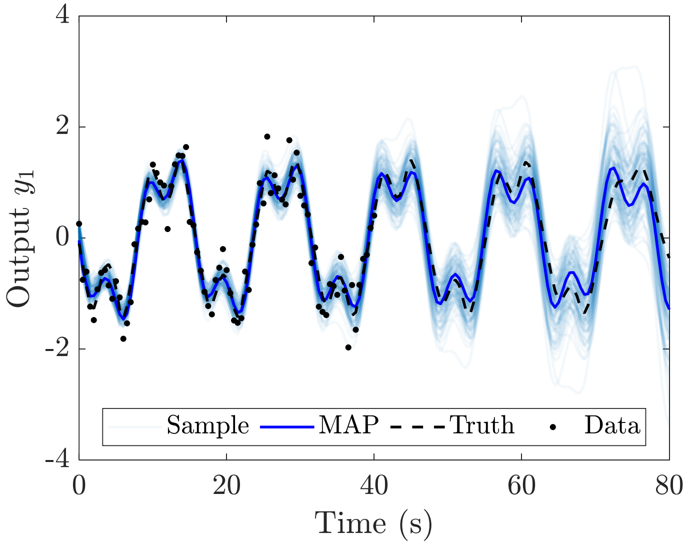
$\alpha = 1, \delta = -0.3, \beta = -1, \gamma = 0.5, \omega = 1.2$

Chaotic solution[1]

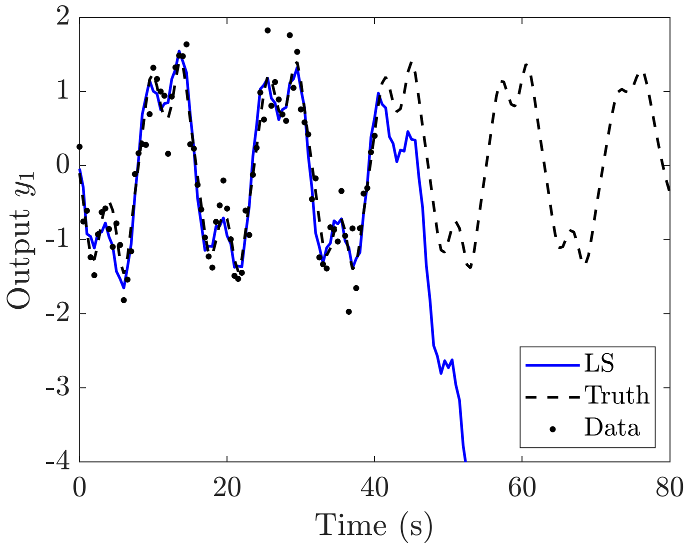
We estimate $A(\theta)$, $B(\theta)$, $H(\theta)$, and $x_0(\theta)$ with dimension $d_x = 6$

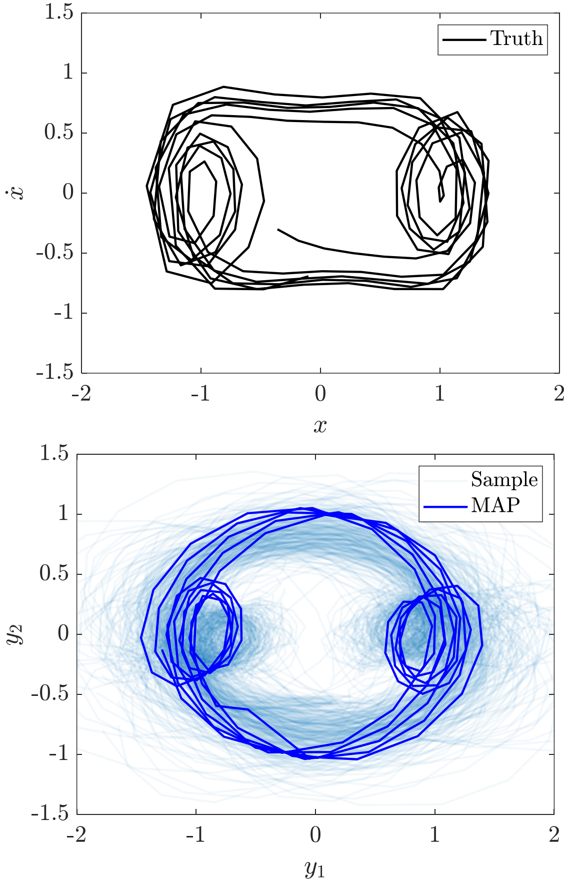
Data:
$\Delta t = 0.5$
$\sigma = 0.3$
$n = 80$

Posterior Objective



Least Squares Objective





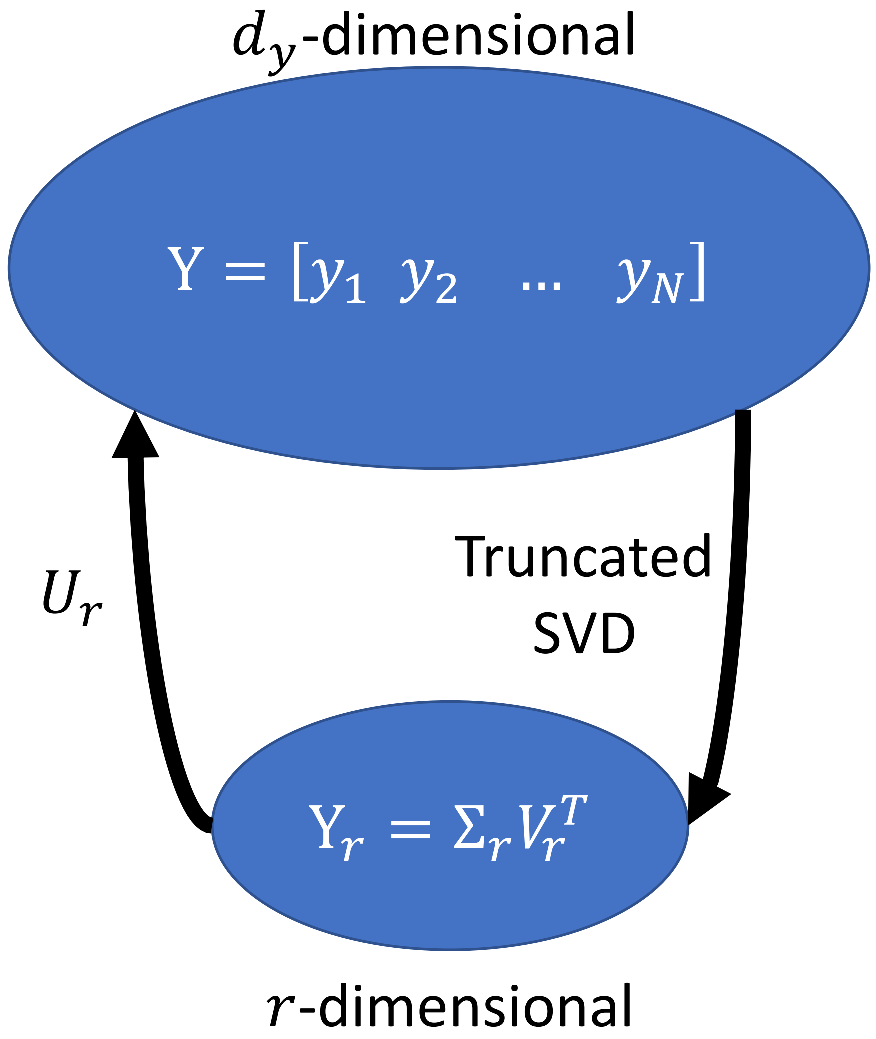
ngalioto@umich.edu

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

High-Dimensional Systems

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

1. Collect all data into one matrix Y
$$\mathrm{Y} = [y_1 \;\; y_2 \;\; \dots \;\; y_N] \in \mathbb{R}^{\mathrm{d}_y \times N}$$
2. Take the SVD
$$\mathrm{Y} = U\Sigma V^T$$
3. Truncate the SVD to rank $r \ll d_y$, and let the transformed data Y_r be defined as
$$\mathrm{Y}_r = \Sigma_r V_r^T \in \mathbb{R}^{r \times N}$$
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

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

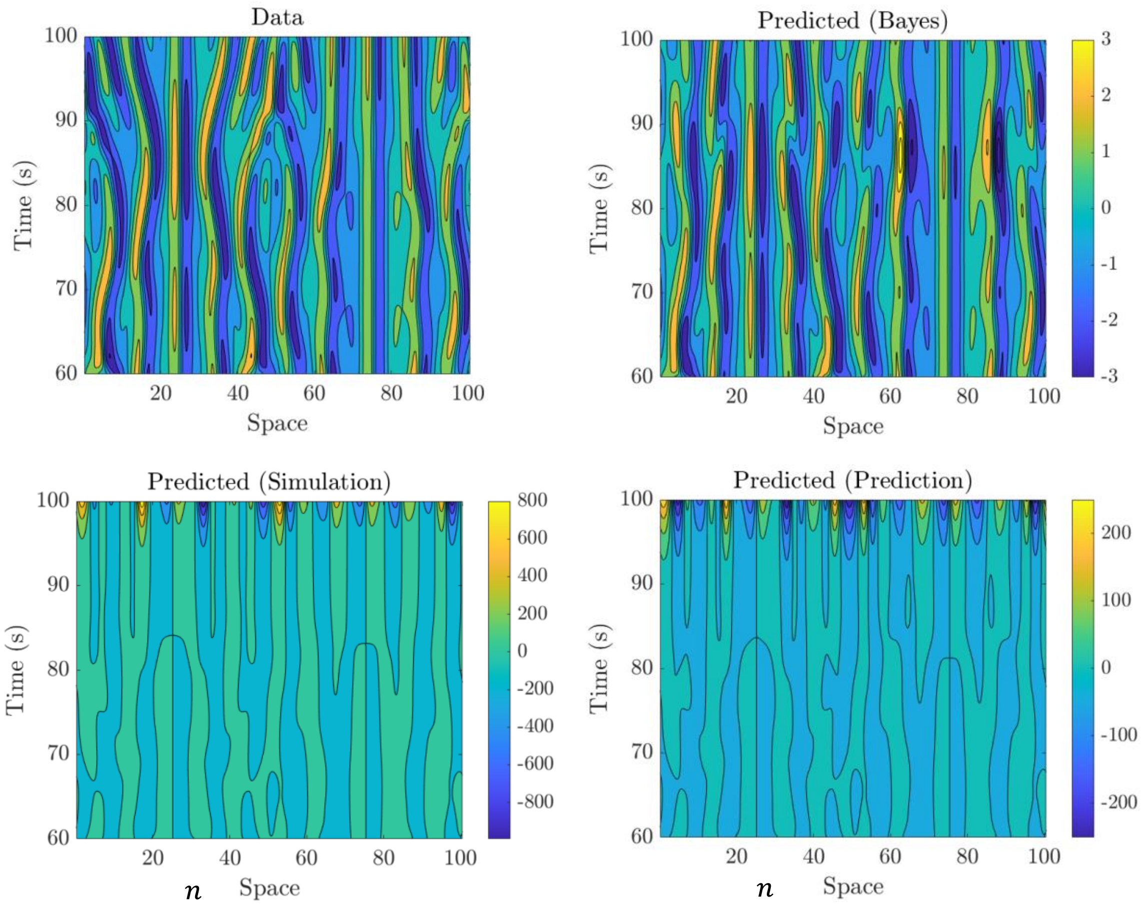
Most positive Lyapunov exponent[1]: $\lambda_1 = 0.088$

Training data[2]: $n = 50$ from $60s$ to $79.6s$

$d_x = 8$

$d_y = 1{,}024$

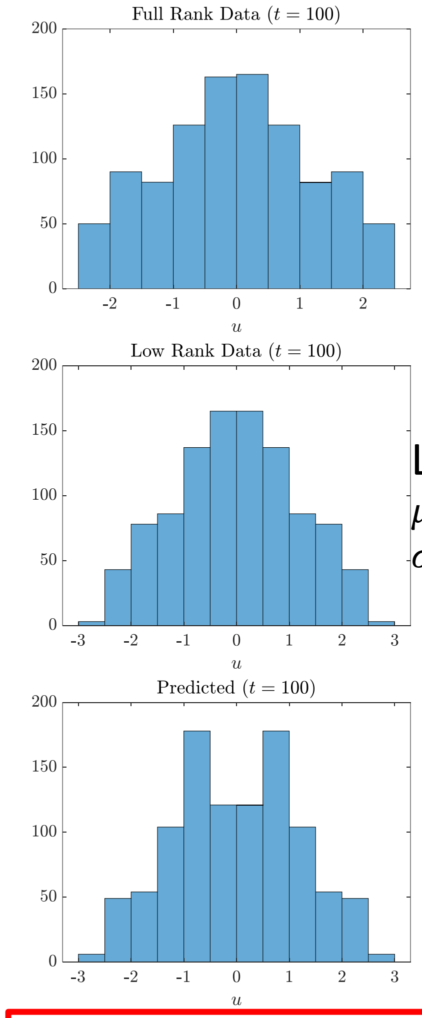
Least squares objectives yield over-fit models. Note the colorbar scale!



$$J(\theta) = \sum_{k=1}^{n} \left\| y_k - A^k(\theta)x_0 \right\|_2^2$$

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

Histograms of $u(t = 100)$



Full rank data:
$\mu = 1.20 \times 10^{-16}$
$\sigma^2 = 1.3972$

Low rank data:
$\mu = -1.98 \times 10^{-16}$
$\sigma^2 = 1.3608$

Predicted:
$\mu = 2.22 \times 10^{-18}$
$\sigma^2 = 1.3924$

Our model accurately predicts the statistical properties

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:
2. Raissi, M. (2018). Deep hidden physics models: Deep learning of nonlinear partial differential equations. *The Journal of Machine Learning Research*, *19*(1), 932-955.

Main Takeaways

- Optimally accounting for different types of uncertainty can lead to robustness even when data are few and/or noisy
- Modeling deterministic systems with stochastic models introduces built-in regularization and optimization benefits

Related Works

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. (2021, May). A New Objective for Identification of Partially Observed Linear Time-Invariant Dynamical Systems from Input-Output Data. In *Learning for Dynamics and Control* (pp. 1180-1191). PMLR.

Funding

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