# Overcoming Model Uncertainty: Integrating machine learning tools into parametric models

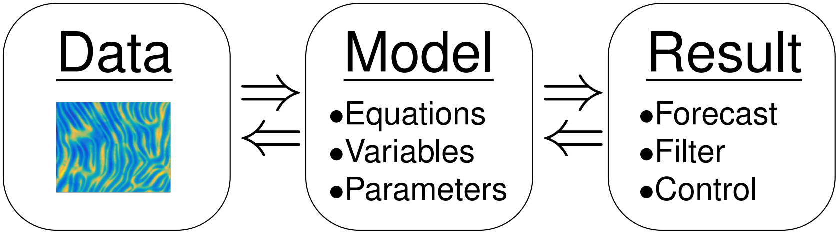
Tyrus Berry

George Mason University

Feb. 21, 2019

Joint work with Franz Hamilton, Tim Sauer, John Harlim (PSU) and Dimitris Giannakis (NYU)

## Parametric Modeling



- **Design Model:** Limited resolution and complexity
- **Assimilate Data:** Fit Parameters/Variables
  - Kalman Filter, EKF, EnKF, Variational methods
  - Observability and noise
  - Model error
- **Study/Apply:** Ensemble Forecast

## WHAT IS THE FILTERING PROBLEM?

- Consider a discrete time dynamical system:

$$\begin{aligned} x_k &= f_k(x_{k-1}, \omega_k) \\ y_k &= h_k(x_k, \nu_k) \end{aligned}$$

- Where $x_k$ is the state variable, $\omega_k$ is stochastic forcing, and the maps $f_k$ define the dynamics
- The maps $h_k$ are called the observation functions, $\nu_k$ is observation noise, and $y_k$ is a noisy observation

## WHAT IS THE FILTERING PROBLEM?

- Consider a discrete time dynamical system:

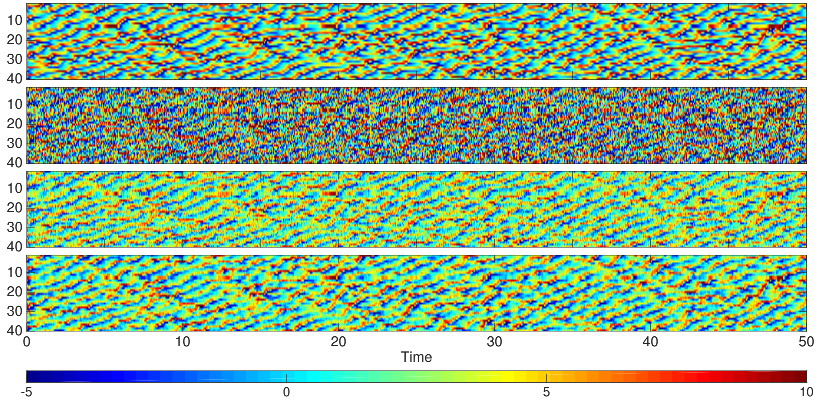
$$\begin{aligned} x_k &= f_k(x_{k-1}, \omega_k) \\ y_k &= h_k(x_k, \nu_k) \end{aligned}$$

- Given the observations $y_1, ..., y_k$ we define three problems:

- **Filtering:** Estimate the current state $p(x_k \mid y_1, ..., y_k)$
- **Forecasting:** Estimate a future state $p(x_{k+\ell} \mid y_1, ..., y_k)$
- **Smoothing:** Estimate a past state $p(x_{k-\ell} \mid y_1, ..., y_k)$

## WHAT IS THE FILTERING PROBLEM?

$$\frac{dx^i}{dt} = -x^{i-2}x^{i-1} + x^{i-1}x^{i+1} - x^i + F$$



## TWO STEP FILTERING TO FIND $p(x_k \mid y_1, ..., y_k)$

- Assume we have $p(x_{k-1} \mid y_1, ..., y_{k-1})$
- **Forecast Step:** Find $p(x_k \mid y_1, ..., y_{k-1})$
- **Assimilation Step:** Perform a Bayesian update,

$$p(x_k \mid y_1, ..., y_k) \propto p(x_k \mid y_1, ..., y_{k-1}) p(y_k \mid x_k, y_1, ..., y_{k-1})$$

**Posterior** $\propto$ **Prior** $\times$ **Likelihood**

## Kalman Filtering: An Intuitive Idea

Filter tracks two things

1. Estimate of state $x$ over time
2. Uncertainty of state estimate, covariance matrix $P$

These two statistics define a multivariate Gaussian.

1. **Predict** an estimate of state ($x_k^-$) and covariance ($P_k^-$)
2. **Observe** data $y_k$
3. **Correct** the mean ($x_k^+$) and covariance ($P_k^+$)

## Kalman Filtering: An Intuitive Idea

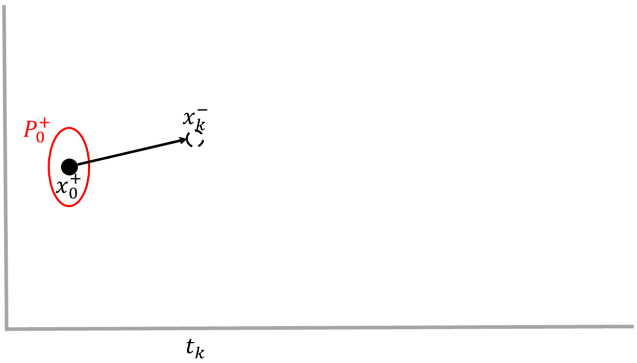
**Initialize!**

## Kalman Filtering: An Intuitive Idea

**Initialize!**

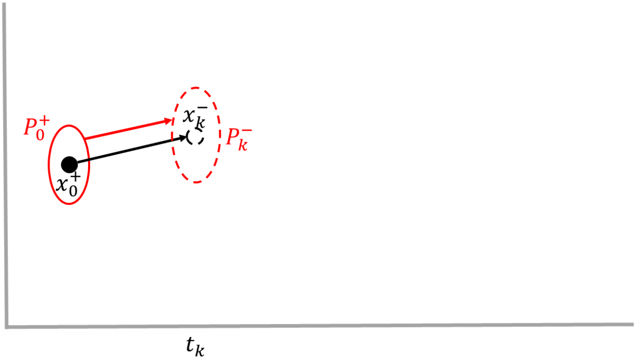
## Kalman Filtering: An Intuitive Idea

**Predict!**



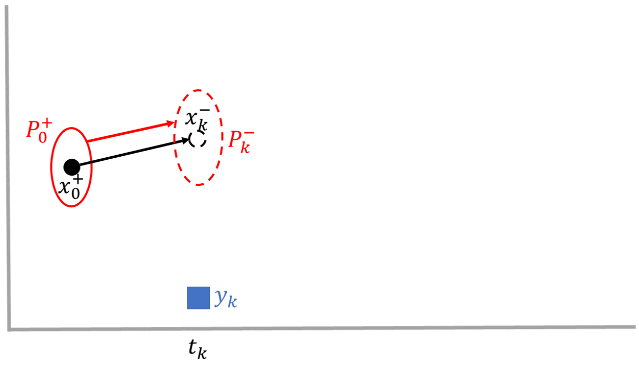
## Kalman Filtering: An Intuitive Idea

**Predict!**



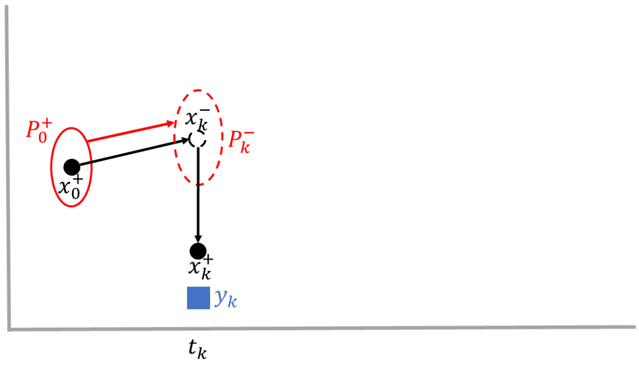
## Kalman Filtering: An Intuitive Idea

**Observe!**



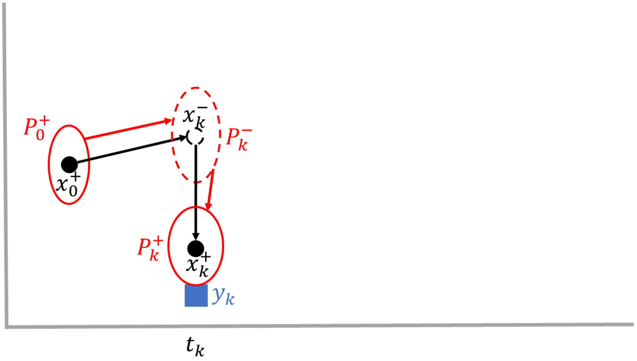
## Kalman Filtering: An Intuitive Idea

**Correct!**



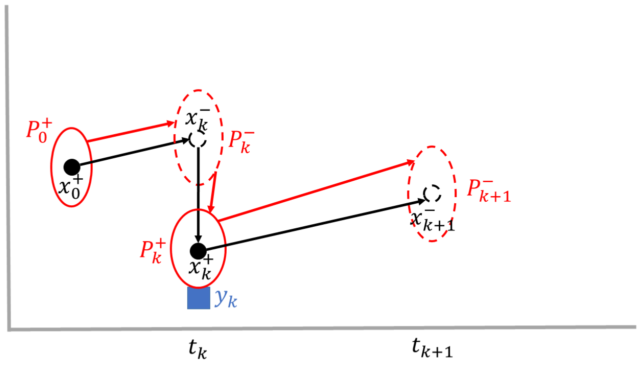
## Kalman Filtering: An Intuitive Idea

**Correct!**



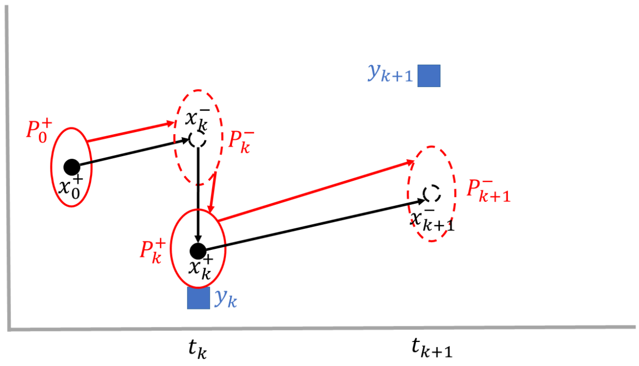
## Kalman Filtering: An Intuitive Idea

Predict!



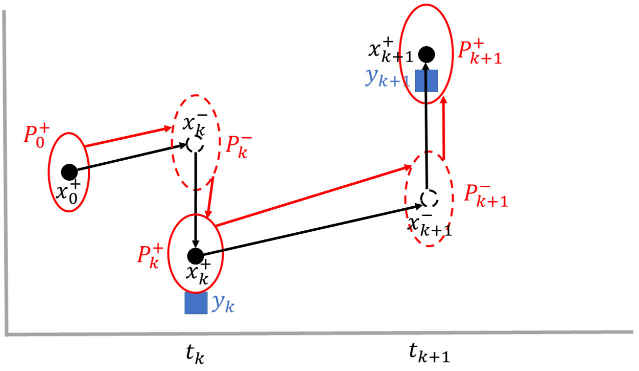
## Kalman Filtering: An Intuitive Idea

**Observe!**

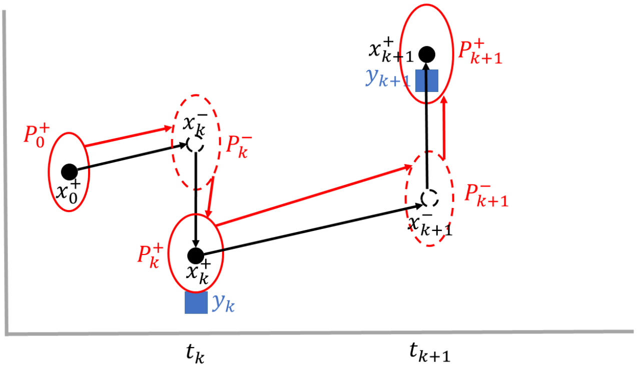


## Kalman Filtering: An Intuitive Idea

**Correct!**



## Kalman Filtering: An Intuitive Idea



And so on, and so on, and so on...

## WHAT ABOUT NONLINEAR SYSTEMS?

- Consider a system of the form:

$$\begin{array}{rcl} x_{k+1} & = & f(x_k) + \omega_{k+1} \qquad \omega_{k+1} \sim \mathcal{N}(0, Q) \\ y_{k+1} & = & h(x_{k+1}) + \nu_{k+1} \qquad \nu_{k+1} \sim \mathcal{N}(0, R) \end{array}$$

- **Extended Kalman Filter (EKF)**:
  - Linearize $F_k = Df(\hat{x}_k^a)$ and $H_k = Dh(\hat{x}_k^f)$
- **Ensemble Kalman Filter (EnKF)**:
  - Implicit linearization via ensemble forecast

## Kalman Filter: Assimilation Step

- **Kalman Equations:** (after some linear algebra...)
  - Kalman Gain: $K_k = P_k^f H_k^\top (P_k^y)^{-1}$
  - Innovation: $\epsilon_k = y_k - y_k^f$
  - Posterior Mean: $\hat{x}_k^a = \hat{x}_k^f + K_k \epsilon_k$
  - Posterior Covariance: $P_k^a = (I - K_k H_k) P_k^f$
- $\hat{x}_k^a$ is the least squares/minimum variance estimator of $x_k$

## PARAMETER ESTIMATION

- When the model has parameters $p$,

$$x_{k+1} = f(x_k, p) + \omega_{k+1}$$

- Can *augment* the state $\tilde{x}_k = [x_k, p_k]$
- Introduce trivial dynamics for $p$

$$\begin{aligned} x_{k+1} &= f(x_k, p_k) + \omega_{k+1} \\ p_{k+1} &= p_k + \omega^p_{k+1} \end{aligned}$$

- Need to tune the covariance of $\omega^p_{k+1}$

## Example of Parameter Estimation

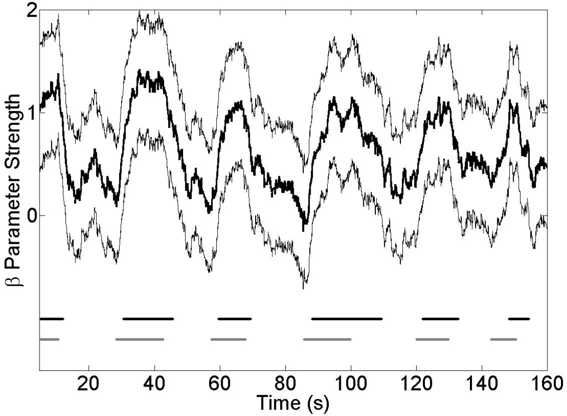
Consider the Hodgkin-Huxley neuron model, expanded to a network of *n* equations

$$\begin{aligned}
\dot{V}_i &= -g_{Na}m^3h(V_i - E_{Na}) - g_K n^4(V_i - E_K) - g_L(V_i - E_L) \\
&\quad + I + \sum_{j\neq i}^{n} \Gamma_{\mathrm{HH}}(V_j)V_j \\
\dot{m}_i &= a_m(V_i)(1-m_i) - b_m(V_i)m_i \\
\dot{h}_i &= a_h(V_i)(1-h_i) - b_h(V_i)h_i \\
\dot{n}_i &= a_n(V_i)(1-n_i) - b_n(V_i)n_i \\
\Gamma_{\mathrm{HH}}(V_j) &= \beta_{ij}/(1+e^{-10(V_j+40)})
\end{aligned}$$

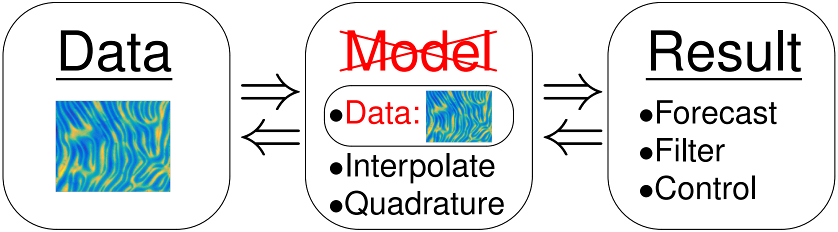
Only observe the voltages $V_i$, recover the hidden variables and the connection parameters $\beta$

## Example of Parameter Estimation

Can even turn connections on and off (grey dashes)
Variance estimate $\Rightarrow$ statistical test (black dashes)



## NONPARAMETRIC MODELING



- **Data IS the model:**
  - Assume a model exists
    - Data lies on/near an unknown sub-manifold
    - Data obeys an unknown dynamical system
  - Represent the model using training data

## KALMAN-TAKENS FILTER

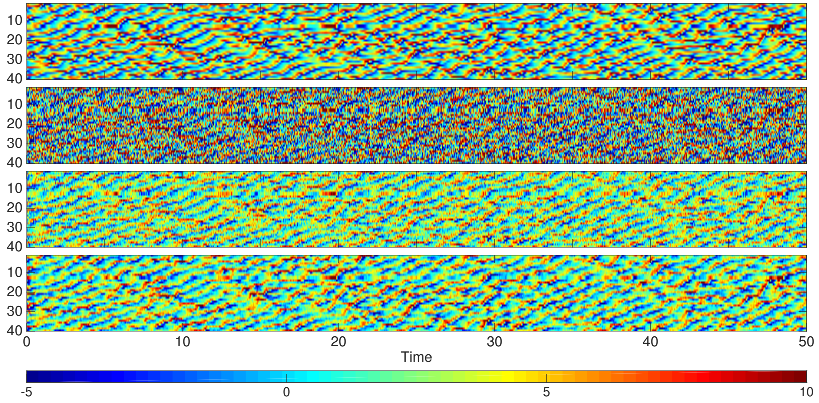
- Linearize $F_k = Df(\hat{x}_k^a)$ and $H_k = Dh(\hat{x}_k^f)$
- Given training data $y_k$, reconstruct state $z_k \sim x_k$ using Takens' delay embedding

$$z_k = [y_k, y_{k-1}, \ldots, y_{k-d}]$$

- Given a new state $y$, build $z$ and find nearest neighbors in $z$-space
- Estimate $F_k$ using local linear regression on training data

## Kalman-Takens vs. True Model

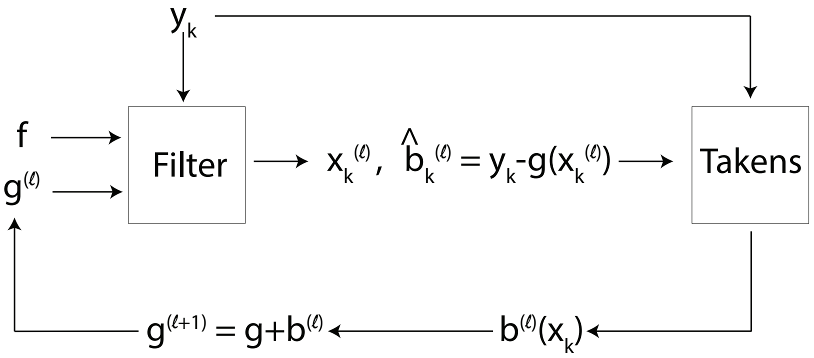
$$\frac{dx^i}{dt} = -x^{i-2}x^{i-1} + x^{i-1}x^{i+1} - x^i + F$$



## CORRECTING BIAS IN THE OBSERVATIONS

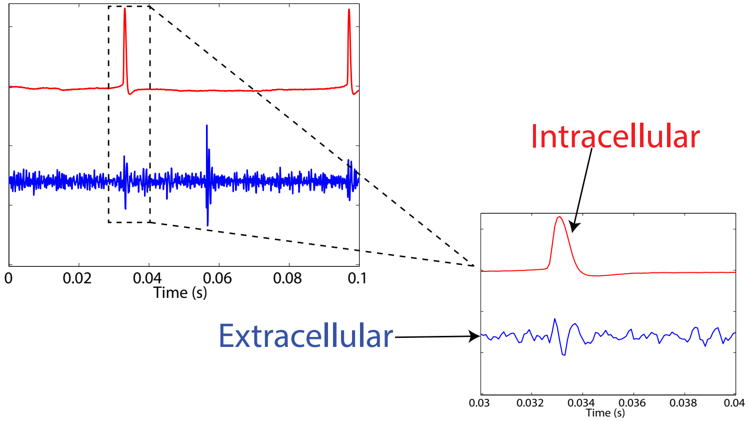
- Given training data $y_k$, run filter with given obs, $h$
- Estimate obs bias by $b(x_k) = y_k - h(x_k)$
- Estimate $b(x)$ using Takens + local linear regression
- Correct the obs functions, $g(x) = h(x) + b(x)$
- Rerun filter and estimate bias again, then repeat

## Bias Correction Method


$y_k$
$f$
$g^{(\ell)}$
Filter
$x_k^{(\ell)}, \ \hat{b}_k^{(\ell)} = y_k - g(x_k^{(\ell)})$
Takens
$b^{(\ell)}(x_k)$
$g^{(\ell+1)} = g + b^{(\ell)}$

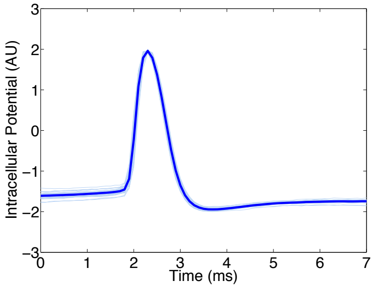
## Reconstructing Intracellular Potential

Mapping from intracellular to extracellular is complex.

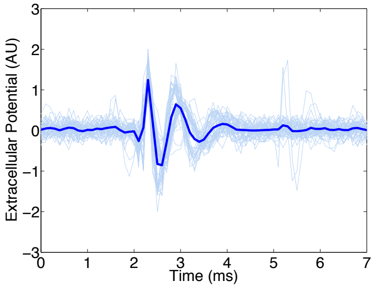


## Reconstructing Intracellular Potential

Models such as FHN similar to intracellular waveform



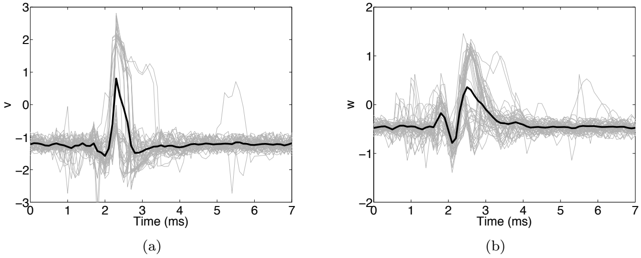
(a)

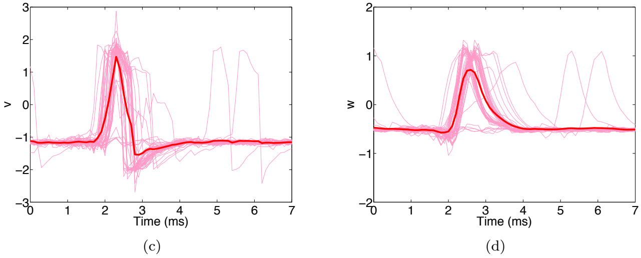


(b)

Henze D, Harris K, Borhegyi Z, Csicsvari J, Mamiya A, Hirase H, et al.. Simultaneous intracellular and extracellular recordings from hippocampus region CA1 of anesthetized rats; 2009. CRCNS.org.

## Apply bias correction with FHN model


(a) (b)

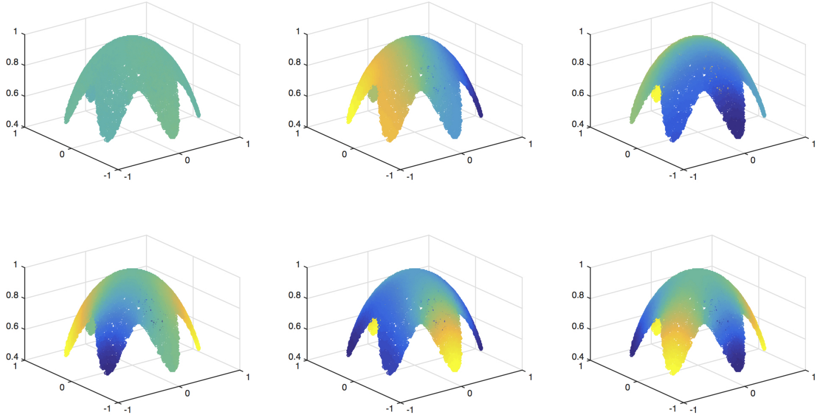

(c) (d)

## What is Manifold Learning?

- **Manifold learning $\Leftrightarrow$ Estimating Laplace Operator**
- Euclidean space:
  - Eigenfunctions of $\Delta$ are $e^{i\vec{\omega}\cdot\vec{x}}$
  - Basis for Fourier transform
- Unit circle:
  - Eigenfunctions of $\Delta$ are $e^{ik\theta}$
  - Basis for Fourier series
- **Theorem**: Eigenfunctions of $\Delta$ give the smoothest basis for square integrable functions on any manifold.

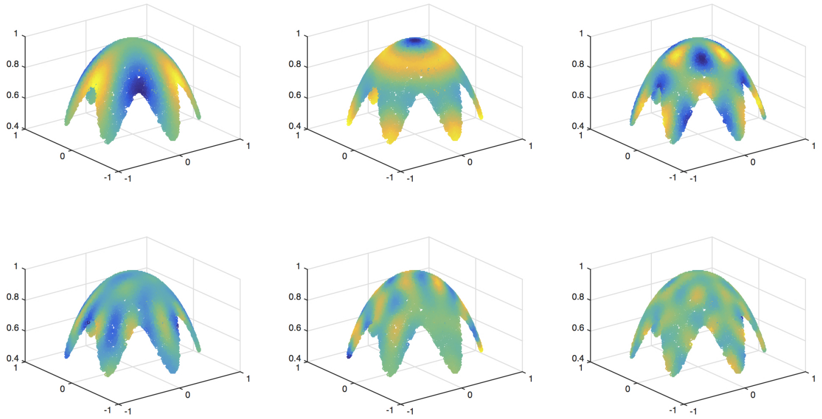
## Harmonic Analysis on Manifolds/Data Sets

- Unit circle: $\Delta = \frac{d^2}{d\theta^2}$ eigenfunctions are Fourier basis
- General manifold or data set $\Rightarrow$ Custom Fourier basis



## Harmonic Analysis on Manifolds/Data Sets

- Unit circle: $\Delta = \frac{d^2}{d\theta^2}$ eigenfunctions are Fourier basis
- General manifold or data set $\Rightarrow$ Custom Fourier basis



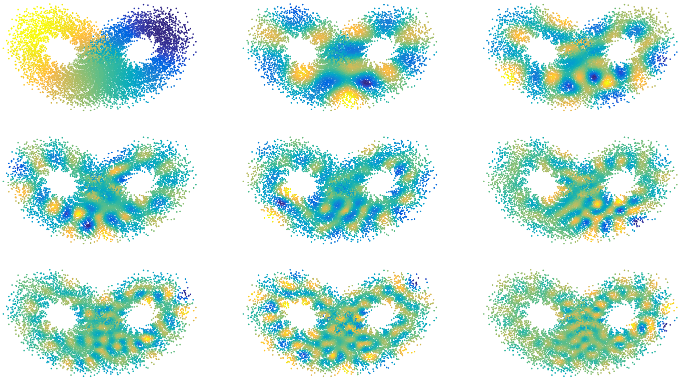
## DIFFUSION FORECAST

- Autonomous SDE: $dx = a(x)\,dt + b(x)\,dW_t$
- Density solves Fokker-Planck PDE: $\frac{\partial}{\partial t}p = \mathcal{L}^* p$
- Shift map: $S(p)(x_i) = p(x_{i+1})$
- Estimates: $\mathbb{E}[S(p)] = e^{\tau\mathcal{L}}p$
- Project onto custom Fourier basis (spectral method)

$$\begin{tikzcd}
p(x,t) \arrow[r, dashed, "\text{Diffusion Forecast}"] \arrow[d, "\langle p,\varphi_j\rangle"] & p(x,t+\tau) = e^{\tau\mathcal{L}^*}p(x,t) \\
\vec{c}(t) \arrow[r, "A_{lj}\equiv\mathbb{E}[\langle\varphi_j,S\varphi_l\rangle_q]"] & \vec{c}(t+\tau) = A\vec{c}(t). \arrow[u, "\sum_j c_j\varphi_j q"']
\end{tikzcd}$$

## MANIFOLD LEARNING $\Rightarrow$ CUSTOM 'FOURIER' BASIS

- **Optimal basis:** Minimum variance $A_{lj} \equiv \mathbb{E}[\langle \varphi_j, S\varphi_l \rangle_q]$



## Diffusion Forecast Example

(Loading Video...)

## Filtering with the Shift Map

Introduce an observable $y = h(x) + \nu$ with $\nu = y - h(x) \sim q$

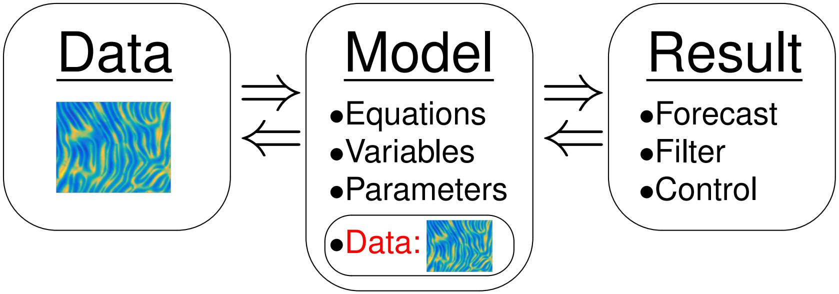
- Likelihood is $p(y \mid x) = q(y - h(x))$
- Bayesian Posterior: $p^a(x_i) \propto p^f(x_i) q(y - h(x_i))$
- Psuedo-spectral method

$$\begin{tikzcd}
p^a(x, t-\tau) \arrow[r, dashed, "\text{Diffusion Forecast}"] \arrow[d, "\langle p^a, \varphi_j \rangle"] & p^f(x, t) \arrow[r, "p^f(x)p(y \mid x)"] & p^a(x, t) \arrow[d, "\langle p^a, \varphi_j \rangle"'] \\
\vec{c}^{\,a}(t-\tau) \arrow[r, "A_{lj} c^a(t-\tau)"] & \vec{c}^{\,f}(t) \arrow[u, "\sum_j c_j^f \varphi_j p_{\mathrm{eq}}"'] \arrow[r, dashed, "\text{Bayesian Update}"] & \vec{c}^{\,a}(t)
\end{tikzcd}$$

## PROBLEM: CURSE OF DIMENSIONALITY

- Nonparametric methods $\rightarrow$ Data required grows like $a^{\text{dim}}$
- Maybe we shouldn't throw out the model...
- Use diffusion forecast to fix model error!

## SEMIPARAMETRIC MODELING



- **Data becomes part of the model:**
  - Start with imperfect parametric model
  - Fit training data with time-varying parameters
  - Query data as part of running model
- **Compensate for model error:**
  - Truncated resolution and complexity
  - Non-analytic expressions
  - Non-stationarity/Inhomogeneity

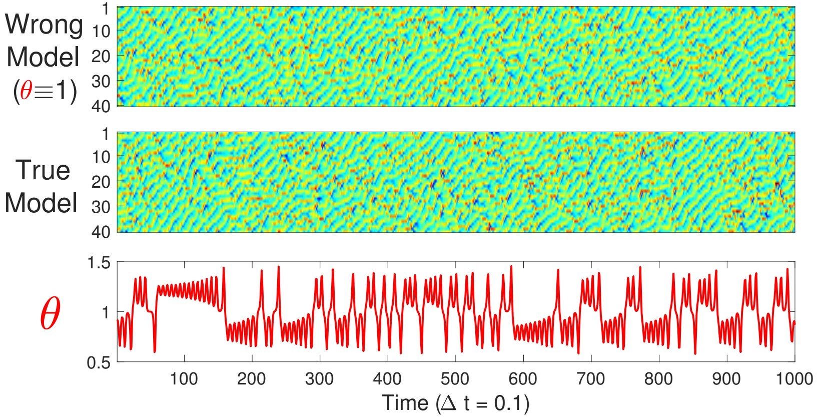
## Semiparametric Forecast Model

- Partially known model $\dot{x} = f(x,\theta)$
- Unknown: $d\theta = a(\theta)\,dt + b(\theta)\,dW_t$
- Apply the Diffusion Forecast to $p(\theta, t)$
- Sample $\theta^k(t) \sim p(\theta,t)$ and pair with ensemble $x^k(t)$

$$\begin{tikzcd}
(x^k(t),\theta^k(t)) \arrow[r, "\dot{x}=f(x,\theta)"] & (x^k(t+\tau),\theta^k(t+\tau)) \\
p(\theta,t) \arrow[u, "\theta^k(t)"'] \arrow[r, dashed, "\text{Diffusion Forecast}"] & p(\theta,t+\tau) \arrow[u, "\theta^k(t+\tau)"']
\end{tikzcd}$$

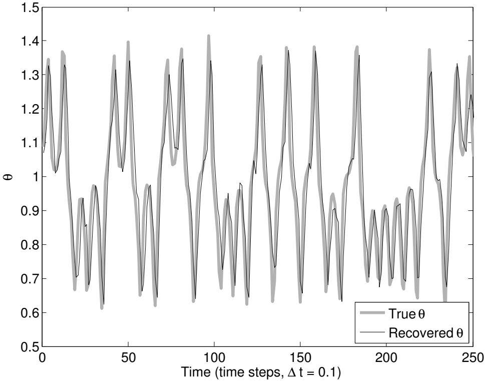
## Example: 40-dimensional Lorenz-96 system

$$\dot{x}_i = \theta x_{i-1}x_{i+1} - x_{i-1}x_{i-2} - x_i + 8$$



## EXAMPLE: 40-DIMENSIONAL LORENZ-96 SYSTEM

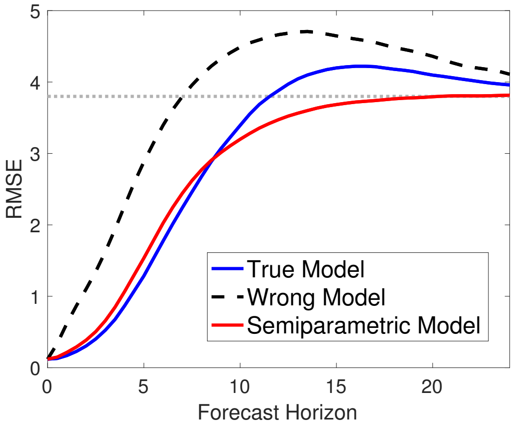
Kalman filter $\Rightarrow$ Estimate time series of $\theta$ (training period)



Using this data, build a diffusion forecast model for $\theta$

## Example: 40-dimensional Lorenz-96 system

$$\dot{x}_i = \theta x_{i-1}x_{i+1} - x_{i-1}x_{i-2} - x_i + 8$$



## Semiparametric Filter: Put it all together...

$$\begin{tikzcd}
\begin{pmatrix} x^{k,a}(t-\tau) \\ \theta^{k,a}(t-\tau) \end{pmatrix} \arrow[r, "\dot{x}=f(x,\theta)"] \arrow[d, "\theta^a"] & \begin{pmatrix} x^{k,f}(t) \\ \theta^{k,f}(t) \end{pmatrix} \arrow[r, "\text{EnKF } y^o(t)"] & \begin{pmatrix} x^{k,a}(t) \\ \theta^{k,a}(t) \end{pmatrix} \arrow[d, "p(\theta^a(t)\,|\,\theta(t))"'] \\
p^a(\theta, t-\tau) \arrow[r, dashed, "\text{Diffusion Forecast}"] \arrow[d, "\langle p^a{,}\varphi_j \rangle"] & p^f(\theta, t) \arrow[u, "\theta^{k,f}(t)"'] \arrow[r, "p^f(\theta)p(y\,|\,\theta)"] & p^a(\theta, t) \arrow[d, "\langle p^a{,}\varphi_j \rangle"'] \\
\vec{c}^{\,a}(t-\tau) \arrow[r, "A_{lj}c^{\,a}(t-\tau)"] & \vec{c}^{\,f}(t) \arrow[u, "\sum_j c_j^f \varphi_j p_{\mathrm{eq}}"'] \arrow[r, dashed, "\text{Bayesian Update}"] & \vec{c}^{\,a}(t)
\end{tikzcd}$$

## For more information: http://math.gmu.edu/˜berry/

### Building the basis

- Coifman and Lafon, *Diffusion maps.*
- B. and Harlim, *Variable Bandwidth Diffusion Kernels.*
- B. and Sauer, *Local Kernels and Geometric Structure of Data.*

### Nonparametric forecast

- B., Giannakis, and Harlim, *Nonparametric forecasting of low-dimensional dynamical systems.*
- B. and Harlim, *Forecasting Turbulent Modes with Nonparametric Diffusion Models.*

### Semiparametric forecast

- B. and Harlim, *Semiparametric forecasting and filtering: correcting low-dimensional model error in parametric models.*