Learning Dynamical Systems with Gaussian Processes

Roger Frigola

University of Cambridge
Machine Learning Group

24th February 2014

Outline

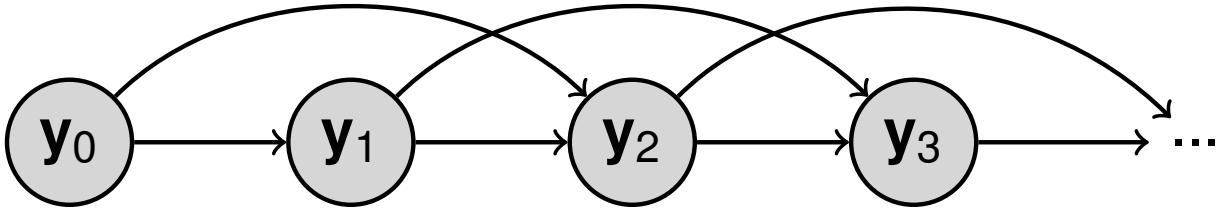
- Time series and dynamical systems.
- GPs to learn from long time series.
- Inference and learning in GP state-space models.
  - Fully Bayesian learning.
  - Stochastic approximation EM.


Cur P/E Level
Est P/E Level
Price USD
High
Low
Average
4JAN02

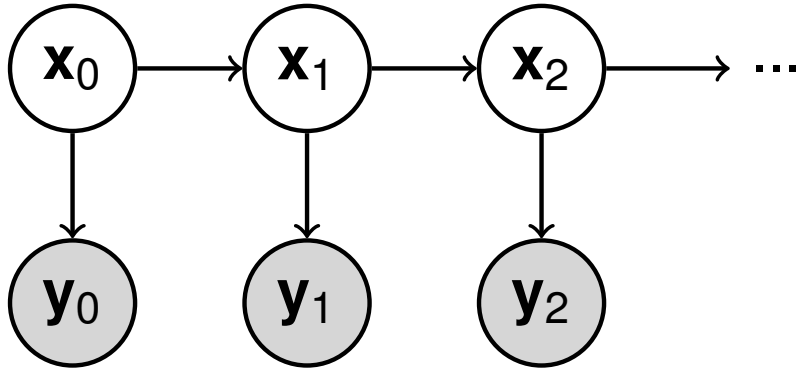

NESPRESSO
NEW ZEALAND
CAMPER
OMEGA
NESPRESSO
NEW ZEALAND
CAMPER
OMEGA
NESPRESSO
OMEGA

Probabilistic Models of Time Series

Auto-regressive model (AR, ARX, NARX...)



State-space models (SSM)



Gaussian Processes for Time Series

Linear auto-regressive and state-space models with Gaussian noise define a Gaussian process

$$y(t) \sim \mathcal{GP}(m(t), k(t, t')) \tag{1}$$

What about nonlinear systems?

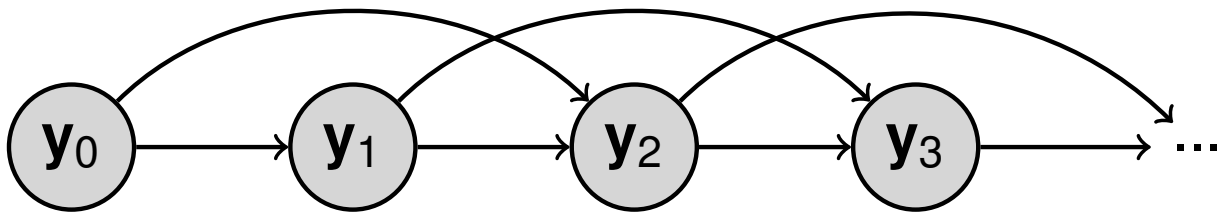
Integrated Pre-Processing for GP-NARX

(Frigola and Rasmussen, CDC 2013)

1. Can we learn nonlinear Bayesian nonparametric models from large datasets?
2. Can we do so in the presence of observation noise?

Nonlinear Auto-regressive Model

$$\mathbf{y}_t = f(\mathbf{y}_{t-1}, ..., \mathbf{y}_{t-n_y}) + \delta_t.$$

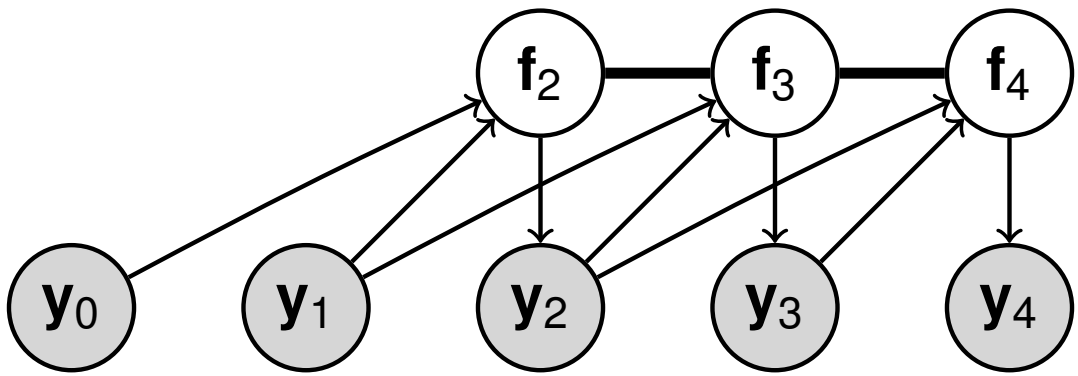


Nonlinear Auto-regressive Model

$$f(\mathbf{Y}_{t-1}) = \mathbf{f}_t \mid \mathbf{Y}_{t-1} \sim \mathcal{GP}\big(m_f(\mathbf{Y}), k_f(\mathbf{Y}, \mathbf{Y}')\big),$$
$$\mathbf{y}_t \mid \mathbf{f}_t \sim p(\mathbf{y}_t \mid \mathbf{f}_t, \theta),$$

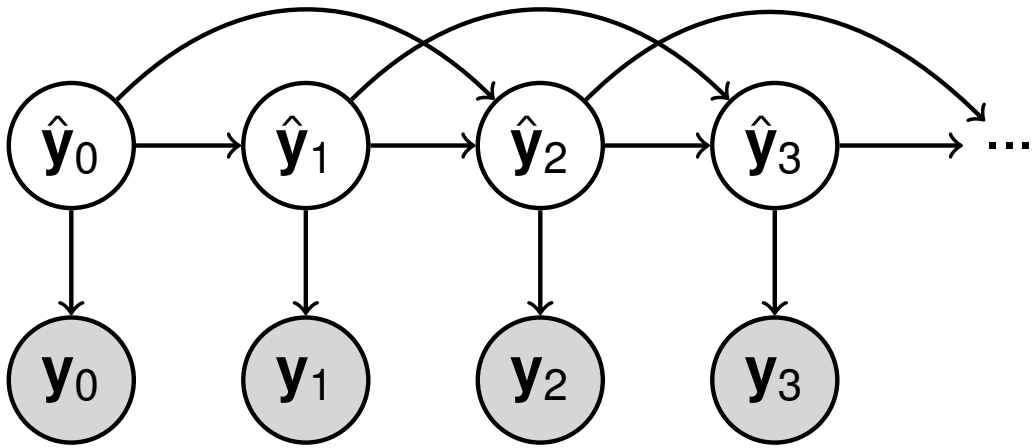
where

$$\mathbf{Y}_{t-1} = \{\mathbf{y}_{t-1}, \ldots, \mathbf{y}_{t-n_y}\}.$$



Integrated Pre-Processing for GP-NARX

Non-Markovian model with hidden states with approximate smoothing.



Use your favourite pre-processing step (e.g. low-pass filter) to obtain a "clean" version of $y_{0:T}$

$$\hat{\mathbf{y}}_{0:T} \approx h(\mathbf{y}_{0:T}, \omega)$$

Joint Pre-Processing and Learning

We have a regression problem where the regressors are the pre-processed signals

$$\mathbf{y}_t = f(\hat{\mathbf{y}}_{t-1}, \hat{\mathbf{y}}_{t-2}, \ldots) + \delta_t.$$

Find a posterior distribution over $f(\cdot)$ using Gaussian process regression.

Joint Pre-Processing and Learning

$$(\omega_{\mathrm{Opt}}, \theta_{\mathrm{Opt}}) = \arg\max_{\omega,\theta} \log p(\mathbf{y}_{0:T}|\hat{X}(\omega), \theta)$$

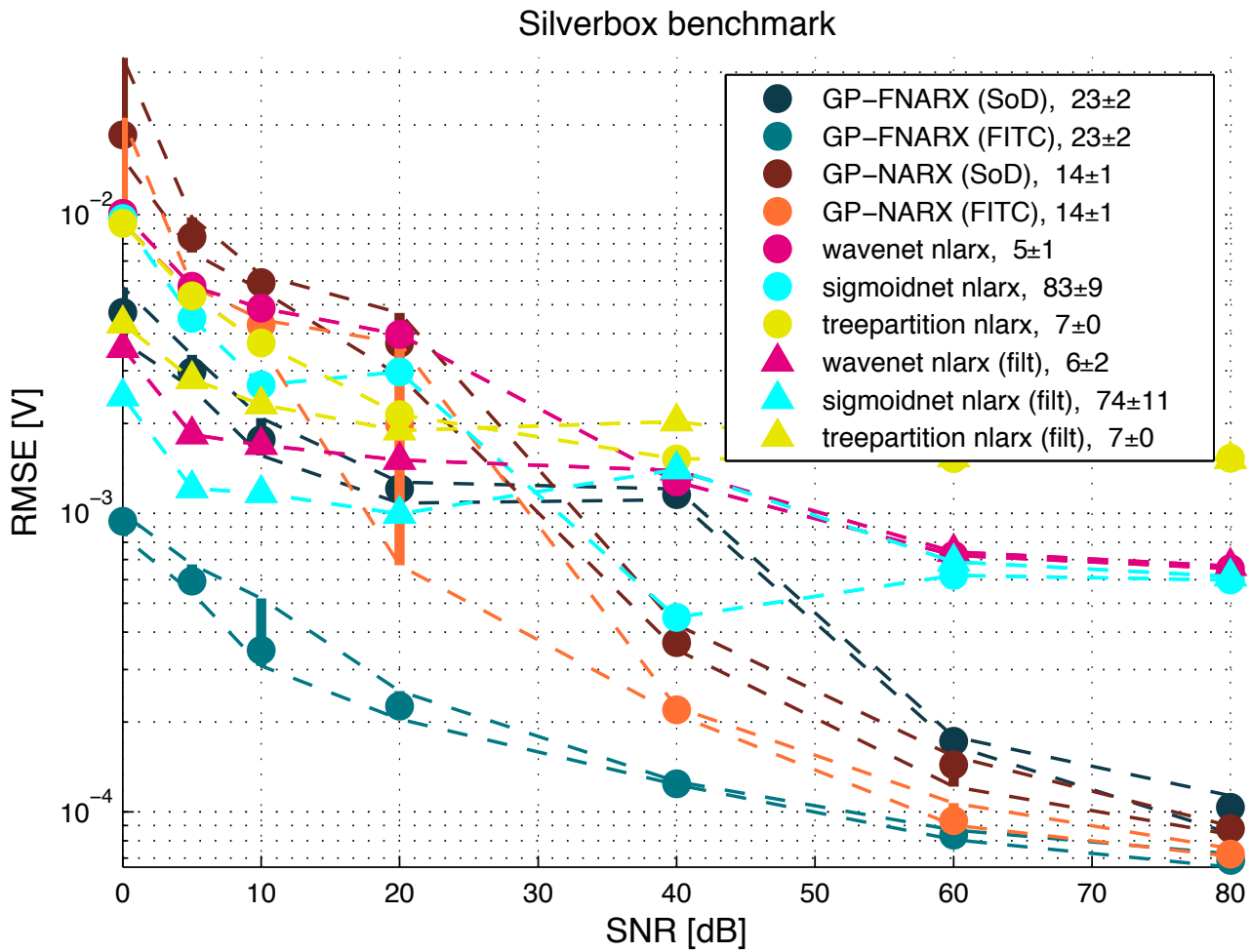
where $\hat{X}(\omega)$ denotes a matrix of filtered regressors.

The marginal likelihood results from integrating analytically the latent variables $\mathbf{f}_{0:T}$

$$p(\mathbf{y}_{0:T}|\hat{X}(\omega), \theta) = \int \underbrace{p(\mathbf{y}_{0:T}|\mathbf{f}_{0:T}, \theta)}_{\text{Likelihood}} \underbrace{p(\mathbf{f}_{0:T}|\hat{X}(\omega), \theta)}_{\text{GP prior}} \, d\mathbf{f}_{0:T}$$

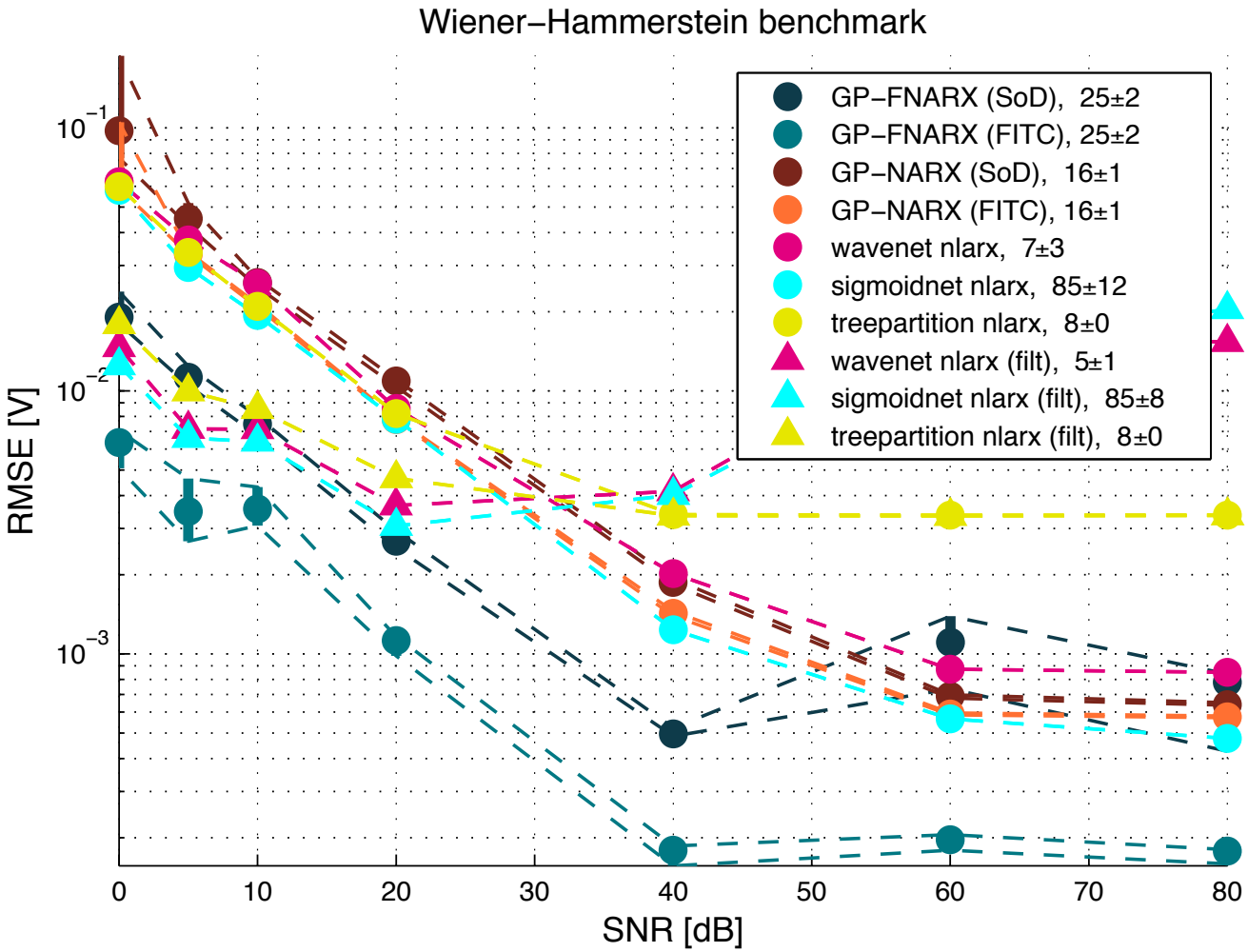
Silverbox benchmark ($T = 1.3 \cdot 10^5$)

Signals contaminated with different levels of Gaussian iid noise.



Wiener-Hammerstein benchmark ($T = 1.9 \cdot 10^5$)

Signals contaminated with different levels of Gaussian iid noise.

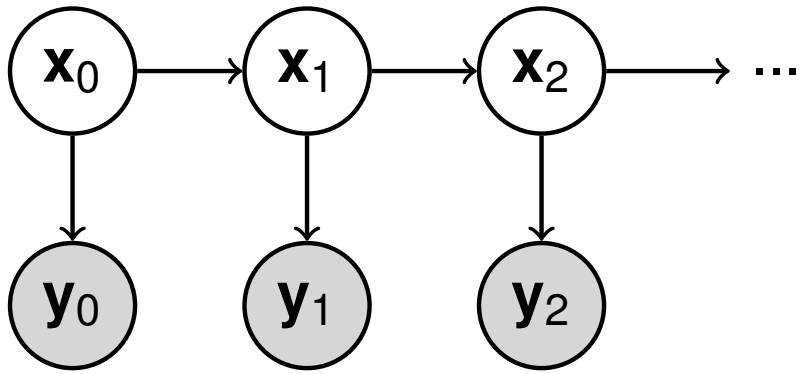


Recap

- Practical Bayesian nonparametric nonlinear system identification for $> 10^5$ data points in a few seconds.
- From raw data to model without human intervention.
- Deals with measurement noise.
- The user can select its own preferred data pre-processing method.

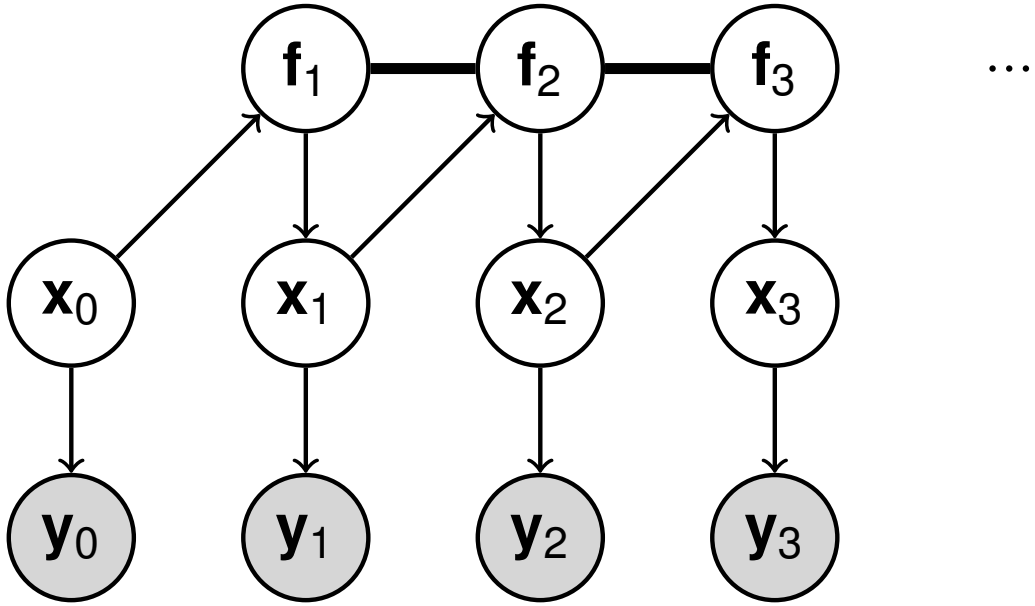
State-Space Models

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t) + \mathbf{v}_t,$$
$$\mathbf{y}_t = g(\mathbf{x}_t, \mathbf{u}_t) + \mathbf{e}_t.$$



Gaussian Process State-Space Models

$$
\begin{aligned}
f(\mathbf{x}_t) = \mathbf{f}_{t+1} \mid \mathbf{x}_t &\sim \mathcal{GP}\big(m_f(\mathbf{x}), k_f(\mathbf{x}, \mathbf{x}')\big), \\
\mathbf{x}_{t+1} \mid \mathbf{f}_{t+1} &\sim \mathcal{N}(\mathbf{x}_{t+1} \mid \mathbf{f}_{t+1}, \mathbf{Q}), \\
\mathbf{y}_t \mid \mathbf{x}_t &\sim p(\mathbf{y}_t \mid \mathbf{x}_t, \theta_y).
\end{aligned}
$$



Fully Bayesian Inference and Learning in GP-SSMs

(Frigola, Lindsten, Schön and Rasmussen, NIPS 2013)

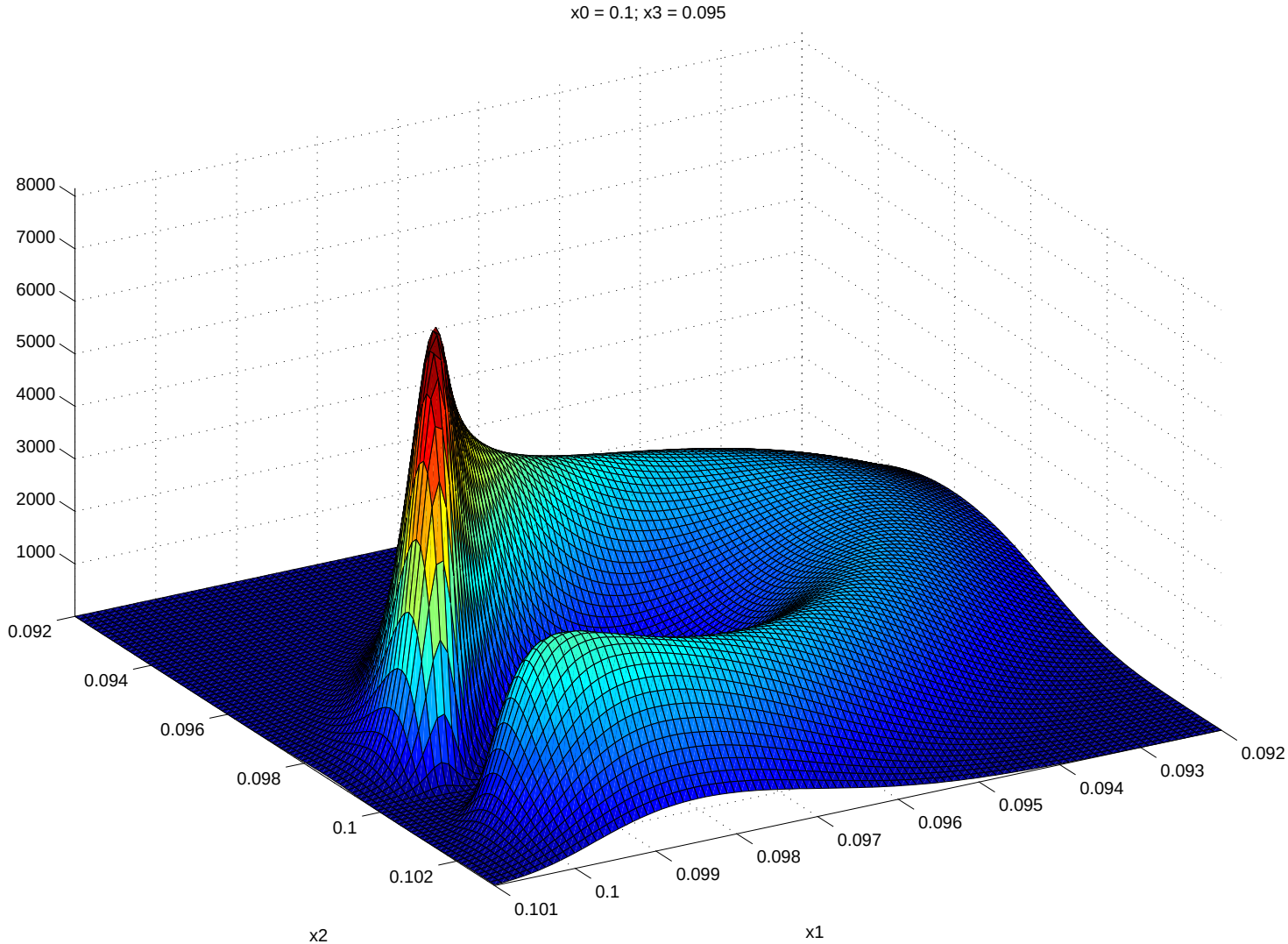
- Prior work had found MAP estimates of $\mathbf{x}_{0:T}$ and θ.
- What if $\dim(\mathbf{x}_t) << \dim(\mathbf{y}_t)$ does NOT hold?
- Can we have a fully Bayesian treatment of this model?

Marginalizing out the State Transition Function

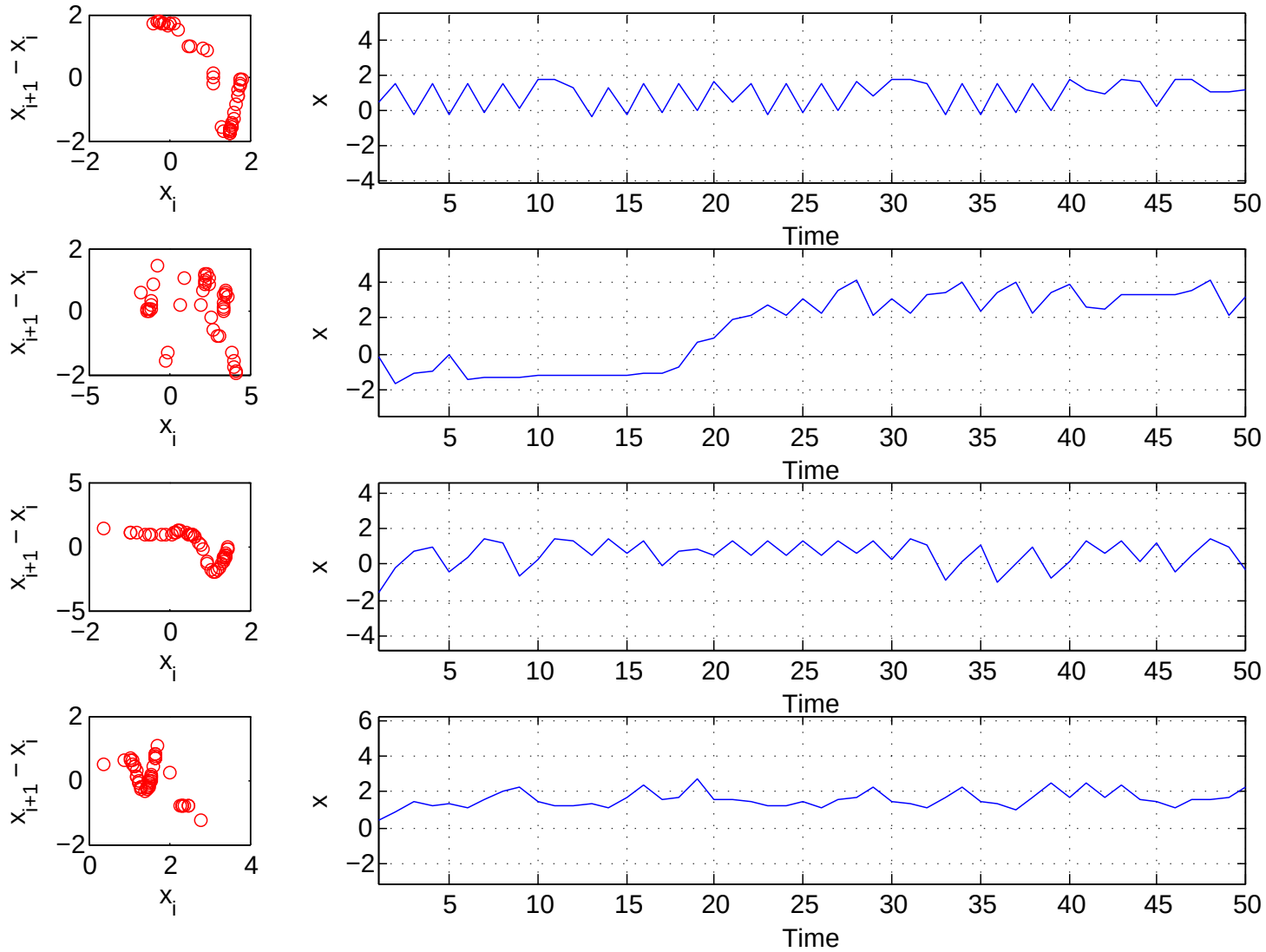
Marginal prior over state trajectories $p(\mathbf{x}_{0:T} \mid \theta)$:

$$
\begin{aligned}
p(\mathbf{x}_{1:T} \mid \theta, \mathbf{x}_0) &= \prod_{t=1}^{T} p(\mathbf{x}_t \mid \theta, \mathbf{x}_{0:t-1}) \\
&= \prod_{t=1}^{T} \mathcal{N}\big(\mathbf{x}_t \mid \boldsymbol{\mu}_t(\mathbf{x}_{0:t-1}), \boldsymbol{\Sigma}_t(\mathbf{x}_{0:t-1})\big),
\end{aligned}
$$

Prior over State Trajectories



Sampling from Prior over State Trajectories



Posterior Sampling with Particle MCMC

Target $p(\mathbf{x}_{0:T}, \theta \mid \mathbf{y}_{0:T})$.

Particle Gibbs with Ancestor Sampling (Lindsten, Jordan and Schön, NIPS 2012) is an efficient PMCMC sampler for non-Markovian problems.

1. Set $\theta[0]$ and $\mathbf{x}_{0:T}[0]$ arbitrarily.
2. **For** $\ell \geq 1$ **do**
   - **2.1** Draw $\theta[\ell] \sim p(\theta \mid \mathbf{x}_{0:T}[\ell - 1], \mathbf{y}_{0:T})$ with slice sampling.
   - **2.2** Run CPF-AS targeting $p(\mathbf{x}_{0:T} \mid \theta[\ell], \mathbf{y}_{0:T})$, conditionally on $\mathbf{x}_{0:T}[\ell - 1]$.
   - **2.3** Sample k with $P(k = i) = w_T^i$ and set $\mathbf{x}_{1:T}[\ell] = \mathbf{x}_{1:T}^k$.
3. **end**

Samples from the Smoothing Distribution Solve the Learning Problem

Making predictions

$$p(\mathbf{f}^* \mid \mathbf{x}^*, \mathbf{y}_{0:T}) = \int p(\mathbf{f}^* \mid \mathbf{x}^*, \mathbf{x}_{0:T}, \theta)\, p(\mathbf{x}_{0:T}, \theta \mid \mathbf{y}_{0:T})\, d\mathbf{x}_{0:T}\, d\theta.$$

Using samples from $p(\mathbf{x}_{0:T}, \theta \mid \mathbf{y}_{0:T})$

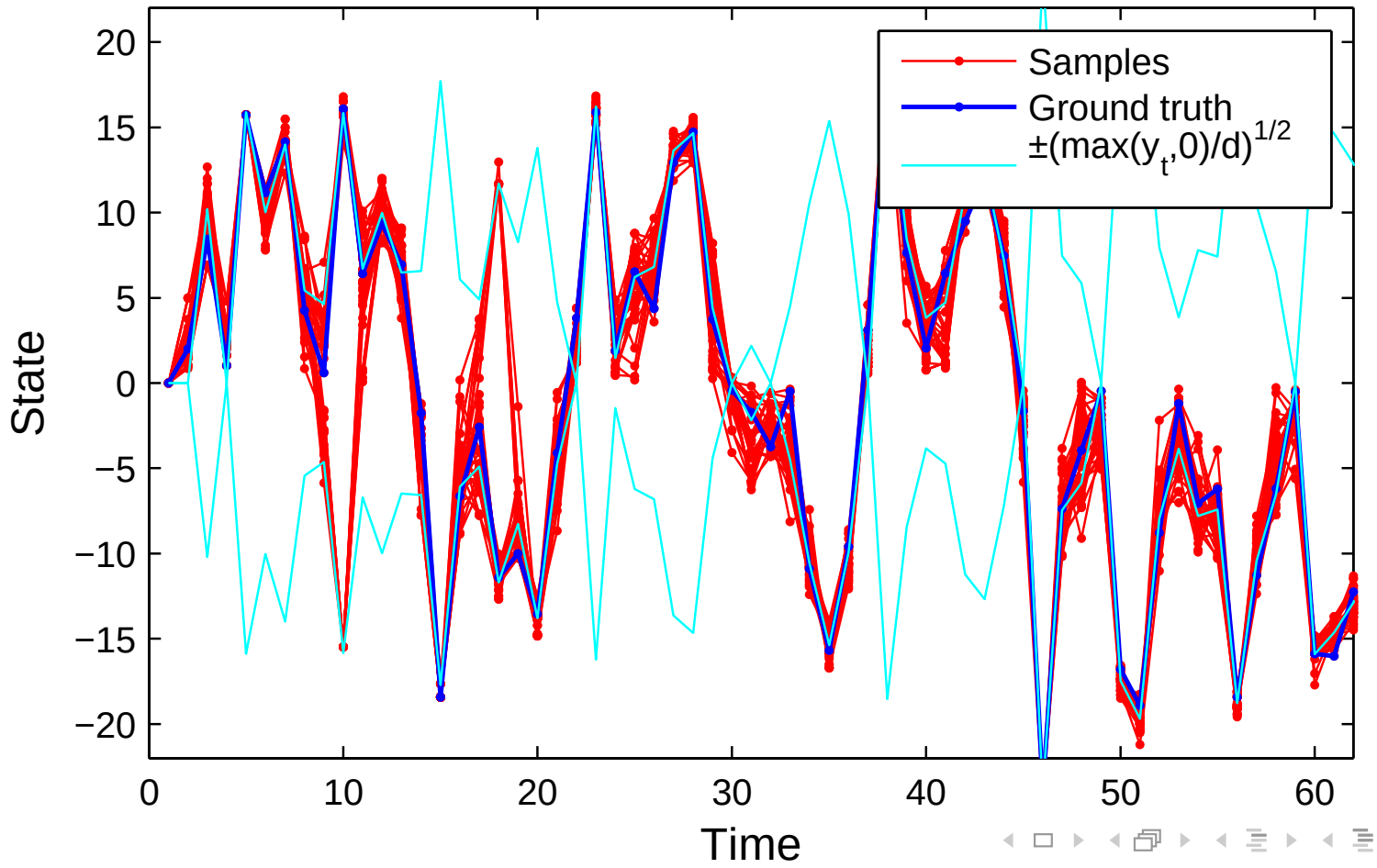
$$\begin{aligned} p(\mathbf{f}^* \mid \mathbf{x}^*, \mathbf{y}_{0:T}) &\approx \frac{1}{L} \sum_{l=1}^{L} p(\mathbf{f}^* \mid \mathbf{x}^*, \mathbf{x}_{0:T}[l], \theta[l]) \\ &= \frac{1}{L} \sum_{l=1}^{L} \mathcal{N}(\mathbf{f}^* \mid \boldsymbol{\mu}^l(\mathbf{x}^*), \boldsymbol{\Sigma}^l(\mathbf{x}^*)), \end{aligned}$$

where $\boldsymbol{\mu}^l(\mathbf{x}^*)$ and $\boldsymbol{\Sigma}^l(\mathbf{x}^*)$ follow the expressions for the predictive distribution in standard GP regression.

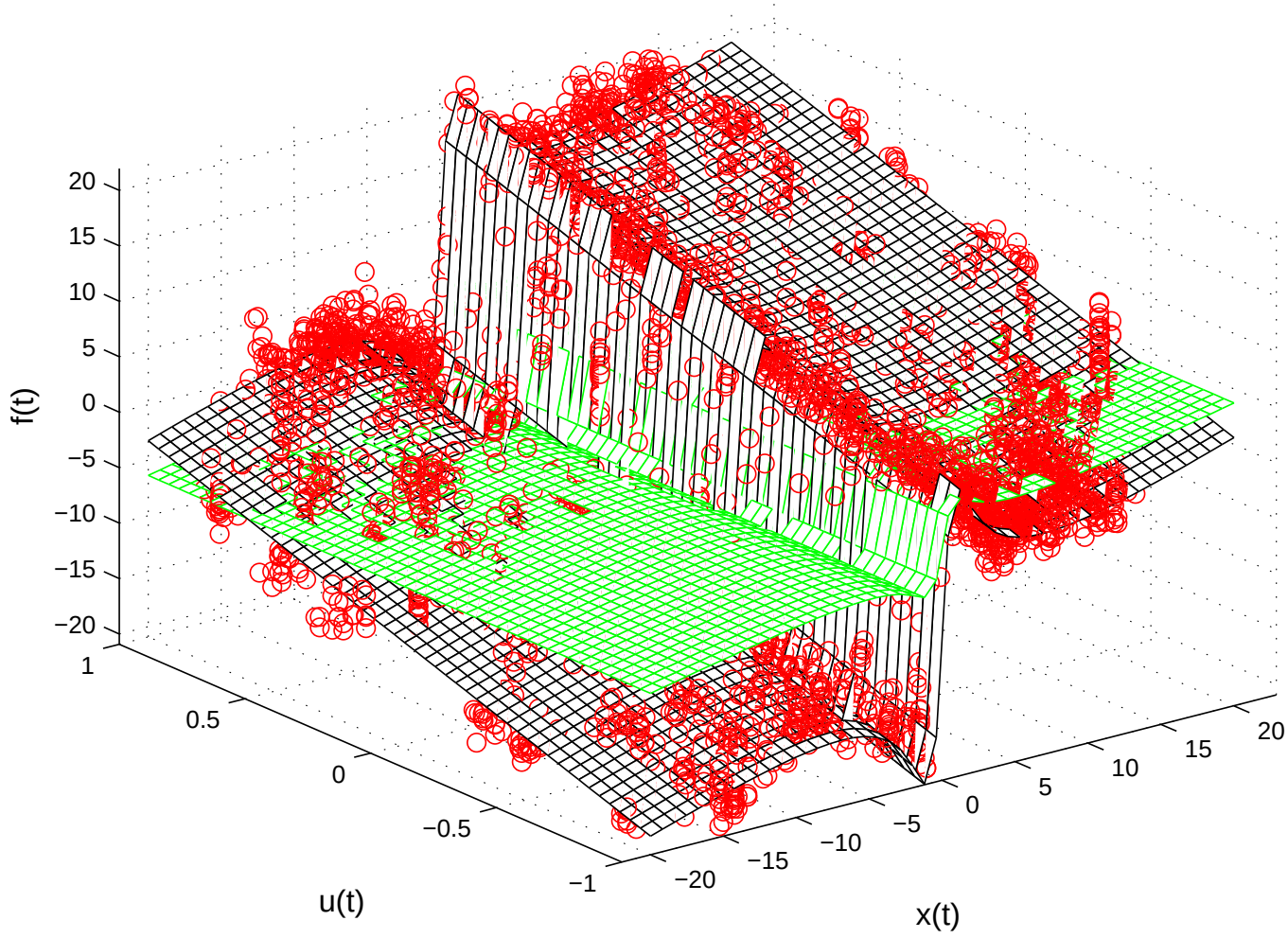
1-dimensional Benchmark System

$$x_{t+1} = ax_t + b\frac{x_t}{1+x_t^2} + cu_t + v_t, \quad v_t \sim \mathcal{N}(0, q),$$

$$y_t = dx_t^2 + e_t, \quad e_t \sim \mathcal{N}(0, r).$$



State transition function

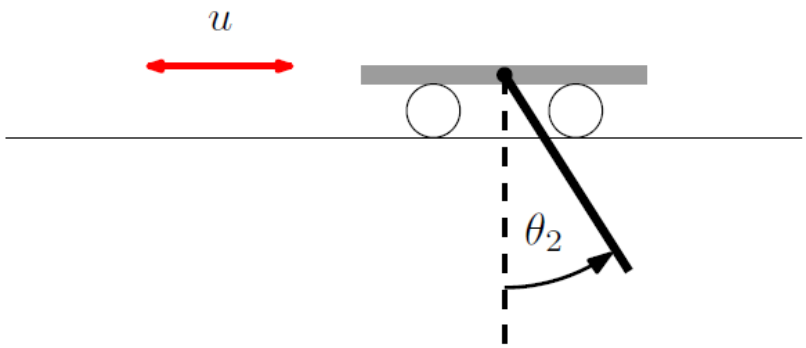


Black: ground truth

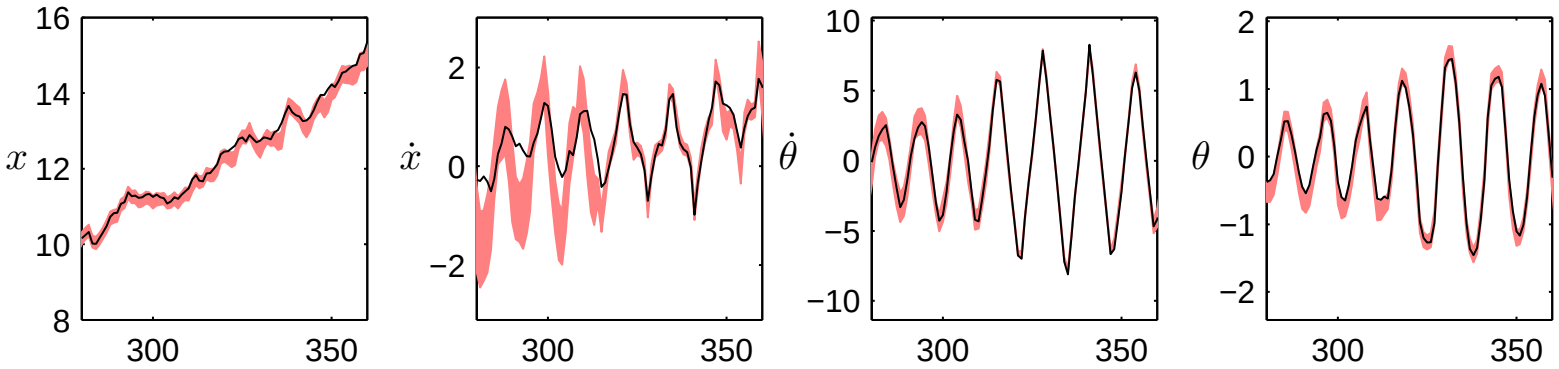
Red: samples from smoothing distribution

Green: GP mean function

4-dimensional Cart and Pole System



One step ahead predictive distribution for each of the states of the cart and pole:



Black: ground truth.

Coloured band: one standard deviation predictive.

Maximum Likelihood for GP-SSMs

(Frigola, Lindsten, Schön and Rasmussen, IFAC World Congress 2014)

$$\widehat{\theta}^{\mathrm{ML}} = \arg\max_{\theta} \; p(\mathbf{y}_{0:T} \mid \theta).$$

Need to integrate out state trajectory ($\mathbf{f}_{0:T}$ already marginalised)

$$p(\mathbf{y}_{0:T} \mid \theta) = \int p(\mathbf{y}_{0:T} \mid \mathbf{x}_{0:T}, \theta)\, p(\mathbf{x}_{0:T} \mid \theta)\, \mathrm{d}\mathbf{x}_{0:T}.$$

EM for GP-SSMs

Surrogate cost function for the ML problem

$$
\begin{aligned}
Q(\boldsymbol{\theta}, \boldsymbol{\theta}') &= \mathbb{E}_{\mathbf{x}_{0:T}|\mathbf{y}_{0:T},\boldsymbol{\theta}'}[\log p(\mathbf{y}_{0:T}, \mathbf{x}_{0:T} \mid \boldsymbol{\theta})] \\
&= \int \log p(\mathbf{y}_{0:T}, \mathbf{x}_{0:T} \mid \boldsymbol{\theta}) p(\mathbf{x}_{0:T} \mid \mathbf{y}_{0:T}, \boldsymbol{\theta}') d\mathbf{x}_{0:T}.
\end{aligned}
$$

EM algorithm, initialise $\boldsymbol{\theta}_0$ and iterate:

(E) Compute $Q(\boldsymbol{\theta}, \boldsymbol{\theta}_{k-1})$.

(M) Compute $\boldsymbol{\theta}_k = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}_{k-1})$.

Particle Stochastic Approximation EM

How to run EM when integrals are intractable?

- Monte Carlo EM
- Stochastic Approximation EM
- Particle Stochastic Approximation EM (Lindsten, 2013)

Particle Stochastic Approximation EM

Stochastic approximation of the auxiliary quantity

$$\hat{Q}_k(\boldsymbol{\theta}) \approx Q(\boldsymbol{\theta}, \boldsymbol{\theta}_{k-1}).$$

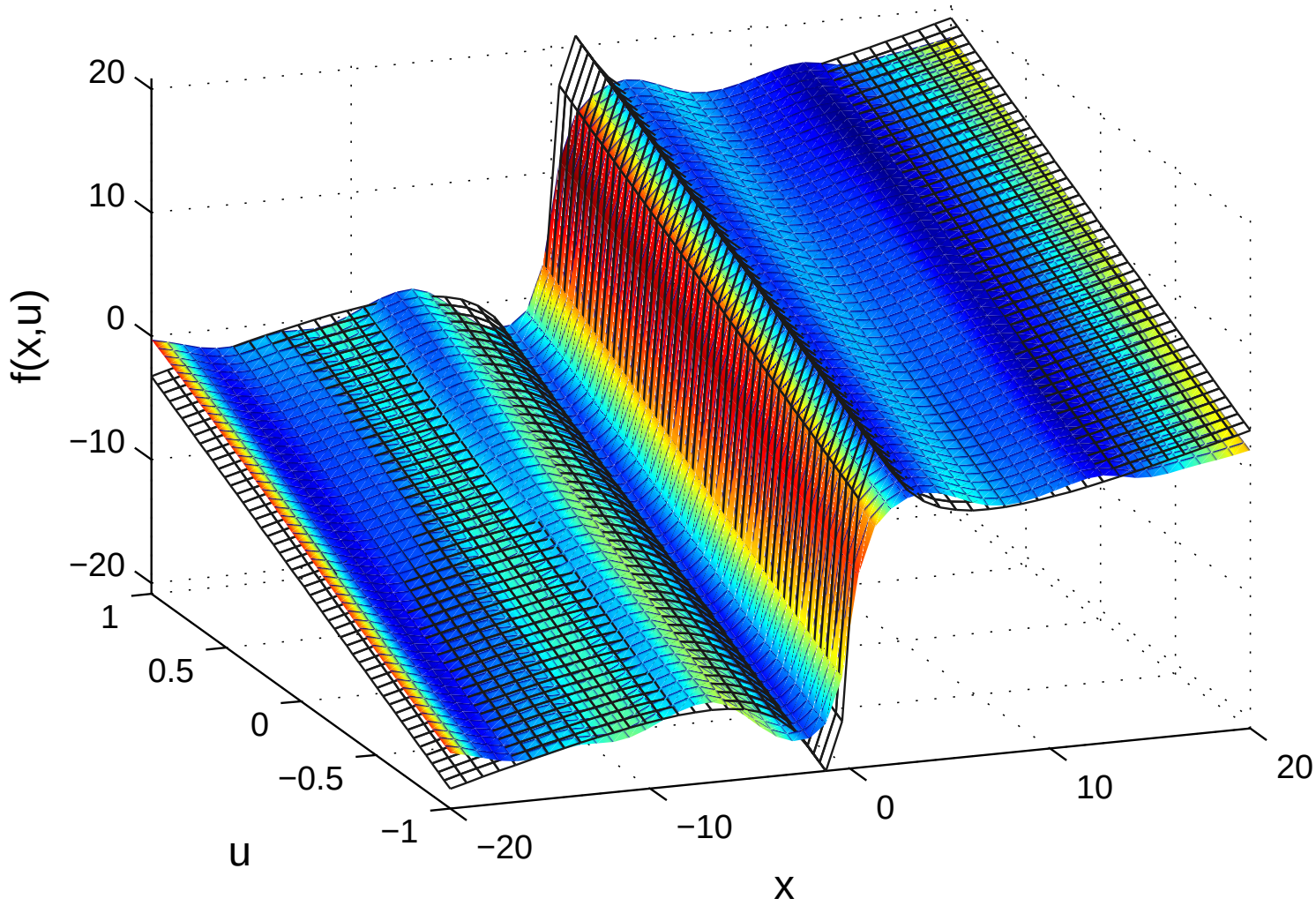
Updates

$$\hat{Q}_k(\boldsymbol{\theta}) = (1 - \gamma_k)\hat{Q}_{k-1}(\boldsymbol{\theta}) + \gamma_k \log p(\mathbf{y}_{0:T}, \mathbf{x}_{0:T}[k] \mid \boldsymbol{\theta})$$

where $\sum_k \gamma_k = \infty$ and $\sum_k \gamma_k^2 < \infty$.

We can reuse old samples!

Maximum Likelihood for GP-SSMs



Black mesh: ground truth.

Coloured surface: mean predictive, coloured according to variance.

Current/Future Work

- Is it possible to learn GP-SSMs using deterministic approximate inference? Variational GPs à la Titsias and Lawrence.
- Inducing inputs for sparse GPs in sampling approaches.
- Training with mini-batches of data.