# State Space Representation of Gaussian Processes

Simo Särkkä

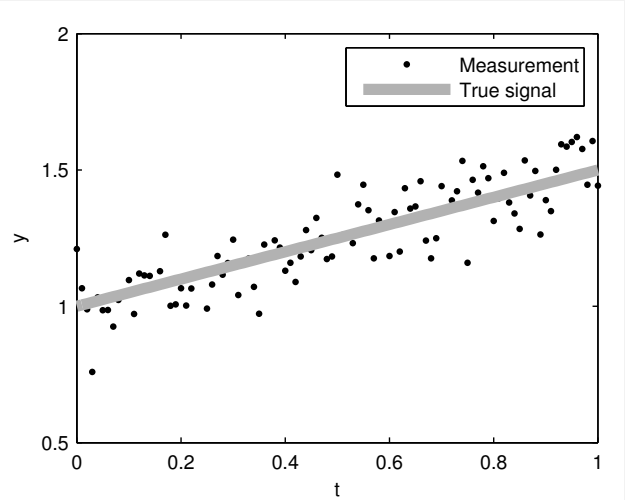
Aalto University, Finland

January 16, 2014

Aalto University

# Contents

# Batch Linear Regression [1/2]




- Consider the linear regression model

$$y_k = \theta_1 + \theta_2\, t_k + \varepsilon_k, \qquad k = 1, \ldots, T,$$

with $\varepsilon_k \sim \mathrm{N}(0, \sigma^2)$ and $\boldsymbol{\theta} = (\theta_1, \theta_2) \sim \mathrm{N}(\mathbf{m}_0, \mathbf{P}_0)$.

- In probabilistic notation this is:

$$p(y_k \,|\, \boldsymbol{\theta}) = \mathrm{N}(y_k \,|\, \mathbf{H}_k\, \boldsymbol{\theta}, \sigma^2)$$
$$p(\boldsymbol{\theta}) = \mathrm{N}(\boldsymbol{\theta} \,|\, \mathbf{m}_0, \mathbf{P}_0),$$

where $\mathbf{H}_k = (1 \;\; t_k)$.

# Batch Linear Regression [2/2]

- The Bayesian batch solution by the Bayes' rule:

$$p(\boldsymbol{\theta} \mid y_{1:T}) \propto p(\boldsymbol{\theta}) \prod_{k=1}^{T} p(y_k \mid \boldsymbol{\theta})$$
$$= \mathrm{N}(\boldsymbol{\theta} \mid \mathbf{m}_0, \mathbf{P}_0) \prod_{k=1}^{T} \mathrm{N}(y_k \mid \mathbf{H}_k \boldsymbol{\theta}, \sigma^2).$$

- The posterior is Gaussian

$$p(\boldsymbol{\theta} \mid y_{1:T}) = \mathrm{N}(\boldsymbol{\theta} \mid \mathbf{m}_T, \mathbf{P}_T).$$

- The mean and covariance are given as

$$\mathbf{m}_T = \left[\mathbf{P}_0^{-1} + \frac{1}{\sigma^2}\mathbf{H}^T\mathbf{H}\right]^{-1} \left[\frac{1}{\sigma^2}\mathbf{H}^T\mathbf{y} + \mathbf{P}_0^{-1}\mathbf{m}_0\right]$$
$$\mathbf{P}_T = \left[\mathbf{P}_0^{-1} + \frac{1}{\sigma^2}\mathbf{H}^T\mathbf{H}\right]^{-1},$$

where $\mathbf{H}_k = (1 \;\; t_k)$, $\mathbf{H} = (\mathbf{H}_1; \mathbf{H}_2; \ldots; \mathbf{H}_T)$, $\mathbf{y} = (y_1; \ldots; y_T)$.

# Recursive Linear Regression [1/3]

- Assume that we have already computed the posterior distribution, which is conditioned on the measurements up to $k-1$:

$$p(\boldsymbol{\theta} \mid y_{1:k-1}) = \mathrm{N}(\boldsymbol{\theta} \mid \mathbf{m}_{k-1}, \mathbf{P}_{k-1}).$$

- Assume that we get the $k$th measurement $y_k$. Using the equations from the previous slide we get

$$\begin{aligned} p(\boldsymbol{\theta} \mid y_{1:k}) &\propto p(y_k \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta} \mid y_{1:k-1}) \\ &\propto \mathrm{N}(\boldsymbol{\theta} \mid \mathbf{m}_k, \mathbf{P}_k). \end{aligned}$$

- The mean and covariance are given as

$$\begin{aligned} \mathbf{m}_k &= \left[\mathbf{P}_{k-1}^{-1} + \frac{1}{\sigma^2}\mathbf{H}_k^T\mathbf{H}_k\right]^{-1}\left[\frac{1}{\sigma^2}\mathbf{H}_k^T y_k + \mathbf{P}_{k-1}^{-1}\mathbf{m}_{k-1}\right] \\ \mathbf{P}_k &= \left[\mathbf{P}_{k-1}^{-1} + \frac{1}{\sigma^2}\mathbf{H}_k^T\mathbf{H}_k\right]^{-1}. \end{aligned}$$

# Recursive Linear Regression [2/3]

- By the matrix inversion lemma (or Woodbury identity):

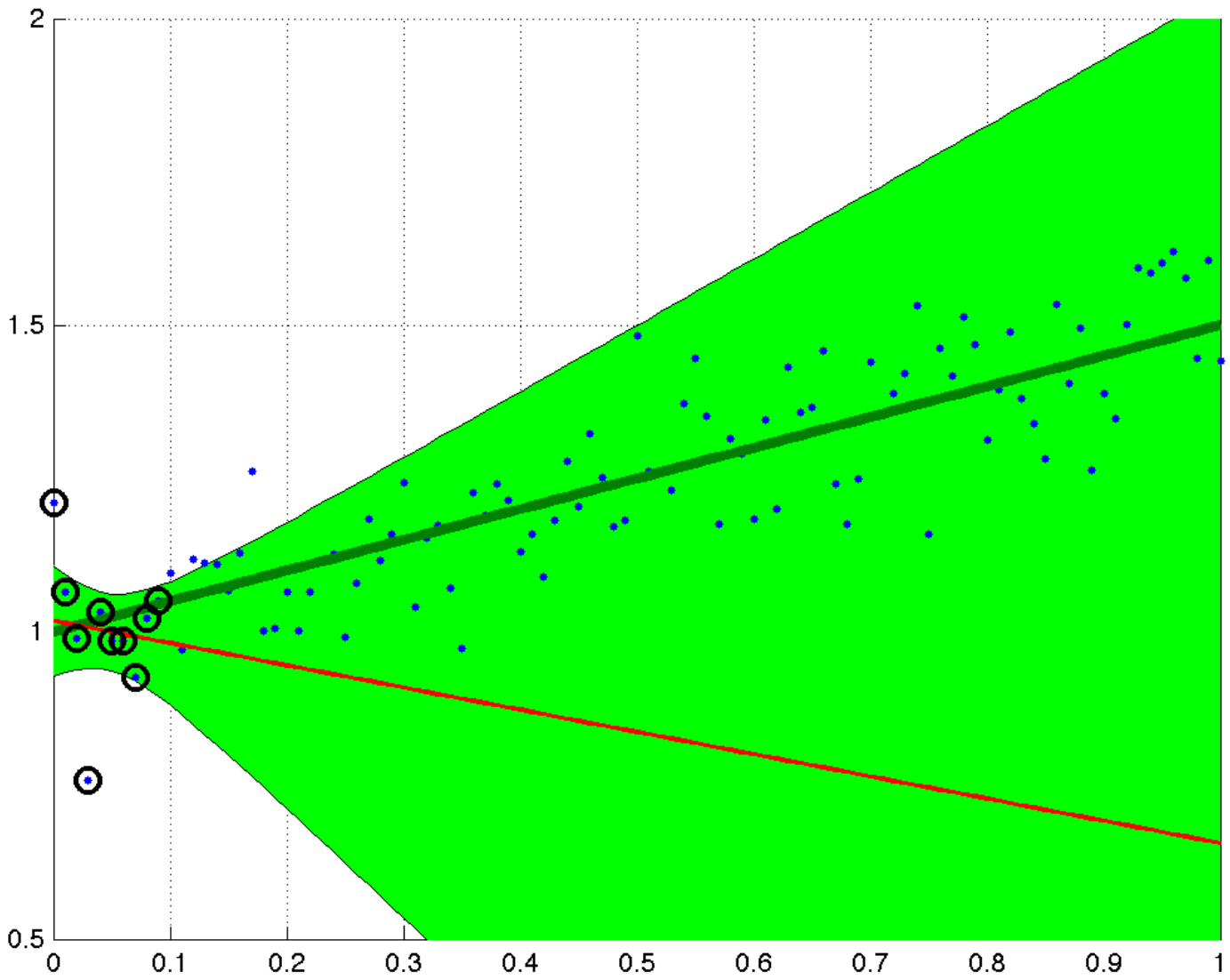
$$\mathbf{P}_k = \mathbf{P}_{k-1} - \mathbf{P}_{k-1}\mathbf{H}_k^T \left[\mathbf{H}_k\mathbf{P}_{k-1}\mathbf{H}_k^T + \sigma^2\right]^{-1} \mathbf{H}_k\mathbf{P}_{k-1}.$$

- Now the equations for the mean and covariance reduce to

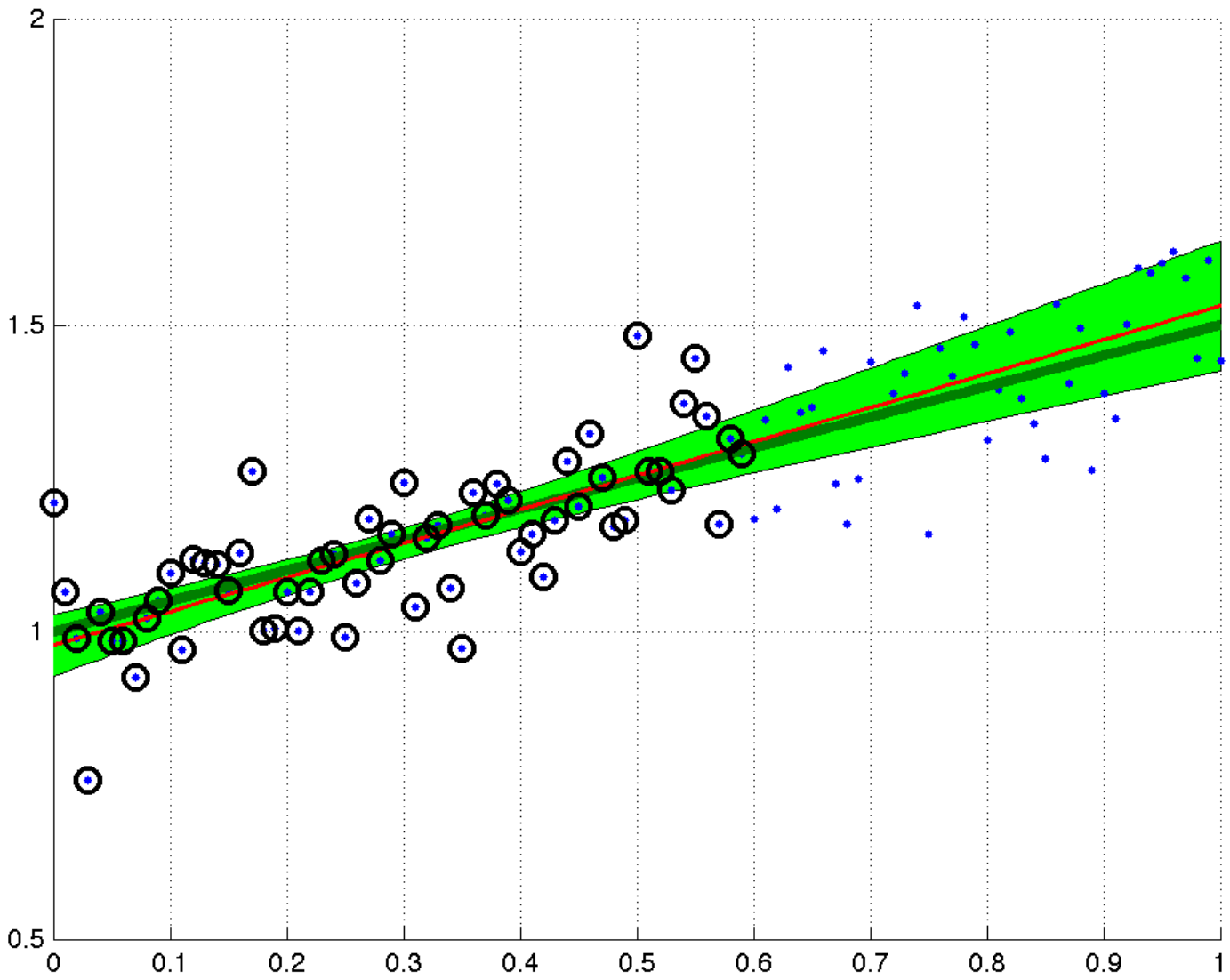
$$\begin{aligned} S_k &= \mathbf{H}_k\mathbf{P}_{k-1}\mathbf{H}_k^T + \sigma^2 \\ \mathbf{K}_k &= \mathbf{P}_{k-1}\mathbf{H}_k^T S_k^{-1} \\ \mathbf{m}_k &= \mathbf{m}_{k-1} + \mathbf{K}_k[y_k - \mathbf{H}_k\mathbf{m}_{k-1}] \\ \mathbf{P}_k &= \mathbf{P}_{k-1} - \mathbf{K}_k S_k \mathbf{K}_k^T. \end{aligned}$$

- Computing these for $k = 0, \ldots, T$ gives exactly the linear regression solution.
- A special case of Kalman filter.

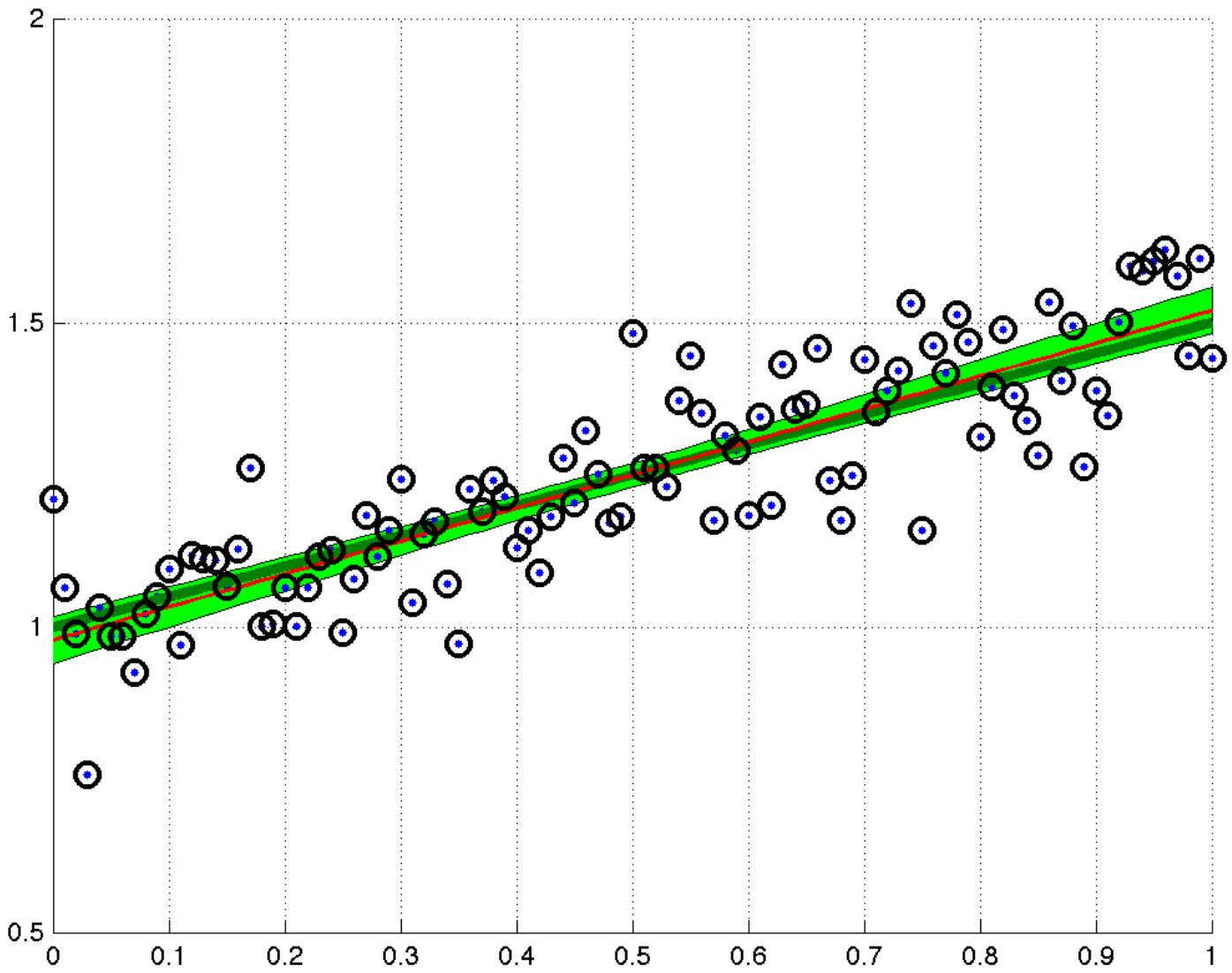
# Recursive Linear Regression [3/3]



# Recursive Linear Regression [3/3]



# Recursive Linear Regression [3/3]



# Batch vs. Recursive Estimation [1/2]

*General batch solution:*

- Specify the measurement model:

$$p(\mathbf{y}_{1:T} \,|\, \boldsymbol{\theta}) = \prod_k p(\mathbf{y}_k \,|\, \boldsymbol{\theta}).$$

- Specify the prior distribution $p(\boldsymbol{\theta})$.
- Compute posterior distribution by the Bayes' rule:

$$p(\boldsymbol{\theta} \,|\, \mathbf{y}_{1:T}) = \frac{1}{Z} p(\boldsymbol{\theta}) \prod_k p(\mathbf{y}_k \,|\, \boldsymbol{\theta}).$$

- Compute point estimates, moments, predictive quantities etc. from the posterior distribution.

# Batch vs. Recursive Estimation [2/2]

*General recursive solution:*

- Specify the measurement likelihood $p(\mathbf{y}_k \,|\, \boldsymbol{\theta})$.
- Specify the prior distribution $p(\boldsymbol{\theta})$.
- Process measurements $\mathbf{y}_1, \ldots, \mathbf{y}_T$ with the Bayes' rule one at a time, starting from the prior:

$$
\begin{aligned}
p(\boldsymbol{\theta} \,|\, \mathbf{y}_1) &= \frac{1}{Z_1} p(\mathbf{y}_1 \,|\, \boldsymbol{\theta}) p(\boldsymbol{\theta}) \\
p(\boldsymbol{\theta} \,|\, \mathbf{y}_{1:2}) &= \frac{1}{Z_2} p(\mathbf{y}_2 \,|\, \boldsymbol{\theta}) p(\boldsymbol{\theta} \,|\, \mathbf{y}_1) \\
&\vdots \\
p(\boldsymbol{\theta} \,|\, \mathbf{y}_{1:T}) &= \frac{1}{Z_T} p(\mathbf{y}_T \,|\, \boldsymbol{\theta}) p(\boldsymbol{\theta} \,|\, \mathbf{y}_{1:T-1}).
\end{aligned}
$$

- The posterior at the last step is the same as the batch solution.

# Drift Model for Linear Regression [1/4]

- Let assume Gaussian random walk between the measurements in the linear regression model:

$$
\begin{aligned}
p(y_k \mid \boldsymbol{\theta}_k) &= \mathrm{N}(y_k \mid \mathbf{H}_k \, \boldsymbol{\theta}_k, \sigma^2) \\
p(\boldsymbol{\theta}_k \mid \boldsymbol{\theta}_{k-1}) &= \mathrm{N}(\boldsymbol{\theta}_k \mid \boldsymbol{\theta}_{k-1}, \mathbf{Q}) \\
p(\boldsymbol{\theta}_0) &= \mathrm{N}(\boldsymbol{\theta}_0 \mid \mathbf{m}_0, \mathbf{P}_0).
\end{aligned}
$$

- Again, assume that we already know

$$
p(\boldsymbol{\theta}_{k-1} \mid y_{1:k-1}) = \mathrm{N}(\boldsymbol{\theta}_{k-1} \mid \mathbf{m}_{k-1}, \mathbf{P}_{k-1}).
$$

- The joint distribution of $\boldsymbol{\theta}_k$ and $\boldsymbol{\theta}_{k-1}$ is (due to Markovianity of dynamics!):

$$
p(\boldsymbol{\theta}_k, \boldsymbol{\theta}_{k-1} \mid y_{1:k-1}) = p(\boldsymbol{\theta}_k \mid \boldsymbol{\theta}_{k-1}) \, p(\boldsymbol{\theta}_{k-1} \mid y_{1:k-1}).
$$

# Drift Model for Linear Regression [2/4]

- Integrating over $\boldsymbol{\theta}_{k-1}$ gives:

$$p(\boldsymbol{\theta}_k \,|\, y_{1:k-1}) = \int p(\boldsymbol{\theta}_k \,|\, \boldsymbol{\theta}_{k-1})\, p(\boldsymbol{\theta}_{k-1} \,|\, y_{1:k-1})\, d\boldsymbol{\theta}_{k-1}.$$

- This equation for Markov processes is called the Chapman-Kolmogorov equation.
- Because the distributions are Gaussian, the result is Gaussian

$$p(\boldsymbol{\theta}_k \,|\, y_{1:k-1}) = \mathrm{N}(\boldsymbol{\theta}_k \,|\, \mathbf{m}_k^-, \mathbf{P}_k^-),$$

where

$$\begin{aligned} \mathbf{m}_k^- &= \mathbf{m}_{k-1} \\ \mathbf{P}_k^- &= \mathbf{P}_{k-1} + \mathbf{Q}. \end{aligned}$$

# Drift Model for Linear Regression [3/4]

- As in the pure recursive estimation, we get

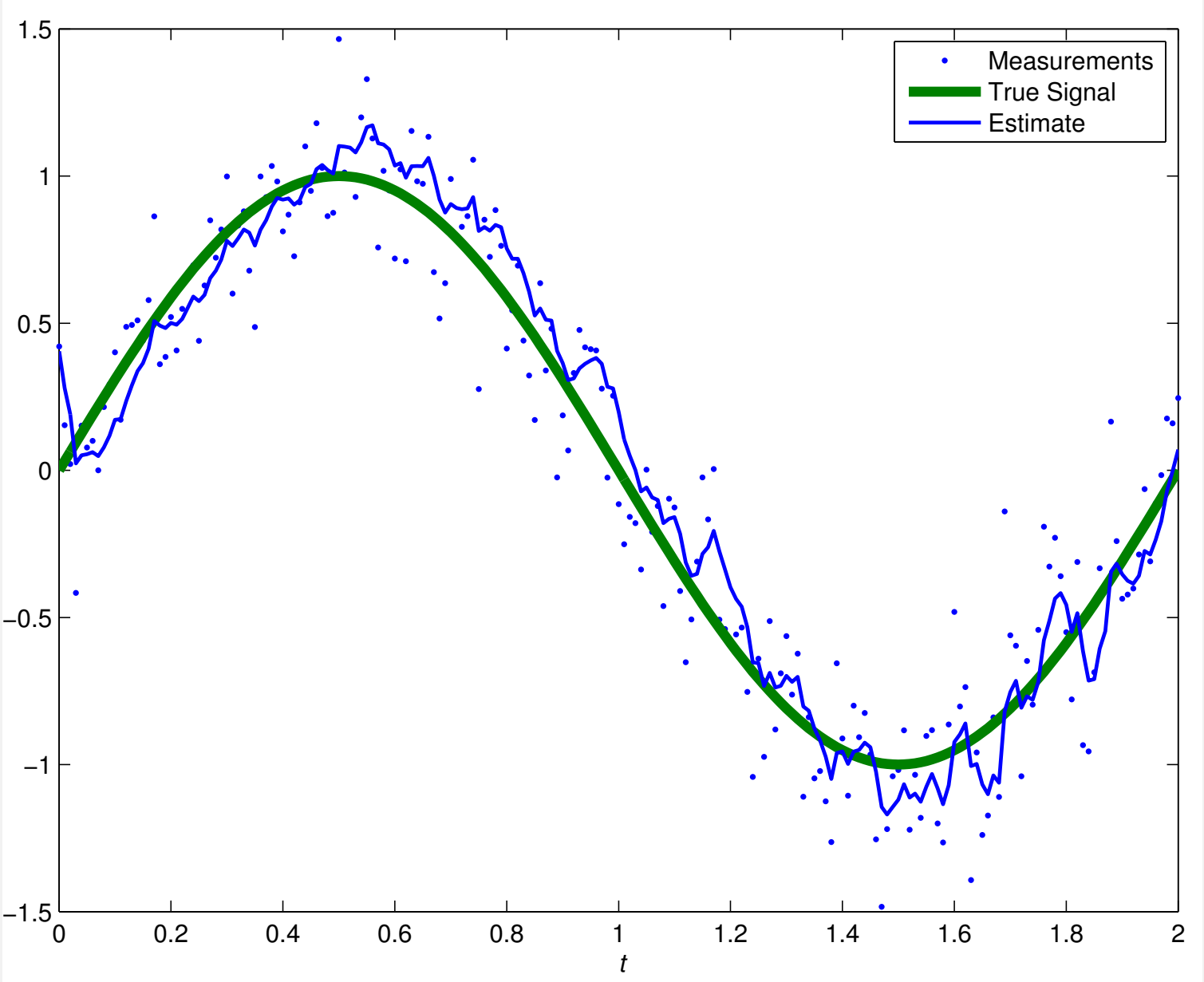
$$\begin{aligned} p(\boldsymbol{\theta}_k \mid y_{1:k}) &\propto p(y_k \mid \boldsymbol{\theta}_k)\, p(\boldsymbol{\theta}_k \mid y_{1:k-1}) \\ &\propto \mathrm{N}(\boldsymbol{\theta}_k \mid \mathbf{m}_k, \mathbf{P}_k). \end{aligned}$$

- After applying the matrix inversion lemma, mean and covariance can be written as

$$\begin{aligned} S_k &= \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \sigma^2 \\ \mathbf{K}_k &= \mathbf{P}_k^- \mathbf{H}_k^T S_k^{-1} \\ \mathbf{m}_k &= \mathbf{m}_k^- + \mathbf{K}_k [y_k - \mathbf{H}_k \mathbf{m}_k^-] \\ \mathbf{P}_k &= \mathbf{P}_k^- - \mathbf{K}_k S_k \mathbf{K}_k^T. \end{aligned}$$

- Again, we have derived a special case of the Kalman filter.
- The batch version of this solution would be much more complicated.

# Drift Model for Linear Regression [4/4]



# State Space Notation

- In the previous slide we formulated the model as

$$p(\boldsymbol{\theta}_k \,|\, \boldsymbol{\theta}_{k-1}) = \mathrm{N}(\boldsymbol{\theta}_k \,|\, \boldsymbol{\theta}_{k-1}, \mathbf{Q})$$
$$p(y_k \,|\, \boldsymbol{\theta}_k) = \mathrm{N}(y_k \,|\, \mathbf{H}_k \, \boldsymbol{\theta}_k, \sigma^2)$$

- But in Kalman filtering and control theory the vector of parameters $\boldsymbol{\theta}_k$ is usually called "state" and denoted as $\mathbf{x}_k$.
- More standard state space notation:

$$p(\mathbf{x}_k \,|\, \mathbf{x}_{k-1}) = \mathrm{N}(\mathbf{x}_k \,|\, \mathbf{x}_{k-1}, \mathbf{Q})$$
$$p(y_k \,|\, \mathbf{x}_k) = \mathrm{N}(y_k \,|\, \mathbf{H}_k \, \mathbf{x}_k, \sigma^2)$$

- Or equivalently

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \mathbf{q}_{k-1}$$
$$y_k = \mathbf{H}_k \, \mathbf{x}_k + r_k,$$

where $\mathbf{q}_{k-1} \sim \mathrm{N}(\mathbf{0}, \mathbf{Q})$, $r_k \sim \mathrm{N}(0, \sigma^2)$.

# Probabilistics State Space Models

- Generally, Markov model for the state:

$$\mathbf{x}_k \sim p(\mathbf{x}_k \,|\, \mathbf{x}_{k-1}).$$

- Likelihood distribution of the measurement:

$$\mathbf{y}_k \sim p(\mathbf{y}_k \,|\, \mathbf{x}_k).$$

- Has the form of hidden Markov model (HMM):

$$\begin{array}{lccccccccc}
\text{observed:} & \mathbf{y}_1 & & \mathbf{y}_2 & & \mathbf{y}_3 & & \mathbf{y}_4 & & \\
 & \uparrow & & \uparrow & & \uparrow & & \uparrow & & \\
\text{hidden:} & \mathbf{x}_1 & \longrightarrow & \mathbf{x}_2 & \longrightarrow & \mathbf{x}_3 & \longrightarrow & \mathbf{x}_4 & \longrightarrow & \ldots
\end{array}$$

# Probabilistics State Space Models: Example

## Example (Gaussian random walk)

Gaussian random walk model can be written as

$$\begin{aligned} x_k &= x_{k-1} + w_{k-1}, \quad & w_{k-1} &\sim \mathrm{N}(0, q) \\ y_k &= x_k + e_k, & e_k &\sim \mathrm{N}(0, r), \end{aligned}$$

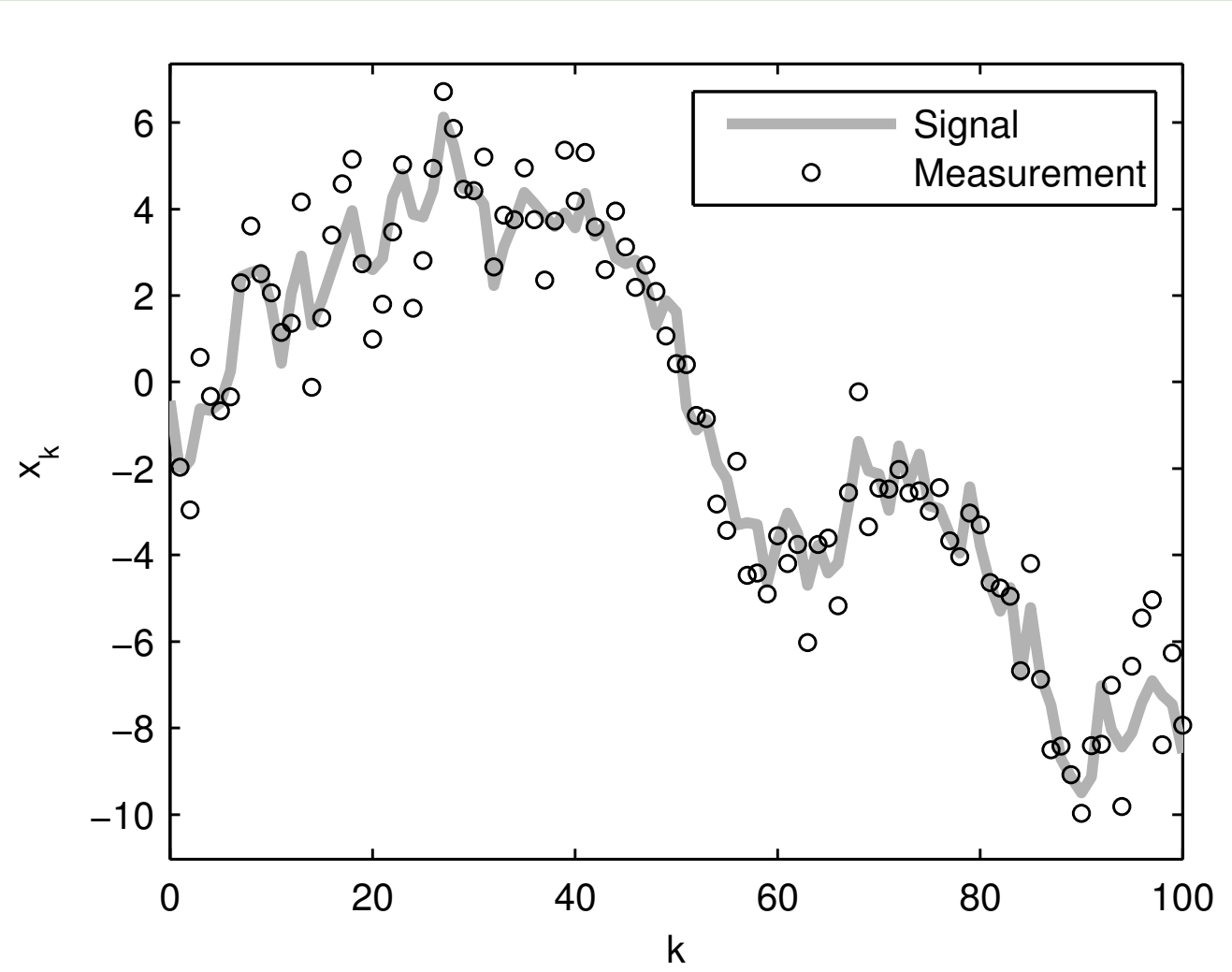
where $x_k$ is the hidden state and $y_k$ is the measurement. In terms of probability densities the model can be written as

$$\begin{aligned} p(x_k \mid x_{k-1}) &= \frac{1}{\sqrt{2\pi q}} \exp\left(-\frac{1}{2q}(x_k - x_{k-1})^2\right) \\ p(y_k \mid x_k) &= \frac{1}{\sqrt{2\pi r}} \exp\left(-\frac{1}{2r}(y_k - x_k)^2\right) \end{aligned}$$

which is a discrete-time state space model.

# Probabilistics State Space Models: Example (cont.)

## Example (Gaussian random walk (cont.))



# Probabilistics State Space Models: Further Examples

- Linear Gauss-Markov model – which defines a Gaussian process:

$$\begin{aligned} \mathbf{x}_k &= \mathbf{A}_{k-1}\,\mathbf{x}_{k-1} + \mathbf{q}_{k-1} \\ \mathbf{y}_k &= \mathbf{H}_k\,\mathbf{x}_k + \mathbf{r}_k, \end{aligned}$$

- Gaussian driven non-linear model – a non-Gaussian process:

$$\begin{aligned} \mathbf{x}_k &= \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{q}_{k-1}) \\ \mathbf{y}_k &= \mathbf{h}(\mathbf{x}_k, \mathbf{r}_k). \end{aligned}$$

- Continuous-discrete-time models – these correspond to GP regression

$$\begin{aligned} \frac{d\mathbf{x}}{dt} &= \mathbf{a}(\mathbf{x}) + \mathbf{L}(\mathbf{x})\,\mathbf{w}(t) \\ \mathbf{y}_k &\sim p(\mathbf{y}_k \,|\, \mathbf{x}(t_k)). \end{aligned}$$

# Bayesian Filtering, Prediction and Smoothing

- In principle, we could just use the (batch) Bayes' rule

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_T \,|\, \mathbf{y}_1, \ldots, \mathbf{y}_T) = \frac{p(\mathbf{y}_1, \ldots, \mathbf{y}_T \,|\, \mathbf{x}_1, \ldots, \mathbf{x}_T)\, p(\mathbf{x}_1, \ldots, \mathbf{x}_T)}{p(\mathbf{y}_1, \ldots, \mathbf{y}_T)},$$

- Curse of computational complexity: complexity grows more than linearly with number of measurements (typically we have $O(T^3)$).
- Hence, we concentrate on the following:
  - Filtering distributions:

$$p(\mathbf{x}_k \,|\, \mathbf{y}_1, \ldots, \mathbf{y}_k), \qquad k = 1, \ldots, T.$$

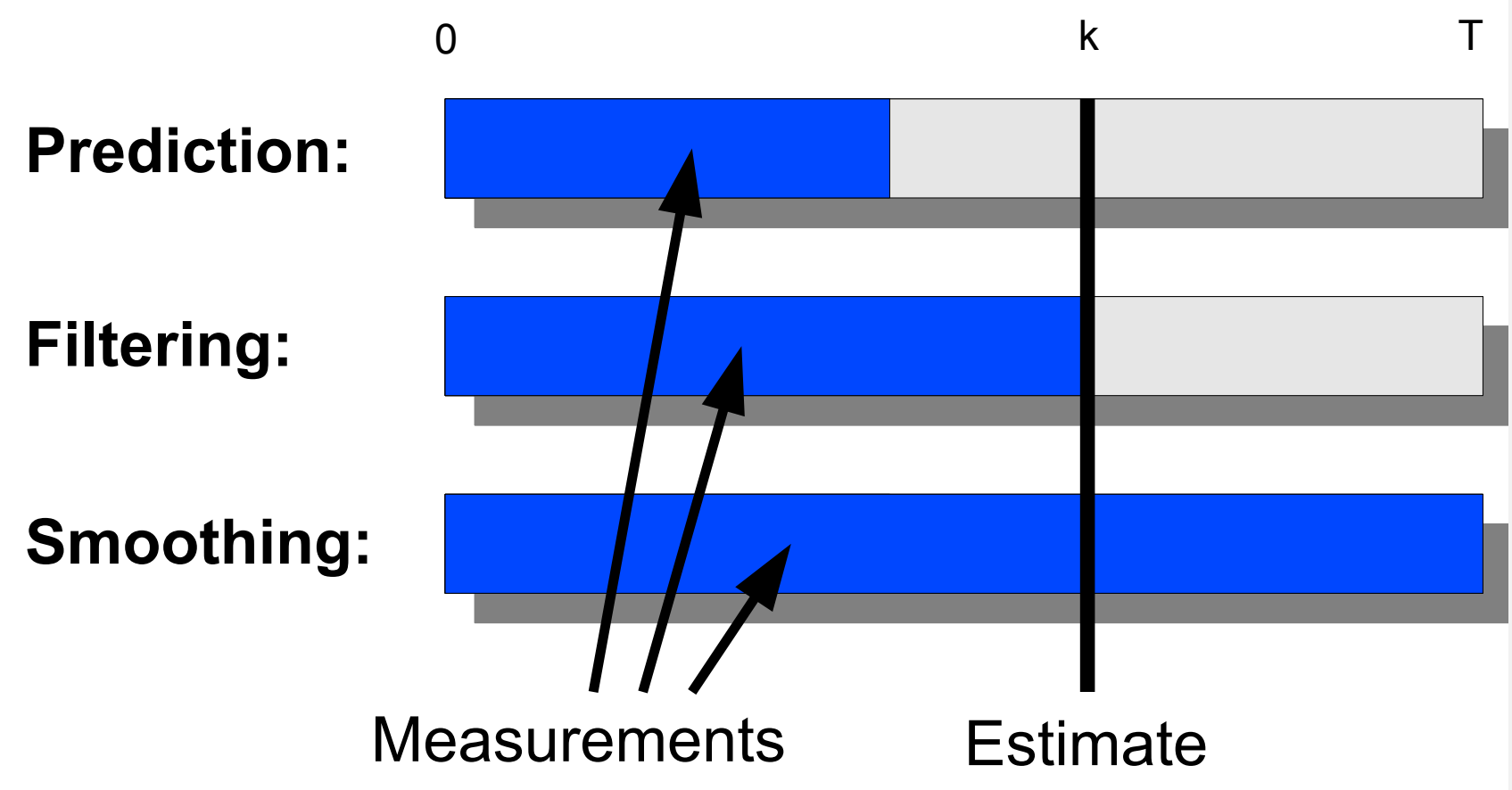
  - Prediction distributions:

$$p(\mathbf{x}_{k+n} \,|\, \mathbf{y}_1, \ldots, \mathbf{y}_k), \qquad k = 1, \ldots, T, \quad n = 1, 2, \ldots,$$

  - Smoothing distributions:

$$p(\mathbf{x}_k \,|\, \mathbf{y}_1, \ldots, \mathbf{y}_T), \qquad k = 1, \ldots, T.$$

# Bayesian Filtering, Prediction and Smoothing (cont.)



# Bayesian Filter: Principle

- Bayesian optimal filter computes the distribution

$$p(\mathbf{x}_k \mid \mathbf{y}_{1:k})$$

- Given the following:
  1. Prior distribution $p(\mathbf{x}_0)$.
  2. State space model:

$$\begin{aligned} \mathbf{x}_k &\sim p(\mathbf{x}_k \mid \mathbf{x}_{k-1}) \\ \mathbf{y}_k &\sim p(\mathbf{y}_k \mid \mathbf{x}_k), \end{aligned}$$

  3. Measurement sequence $\mathbf{y}_{1:k} = \mathbf{y}_1, \ldots, \mathbf{y}_k$.
- Computation is based on recursion rule for incorporation of the new measurement $\mathbf{y}_k$ into the posterior:

$$p(\mathbf{x}_{k-1} \mid \mathbf{y}_{1:k-1}) \longrightarrow p(\mathbf{x}_k \mid \mathbf{y}_{1:k})$$

# Bayesian Optimal Filter: Formal Equations

## Optimal filter

- Initialization: The recursion starts from the prior distribution $p(\mathbf{x}_0)$.
- Prediction: by the Chapman-Kolmogorov equation

$$p(\mathbf{x}_k \,|\, \mathbf{y}_{1:k-1}) = \int p(\mathbf{x}_k \,|\, \mathbf{x}_{k-1})\, p(\mathbf{x}_{k-1} \,|\, \mathbf{y}_{1:k-1})\, \mathrm{d}\mathbf{x}_{k-1}.$$

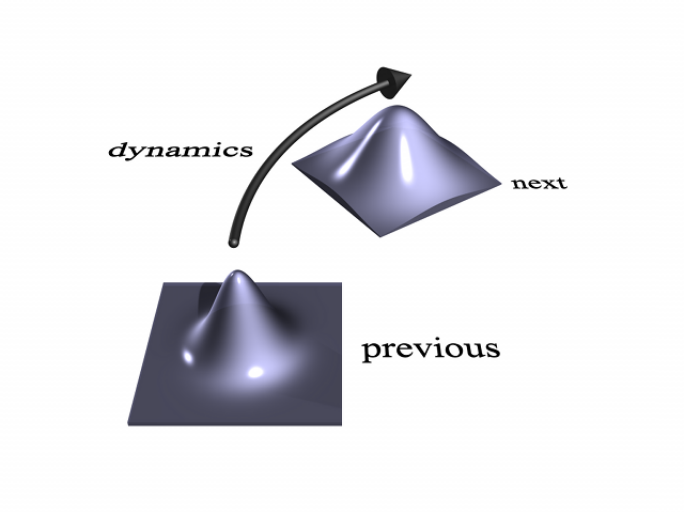
- Update: by the Bayes' rule

$$p(\mathbf{x}_k \,|\, \mathbf{y}_{1:k}) = \frac{1}{Z_k} p(\mathbf{y}_k \,|\, \mathbf{x}_k)\, p(\mathbf{x}_k \,|\, \mathbf{y}_{1:k-1}).$$

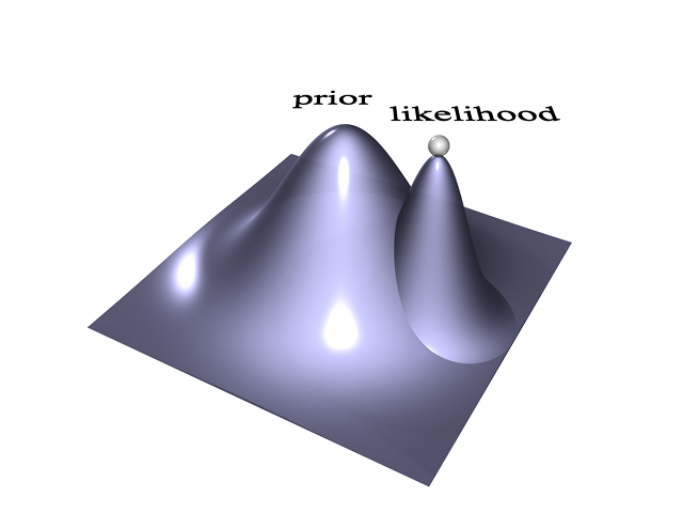
- The normalization constant $Z_k = p(\mathbf{y}_k \,|\, \mathbf{y}_{1:k-1})$ is given as

$$Z_k = \int p(\mathbf{y}_k \,|\, \mathbf{x}_k)\, p(\mathbf{x}_k \,|\, \mathbf{y}_{1:k-1})\, \mathrm{d}\mathbf{x}_k.$$

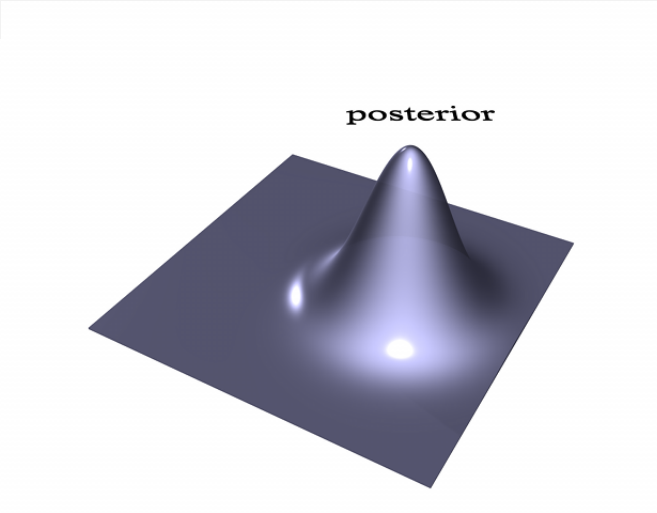
# Bayesian Optimal Filter: Graphical Explanation




On prediction step the distribution of previous step is propagated through the dynamics.




Prior distribution from prediction and the likelihood of measurement.

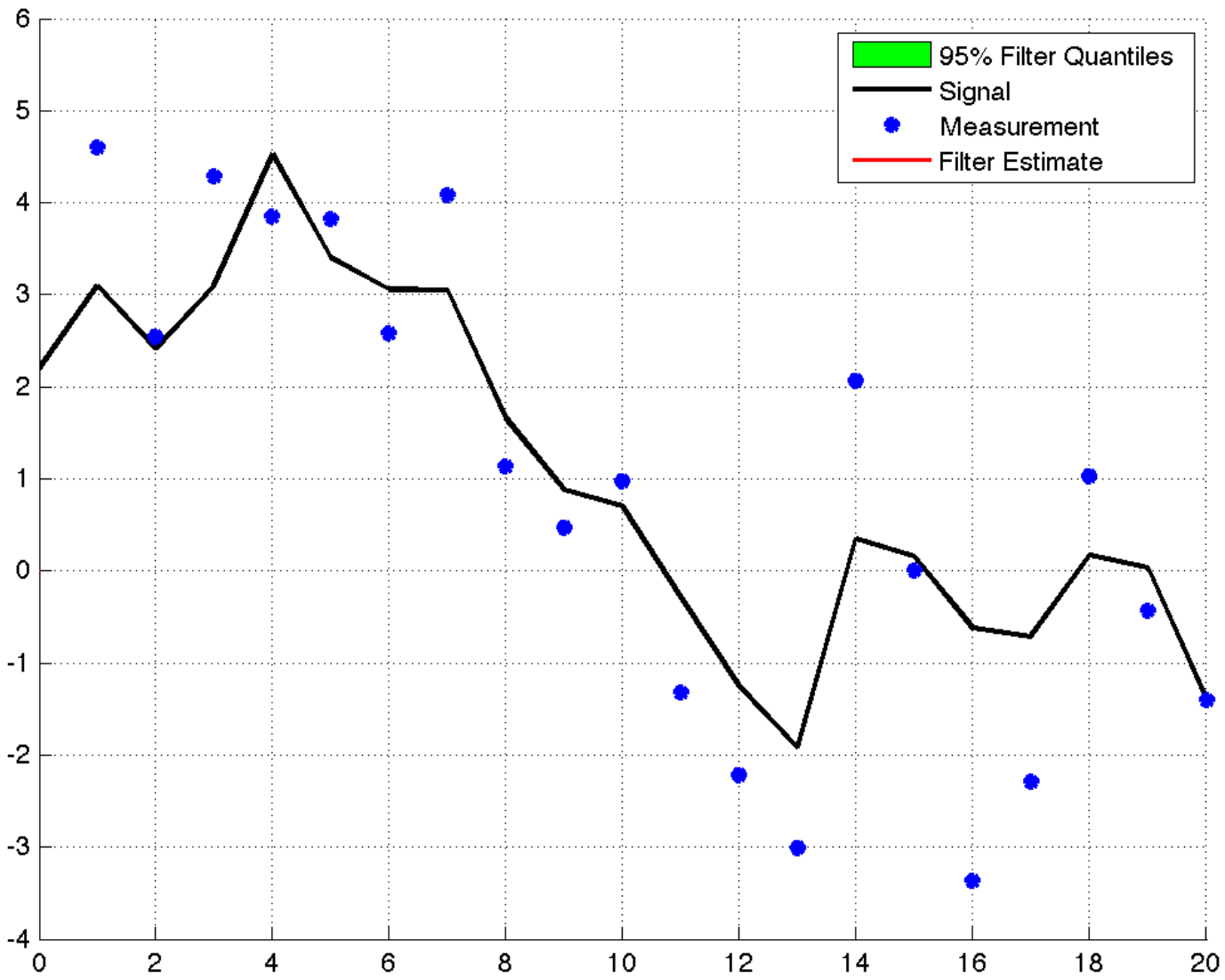



The posterior distribution after combining the prior and likelihood by Bayes' rule.

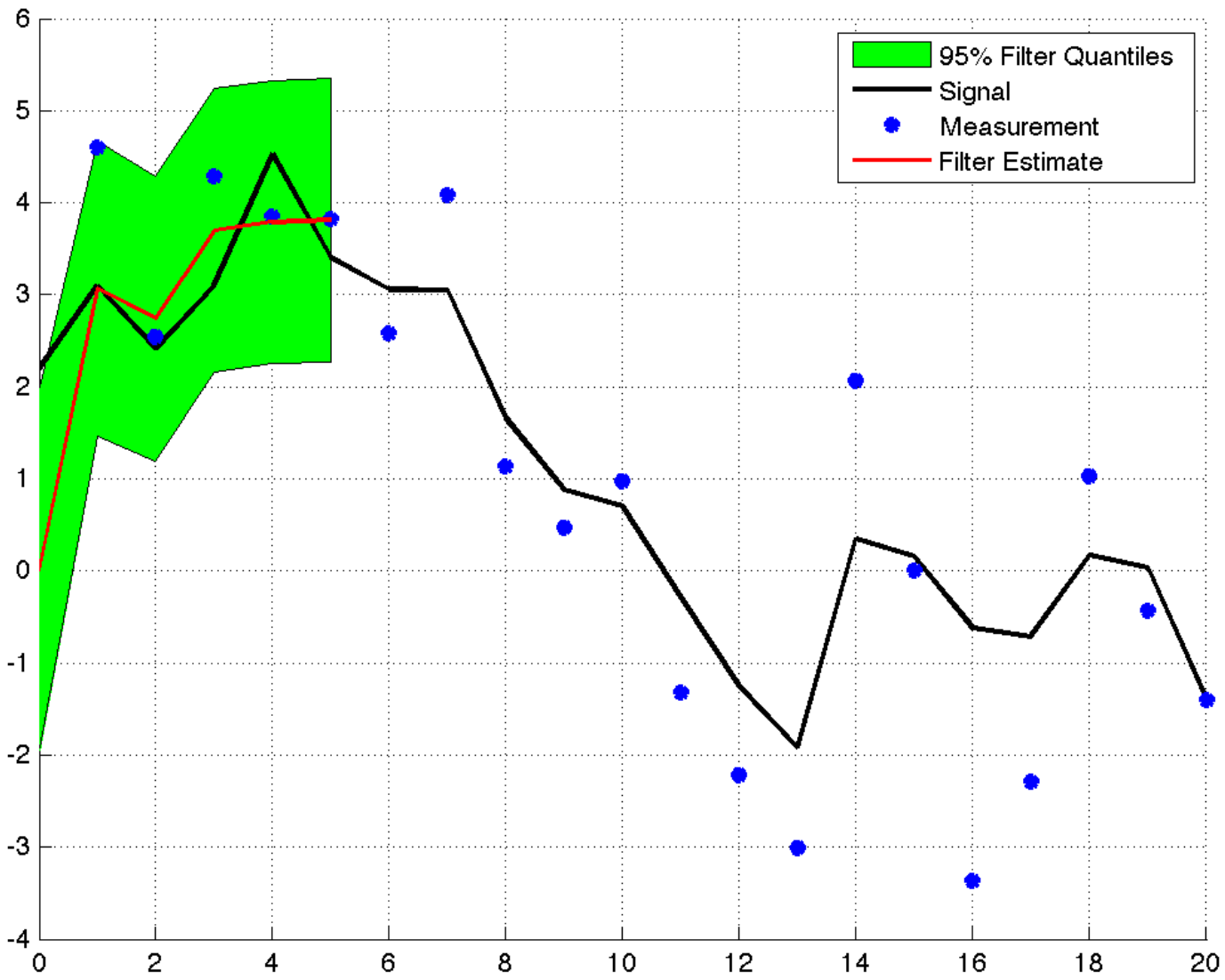
# Filtering Algorithms

- Kalman filter is the classical optimal filter for linear-Gaussian models.
- Extended Kalman filter (EKF) is linearization based extension of Kalman filter to non-linear models.
- Unscented Kalman filter (UKF) is sigma-point transformation based extension of Kalman filter.
- Gauss-Hermite and Cubature Kalman filters (GHKF/CKF) are numerical integration based extensions of Kalman filter.
- Particle filter forms a Monte Carlo representation (particle set) to the distribution of the state estimate.
- Grid based filters approximate the probability distributions on a finite grid.
- Mixture Gaussian approximations are used, for example, in multiple model Kalman filters and Rao-Blackwellized Particle filters.

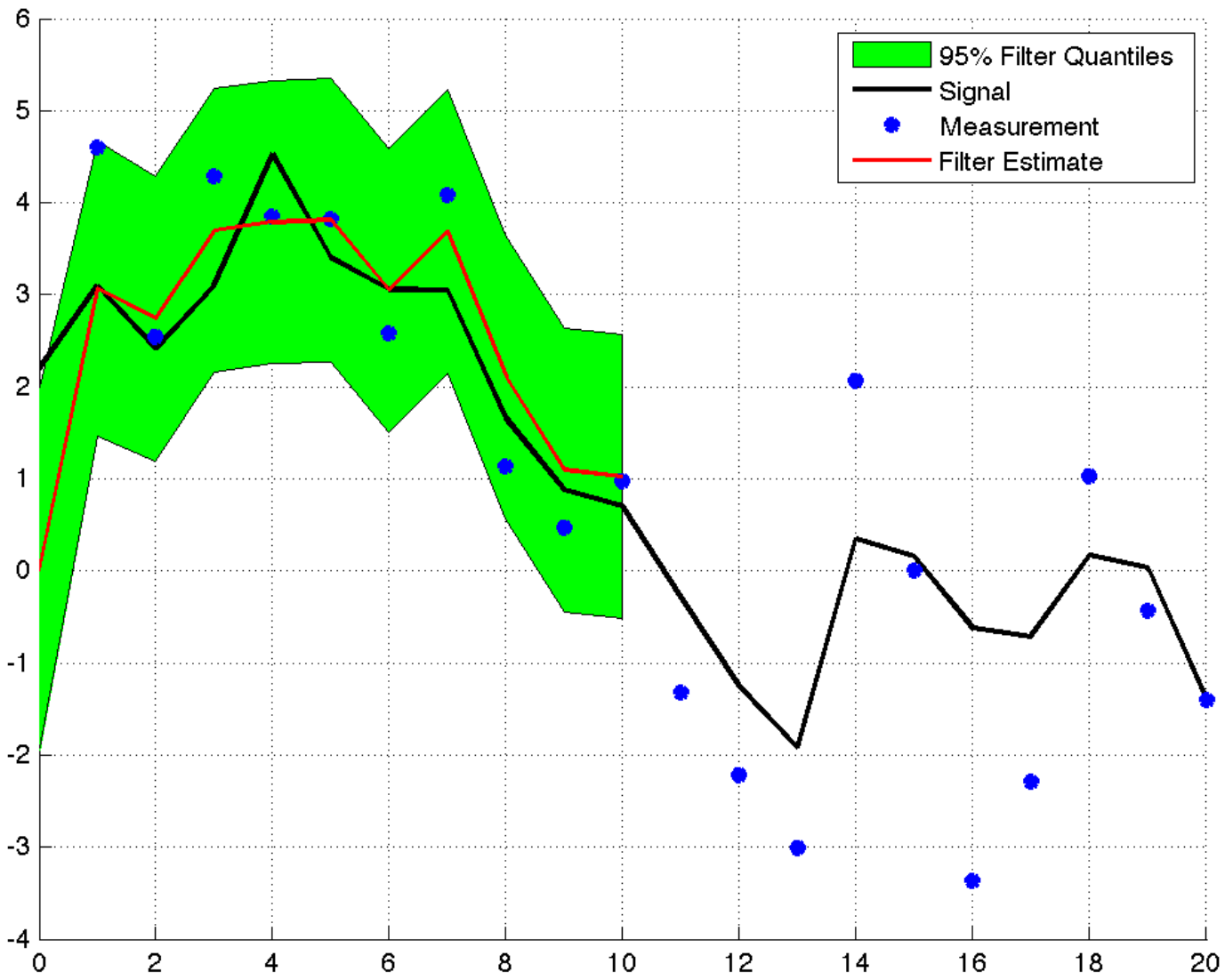
# Kalman Filter Example



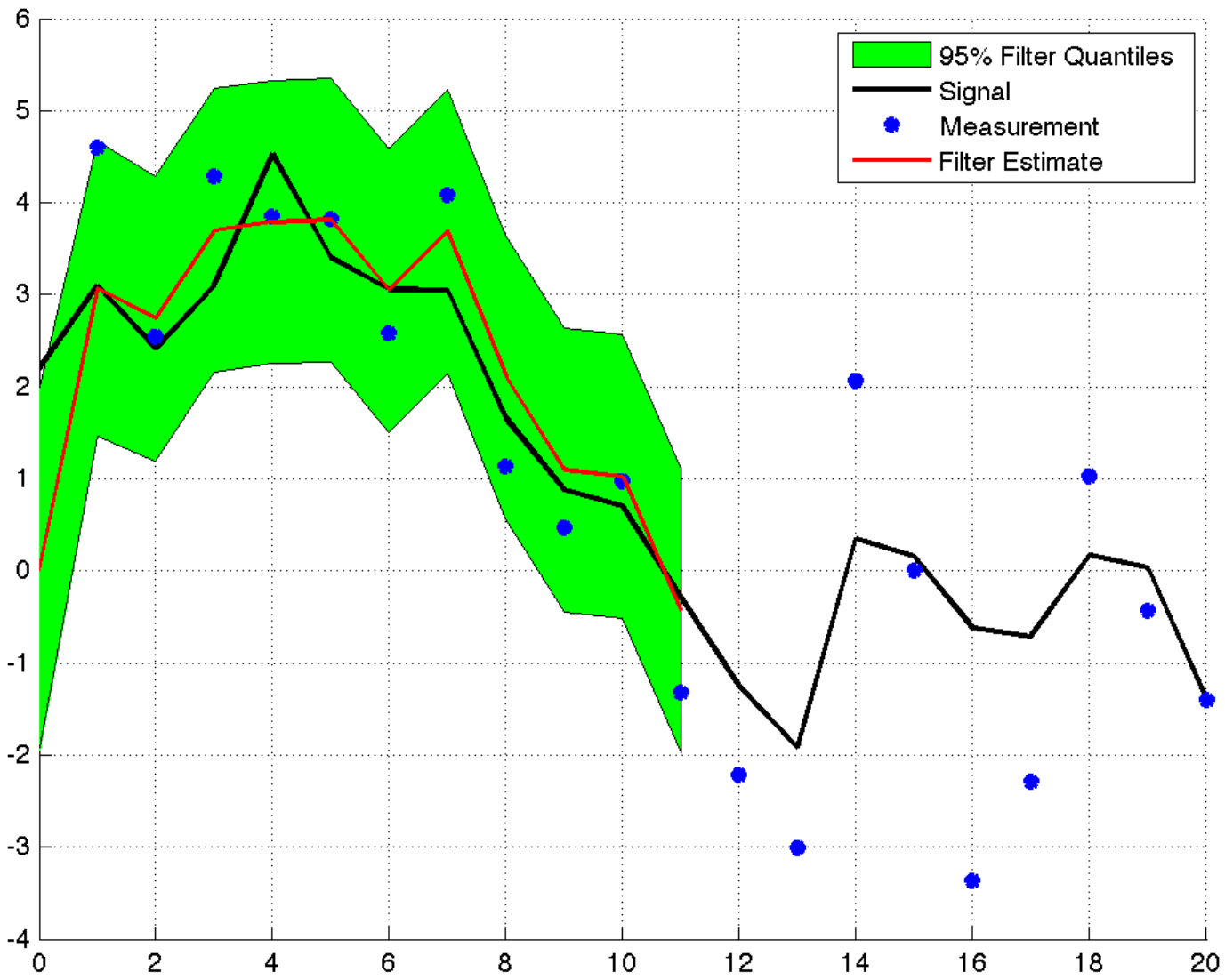
# Kalman Filter Example



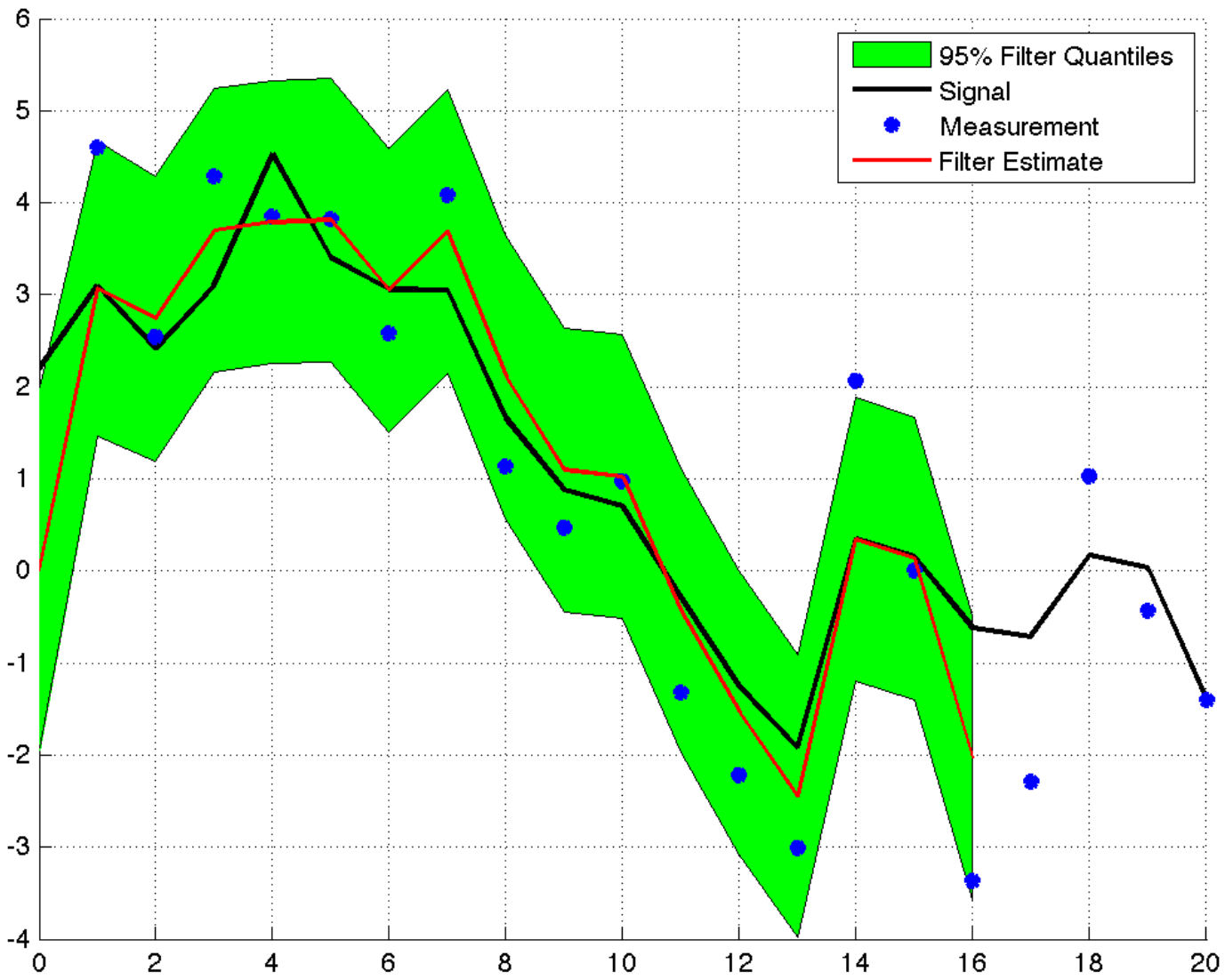
# Kalman Filter Example



# Kalman Filter Example



# Kalman Filter Example



# Bayesian Smoothing Problem

- We have a probabilistic state space model:

$$\mathbf{y}_k \sim p(\mathbf{y}_k \mid \mathbf{x}_k)$$
$$\mathbf{x}_k \sim p(\mathbf{x}_k \mid \mathbf{x}_{k-1})$$

- Assume that the filtering distributions $p(\mathbf{x}_k \mid \mathbf{y}_{1:k})$ have already been computed for all $k = 0, \ldots, T$.
- We want recursive equations of computing the smoothing distribution for all $k < T$:

$$p(\mathbf{x}_k \mid \mathbf{y}_{1:T}).$$

- The recursion will go backwards in time, because on the last step, the filtering and smoothing distributions coincide:

$$p(\mathbf{x}_T \mid \mathbf{y}_{1:T}).$$

# Bayesian Smoothing Equations

## Bayesian Smoothing Equations

The Bayesian smoothing equations consist of prediction step and backward update step:

$$p(\mathbf{x}_{k+1} \,|\, \mathbf{y}_{1:k}) = \int p(\mathbf{x}_{k+1} \,|\, \mathbf{x}_k)\, p(\mathbf{x}_k \,|\, \mathbf{y}_{1:k})\, \mathrm{d}\mathbf{x}_k$$

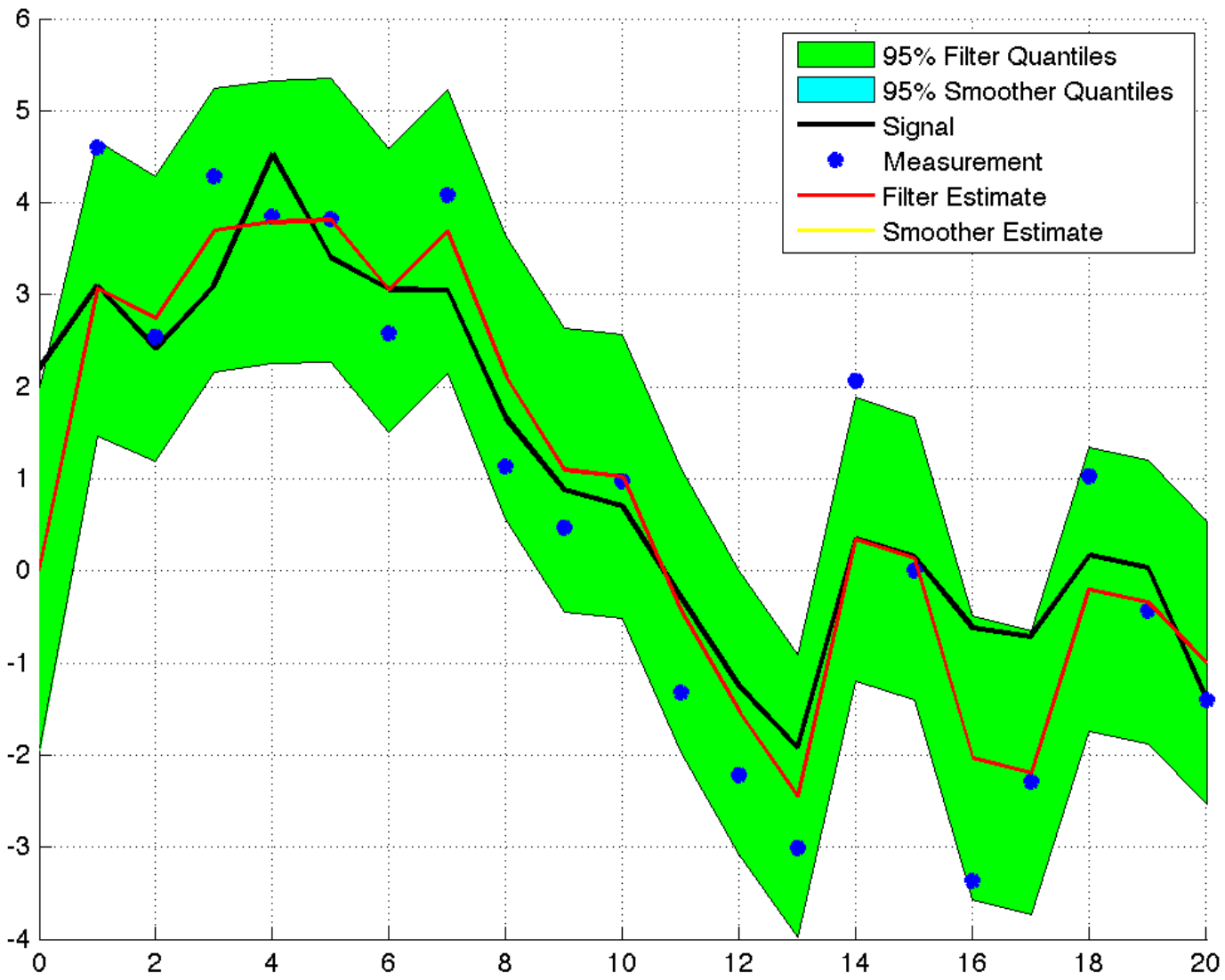
$$p(\mathbf{x}_k \,|\, \mathbf{y}_{1:T}) = p(\mathbf{x}_k \,|\, \mathbf{y}_{1:k}) \int \left[ \frac{p(\mathbf{x}_{k+1} \,|\, \mathbf{x}_k)\, p(\mathbf{x}_{k+1} \,|\, \mathbf{y}_{1:T})}{p(\mathbf{x}_{k+1} \,|\, \mathbf{y}_{1:k})} \right] \mathrm{d}\mathbf{x}_{k+1}$$

The recursion is started from the filtering (and smoothing) distribution of the last time step $p(\mathbf{x}_T \,|\, \mathbf{y}_{1:T})$.

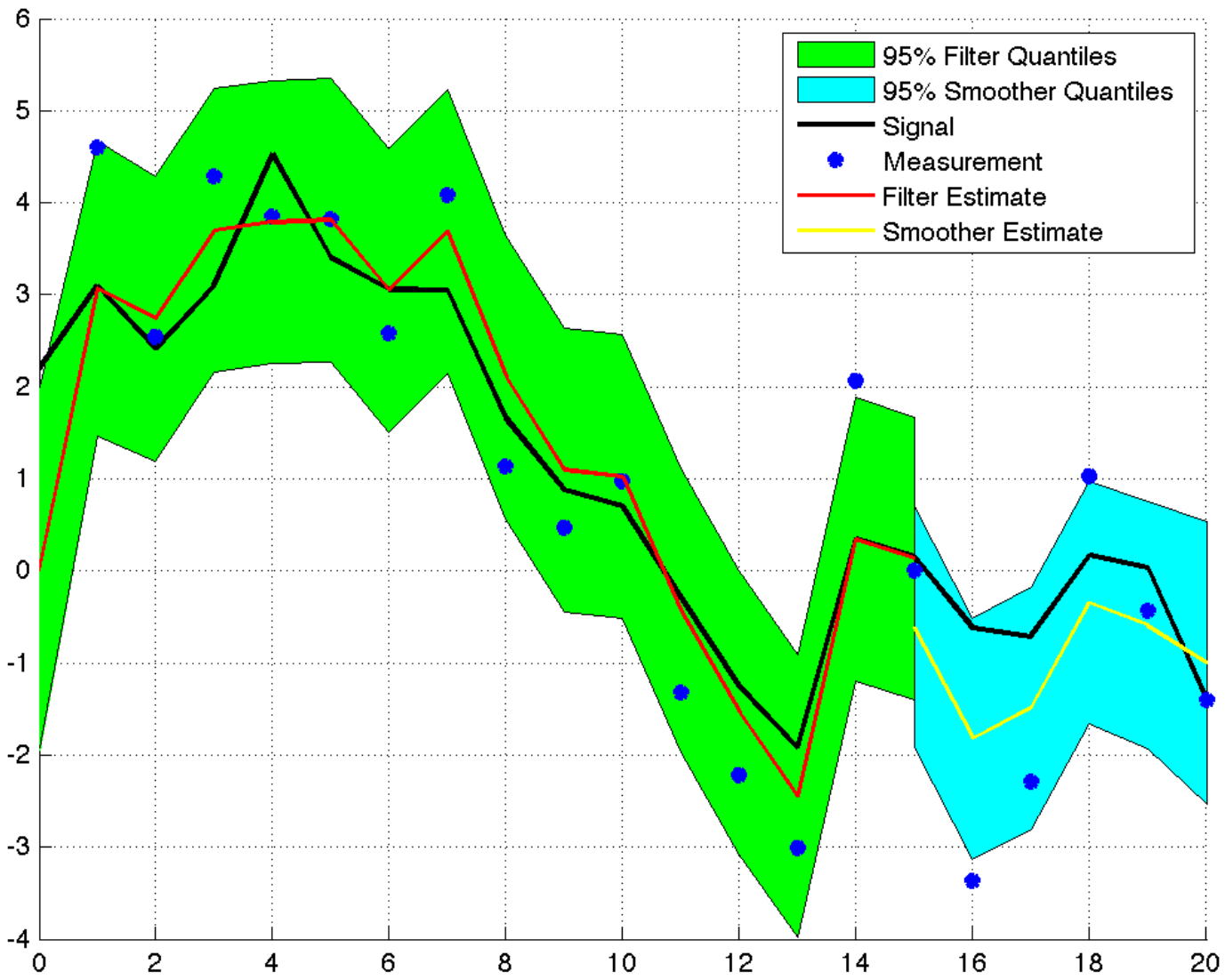
# Smoothing Algorithms

- Rauch-Tung-Striebel (RTS) smoother is the closed form smoother for linear Gaussian models.
- Extended, statistically linearized and unscented RTS smoothers are the approximate nonlinear smoothers corresponding to EKF, SLF and UKF.
- Gaussian RTS smoothers: cubature RTS smoother, Gauss-Hermite RTS smoothers and various others
- Particle smoothing is based on approximating the smoothing solutions via Monte Carlo.
- Rao-Blackwellized particle smoother is a combination of particle smoothing and RTS smoothing.

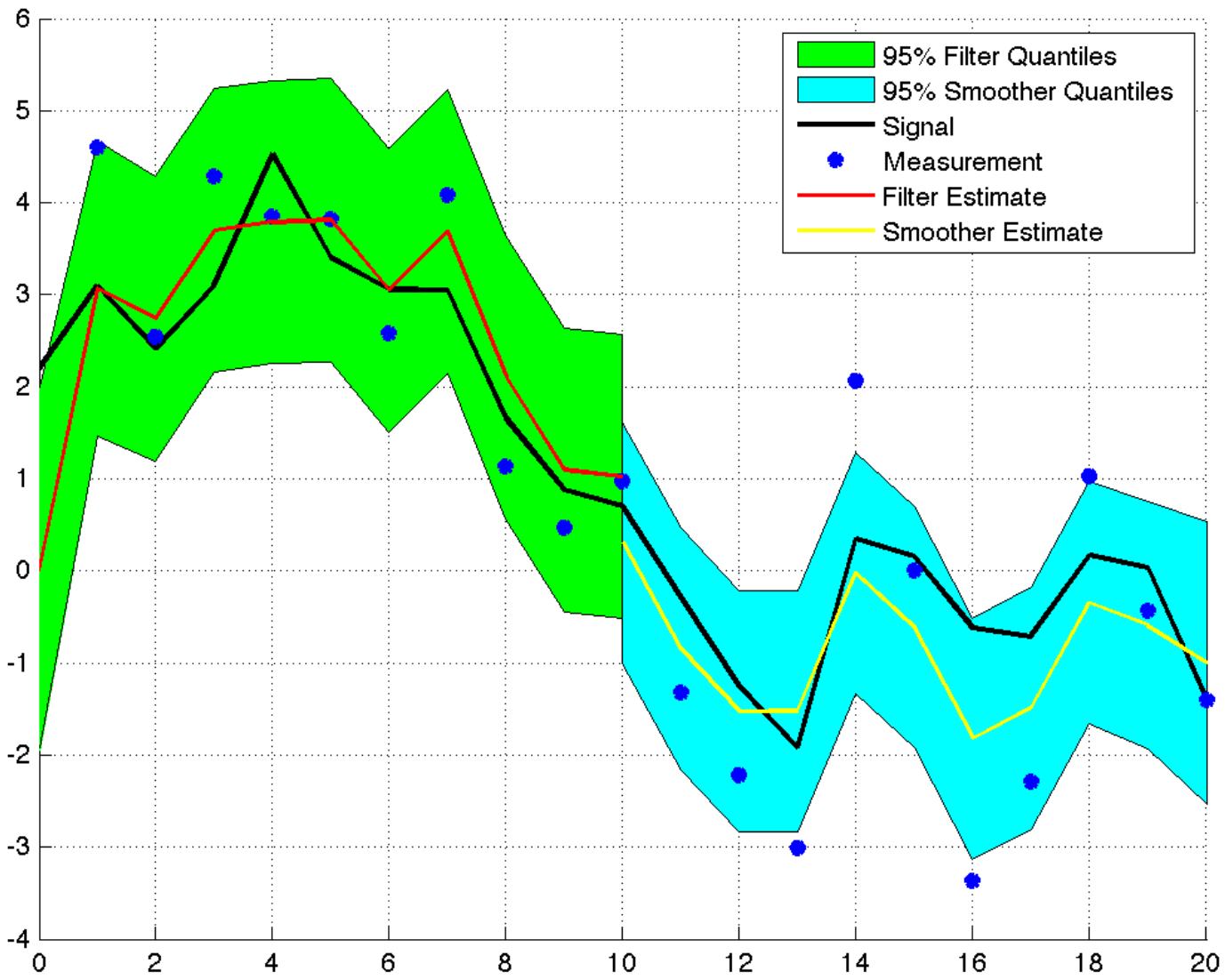
# Rauch-Tung-Striebel Smoother Example



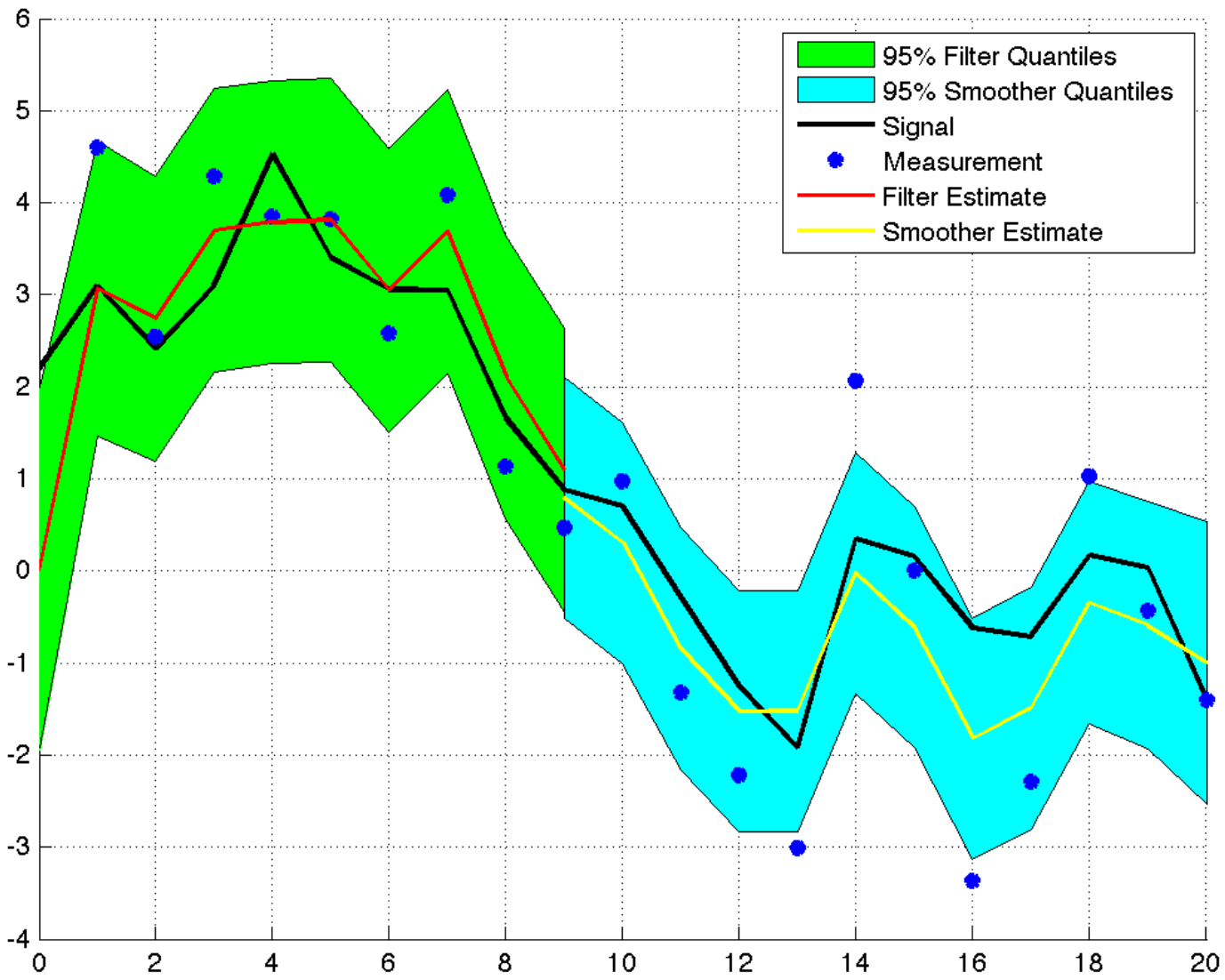
# Rauch-Tung-Striebel Smoother Example



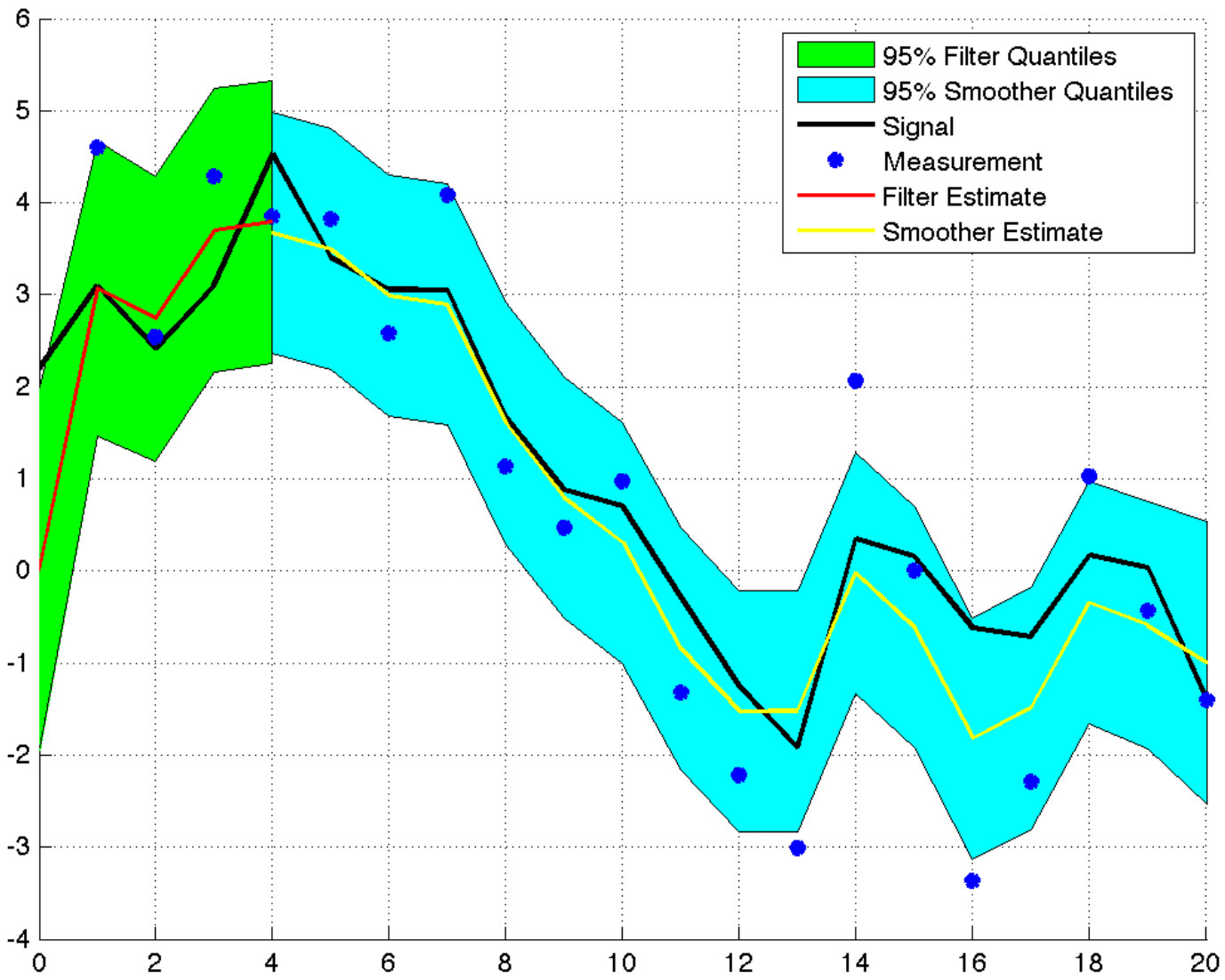
# Rauch-Tung-Striebel Smoother Example



# Rauch-Tung-Striebel Smoother Example



# Rauch-Tung-Striebel Smoother Example



# Continuous-time Limit of the Random Walk Model [1/3]

- Our discrete-time random walk model had the form

$$\begin{aligned} x_k &= x_{k-1} + w_{k-1}, \quad & w_{k-1} &\sim \mathrm{N}(0, q) \\ y_k &= x_k + e_k, & e_k &\sim \mathrm{N}(0, r), \end{aligned}$$

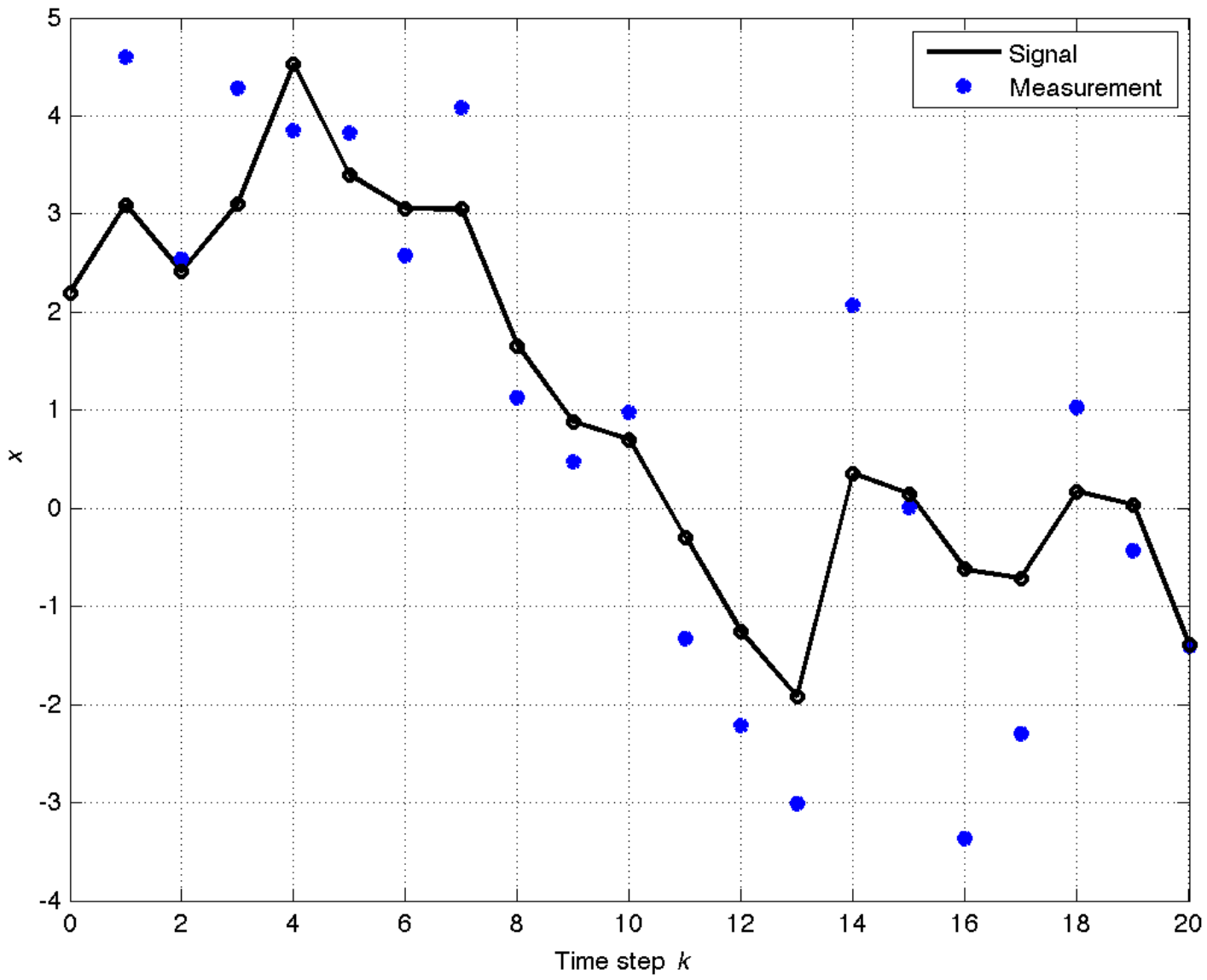
- Assume that, in fact, it was a discretely-observed random walk model

$$\begin{aligned} x(t_k) &= x(t_{k-1}) + w_{k-1}, \quad & w_{k-1} &\sim \mathrm{N}(0, q) \\ y_k &= x(t_k) + e_k, & e_k &\sim \mathrm{N}(0, r). \end{aligned}$$

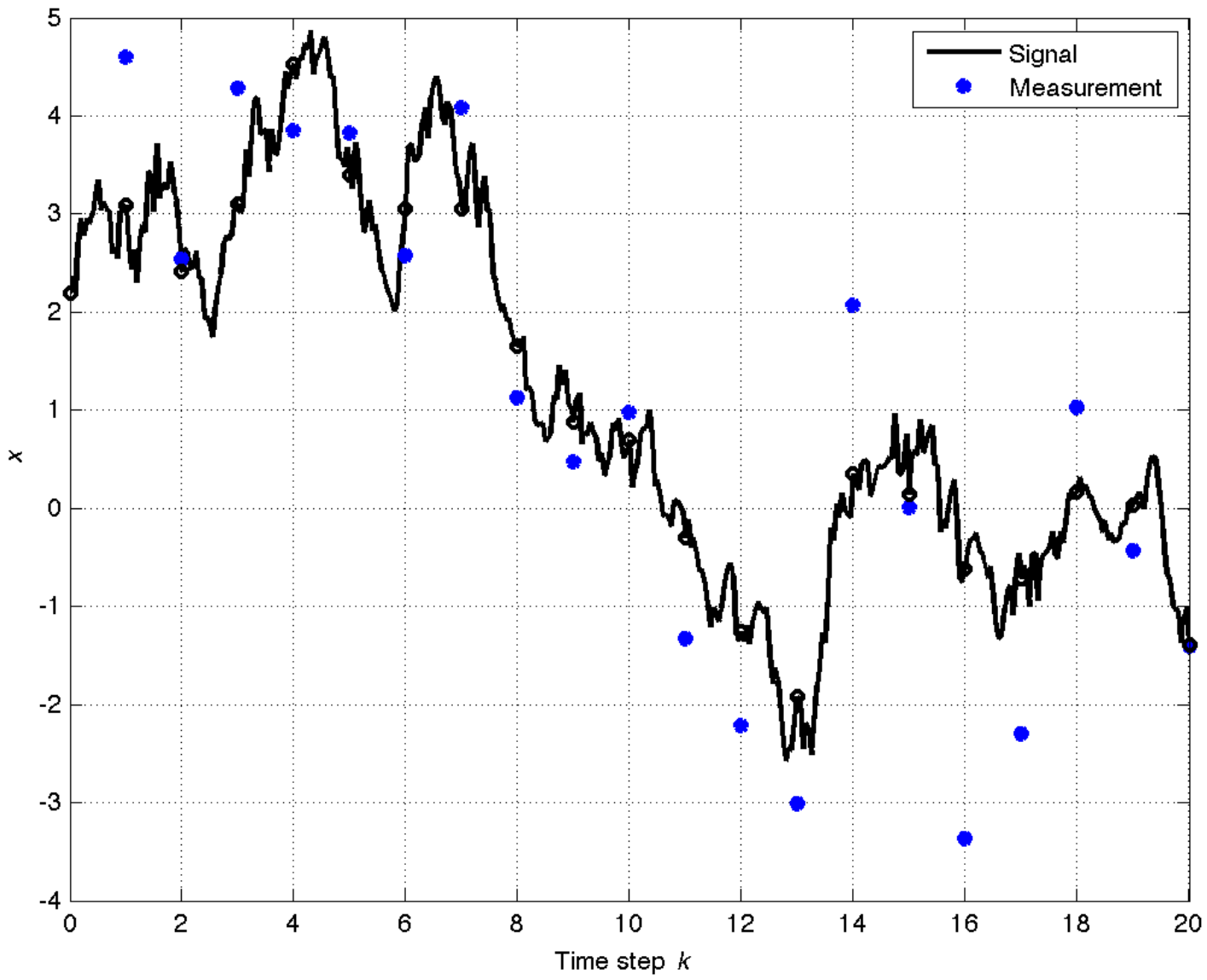
- Between the measurements we actually have $n$ intermediate steps:

$$\begin{array}{ccccccccc} \mathbf{y}_1 & & & & \mathbf{y}_2 & & & \mathbf{y}_3 & \dots \\ \uparrow & & & & \uparrow & & & \uparrow & \\ \mathbf{x}(t_1) & \succ & \mathbf{x}(t_1') & \succ & \mathbf{x}(t_1'') & \dots & \mathbf{x}(t_2) \succ \mathbf{x}(t_2') & \dots & \mathbf{x}(t_3) \quad \dots \end{array}$$

# Continuous-time Limit of the Random Walk Model [2/3]



# Continuous-time Limit of the Random Walk Model [2/3]



# Continuous-time Limit of the Random Walk Model [3/3]

- With $n = \infty$ intermediate steps we get a discretely observed stochastic differential equation (SDE) model:

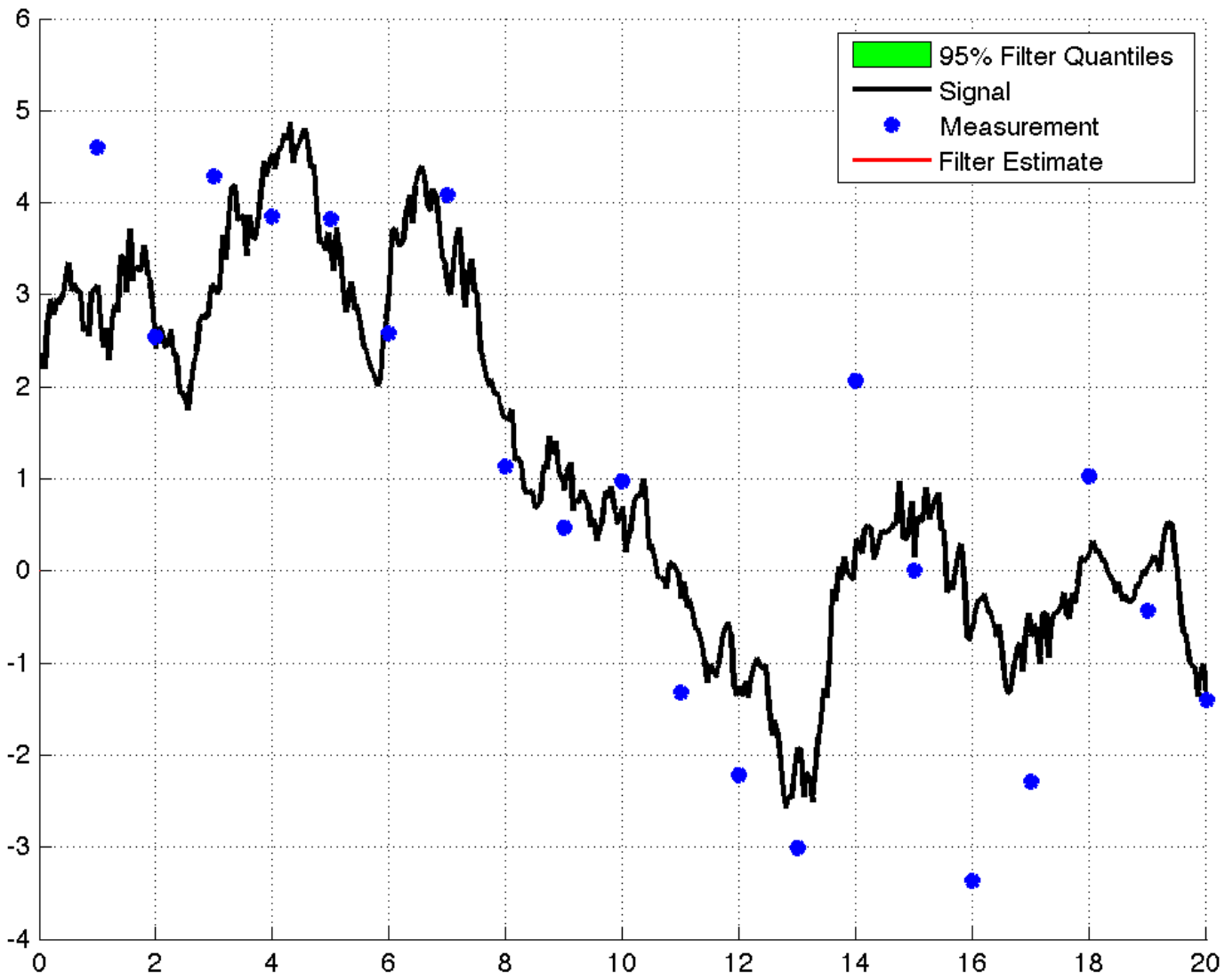
$$\frac{dx(t)}{dt} = w(t),$$
$$y_k = x(t_k) + e_k, \quad e_k \sim \mathrm{N}(0, r).$$

- Now $w(t)$ is a continuous-time white noise with spectral density $q$.
- The process defined as $dx(t)/dt = w(t)$ is called Brownian motion.
- Hence, we actually have a Gaussian process regression model

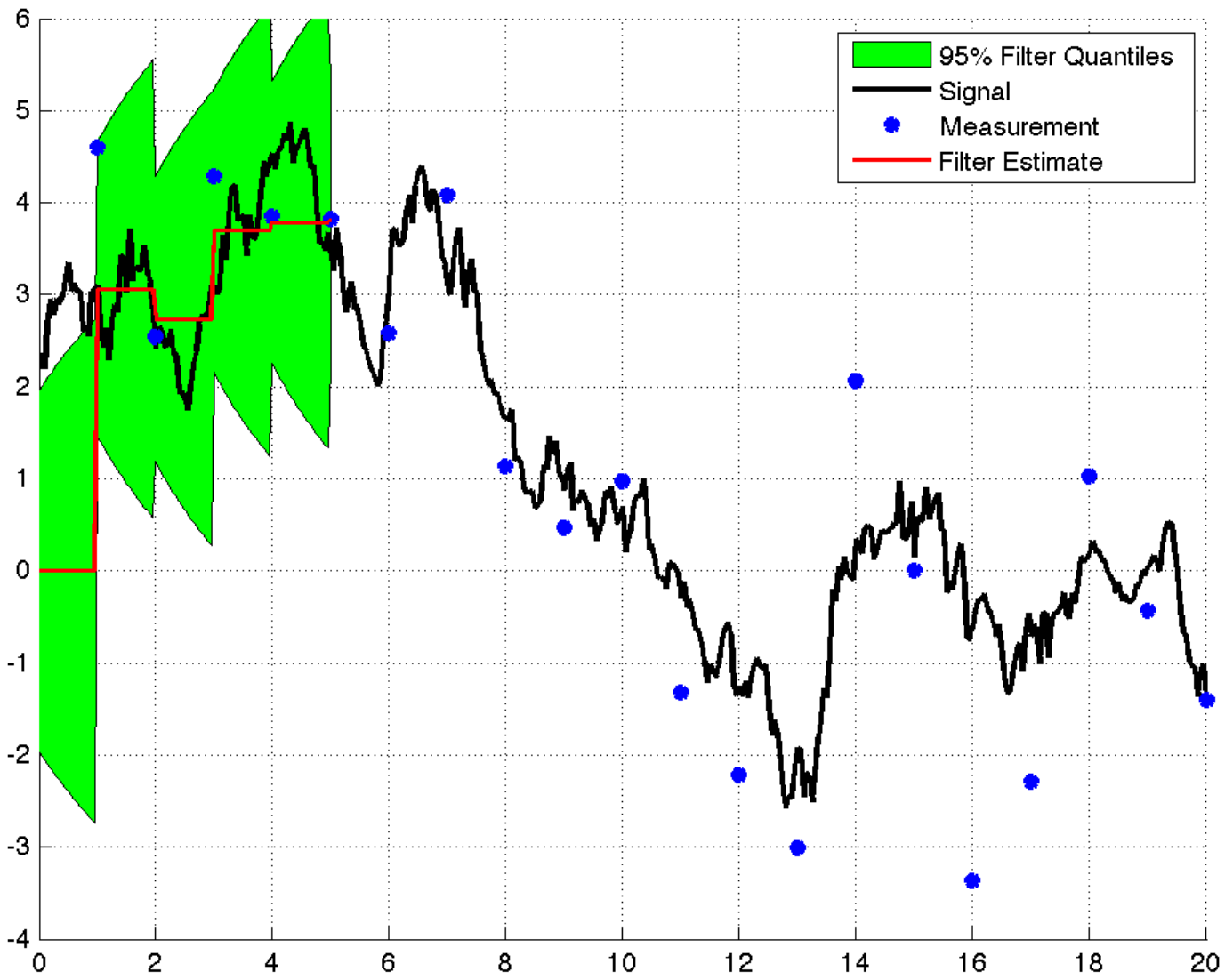
$$x(t) \sim \mathcal{GP}(0, q \min(t, t')),$$
$$y_k = x(t_k) + e_k, \qquad e_k \sim \mathrm{N}(0, r).$$

- Kalman filter and RTS smoother can still be used to compute the posterior of $x(t)$ given the observations $y_1, \ldots, y_T$ ...which is the GP regression solution!

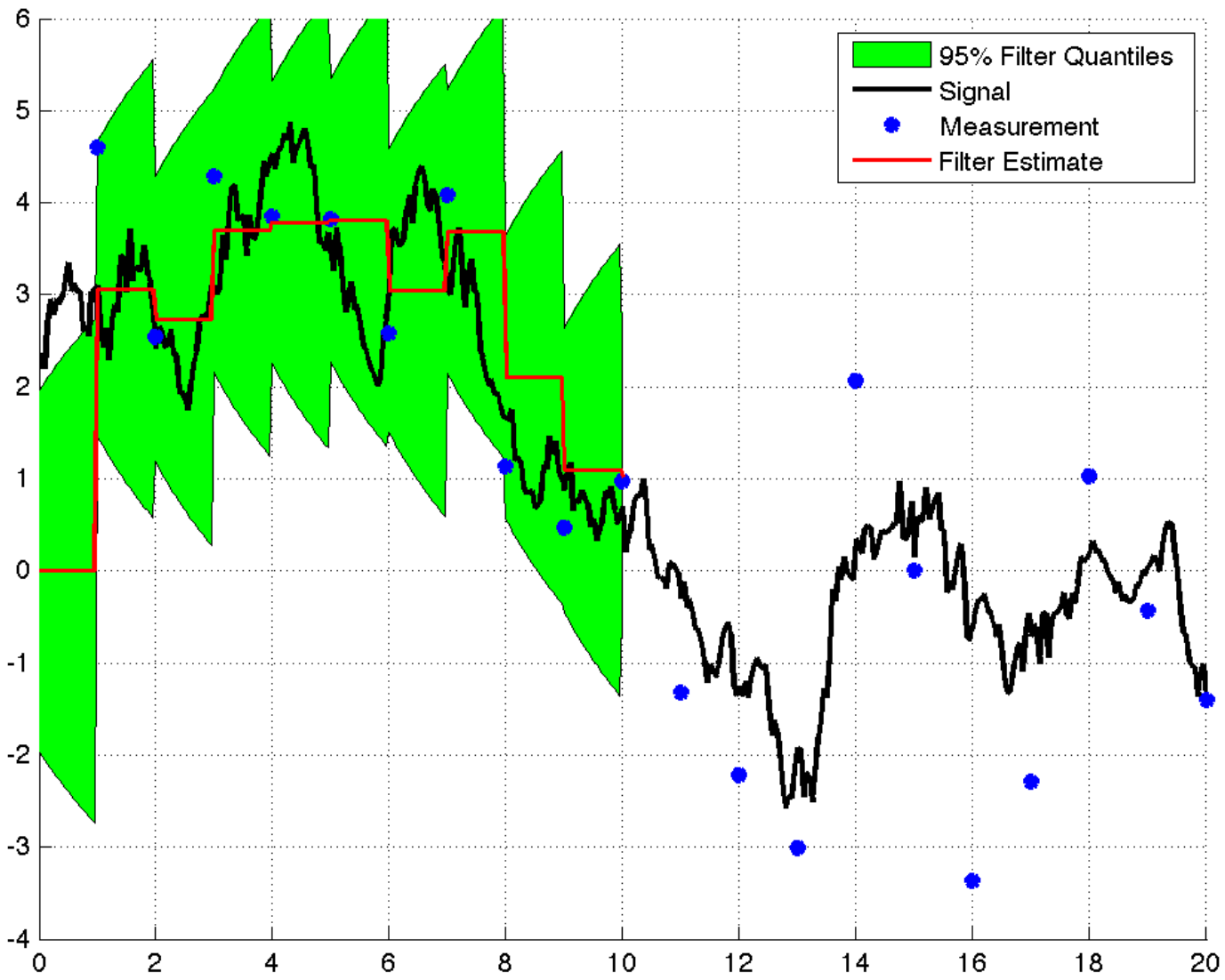
# Continuous-discrete (-time) Filter and Smoother Example



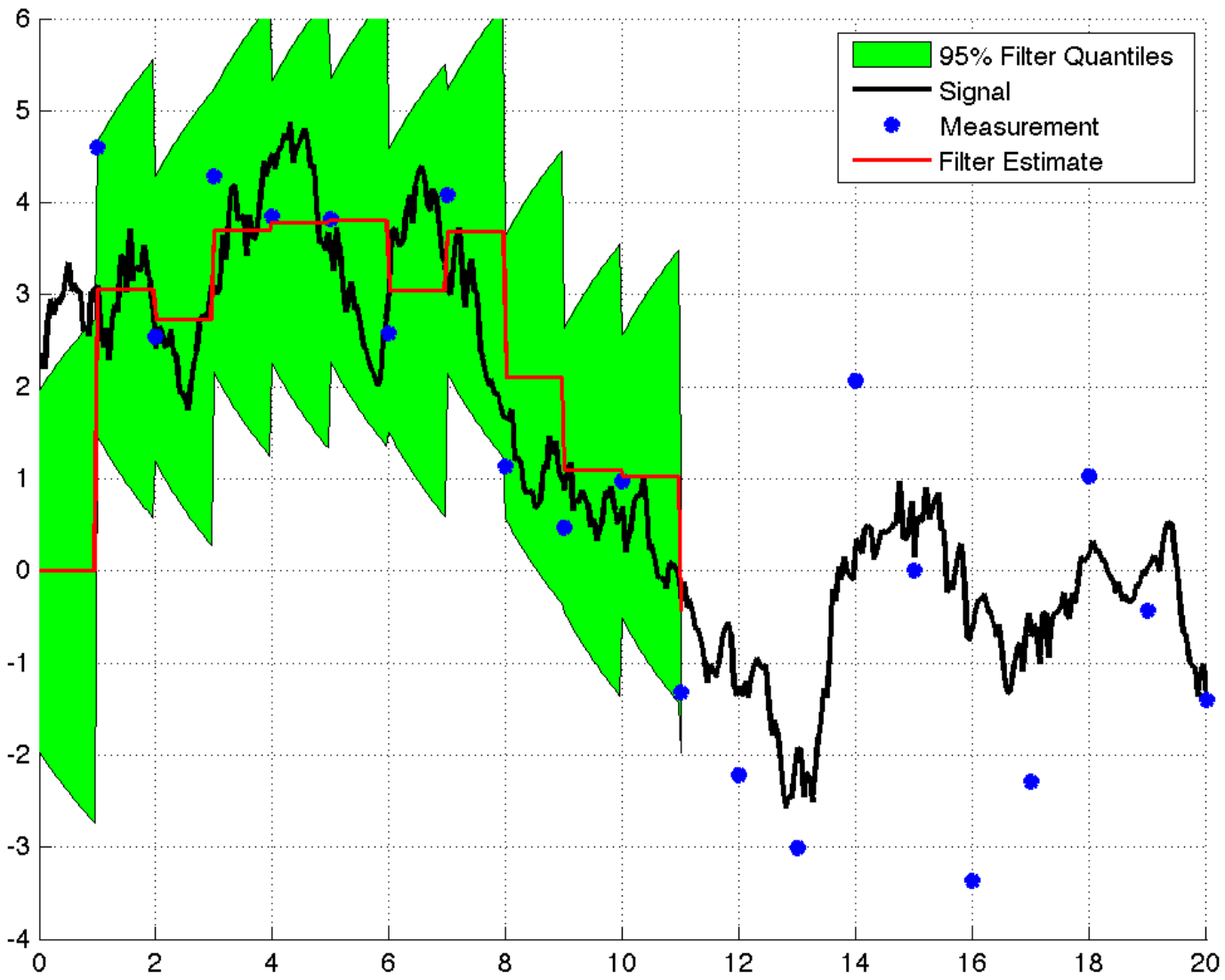
# Continuous-discrete (-time) Filter and Smoother Example



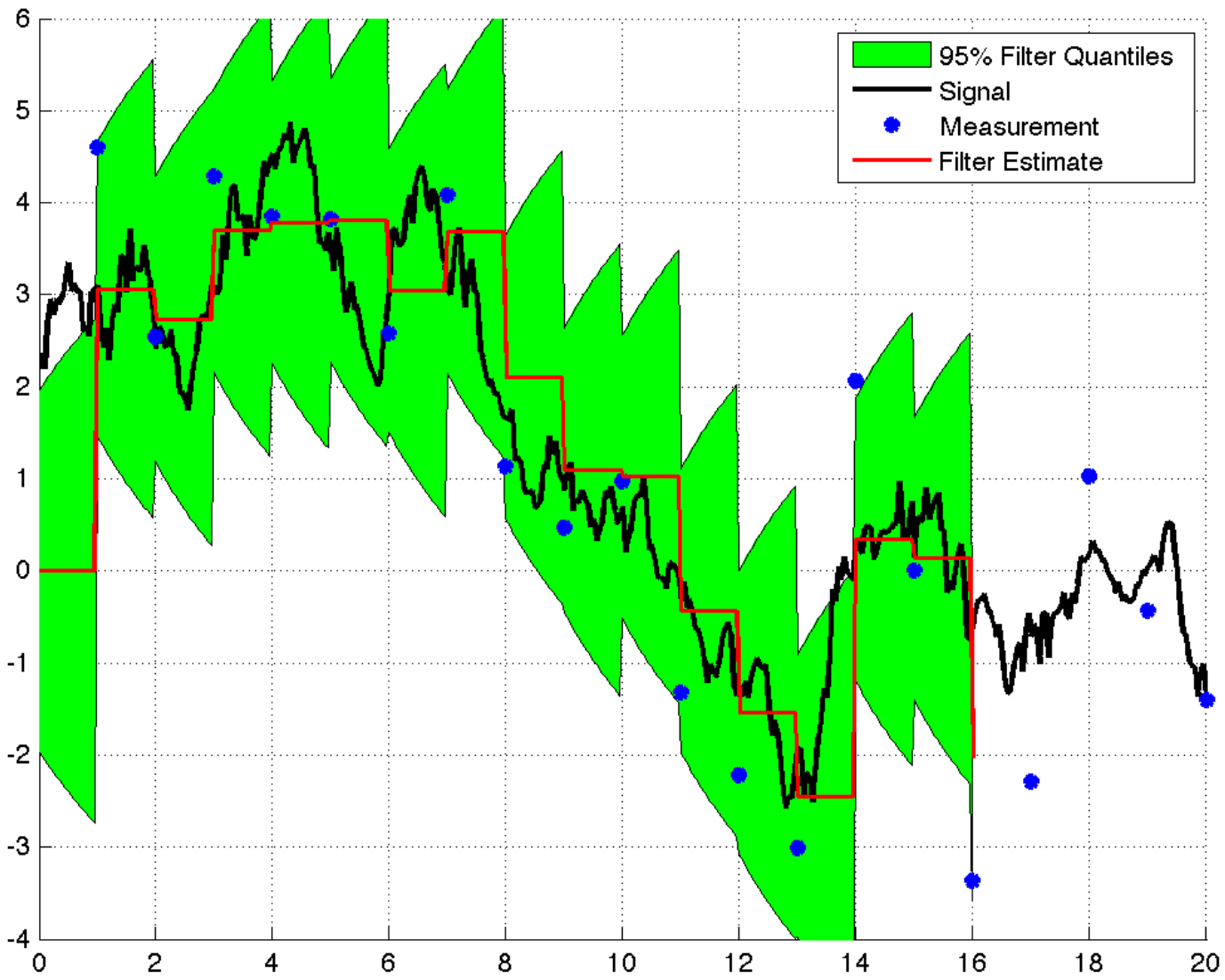
# Continuous-discrete (-time) Filter and Smoother Example



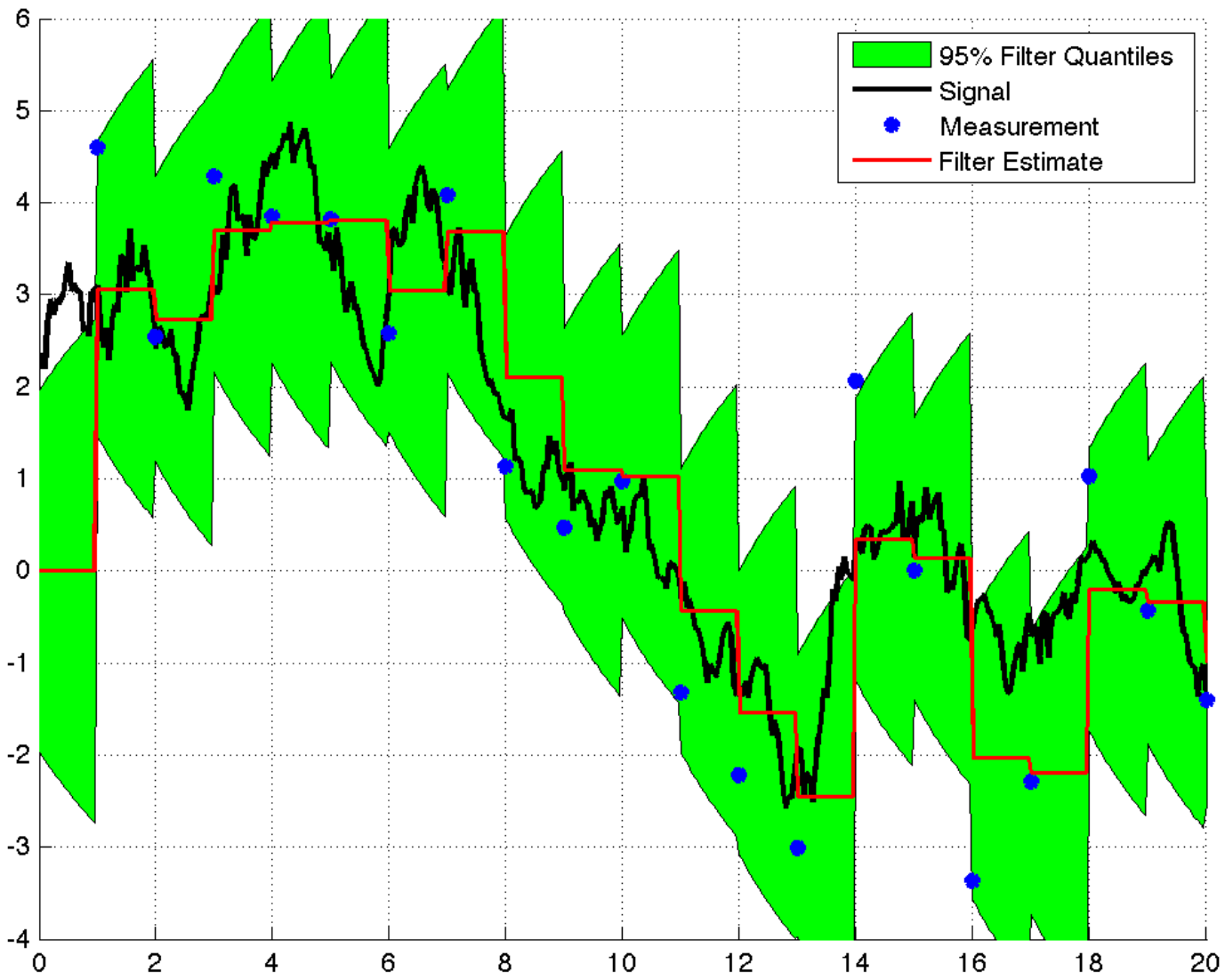
# Continuous-discrete (-time) Filter and Smoother Example



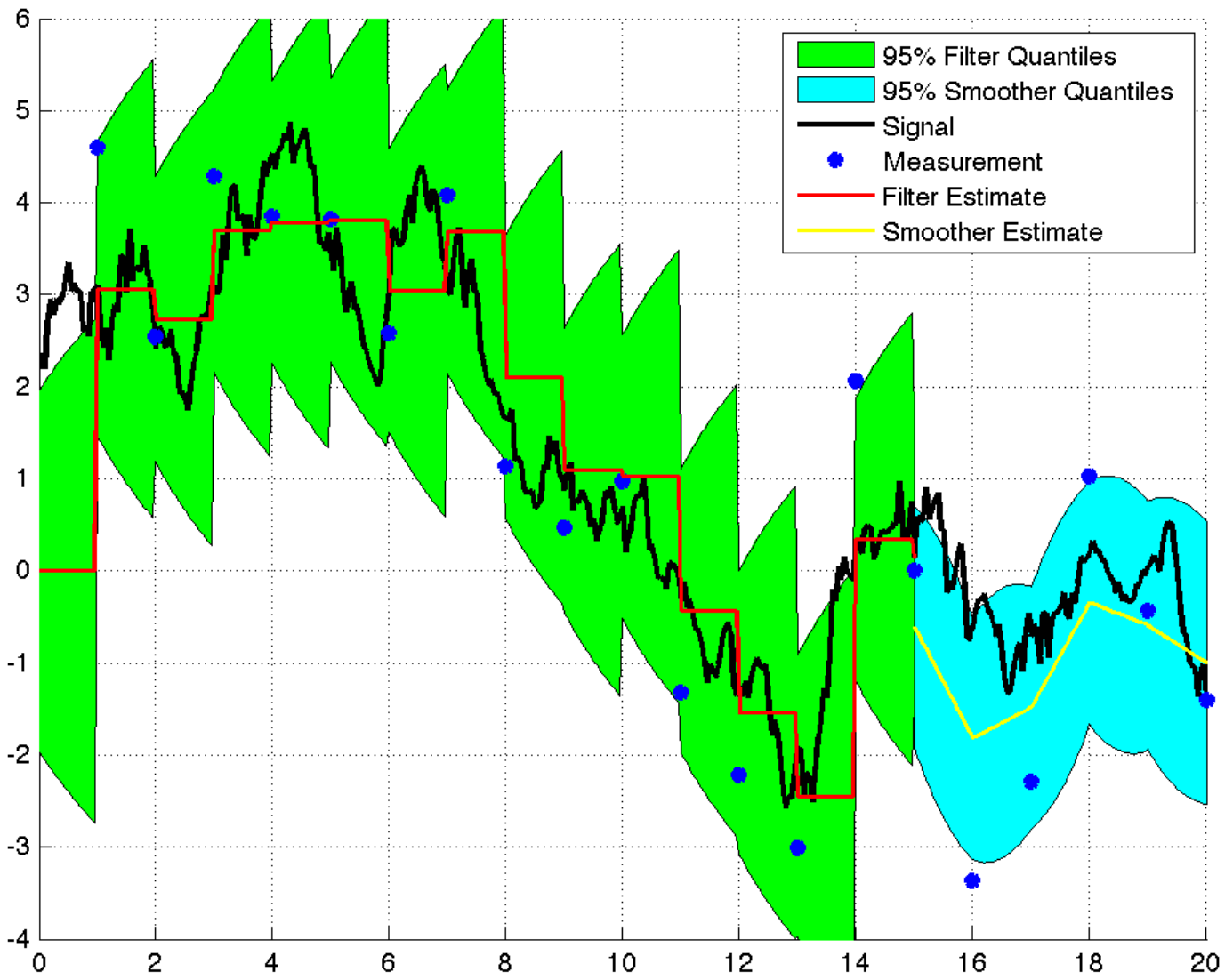
# Continuous-discrete (-time) Filter and Smoother Example



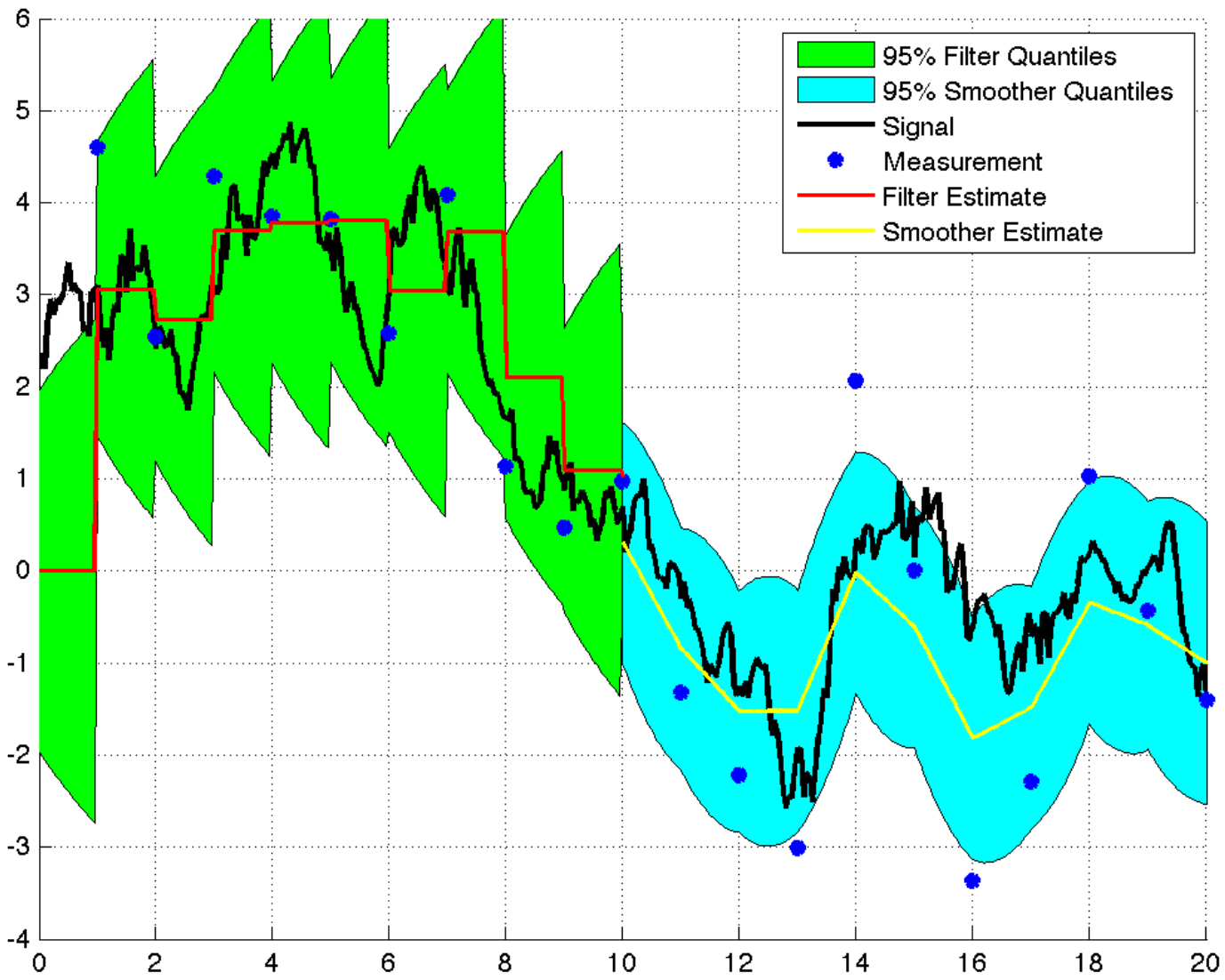
# Continuous-discrete (-time) Filter and Smoother Example



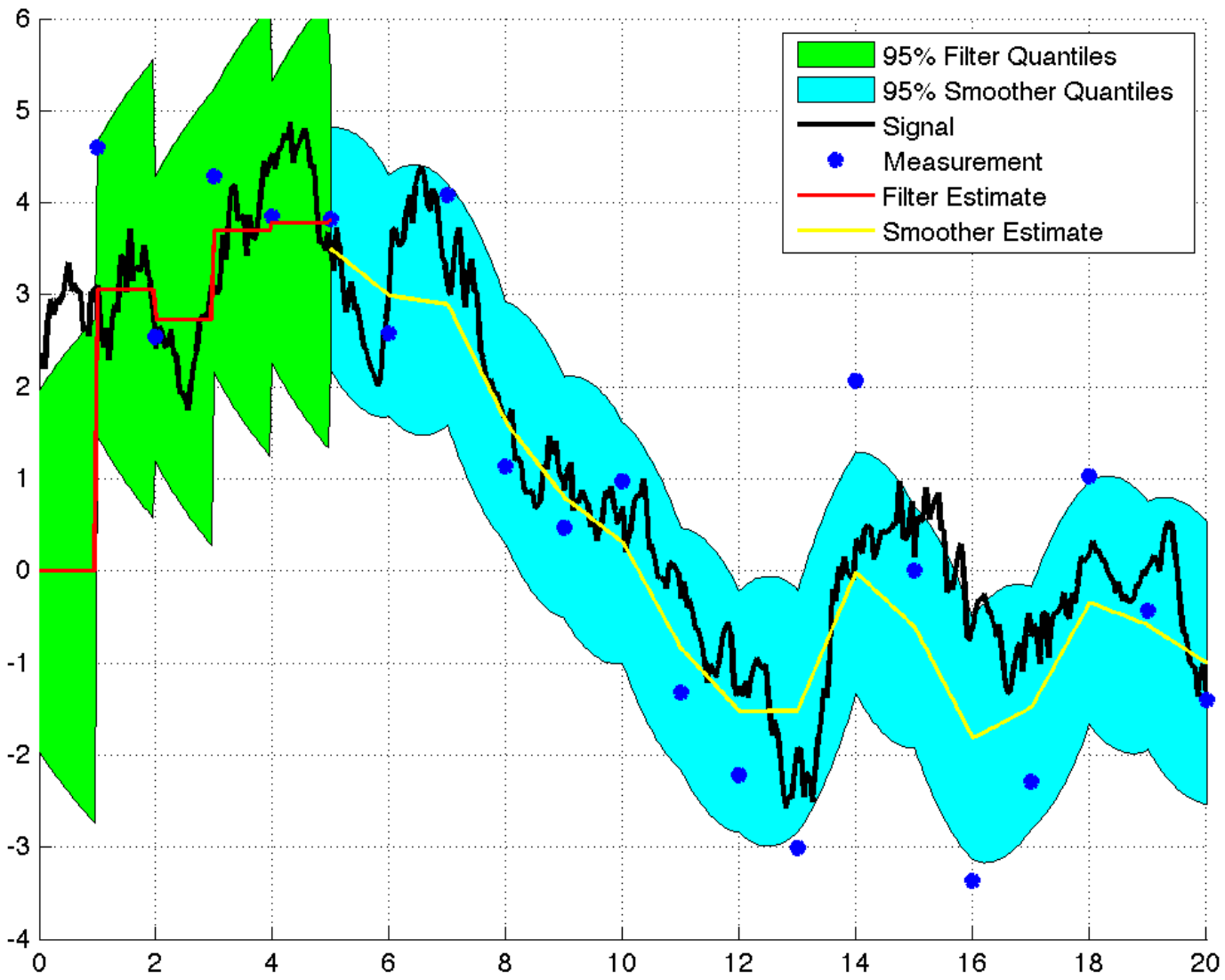
# Continuous-discrete (-time) Filter and Smoother Example



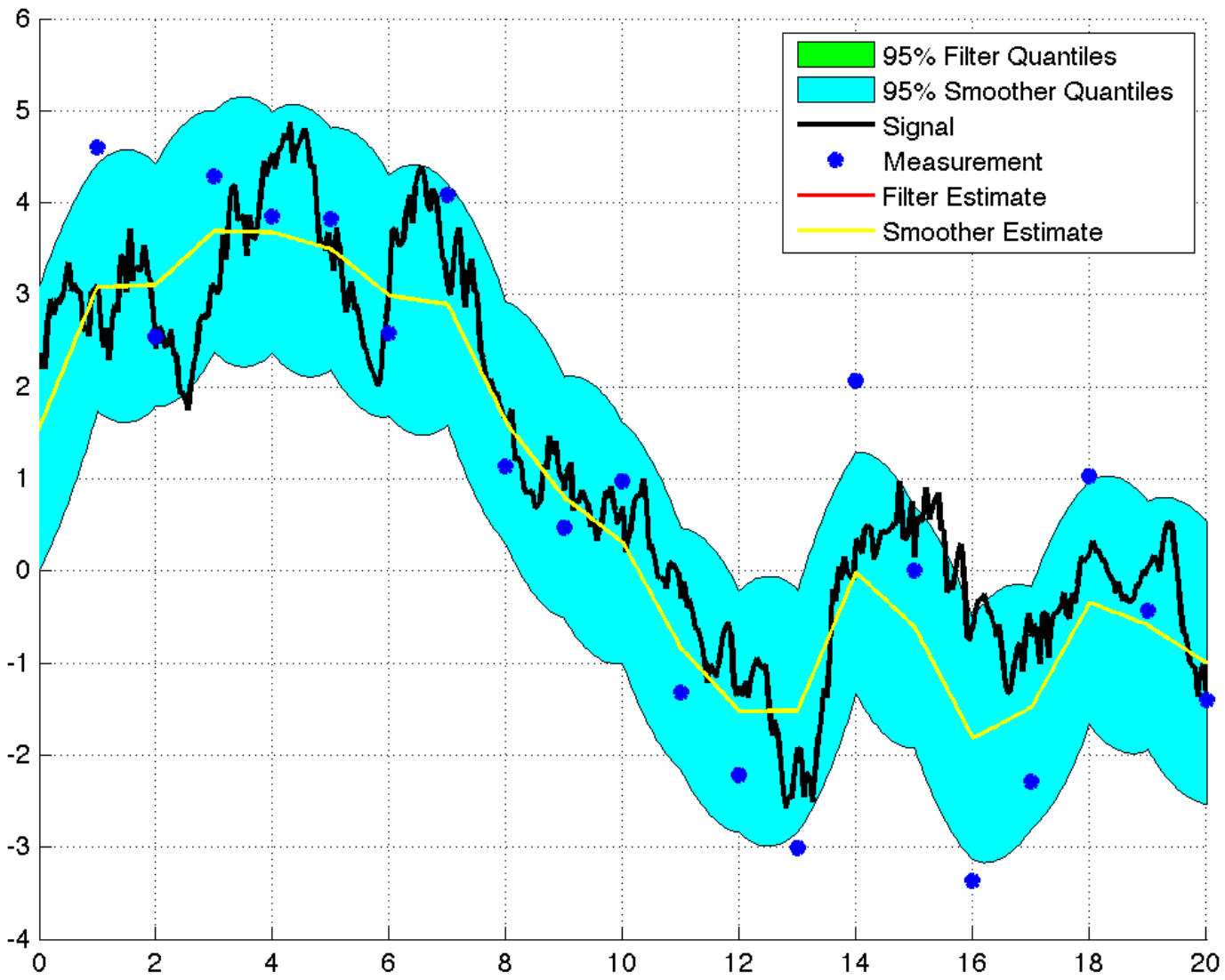
# Continuous-discrete (-time) Filter and Smoother Example



# Continuous-discrete (-time) Filter and Smoother Example



# Continuous-discrete (-time) Filter and Smoother Example



# Continuous-Discrete Baysian Filtering

- General continuous-discrete filtering model:

$$\frac{d\mathbf{x}}{dt} = \mathbf{a}(\mathbf{x}) + \mathbf{L}(\mathbf{x})\,\mathbf{w}(t)$$
$$\mathbf{y}_k \sim p(\mathbf{y}_k \,|\, \mathbf{x}(t_k)).$$

## Continuous-Discrete Bayesian Optimal filter

1. Prediction step: Solve the Fokker-Planck-Kolmogorov PDE

$$\frac{\partial p}{\partial t} = -\sum_i \frac{\partial}{\partial x_i}\left(a_i(\mathbf{x})\,p\right) + \frac{1}{2}\sum_{ij}\frac{\partial^2}{\partial x_i \partial x_j}\left([\mathbf{L}(\mathbf{x})\,\mathbf{Q}\,\mathbf{L}^\mathsf{T}(\mathbf{x})]_{ij}\,p\right)$$

2. Update step: Apply the Bayes' rule.

$$p(\mathbf{x}(t_k) \,|\, \mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k \,|\, \mathbf{x}(t_k))\,p(\mathbf{x}(t_k) \,|\, \mathbf{y}_{1:k-1})}{\int p(\mathbf{y}_k \,|\, \mathbf{x}(t_k))\,p(\mathbf{x}(t_k) \,|\, \mathbf{y}_{1:k-1})\,d\mathbf{x}(t_k)}$$

# Continuous-Discrete Baysian Filtering (cont.)

- Continuous-discrete non-linear Kalman filters are considered with models of the form:

$$\frac{d\mathbf{x}}{dt} = \mathbf{a}(\mathbf{x}) + \mathbf{L}(\mathbf{x})\,\mathbf{w}(t)$$
$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}(t_k)) + \mathbf{r}_k.$$

- Kalman filters form a Gaussian (process) approximation to the posterior of the process $\mathbf{x}(t)$.
- The resulting approximation is of the form

$$p(\mathbf{x}(t) \,|\, \mathbf{y}_{1:k}) \approx \mathrm{N}(\mathbf{x}(t) \,|\, \mathbf{m}(t), \mathbf{P}(t)), \quad t \in [t_k, t_{k+1}),$$

  where $\mathbf{m}(t)$ and $\mathbf{P}(t)$ are computed by the non-linear Kalman filter.
- Different brands: EKF, UKF, CKF, GHKF, etc.

# Continuous-Discrete Baysian Smoothing

- Continuous-discrete (-time) smoothing refers to recursive computation of the distributions

$$p(\mathbf{x}(t) \mid \mathbf{y}_{1:T}), \qquad t \in [t_0, t_T].$$

- Discrete-time smoothing: compute $p(\mathbf{x}(t_k) \mid \mathbf{y}_{1:T})$ for $k = 1, \ldots, T$.
- The (discrete-time) Bayesian smoothing equation is

$$p(\mathbf{x}(t_k) \mid \mathbf{y}_{1:T}) = p(\mathbf{x}(t_k) \mid \mathbf{y}_{1:k}) \times \int \frac{p(\mathbf{x}(t_{k+1}) \mid \mathbf{y}_{1:T})\, p(\mathbf{x}(t_{k+1}) \mid \mathbf{x}(t_k))}{p(\mathbf{x}(t_{k+1}) \mid \mathbf{y}_{1:k})}\, d\mathbf{x}(t_{k+1}).$$

- The continuous-time version of the above is a quite complicated partial differential equation (PDE).
- Continuous-discrete non-linear Gaussian smoother can be derived by computing the continuous-time limit of the Gaussian discrete-time smoother.

# Definition and Notation of Gaussian Processes in Regression

- **Spatial** Gaussian process (GP) or Gaussian field is a random function $\mathbf{f}(\mathbf{x})$, such that all finite-dimensional distributions $p(\mathbf{f}(\mathbf{x}_1), \ldots, \mathbf{f}(\mathbf{x}_n))$ are Gaussian.
- Note that $\mathbf{x}$ is the input – not the state! – BEWARE of the notation!
- Can be defined in terms of mean and covariance functions:

$$\begin{aligned} \mathbf{m}(\mathbf{x}) &= \mathrm{E}[\mathbf{f}(\mathbf{x})] \\ \mathbf{K}(\mathbf{x}, \mathbf{x}') &= \mathrm{E}[(\mathbf{f}(\mathbf{x}) - \mathbf{m}(\mathbf{x}))\,(\mathbf{f}(\mathbf{x}') - \mathbf{m}(\mathbf{x}'))^{\mathsf{T}}]. \end{aligned}$$

- The joint distribution of an arbitrary collection of random variables $\mathbf{f}(\mathbf{x}_1), \ldots, \mathbf{f}(\mathbf{x}_n)$ is then given as

$$\begin{pmatrix} \mathbf{f}(\mathbf{x}_1) \\ \vdots \\ \mathbf{f}(\mathbf{x}_n) \end{pmatrix} \sim \mathrm{N}\left( \begin{pmatrix} \mathbf{m}(\mathbf{x}_1) \\ \vdots \\ \mathbf{m}(\mathbf{x}_n) \end{pmatrix}, \begin{pmatrix} \mathbf{K}(\mathbf{x}_1, \mathbf{x}_1) & \ldots & \mathbf{K}(\mathbf{x}_1, \mathbf{x}_n) \\ \vdots & \ddots & \\ \mathbf{K}(\mathbf{x}_n, \mathbf{x}_1) & & \mathbf{K}(\mathbf{x}_n, \mathbf{x}_n) \end{pmatrix} \right)$$

# Definition and Notation of Gaussian Processes in Regression (cont.)

- **Gaussian process regression** (Rasmussen & Williams, 2006):
  - GPs are used as non-parametric prior models for "learning" input-output $\mathbb{R}^d \mapsto \mathbb{R}^m$ mappings in form $\mathbf{y} = \mathbf{f}(\mathbf{x})$.
  - A set of noisy training samples $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n)\}$ given.
  - The values of function $\mathbf{f}(\mathbf{x})$ at measurement points and test points are of interest.
- **Temporal** Gaussian process (GP) is a temporal random function $\mathbf{f}(t)$, such that joint distribution of $\mathbf{f}(t_1), \ldots, \mathbf{f}(t_n)$ is always Gaussian.
- In this case the input is the time $t$ and thus our regressor functions have the form $\mathbf{y} = \mathbf{f}(t)$.
- Mean and covariance functions have the form:

$$\begin{aligned} \mathbf{m}(t) &= \mathrm{E}[\mathbf{f}(t)] \\ \mathbf{K}(t, t') &= \mathrm{E}[(\mathbf{f}(t) - \mathbf{m}(t))\,(\mathbf{f}(t') - \mathbf{m}(t'))^{\mathsf{T}}]. \end{aligned}$$

# Gaussian Process Regression as State Estimation

- Consider a GP regression problem with measurements $y_1, \ldots, y_T$:

$$y_k = f(t_k) + e_k, \qquad e_k \sim \mathrm{N}(0, \sigma^2).$$

- The covariance function of $f(t)$ can be, e.g.,

$$K(t, t') = s^2 \exp\left(-\frac{1}{2\ell^2}||t - t'||^2\right),$$

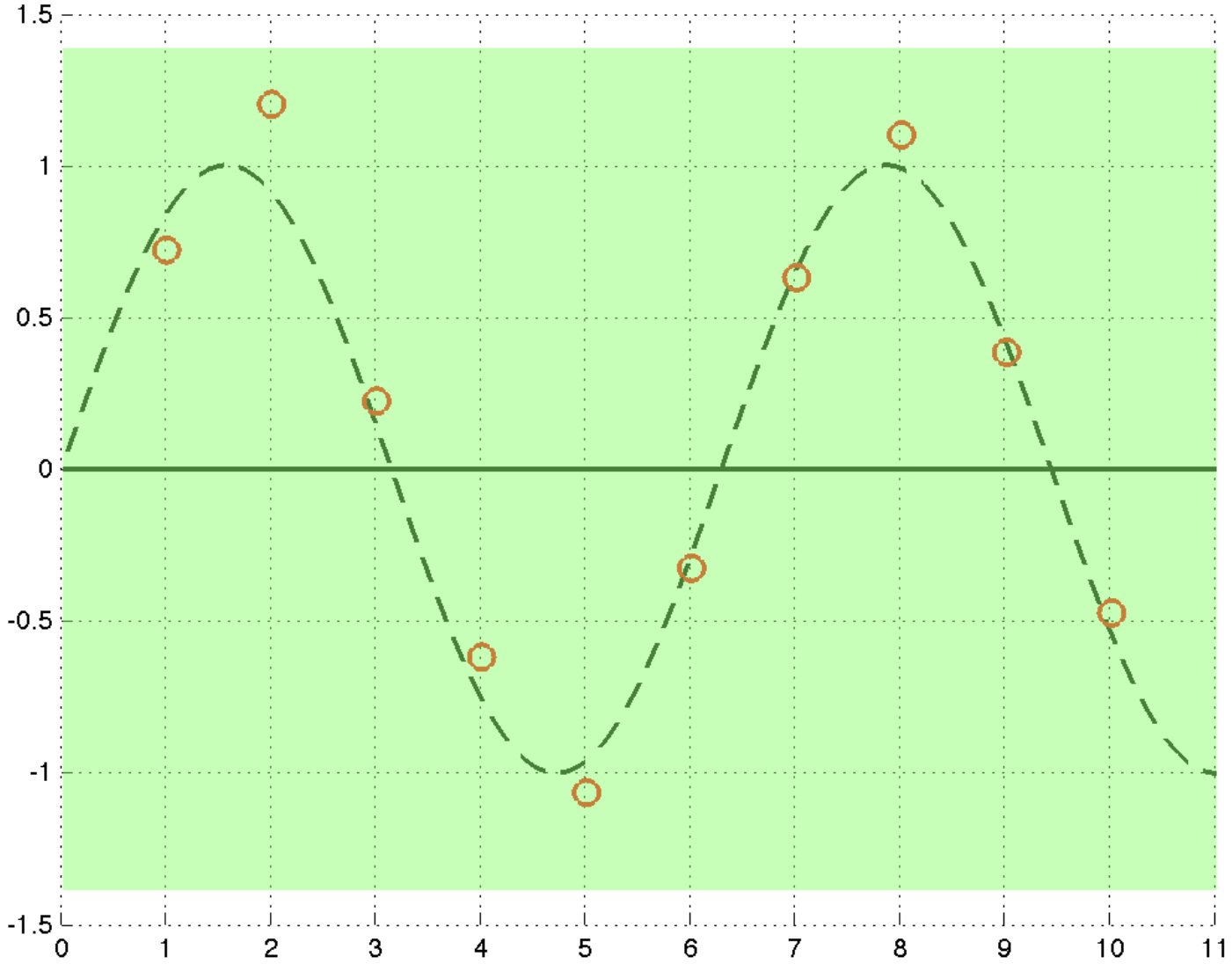
- We now assert that it is possible to form a state space model

$$\begin{aligned} \frac{d\mathbf{f}(t)}{dt} &= \mathbf{A}\,\mathbf{f}(t) + \mathbf{L}\,w(t) \\ y_k &= \mathbf{H}\,\mathbf{f}(t_k) + e_k. \end{aligned}$$

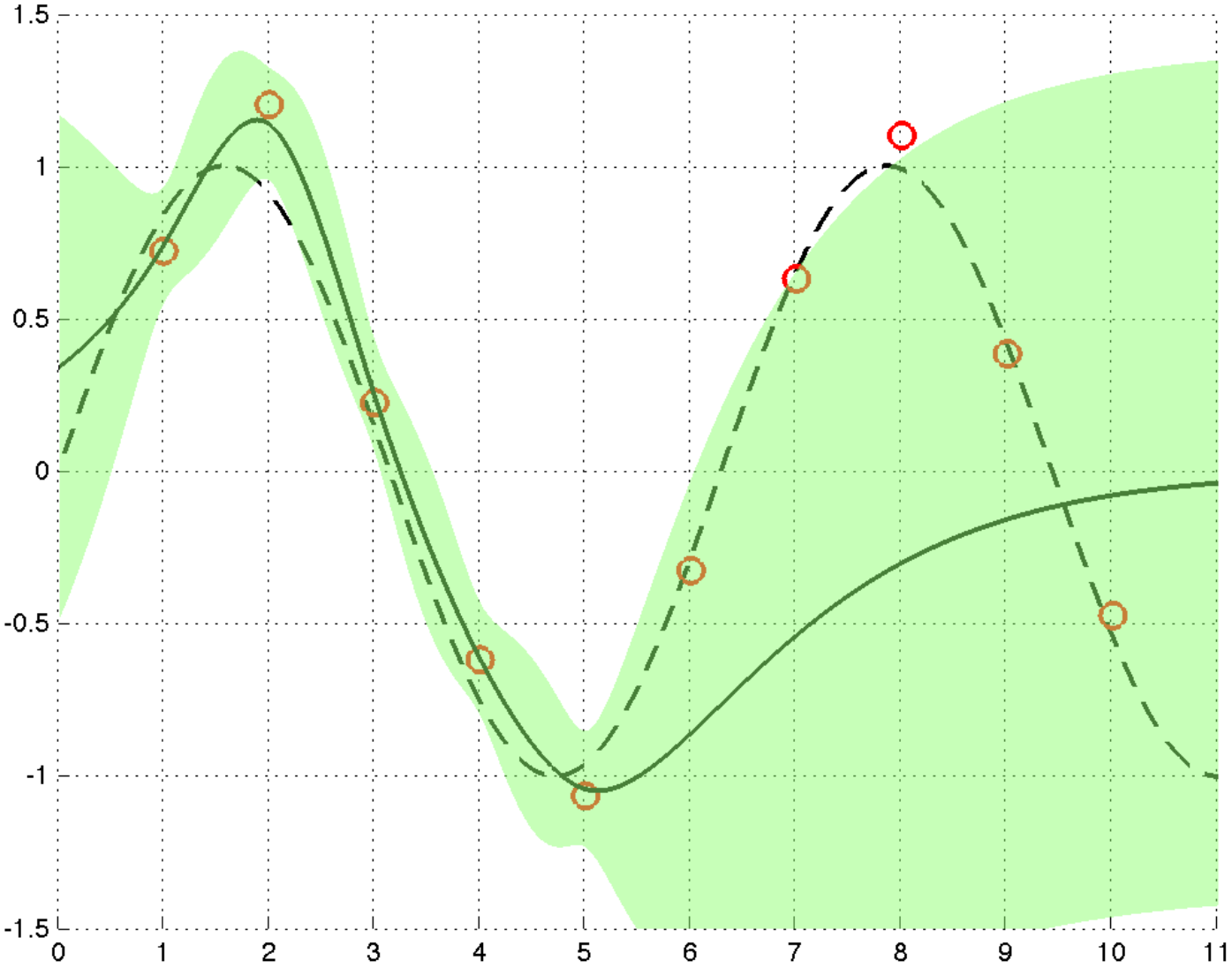
  such that a linear Kalman filter/smoother computes the GP-regression solution.

- The solution has complexity $O(T)$ when GP-regression has $O(T^3)$.

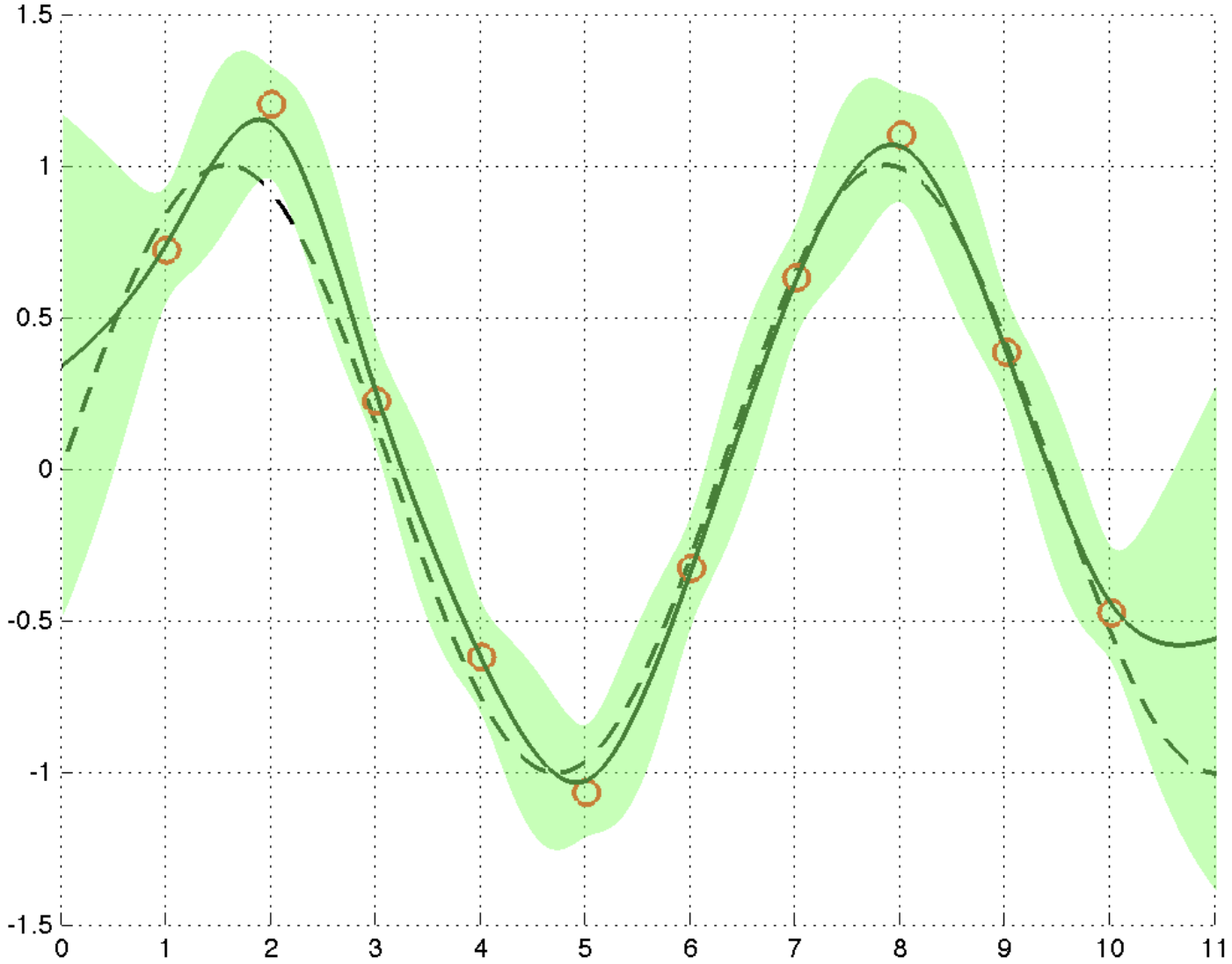
# Gaussian Process Regression Example



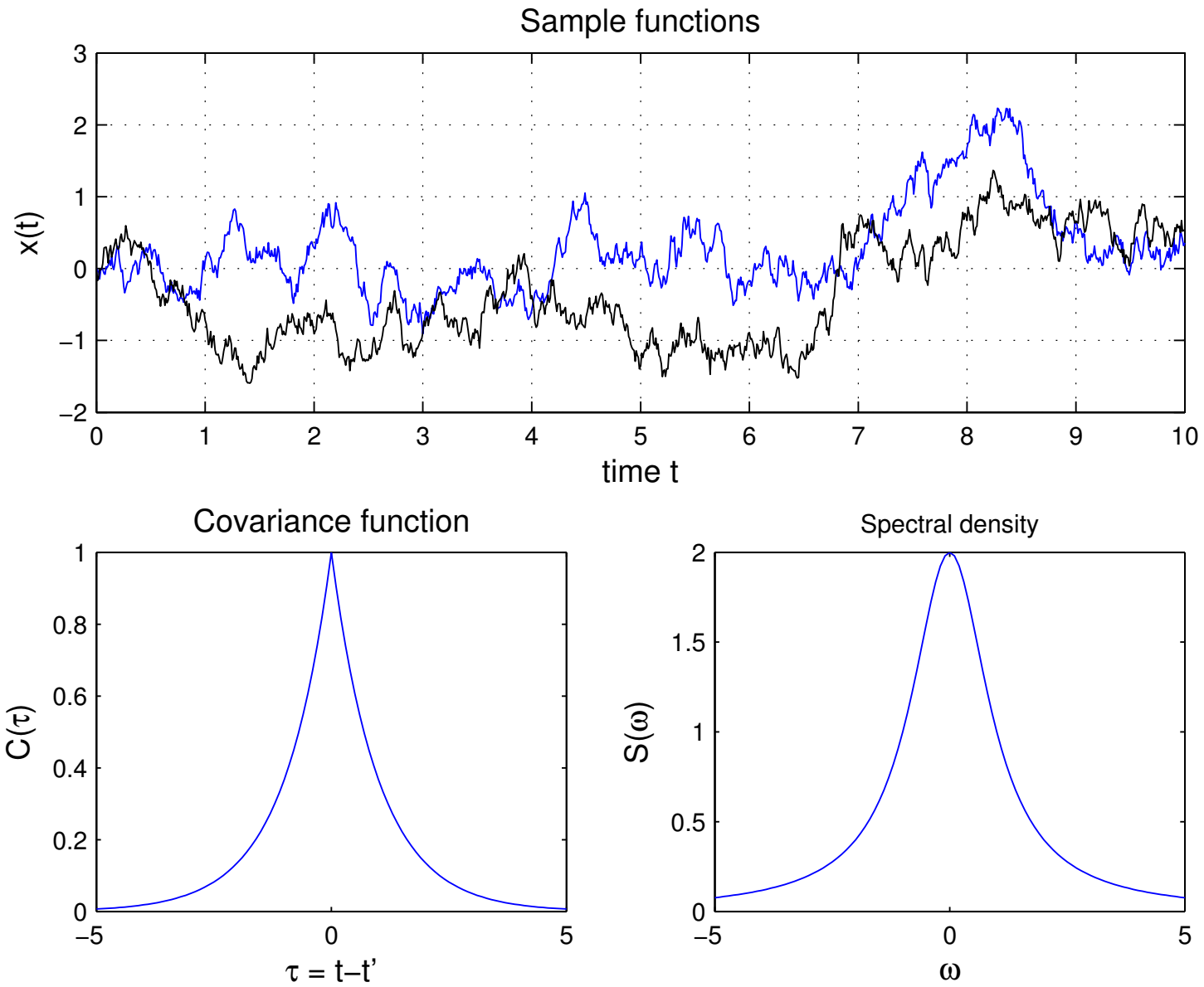
# Gaussian Process Regression Example



# Gaussian Process Regression Example



# Representations of Temporal Gaussian Processes



# Representations of Temporal Gaussian Processes

- Example: Ornstein-Uhlenbeck process – path representation as a stochastic differential equation (SDE):

$$\frac{df(t)}{dt} = -\lambda\, f(t) + w(t).$$

  where $w(t)$ is a white noise process.
- The mean and covariance functions:

$$m(t) = 0$$
$$k(t, t') = \exp(-\lambda |t - t'|)$$

- Spectral density:

$$S(\omega) = \frac{2\lambda}{\omega^2 + \lambda^2}$$

- Ornstein-Uhlenbeck process $f(t)$ is Markovian in the sense that given $f(t)$ the past $\{f(s), s < t\}$ does not affect the distribution of the future $\{f(s'), s' > t\}$.

# State Space Form of Linear Time-Invariant SDEs

- Consider a $N$th order LTI SDE of the form

$$\frac{d^N f}{dt^N} + a_{N-1}\frac{d^{N-1} f}{dt^{N-1}} + \cdots + a_0 f = w(t).$$

- If we define $\mathbf{f} = (f, \ldots, d^{N-1}f/dt^{N-1})$, we get a state space model:

$$\frac{d\mathbf{f}}{dt} = \underbrace{\begin{pmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & 0 & 1 \\ -a_0 & -a_1 & \ldots & -a_{N-1} \end{pmatrix}}_{\mathbf{A}} \mathbf{f} + \underbrace{\begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}}_{\mathbf{L}} w(t)$$

$$f(t) = \underbrace{\begin{pmatrix} 1 & 0 & \cdots & 0 \end{pmatrix}}_{\mathbf{H}} \mathbf{f}.$$

- The vector process $\mathbf{f}(t)$ is now time-Markovian although $f(t)$ is not.

# Spectra of Linear Time-Invariant SDEs

- By taking the Fourier transform of the LTI SDE, we can derive the spectral density which has the form:

$$S(\omega) = \frac{(\text{constant})}{(\text{polynomial in } \omega^2)}$$

- It turns out that we can also do this conversion to the other direction:
  - With certain parameter values, the Matérn has the form:

$$S(\omega) \propto (\lambda^2 + \omega^2)^{-(p+1)}.$$

  - Many non-rational spectral densities can be approximated, e.g.:

$$S(\omega) = \sigma^2 \sqrt{\frac{\pi}{\kappa}} \exp\left(-\frac{\omega^2}{4\kappa}\right) \approx \frac{(\text{const})}{N!/0!(4\kappa)^N + \cdots + \omega^{2N}}$$

- For the conversion of a rational spectral density to a Markovian (state-space) model, we can use the classical spectral factorization –

# Converting Covariance Functions to State Space Models

- Spectral factorization finds rational stable transfer function

$$G(i\,\omega) = \frac{b_m\,(i\,\omega)^m + \cdots + b_1\,(i\,\omega) + b_0}{a_n\,(i\,\omega)^n + \cdots + a_1\,(i\,\omega) + a_0}$$

  such that

$$S(\omega) = G(i\,\omega)\,q_c\,G(-i\,\omega).$$

- The procedure practice:
  - Compute the roots of the numerator and denominator polynomials.
  - Construct the numerator and denominator polynomials of the transfer function $G(i\,\omega)$ from the positive-imaginary-part roots only.
- The SDE is then the inverse Fourier transform of

$$F(i\,\omega) = G(i\,\omega)\,W(i\,\omega).$$

- Can be further converted into a state space model whose (vector-valued) state is Markovian.

# GP to State Spate Conversion in Practice

## Converting a covariance function into state space model

1. Compute the corresponding spectral density $S(\omega)$ by computing the Fourier transform of $K(\tau)$.
2. If $S(\omega)$ is not a rational function, approximate it with such a function, e.g., via Taylor series expansions or Padé approximants.
3. Find a *stable* rational transfer function $G(i\,\omega)$ and constant $q_c$ such that
$$S(\omega) = G(i\,\omega)\,q_c\,G(-i\,\omega).$$
via the spectral factorization.
4. Use the methods from control theory to convert the transfer function model into an equivalent state space model.

# Application to Gaussian Process Regression

- Consider a Gaussian process regression problem of the form

$$
\begin{aligned}
f(x) &\sim \mathcal{GP}(0, k(x, x')) \\
y_k &= f(x_k) + e_k, \qquad e_k \sim \mathrm{N}(0, \sigma^2_{\text{noise}}).
\end{aligned}
$$

- Renaming $x$ into time $t$ gives:

$$
\begin{aligned}
f(t) &\sim \mathcal{GP}(0, k(t, t')) \\
y_k &= f(t_k) + e_k, \qquad e_k \sim \mathrm{N}(0, \sigma^2_{\text{noise}}).
\end{aligned}
$$

- We can use the method to convert this to state estimation problem:

$$
\begin{aligned}
\frac{d\mathbf{f}(t)}{dt} &= \mathbf{A}\,\mathbf{f}(t) + \mathbf{L}\,w(t) \\
y_k &= \mathbf{H}\,\mathbf{f}(t_k) + e_k.
\end{aligned}
$$

# Kalman Filter and RTS Smoother Solution

- Kalman filter and RTS smoother can be used for efficiently computing posteriors of models of the state space form

$$\frac{d\mathbf{f}(t)}{dt} = \mathbf{A}\,\mathbf{f}(t) + \mathbf{L}\,\mathbf{w}(t)$$
$$\mathbf{y}_k = \mathbf{H}\,\mathbf{f}(t_k) + \mathbf{e}_k,$$

  where $\mathbf{y}_k$ is the measurement and $\mathbf{e}_k \sim \mathrm{N}(0, \mathbf{R}_k)$.
- With $n$ measurements, complexity of KF/RTS is $O(n)$, when the brute-force GPR solution is $O(n^3)$.
- $\Rightarrow$ Many Gaussian process regression (or e.g. Kriging/inverse) problems can be more efficiently solved with KF & RTS

# Example: Matérn Covariance Function

## Example (1D Matérn covariance function)

- 1D Matérn family is ($\tau = |t - t'|$):

$$k(\tau) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu}\,\frac{\tau}{l}\right)^{\nu} K_\nu\left(\sqrt{2\nu}\,\frac{\tau}{l}\right),$$

  where $\nu, \sigma, l > 0$ are the smoothness, magnitude and length scale parameters, and $K_\nu(\cdot)$ the modified Bessel function.

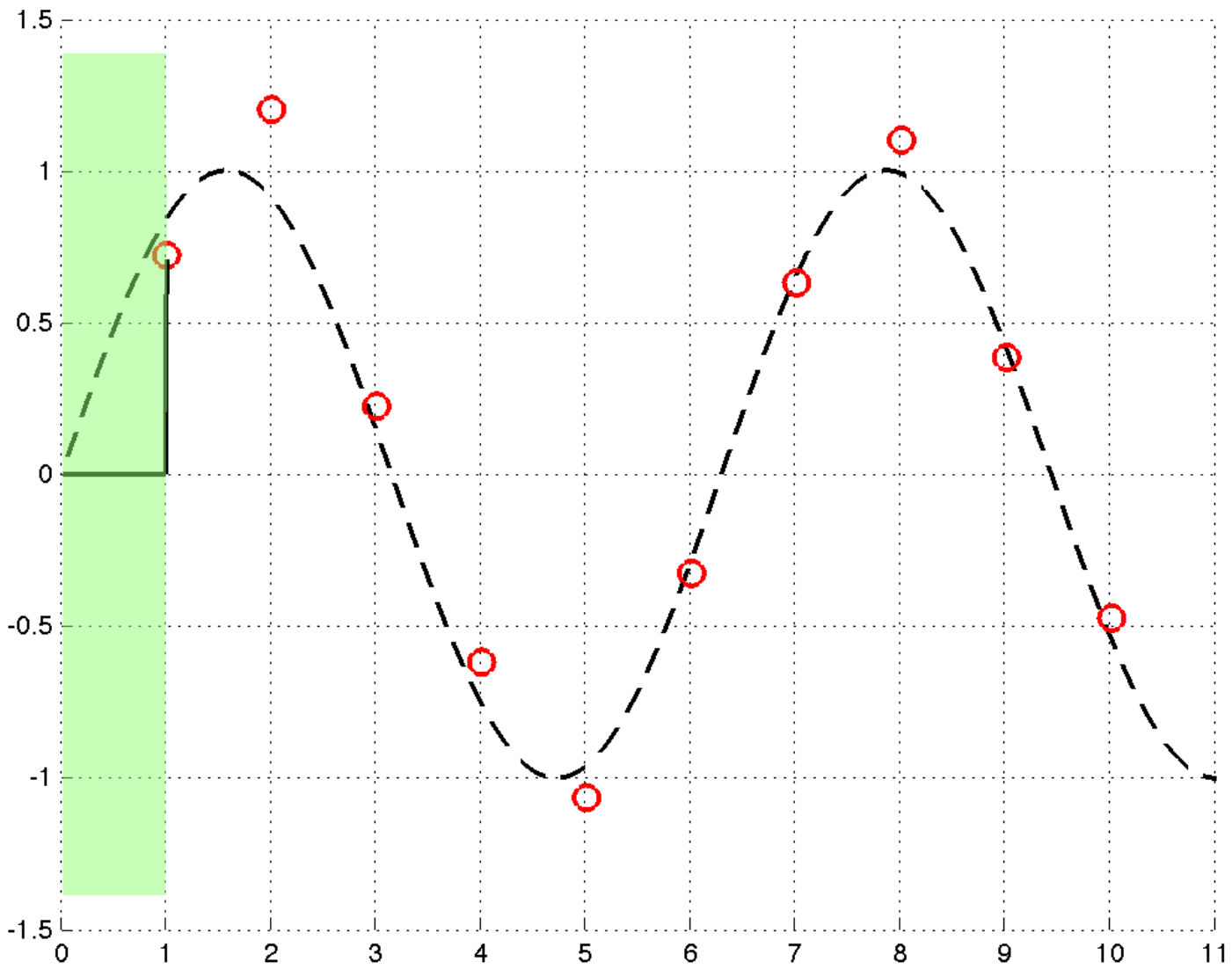
- For example, when $\nu = 5/2$, we get

$$\frac{\mathrm{d}\mathbf{f}(t)}{dt} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -\lambda^3 & -3\lambda^2 & -3\lambda \end{pmatrix} \mathbf{f}(t) + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} w(t).$$

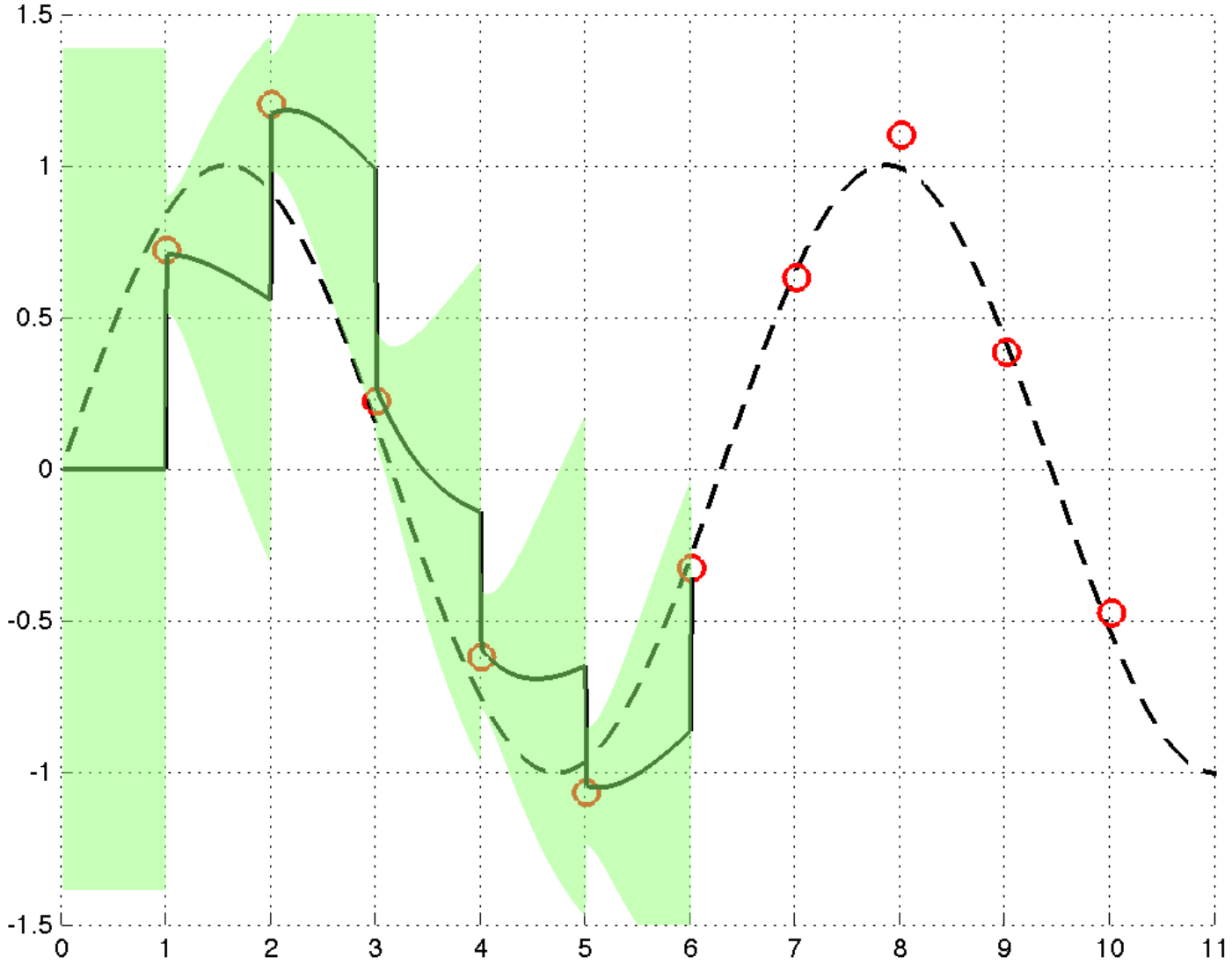
# Inference in Practice

- Conventional GP regression:
  1. Evaluate the covariance function at the training and test set points.
  2. Use GP regression formulas to compute the posterior process statistics.
  3. Use the mean function as the prediction.
- State-space GP regression:
  1. Form the state space model.
  2. Run Kalman filter through the measurement sequence.
  3. Run RTS smoother through the filter results.
  4. Use the smoother mean function as the prediction.
- With both GP regression and state-space formulation we have the corresponding parameter estimation methods – see, e.g., Rasmussen & Williams (2006) and Särkkä (2013), respectively.

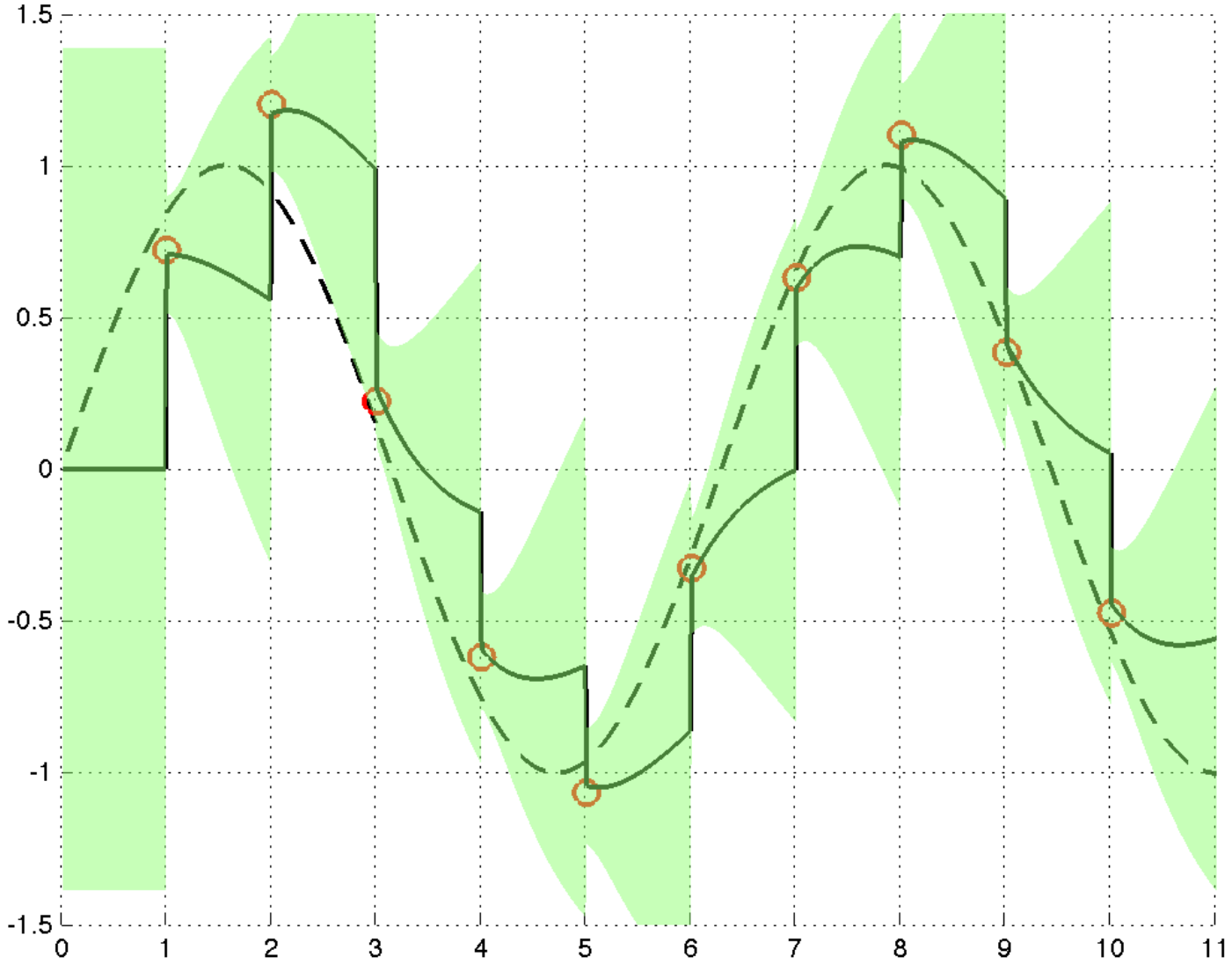
# State-Space GP Regression Demo



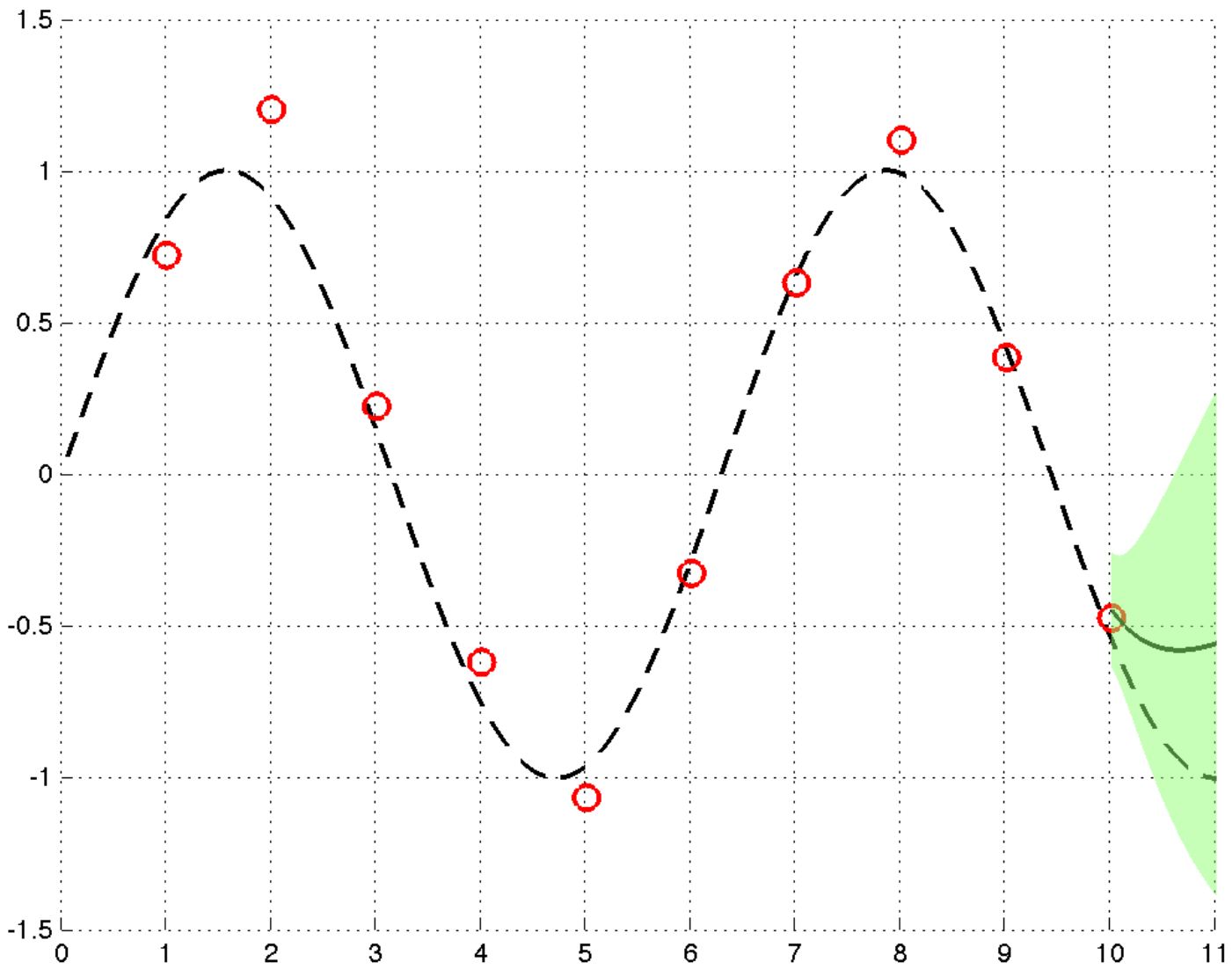
# State-Space GP Regression Demo



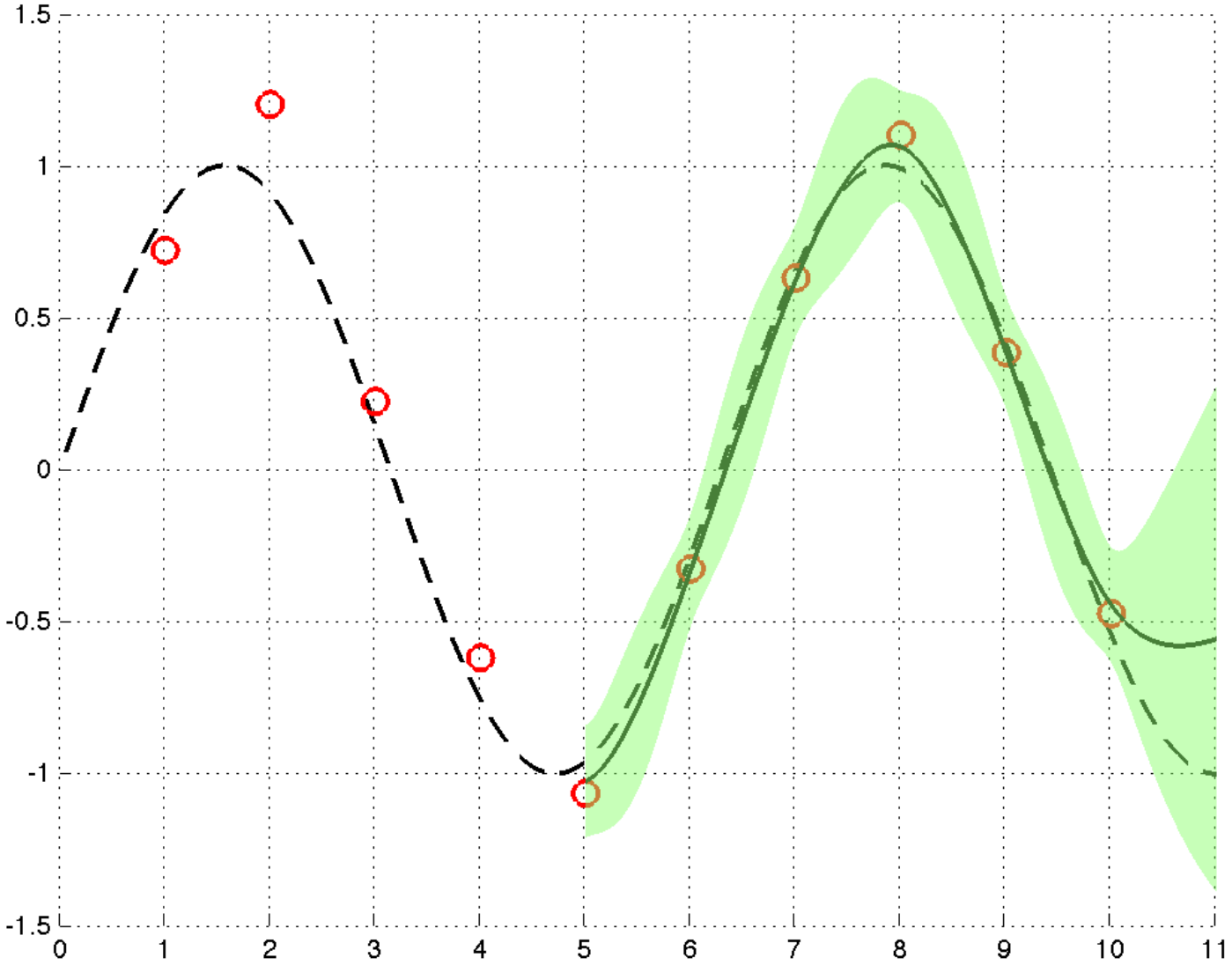
# State-Space GP Regression Demo



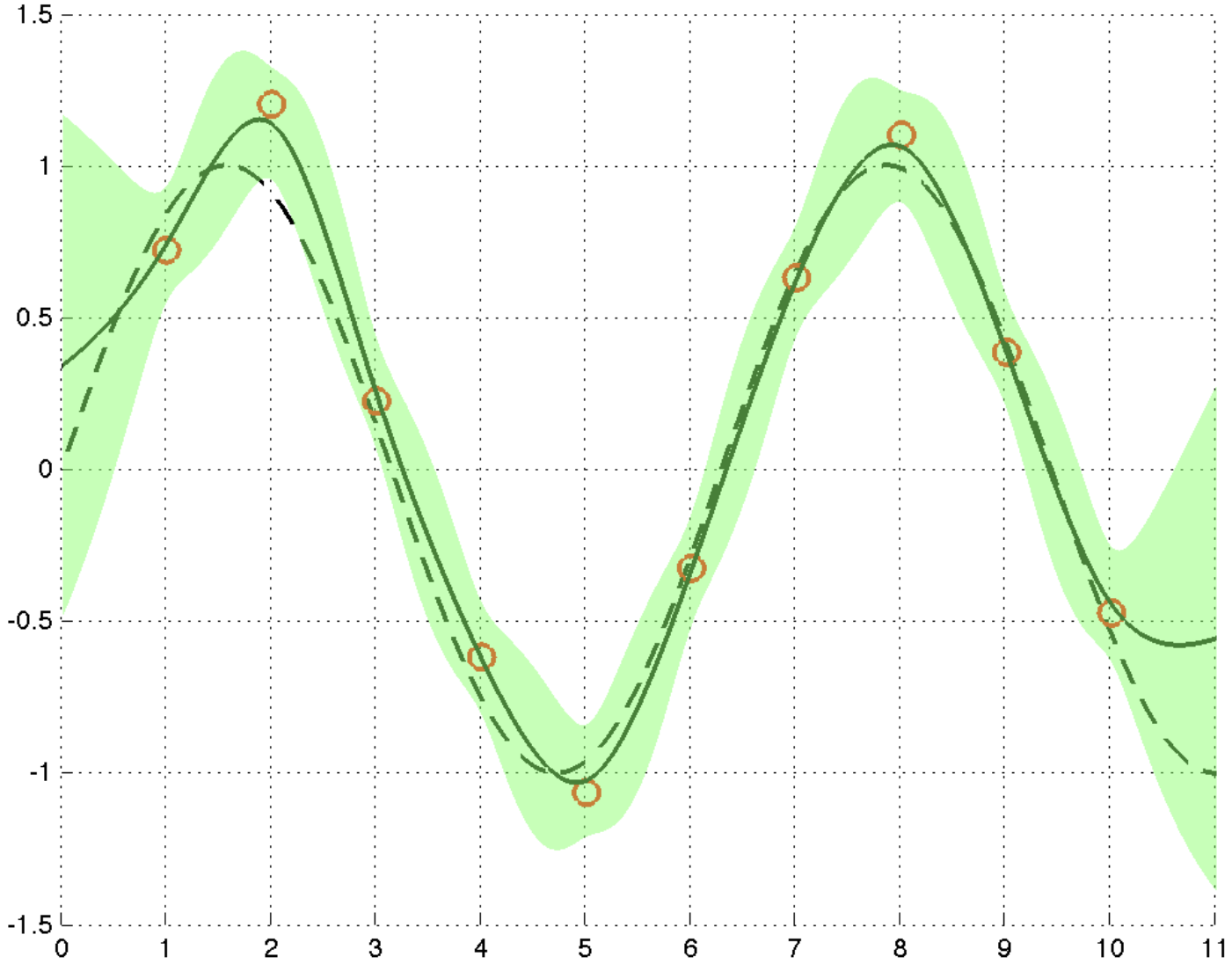
# State-Space GP Regression Demo



# State-Space GP Regression Demo

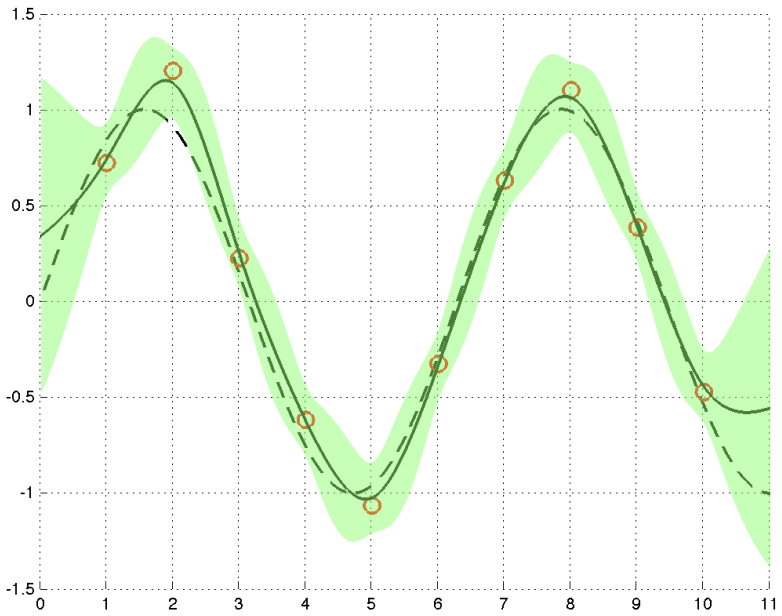


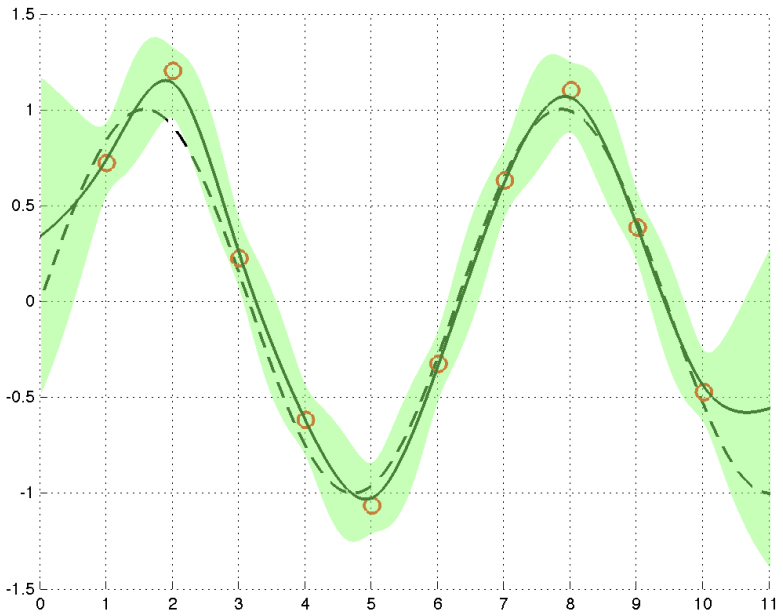
# State-Space GP Regression Demo



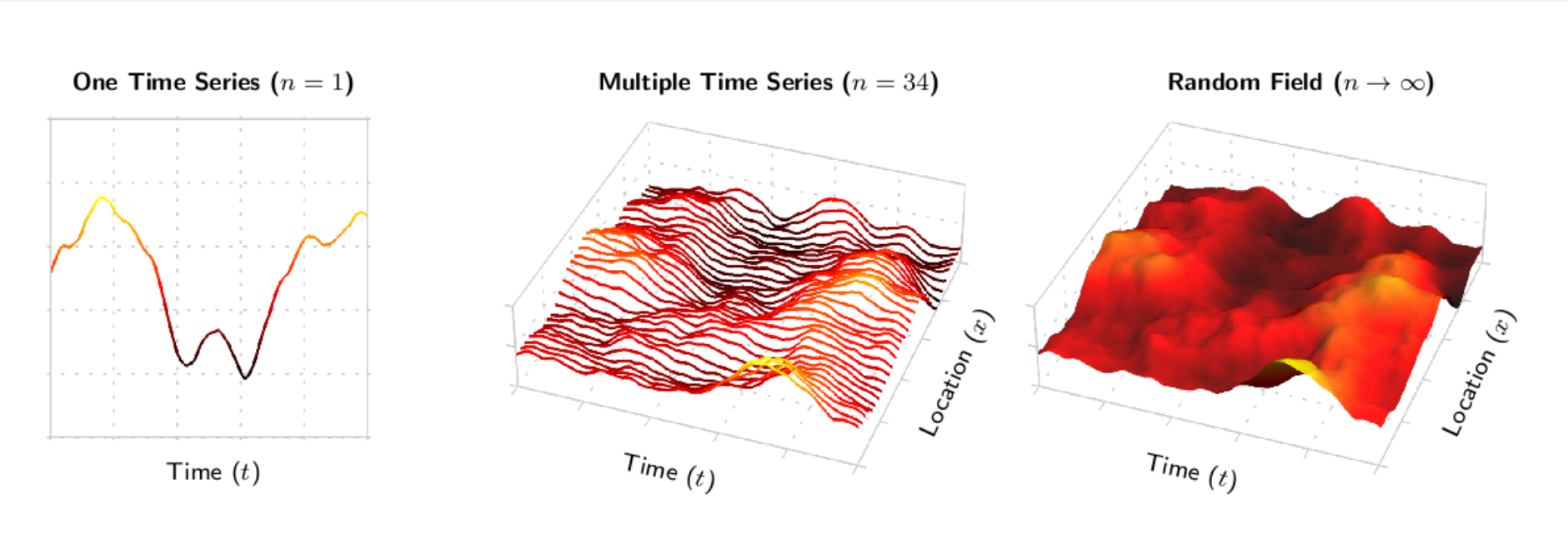
# State-Space GP Regression Demo (cont.)

## Comparison of GP regression (L) and RTS smoother (R) results



# From Temporal to Spatio-Temporal Processes




The temporal vector-valued process becomes an infinite-dimensional function (Hilbert space) -valued process:

$$\mathbf{f}(t) = \begin{pmatrix} f_1(t) \\ \vdots \\ f_n(t) \end{pmatrix} \to \begin{pmatrix} f(\mathbf{x}_1, t) \\ \vdots \\ f(\mathbf{x}_n, t) \end{pmatrix} \to f(\mathbf{x}, t), \quad \mathbf{x} \in \mathbb{R}^d.$$

# State Space Inference in Spatio-Temporal Processes

- Spatio-temporal GP-regression problem:

$$\mathbf{f}(\mathbf{x}, t) \sim \mathcal{GP}(0, \mathbf{K}(\mathbf{x}, \mathbf{x}'; t, t'))$$
$$\mathbf{y}_k = \mathbf{f}(\mathbf{x}_k, t_k) + \mathbf{e}_k$$

- We can convert this into infinite-dimensional state-space model with operators $\mathcal{A}$ and $\mathcal{H}_k$:

$$\frac{\partial \mathbf{f}(\mathbf{x}, t)}{\partial t} = \mathcal{A}\,\mathbf{f}(\mathbf{x}, t) + \mathbf{L}\,\mathbf{w}(\mathbf{x}, t)$$
$$\mathbf{y}_k = \mathcal{H}_k\,\mathbf{f}(\mathbf{x}, t_k) + \mathbf{e}_k$$

- We can use the infinite-dimensional Kalman filter and RTS smoother – scale linearly in time dimension.
- We can approximate with PDE methods such as basis function expansions, FEM, finite-differences, spectral methods, etc.
- Above the infinite-dimensional process $t \mapsto \mathbf{f}(\cdot, t)$ is Markovian.

# Example: 2D Matérn Covariance Function

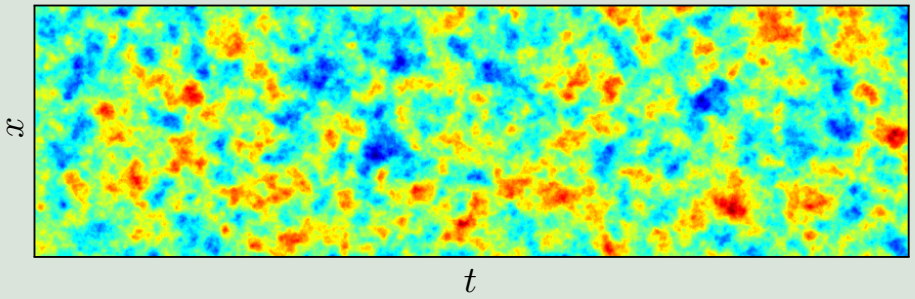
## Example (2D Matérn covariance function)

- The multidimensional Matérn covariance function is the following ($r = ||\boldsymbol{\xi} - \boldsymbol{\xi}'||$, for $\boldsymbol{\xi} = (x_1, x_2, \ldots, x_{d-1}, t) \in \mathbb{R}^d$):

$$k(r) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu}\,\frac{r}{l}\right)^{\nu} K_\nu \left(\sqrt{2\nu}\,\frac{r}{l}\right).$$

- For example, if $\nu = 1$ and $d = 2$, we get the following:

$$\frac{\partial \mathbf{f}(x,t)}{\partial t} = \begin{pmatrix} 0 & 1 \\ \partial^2/\partial x^2 - \lambda^2 & -2\sqrt{\lambda^2 - \partial^2/\partial x^2} \end{pmatrix} \mathbf{f}(x,t) + \begin{pmatrix} 0 \\ 1 \end{pmatrix} w(x,t).$$



# State Space Inference in Latent Force Models

- A latent force model is of the form

$$\frac{dx_f(t)}{dt} = g(x_f(t)) + u(t),$$

  where $u(t)$ is the latent force.
- We measure the system at discrete instants of time:

$$y_k = x_f(t_k) + r_k$$

- Let's now model $u(t)$ as a Gaussian process of Matern type

$$K(\tau) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu}\,\frac{\tau}{l}\right)^{\nu} K_\nu \left(\sqrt{2\nu}\,\frac{\tau}{l}\right)$$

- Recall that if, for example, $\nu = 1/2$ then the GP can be expressed as the solution of the stochastic differential equation (SDE)

$$\frac{\mathrm{d}u(t)}{\mathrm{d}t} = -\lambda\, u(t) + w(t)$$

# State Space Inference in Latent Force Models (cont.)

- If we define $\mathbf{x} = (x_f, u)$, we get a two-dimensional SDE

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \underbrace{\begin{pmatrix} g(x_1(t)) + x_2(t) \\ -\lambda\, x_2(t) \end{pmatrix}}_{\mathbf{a}(\mathbf{x})} + \underbrace{\begin{pmatrix} 0 \\ 1 \end{pmatrix}}_{\mathbf{L}} w(t)$$

- We can now rewrite the measurement model as

$$y_k = \underbrace{\begin{pmatrix} 1 & 0 \end{pmatrix}}_{\mathbf{H}} \mathbf{x}(t_k) + r_k$$

- Thus the result is a model of the generic form

$$\begin{aligned} \frac{d\mathbf{x}}{dt} &= \mathbf{a}(\mathbf{x}) + \mathbf{L}\,\mathbf{w}(t) \\ \mathbf{y}_k &= \mathbf{H}\,\mathbf{x}(t_k) + \mathbf{r}_k. \end{aligned}$$

- This model can now be efficiently tackled with non-linear Kalman filtering and smoothing.

# Other Extensions

- Change-detection via combining state-space GPs with switching models and methods such as IMM or EC.
- Distributed latent force models, i.e., combined partial differential equation (PDE) and GP models.
- Outlier-robust GPs via Student-t measurement model or via classical clutter rejection methods.
- Online parameter estimation via sequential Monte Carlo, state-augmentation, online-EM, variational Bayes, and other methods.
- Classification corresponds to replacing the linear-Gaussian measurement model with a non-Gaussian one.
- Non-linear spatio-temporal (PDE) models via using non-linear infinite-dimensional Kalman filter and smoothers.

# GPS Satellite Orbit Prediction

- Accurate orbit prediction improves Time To First Fix (TTFF) when network is not available for A-GPS.
- The equation of motion for the satellite can be written as

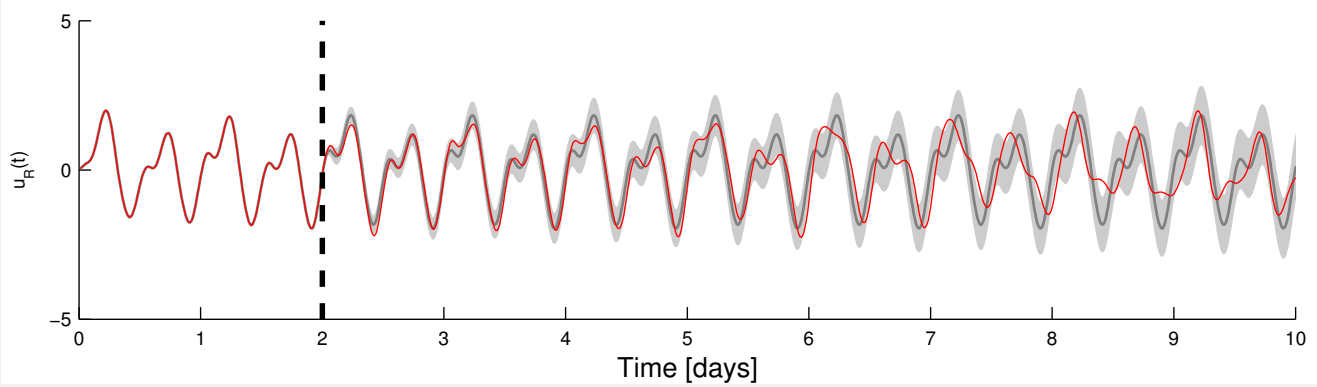
$$\frac{d}{dt}\begin{bmatrix}\mathbf{r}\\ \mathbf{v}\end{bmatrix} = \begin{bmatrix}\mathbf{v}\\ \mathbf{a}(\mathbf{r},t)+\mathbf{u}(\mathbf{r},\mathbf{v},t)\end{bmatrix}.$$

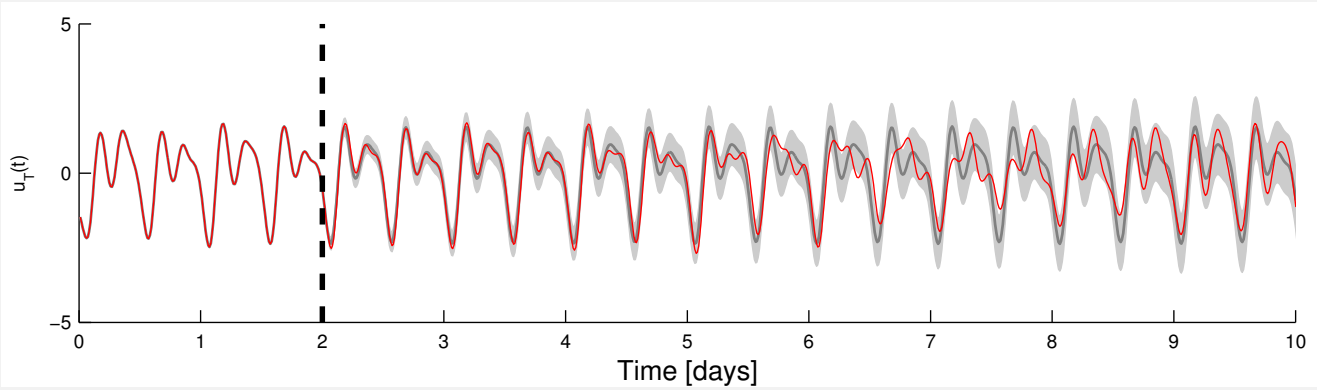
- The deterministic model for acceleration is

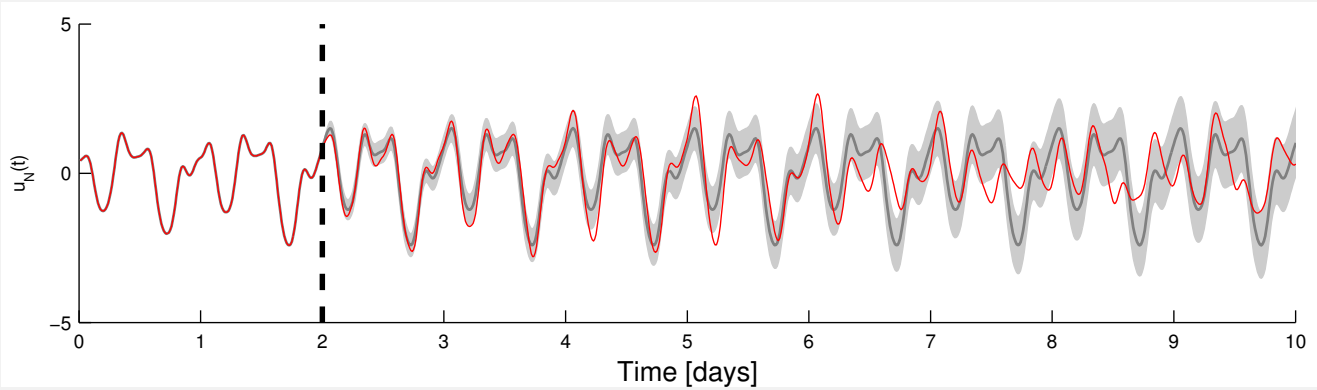
$$\mathbf{a}(\mathbf{r},t) = \mathbf{a}_{\mathrm{g}} + \mathbf{a}_{\mathrm{moon}} + \mathbf{a}_{\mathrm{sun}} + \mathbf{a}_{\mathrm{srp}}.$$

- Most modeling errors reside in the solar radiation pressure $\mathbf{a}_{\mathrm{srp}}$
- Unknown forces $\mathbf{u}(\mathbf{r},\mathbf{v},t)$ modeled as state-space GPs.

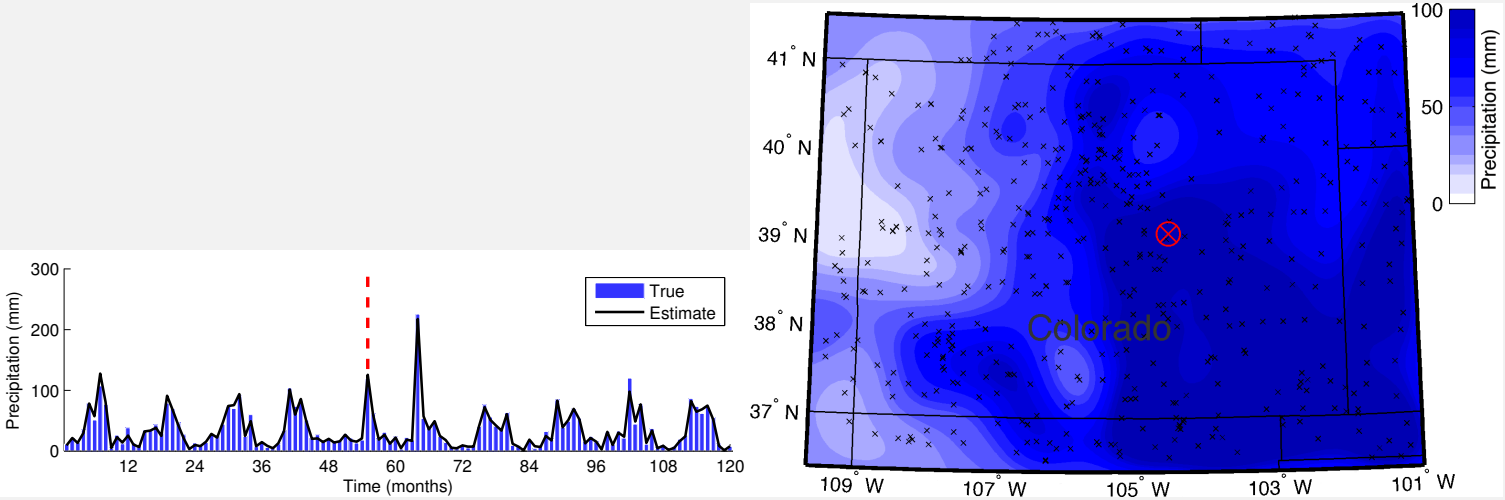
# GPS Satellite Orbit Prediction: Prediction Results



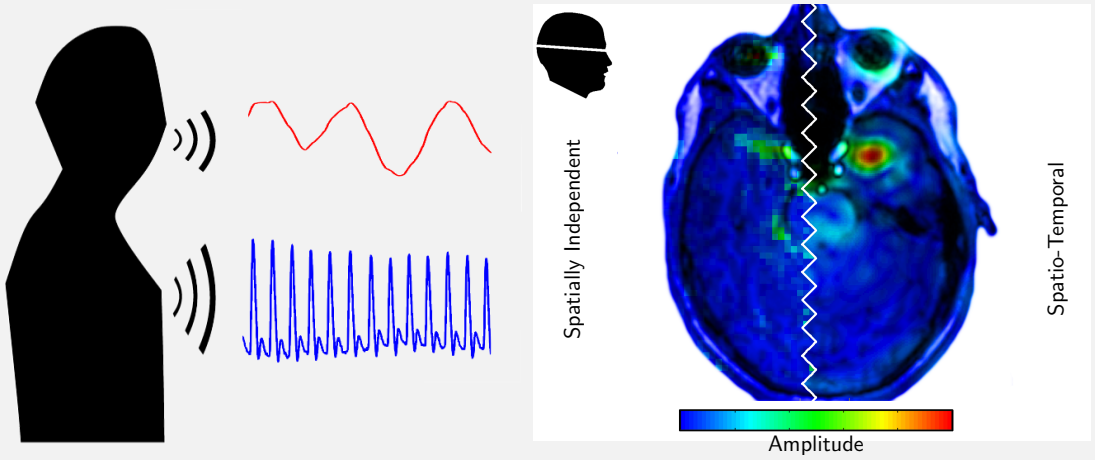



# Spatio-Temporal Modeling of Precipitation




- Spatio-temporal interpolation of precipitation levels based on monthly data, years 1895–1997, Colorado, US.
- We used an infinite-dimensional state-space GP model with the non-separable spatio-temporal Matérn covariance function.
- Truncated eigenfunction expansion of the Laplace operator with 384 eigenfunctions.

# Oscillatory Structures in fMRI Brain Data




- Spatio-temporal estimation of heart beat induced oscillations in fMRI brain data (measured at AMI centre, Finland).
- Superposition of spatio-temporal oscillators (GPs as well):

$$\frac{\partial^2 f_j(\mathbf{x}, t)}{\partial t^2} + \mathcal{A}_j \frac{\partial f_j(\mathbf{x}, t)}{\partial t} + \mathcal{B}_j f_j(\mathbf{x}, t) = \xi_j(\mathbf{x}, t).$$

- Spatial smoothness controlled by the spectral density kernel of $\xi_j(\cdot, t)$.

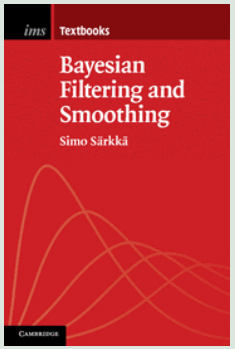
# Summary

- Bayesian filtering and smoothing methods solve dynamic Bayesian inference problems recursively.
- Bayesian continuous-discrete filtering and smoothing = state estimation in non-linear stochastic differential equation models.
- GP regression models can be often recasted as state space models and solved via Kalman filtering and smoothing.
- The infinite-dimensional generalization leads to space-space representations of spatio-temporal processes.
- Using state-space GPs as models for latent functions in differential equations leads to state-space latent force models.
- The Bayesian estimation in these half-nonparametric models can be implemented with non-linear Kalman filtering and smoothing.

# Key References

## Bayesian Methods for State-Space Models (Kalman/particle/etc.)

**S. Särkkä (2013).** *Bayesian Filtering and Smoothing*. Cambridge University Press. Also available ONLINE at `becs.aalto.fi/~ssarkka/`




## Time-Markovian Representation of Covariance Functions

**S. Särkkä, A. Solin, and J. Hartikainen (2013).** *Spatio-Temporal Learning via Infinite-Dimensional Bayesian Filtering and Smoothing*. IEEE Sig.Proc.Mag., 30(5):51–61.

# Additional references


- **Continuous-discrete state-space methods:** A. H. Jazwinski (1970). Stochastic Processes and Filtering Theory. Academic Press.
- **More recent continuous-discrete state-space methods:** S. Särkkä (2006). Recursive Bayesian Inference on Stochastic Differential Equations. Doctoral dissertation, Helsinki University of Technology.
- **Gaussian process regression and classification:** C. E. Rasmussen and C. K. I. Williams (2006). Gaussian Processes for Machine Learning. MIT Press.
- **Temporal state-space GPs:** J. Hartikainen and S. Särkkä (2010). Kalman Filtering and Smoothing Solutions to Temporal Gaussian Process Regression Models. Proc. MLSP.
- **State-space latent force models & change-point models with EC:** J. Hartikainen and S. Särkkä (2011). Sequential Inference for Latent Force Models. Proc. UAI 2011.
- **Spatio-temporal state-space GPs with non-Gaussian likelihoods & EP:** J. Hartikainen, J. Riihimäki and S. Särkkä (2011). Sparse Spatio-Temporal Gaussian Processes with General Likelihoods. Proc. ICANN.
- **Non-linear state-space latent force models with sigma-point methods:** J. Hartikainen, M. Seppänen and S. Särkkä (2012). State-Space Inference for Non-Linear Latent Force Models with Application to Satellite Orbit Prediction. Proc. ICML.
- **Student-t likelihoods:** R. Piché, S. Särkkä and J. Hartikainen (2012). Recursive Outlier-Robust Filtering and Smoothing for Nonlinear Systems Using the Multivariate Student-t Distribution. Proc. MLSP.
- **Sigma-point continuous-discrete filtering and smoothing:** S. Särkkä and J. Sarmavuori (2013). Gaussian Filtering and Smoothing for Continuous-Discrete Dynamic Systems. Sig.Proc., 93(2):500–510.
- **Spatio-temporal latent oscillation models:** A. Solin and S. Särkkä (2013). Infinite-Dimensional Bayesian Filtering for Detection of Quasi-Periodic Phenomena in Spatio-Temporal Data. Phys.Rev.E, Volume 88, Issue 5, 052909.