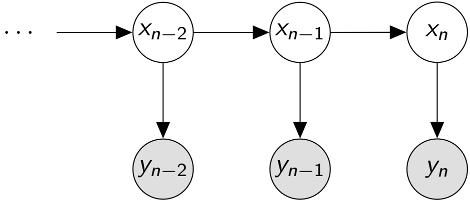
Bayesian Filtering – example

Václav Šmídl

April 22, 2020

State space model



Bayesian filtering:

$$p(\boldsymbol{x}_n|\boldsymbol{y}_{1:n-1}) = \int p(\boldsymbol{x}_n|\boldsymbol{x}_{n-1})p(\boldsymbol{x}_{n-1}|\boldsymbol{y}_{1:n-1})d\boldsymbol{x}_{n-1} \qquad \text{prediction}$$

$$p(\boldsymbol{x}_n|\boldsymbol{y}_{1:n}) = \frac{p(\boldsymbol{y}_n|\boldsymbol{x}_n)p(\boldsymbol{x}_n|\boldsymbol{y}_{1:n-1})}{p(\boldsymbol{y}_n|\boldsymbol{y}_{1:n-1})}. \qquad \text{update}$$

Conditions of state-space

- State space formalism allows to use standard methods

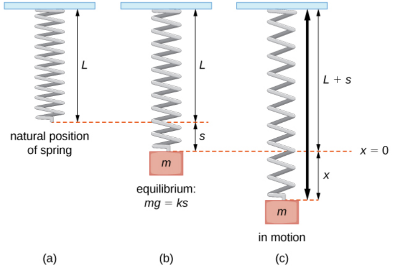
1. State-space has to satisfy conditional independence $p(x_n|x_{n-1})$,
   - state variable should contain all information needed for future prediction of the system.
   - every time-invariant variable has to have a model of its evolution (dynamics).
2. It is a first order system.
   - Higher order models has to be transformed to first order.

Spring-mass system

Consider system spring mass system

$$ma = m\frac{d^2x}{dt} = -kx = F$$

where we want to track the position x from noisy measurements.

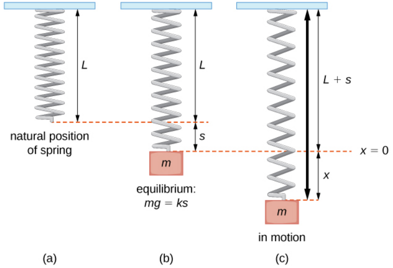


Spring-mass system

Consider system spring mass system

$$ma = m\frac{d^2x}{dt} = -kx = F$$

where we want to track the position x from noisy measurements.



1. First order system

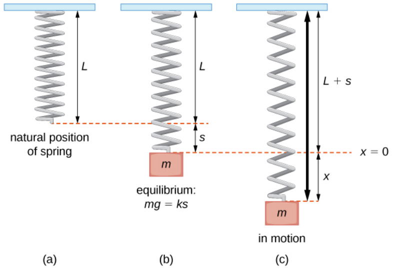
$$\frac{d}{dt}\begin{bmatrix} x \\ x' \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k/m & 0 \end{bmatrix}\begin{bmatrix} x \\ x' \end{bmatrix}. \quad \Leftrightarrow \quad \frac{d}{dt}\boldsymbol{x} = A_c\boldsymbol{x}, \boldsymbol{x} = \begin{bmatrix} x \\ x' \end{bmatrix},$$

Spring-mass system

Consider system spring mass system

$$ma = m\frac{d^2x}{dt} = -kx = F$$

where we want to track the position x from noisy measurements.



1. First order system

$$\frac{d}{dt}\begin{bmatrix} x \\ x' \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k/m & 0 \end{bmatrix}\begin{bmatrix} x \\ x' \end{bmatrix}. \quad \Leftrightarrow \quad \frac{d}{dt}\boldsymbol{x} = A_c\boldsymbol{x}, \boldsymbol{x} = \begin{bmatrix} x \\ x' \end{bmatrix},$$

2. Discretization (Euler) $\boldsymbol{x}_n = \boldsymbol{x}(t), x_{n+1} = \boldsymbol{x}(t + \Delta t)$,

$$\frac{\boldsymbol{x}_{n+1} - \boldsymbol{x}_n}{\Delta t} = A_c\boldsymbol{x}, \implies \qquad \boldsymbol{x}_{n+1} = \underbrace{(I + \Delta t A_c)}_{A}\boldsymbol{x}_n,$$

Spring mass system

Exact solution of linear ODE:

$$\boldsymbol{x}(t+\Delta t) = e^{A_c \Delta t}\boldsymbol{x}(t),$$
$$\boldsymbol{x}_{n+1} = A\boldsymbol{x}_n,$$

Spring mass system

Exact solution of linear ODE:

$$\boldsymbol{x}(t + \Delta t) = e^{A_c \Delta t} \boldsymbol{x}(t),$$
$$\boldsymbol{x}_{n+1} = A \boldsymbol{x}_n,$$

Properties:

$$\operatorname{eig}(A_c) = \pm\sqrt{-\frac{k}{m}}, \quad \omega = \sqrt{-\frac{k}{m}},$$
$$\boldsymbol{x}(t) = \begin{bmatrix} \cos(\omega t) & \frac{1}{\omega}\sin(\omega t) \\ -\omega \sin(\omega t) & \cos(\omega t) \end{bmatrix} \boldsymbol{x}(0)$$

Spring mass system

Exact solution of linear ODE:

$$\boldsymbol{x}(t+\Delta t) = e^{A_c \Delta t}\boldsymbol{x}(t),$$
$$\boldsymbol{x}_{n+1} = A\boldsymbol{x}_n,$$

Properties:

$$\mathrm{eig}(A_c) = \pm\sqrt{-\frac{k}{m}}, \quad \omega = \sqrt{-\frac{k}{m}},$$
$$\boldsymbol{x}(t) = \begin{bmatrix} \cos(\omega t) & \frac{1}{\omega}\sin(\omega t) \\ -\omega\sin(\omega t) & \cos(\omega t) \end{bmatrix} \boldsymbol{x}(0)$$

Observation (with noise):

$$\boldsymbol{y}_n = [x, x']^\top = \boldsymbol{I}\boldsymbol{x}_n + \epsilon \qquad \text{complete,}$$
$$\boldsymbol{y}_n = [x] = [1, 0]\boldsymbol{x}_n + \epsilon \qquad \text{position only,}$$
$$\boldsymbol{y}_n = [x'] = [0, 1]\boldsymbol{x}_n + \epsilon \qquad \text{velocity only}$$

Linear Gaussian State space

General linear Gaussian state-space model

$$
\begin{aligned}
\boldsymbol{x}_{n+1} &= A\boldsymbol{x}_n + B\boldsymbol{u}_n + v_n, & v_n &\sim \mathcal{N}(0, Q), \\
\boldsymbol{y}_n &= C\boldsymbol{x}_n + D\boldsymbol{u}_n + w_n, & w_n &\sim \mathcal{N}(0, R).
\end{aligned}
$$

where $\boldsymbol{x}_n$ is the state variable, and $\boldsymbol{y}_n$ is the observation at time n.

Linear Gaussian State space

General linear Gaussian state-space model

$$\begin{aligned} \boldsymbol{x}_{n+1} &= A\boldsymbol{x}_n + B\boldsymbol{u}_n + v_n, & v_n &\sim \mathcal{N}(0, Q), \\ \boldsymbol{y}_n &= C\boldsymbol{x}_n + D\boldsymbol{u}_n + w_n, & w_n &\sim \mathcal{N}(0, R). \end{aligned}$$

where $\boldsymbol{x}_n$ is the state variable, and $\boldsymbol{y}_n$ is the observation at time n.

- violates conditional independence.

Linear Gaussian State space

General linear Gaussian state-space model

$$\begin{aligned} \boldsymbol{x}_{n+1} &= A\boldsymbol{x}_n + B\boldsymbol{u}_n + v_n, & v_n &\sim \mathcal{N}(0, Q), \\ \boldsymbol{y}_n &= C\boldsymbol{x}_n + D\boldsymbol{u}_n + w_n, & w_n &\sim \mathcal{N}(0, R). \end{aligned}$$

where $\boldsymbol{x}_n$ is the state variable, and $\boldsymbol{y}_n$ is the observation at time n.

- violates conditional independence.
- not an issue for filtering, if u_n is known exactly, $p(\boldsymbol{x}_{n+1}|\boldsymbol{x}_n, \boldsymbol{u}_n)$

Linear Gaussian State space

General linear Gaussian state-space model

$$\begin{aligned} \boldsymbol{x}_{n+1} &= A\boldsymbol{x}_n + B\boldsymbol{u}_n + v_n, & v_n &\sim \mathcal{N}(0, Q), \\ \boldsymbol{y}_n &= C\boldsymbol{x}_n + D\boldsymbol{u}_n + w_n, & w_n &\sim \mathcal{N}(0, R). \end{aligned}$$

where $\boldsymbol{x}_n$ is the state variable, and $\boldsymbol{y}_n$ is the observation at time n.

- violates conditional independence.
- not an issue for filtering, if u_n is known exactly, $p(\boldsymbol{x}_{n+1}|\boldsymbol{x}_n, \boldsymbol{u}_n)$
- $\boldsymbol{u}_n$ is an "exogenous" variable – typically controlled input,
  - e.g. $u_n = -Lx_n$

Kalman filtering

General linear state-space model

$$\begin{aligned} \boldsymbol{x}_{n+1} &= A\boldsymbol{x}_n + B\boldsymbol{u}_n + v_n, & v_n &\sim \mathcal{N}(0, Q), \\ \boldsymbol{y}_n &= C\boldsymbol{x}_n + D\boldsymbol{u}_n + w_n, & w_n &\sim \mathcal{N}(0, R). \end{aligned}$$

where $\boldsymbol{x}_n$ is the state variable, and $\boldsymbol{y}_n$ is the observation at time n.
Filtering result:

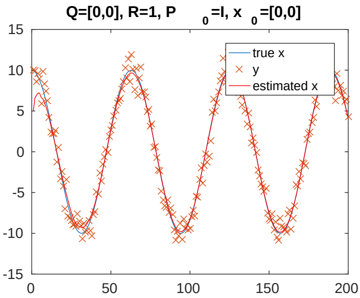
$$\begin{aligned} p(\boldsymbol{x}_n|\boldsymbol{y}_{1:n}) &= \mathcal{N}(\hat{\boldsymbol{x}}_{n|n}, P_{n|n}), & p(\boldsymbol{x}_n|\boldsymbol{y}_{1:n-1}) &= \mathcal{N}(\hat{\boldsymbol{x}}_{n|n-1}, P_{n|n-1}), \\ \hat{\boldsymbol{x}}_{n|n} &= \hat{\boldsymbol{x}}_{n|n-1} + K(\boldsymbol{y}_n - \hat{\boldsymbol{y}}_n), & \hat{\boldsymbol{x}}_{n|n-1} &= A\boldsymbol{x}_{n-1} + B\boldsymbol{u}_{n-1} \\ P_{n|n} &= (I - KC)P_{n|n-1}, & P_{n|n-1} &= AP_{n-1|n-1}A^T + Q \\ \hat{\boldsymbol{y}}_n &= C\hat{\boldsymbol{x}}_{n|n-1} + D\boldsymbol{u}_n, \\ K &= P_{n|n-1}C^T R_y^{-1}, \\ R_y &= C^T P_{n|n-1} C + R, \end{aligned}$$

Simulation #1

System

$$
\begin{aligned}
\boldsymbol{x}_n &= A\boldsymbol{x}_{n-1} + 0v_n \\
y_n &= [1, 0]\boldsymbol{x} + 1w_n \\
\boldsymbol{x}_0 &= [10, 0]
\end{aligned}
$$

estimation, KF $p(\boldsymbol{x}_0) = \mathcal{N}(0, I)$

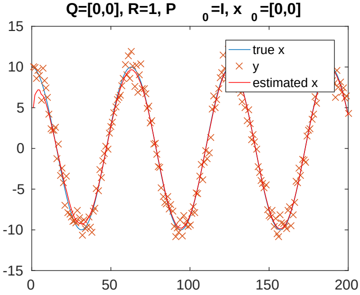


Simulation #1

System

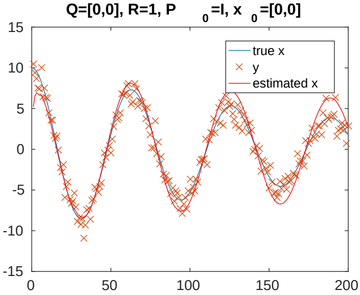
$$\begin{aligned} \boldsymbol{x}_n &= A\boldsymbol{x}_{n-1} + 0v_n \\ y_n &= [1,0]\boldsymbol{x} + 1w_n \\ \boldsymbol{x}_0 &= [10,0] \end{aligned}$$

estimation, KF $p(\boldsymbol{x}_0) = \mathcal{N}(0, I)$



Imprecise system

$$\begin{aligned} \boldsymbol{x}_n &= A\boldsymbol{x}_{n-1} + 0v_n \\ y_n &= [1,0]\boldsymbol{x} + 1w_n \end{aligned}$$

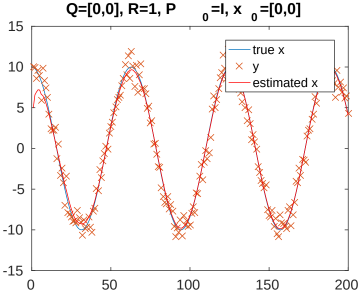


Simulation #1

System

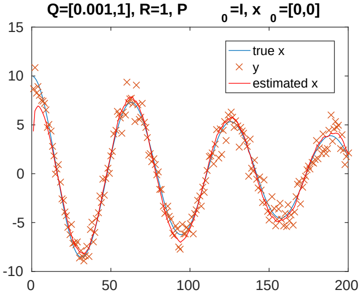
$$
\begin{aligned}
\boldsymbol{x}_n &= A\boldsymbol{x}_{n-1} + 0v_n \\
y_n &= [1,0]\boldsymbol{x} + 1w_n \\
\boldsymbol{x}_0 &= [10,0]
\end{aligned}
$$

estimation, KF $p(\boldsymbol{x}_0) = \mathcal{N}(0, I)$



Imprecise system

$$
\begin{aligned}
\boldsymbol{x}_n &= A\boldsymbol{x}_{n-1} + [0.1, 1]^\top \circ v_n \\
y_n &= [1,0]\boldsymbol{x} + 1w_n
\end{aligned}
$$



Damped oscillator

With damping

$$\frac{d}{dt}\begin{bmatrix} x \\ x' \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & -\gamma \end{bmatrix}\begin{bmatrix} x \\ x' \end{bmatrix}. \quad \overset{A=(I-\Delta t A_c)}{\Leftrightarrow} \quad \boldsymbol{x}_n = \begin{bmatrix} 1 & \Delta t \\ -\Delta t\omega^2 & 1-\Delta t\gamma \end{bmatrix}\boldsymbol{x}_{n-1}.$$

Damped oscillator

With damping

$$\frac{d}{dt}\begin{bmatrix} x \\ x' \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & -\gamma \end{bmatrix}\begin{bmatrix} x \\ x' \end{bmatrix}. \quad \overset{A=(I-\Delta t A_c)}{\Leftrightarrow} \quad \boldsymbol{x}_n = \begin{bmatrix} 1 & \Delta t \\ -\Delta t\omega^2 & 1-\Delta t\gamma \end{bmatrix}\boldsymbol{x}_{n-1}.$$

The residue is $v_n = -\Delta t \boldsymbol{x}_{n-1}$.

Damped oscillator

With damping

$$\frac{d}{dt}\begin{bmatrix} x \\ x' \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & -\gamma \end{bmatrix}\begin{bmatrix} x \\ x' \end{bmatrix}. \quad \overset{A=(I-\Delta t A_c)}{\Leftrightarrow} \quad \boldsymbol{x}_n = \begin{bmatrix} 1 & \Delta t \\ -\Delta t\omega^2 & 1-\Delta t\gamma \end{bmatrix}\boldsymbol{x}_{n-1}.$$

The residue is $v_n = -\Delta t \boldsymbol{x}_{n-1}$. Not a Gaussian!

Damped oscillator

With damping

$$\frac{d}{dt}\begin{bmatrix} x \\ x' \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & -\gamma \end{bmatrix}\begin{bmatrix} x \\ x' \end{bmatrix}. \quad \overset{A=(I-\Delta t A_c)}{\Leftrightarrow} \quad \boldsymbol{x}_n = \begin{bmatrix} 1 & \Delta t \\ -\Delta t\omega^2 & 1-\Delta t\gamma \end{bmatrix}\boldsymbol{x}_{n-1}.$$

The residue is $v_n = -\Delta t \boldsymbol{x}_{n-1}$. Not a Gaussian!
What if we know it is damping with unknown value of γ?

Damped oscillator

With damping

$$\frac{d}{dt}\begin{bmatrix} x \\ x' \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & -\gamma \end{bmatrix}\begin{bmatrix} x \\ x' \end{bmatrix}. \quad \overset{A=(I-\Delta t A_c)}{\Leftrightarrow} \quad \boldsymbol{x}_n = \begin{bmatrix} 1 & \Delta t \\ -\Delta t\omega^2 & 1-\Delta t\gamma \end{bmatrix}\boldsymbol{x}_{n-1}.$$

The residue is $v_n = -\Delta t \boldsymbol{x}_{n-1}$. Not a Gaussian!
What if we know it is damping with unknown value of γ?

- state augmentation: $\boldsymbol{x}_n = [x_n, x'_n, \gamma_n]$

Damped oscillator

With damping

$$\frac{d}{dt}\begin{bmatrix} x \\ x' \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & -\gamma \end{bmatrix}\begin{bmatrix} x \\ x' \end{bmatrix}. \quad \overset{A=(I-\Delta t A_c)}{\Leftrightarrow} \quad \boldsymbol{x}_n = \begin{bmatrix} 1 & \Delta t \\ -\Delta t\omega^2 & 1-\Delta t\gamma \end{bmatrix}\boldsymbol{x}_{n-1}.$$

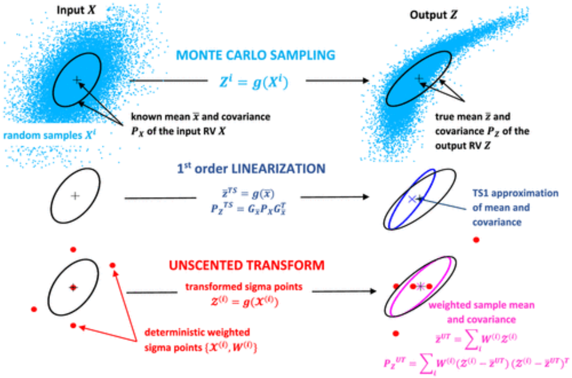
The residue is $v_n = -\Delta t \boldsymbol{x}_{n-1}$. Not a Gaussian!
What if we know it is damping with unknown value of γ?

- state augmentation: $\boldsymbol{x}_n = [x_n, x_n', \gamma_n]$
- parameter dynamics $\gamma_n = \gamma_{n-1} + (\epsilon)$,
- is no longer linear
  - particle filter (sequential MC)
  - Kalman filter with local linearization

Local Linearization

Local linearization: transformation of standard Gaussian $\boldsymbol{x}_{n-1} \sim \mathcal{N}(\hat{\boldsymbol{x}}, P)$

$$\boldsymbol{x}_n = A\boldsymbol{x}_{n-1}, \qquad \boldsymbol{x}_{n|n-1} \sim \mathcal{N}(A\hat{\boldsymbol{x}}, APA'),$$



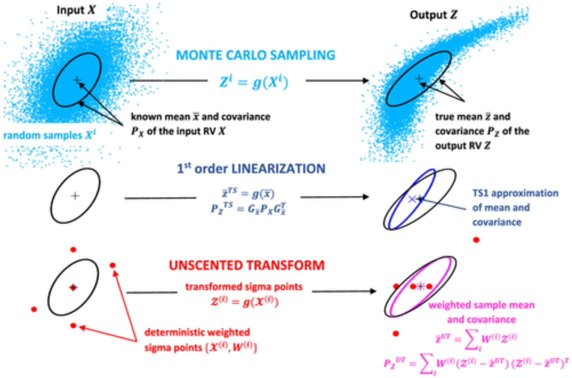
Local Linearization

Local linearization: transformation of standard Gaussian $\boldsymbol{x}_{n-1} \sim \mathcal{N}(\hat{\boldsymbol{x}}, P)$

$$\boldsymbol{x}_n = A\boldsymbol{x}_{n-1}, \qquad \boldsymbol{x}_{n|n-1} \sim \mathcal{N}(A\hat{\boldsymbol{x}}, APA'),$$

for non-linear systems

$$\boldsymbol{x}_n = f(\boldsymbol{x}_{n-1}), \qquad \boldsymbol{x}_{n|n-1} \sim \mathcal{N}(f(\hat{\boldsymbol{x}}), APA'), \qquad A = \frac{\partial f}{\partial \boldsymbol{x}_{n-1}}|_{\boldsymbol{x}=\hat{\boldsymbol{x}}}$$



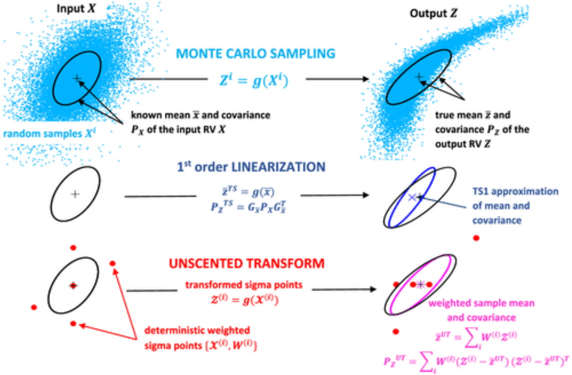
Local Linearization

Local linearization: transformation of standard Gaussian $\boldsymbol{x}_{n-1} \sim \mathcal{N}(\hat{\boldsymbol{x}}, P)$

$$\boldsymbol{x}_n = A\boldsymbol{x}_{n-1}, \qquad \boldsymbol{x}_{n|n-1} \sim \mathcal{N}(A\hat{\boldsymbol{x}}, APA'),$$

for non-linear systems

$$\boldsymbol{x}_n = f(\boldsymbol{x}_{n-1}), \qquad \boldsymbol{x}_{n|n-1} \sim \mathcal{N}(f(\hat{\boldsymbol{x}}), APA'), \qquad A = \frac{\partial f}{\partial \boldsymbol{x}_{n-1}}|_{\boldsymbol{x}=\hat{\boldsymbol{x}}}$$



Extended Kalman Filter

Extended Kalman filter:

$$
\begin{aligned}
\boldsymbol{x}_n &= f(\boldsymbol{x}_{n-1}, \boldsymbol{u}_n) + v_n, \\
\boldsymbol{y}_n &= h(\boldsymbol{x}_n, \boldsymbol{u}_n) + w_n,
\end{aligned}
$$

Update equations:

$$
\begin{aligned}
p(\boldsymbol{x}_n|\boldsymbol{y}_{1:n}) &= \mathcal{N}(\hat{\boldsymbol{x}}_{n|n}, P_{n|n}), \\
\hat{\boldsymbol{x}}_{n|n} &= \hat{\boldsymbol{x}}_{n|n-1} + K(\boldsymbol{y}_n - \hat{\boldsymbol{y}}_n) \\
P_{n|n} &= (I - KC)P_{n|n-1}, \\
\hat{\boldsymbol{y}}_n &= h(\hat{\boldsymbol{x}}_{n|n-1}, \boldsymbol{u}_n), \\
K &= P_{n|n-1}C^T R_y^{-1}, \\
R_y &= C^T P_{n|n-1} C + R,
\end{aligned}
$$

$$
\begin{aligned}
p(\boldsymbol{x}_n|\boldsymbol{y}_{1:n-1}) &= \mathcal{N}(\hat{\boldsymbol{x}}_{n|n-1}, P_{n|n-1}), \\
\hat{\boldsymbol{x}}_{n|n-1} &= f(\hat{\boldsymbol{x}}_{n-1}, \boldsymbol{u}_{n-1}) \\
P_{n|n-1} &= AP_{n-1|n-1}A^T + Q
\end{aligned}
$$

$$
\begin{aligned}
A &= \frac{\partial f(\boldsymbol{x}_n, \boldsymbol{u}_n)}{\partial \boldsymbol{x}_n}\Big|_{\hat{x}_{n-1|n-1}} \\
C &= \frac{\partial h(\boldsymbol{x}_n, \boldsymbol{u}_n)}{\partial \boldsymbol{x}_n}\Big|_{\hat{x}_{n|n-1}}
\end{aligned}
$$

Application to damped oscillator

Dynamics

$$\begin{bmatrix} x_n \\ x'_n \\ \gamma_n \end{bmatrix} = \begin{bmatrix} 1 & \Delta t & 0 \\ -\Delta t\omega^2 & 1-\Delta t\gamma_{n-1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{n-1} \\ x'_{n-1} \\ \gamma_{n-1} \end{bmatrix}$$

$$= \begin{bmatrix} x_{n-1}+\Delta t x'_{n-1} \\ -\Delta t\omega^2 x_{n-1} + (1-\Delta t\gamma_{n-1})x'_{n-1} \\ \gamma_{n-1} \end{bmatrix}$$

$$\frac{df}{d\boldsymbol{x}}\Big|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{n-1}} = \begin{bmatrix} 1 & \Delta t & 0 \\ -\Delta t\omega^2 & 1-\Delta t\hat{\gamma}_{n-1} & -\Delta t\hat{x}'_{n-1} \\ 0 & 0 & 1 \end{bmatrix}.$$

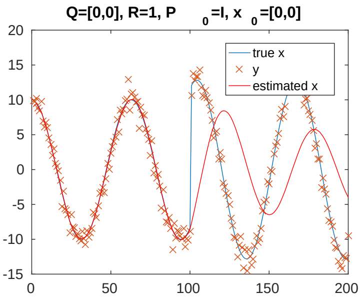
Observation

$$y_n = [1,0,0]\boldsymbol{x}_n$$

Simulation #2

System

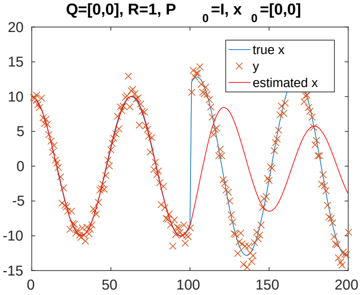
$$\begin{aligned} \boldsymbol{x}_n &= A\boldsymbol{x}_{n-1} \\ y_n &= [1,0]\boldsymbol{x}_n + 1w_n \\ x_{100} &= x_{99} + 20. \end{aligned}$$



Simulation #2

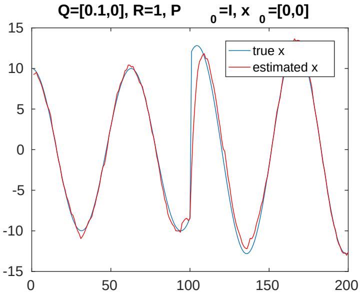
System

$$\begin{aligned} \boldsymbol{x}_n &= A\boldsymbol{x}_{n-1} \\ y_n &= [1,0]\boldsymbol{x}_n + 1w_n \\ x_{100} &= x_{99} + 20. \end{aligned}$$



System

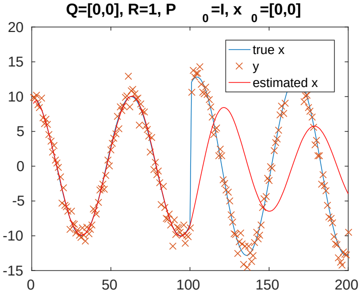
$$\begin{aligned} \boldsymbol{x}_n &= A\boldsymbol{x}_{n-1} + [0.3, 0]^\top \circ v_n \\ y_n &= [1,0]\boldsymbol{x} + 1w_n \end{aligned}$$



Simulation #2

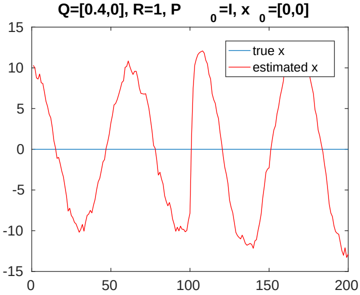
System

$$
\begin{aligned}
\boldsymbol{x}_n &= A\boldsymbol{x}_{n-1} \\
y_n &= [1,0]\boldsymbol{x}_n + 1w_n \\
x_{100} &= x_{99} + 20.
\end{aligned}
$$



System

$$
\begin{aligned}
\boldsymbol{x}_n &= A\boldsymbol{x}_{n-1} + [0.6, 0]^\top \circ v_n \\
y_n &= [1,0]\boldsymbol{x} + 1w_n
\end{aligned}
$$



Severe disturbance

- What is distribution of a sequence: $0, 0, 0, \ldots, 0, 20, 0, \ldots$

Severe disturbance

- What is distribution of a sequence: $0, 0, 0, \ldots, 0, 20, 0, \ldots$
- What is a more suitable distribution?

Severe disturbance

- What is distribution of a sequence: $0, 0, 0, \dots, 0, 20, 0, \dots$
- What is a more suitable distribution?
  - spike and slab

$$p(w_n) = \alpha\mathcal{N}(0, \sigma_1 I) + (1-\alpha)\mathcal{N}(0, \sigma_2 I),$$

  - Normal-Gamma

$$p(w_n) = \mathcal{N}(0, \sigma_n Q),$$
$$p(\sigma_n) = \mathcal{G}(\alpha_0, \beta_0)$$

  - generalized Normal, i.e. $\exp((-|x|/\alpha)^{\beta})$

Kalman filter with Spike & Slab innovations

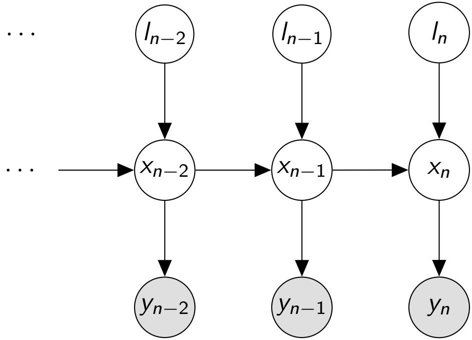
Rewriting:

$$p(w_n) = \alpha\mathcal{N}(0, \sigma_1 I) + (1-\alpha)\mathcal{N}(0, \sigma_2 I),$$

Kalman filter with Spike & Slab innovations

Rewriting:

$$
\begin{aligned}
p(w_n) &= \alpha\mathcal{N}(0, \sigma_1 I) + (1-\alpha)\mathcal{N}(0, \sigma_2 I), \\
p(w_n, l_n) &= p(w_n|l_n)p(l_n) \\
p(w_n|l_n = 1) &= \mathcal{N}(0, Q_1), \quad p(l_n = 1) = \alpha \\
p(w_n|l_n = 2) &= \mathcal{N}(0, Q_2), \quad p(l_n = 2) = 1-\alpha
\end{aligned}
$$



EM:

EM algorithm

Local approximation:
In each step n:

- compute $p(y_n|x_{1:n}, l_n = 1)$, $p(y_n|x_{1:n}, l_n = 2)$
- estimate $\hat{l}_n = \arg\max_{l_n}(p(y_n|x_{1:n}, l_n))$
- Run KF with $Q_n = \begin{cases} Q_1 & \text{if } l_n = 1 \\ Q_2 & \text{otherwise} \end{cases}$

EM algorithm

Local approximation:
In each step n:

- compute $p(y_n|x_{1:n}, l_n = 1)$, $p(y_n|x_{1:n}, l_n = 2)$
- estimate $\hat{l}_n = \arg\max_{l_n}(p(y_n|x_{1:n}, l_n))$
- Run KF with $Q_n = \begin{cases} Q_1 & \text{if } l_n = 1 \\ Q_2 & \text{otherwise} \end{cases}$

Local approximation. What error are we making?

Particle filter

Consider $p(\boldsymbol{x}_{n-1}|y_{1:n-1})$ in the form of weighted empirical density:

$$p(\boldsymbol{x}_{n-1}|y_{1:n-1}) = \sum_{i=1}^{I} w_{n-1}^{(i)} \delta(\boldsymbol{x}_{n-1} - \boldsymbol{x}_{n-1}^{(i)}),$$

Prediction:

$$p(\boldsymbol{x}_n|y_{1:n-1}) = \int \sum_{i=1}^{I} w_{n-1}^{(i)} \delta(\boldsymbol{x}_{n-1} - \boldsymbol{x}_{n-1}^{(i)}) p(\boldsymbol{x}_n|\boldsymbol{x}_{n-1}) = \sum_{i=1}^{I} w_{n-1}^{(i)} p(\boldsymbol{x}_n|\boldsymbol{x}_{n-1}^{(i)}),$$

Correction:

$$p(\boldsymbol{x}_n|y_{1:n}) \propto p(y_n|\boldsymbol{x}_n) p(\boldsymbol{x}_n|y_{1:n-1})$$

Particle filter

Consider $p(\boldsymbol{x}_{n-1}|y_{1:n-1})$ in the form of weighted empirical density:

$$p(\boldsymbol{x}_{n-1}|y_{1:n-1}) = \sum_{i=1}^{I} w_{n-1}^{(i)} \delta(\boldsymbol{x}_{n-1} - \boldsymbol{x}_{n-1}^{(i)}),$$

Prediction:

$$p(\boldsymbol{x}_n|y_{1:n-1}) = \int \sum_{i=1}^{I} w_{n-1}^{(i)} \delta(\boldsymbol{x}_{n-1} - \boldsymbol{x}_{n-1}^{(i)}) p(\boldsymbol{x}_n|\boldsymbol{x}_{n-1}) = \sum_{i=1}^{I} w_{n-1}^{(i)} p(\boldsymbol{x}_n|\boldsymbol{x}_{n-1}^{(i)}),$$

Correction:

$$p(\boldsymbol{x}_n|y_{1:n}) \propto p(y_n|\boldsymbol{x}_n) p(\boldsymbol{x}_n|y_{1:n-1}) \frac{q(\boldsymbol{x}_n|y_{1:n-1})}{q(\boldsymbol{x}_n|y_{1:n-1})}$$

using $p(\boldsymbol{x}_n|\boldsymbol{x}_{n-1}^{(i)})$ as the proposal, and sampling one $\boldsymbol{x}_n^{(i)}$ for each $\boldsymbol{x}_{n-1}^{(i)}$ we obtain

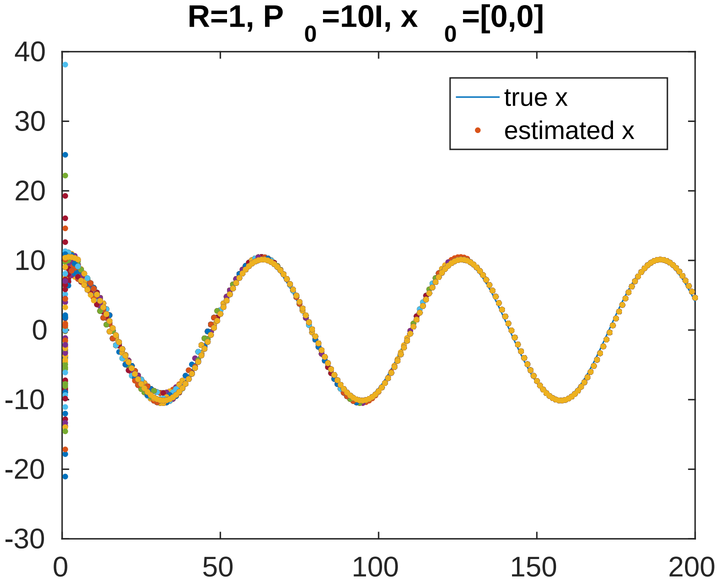
$$w_n^{(i)} \propto p(y_n|\boldsymbol{x}_n) w_{n-1}^{(i)}$$

Particle filter for oscillator

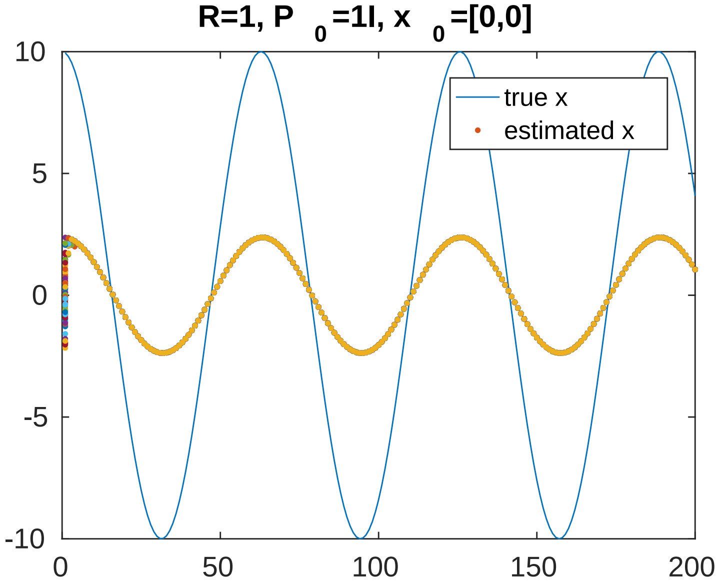
Initial distribution: sample $\boldsymbol{x}_0^{(i)} \sim \mathcal{N}(0, I)$
In each step:

- predict every particle: $\boldsymbol{x}_n^{(i)} = A\boldsymbol{x}_{n-1}^{(i)}$
- compute weight
$$w_n^{(i)} \propto p(y_n | x_n^{(i)}) w_{n-1}^{(i)},$$
- resample
  - generate new indexes of copied particles
  - copy particle values $\boldsymbol{x}_n^{(i)} \leftarrow \boldsymbol{x}_n^{(i_{resampled})}$
  - set $w_n = \frac{1}{I}$

Simulation PF with I =100:

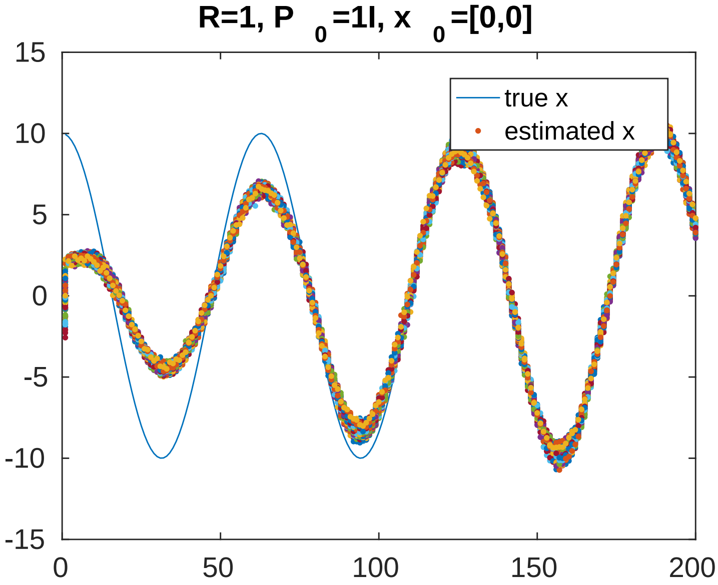


Simulation PF with $I = 100$:



Simulation PF with artificial dynamics:

- Use some q in prediction $x_n = Ax_{n-1} + q \circ v_n$.



Take home message

- Probability calculus can be applied to time series
- The probability language is like a toolbox.
- Tricks from other areas are transferable
  - EM algorithm,
  - importance sampling,
  - MCMC

Assignment

KF and PF in matlab available on webpage

Assignment	points
EKF for damping	10
EM in KF	10
PF for damping	15
PF for step change	15