Bootstrap Particle Filtering

Roland Hostettler

November 28, 2018

Recap: Extended Kalman Filter

- Model approximation:

$$\boldsymbol{x}_n = f(\boldsymbol{x}_{n-1}) + \boldsymbol{q}_n \approx f(\hat{\boldsymbol{x}}_{n-1|n-1}) + \boldsymbol{F}_x(\boldsymbol{x}_{n-1} - \hat{\boldsymbol{x}}_{n-1|n-1}) + \boldsymbol{q}_n$$
$$\boldsymbol{y}_n = g(\boldsymbol{x}_n) + \boldsymbol{r}_n \approx g(\hat{\boldsymbol{x}}_{n|n-1}) + \boldsymbol{G}_x(\boldsymbol{x}_n - \hat{\boldsymbol{x}}_{n|n-1}) + \boldsymbol{r}_n$$

- Prediction:

$$\begin{aligned} \hat{\boldsymbol{x}}_{n|n-1} &= f(\hat{\boldsymbol{x}}_{n-1|n-1}), \\ \boldsymbol{P}_{n|n-1} &= \boldsymbol{F}_x \boldsymbol{P}_{n-1|n-1} \boldsymbol{F}_x^{\mathsf{T}} + \boldsymbol{Q}_n, \end{aligned}$$

- Measurement update:

$$\begin{aligned} \boldsymbol{K}_n &= \boldsymbol{P}_{n|n-1} \boldsymbol{G}_x^{\mathsf{T}} (\boldsymbol{G}_x \boldsymbol{P}_{n|n-1} \boldsymbol{G}_x^{\mathsf{T}} + \boldsymbol{R}_n)^{-1}, \\ \hat{\boldsymbol{x}}_{n|n} &= \hat{\boldsymbol{x}}_{n|n-1} + \boldsymbol{K}_n(\boldsymbol{y}_n - g(\hat{\boldsymbol{x}}_{n|n-1})), \\ \boldsymbol{P}_{n|n} &= \boldsymbol{P}_{n|n-1} - \boldsymbol{K}_n(\boldsymbol{G}_x \boldsymbol{P}_{n|n-1} \boldsymbol{G}_x^{\mathsf{T}} + \boldsymbol{R}_n) \boldsymbol{K}_n^{\mathsf{T}}. \end{aligned}$$

Recap: Unscented Kalman Filter

- Uses a nonlinear transformation of deterministic sampling points
- Prediction:
  - Calculate the sigma-points using $\hat{\boldsymbol{x}}_{n-1|n-1}$ and $\boldsymbol{P}_{n-1|n-1}$
  - Propagate the sigma-points $\boldsymbol{x}_n^j = f(\boldsymbol{x}_{n-1}^j)$
  - Calculate the mean and covariance $\hat{\boldsymbol{x}}_{n|n-1}$, $\boldsymbol{P}_{n|n-1}$
- Measurement update:
  - Calculate the sigma-points using $\hat{\boldsymbol{x}}_{n|n-1}$ and $\boldsymbol{P}_{n|n-1}$
  - Propagate the sigma-points $\boldsymbol{y}_n^j = g(\boldsymbol{x}_n^j)$
  - Calculate the mean and covariance $\mathrm{E}\{\boldsymbol{y}_n \mid \boldsymbol{y}_{1:n-1}\}$, $\mathrm{Cov}\{\boldsymbol{y}_n \mid \boldsymbol{y}_{1:n-1}\}$, $\mathrm{Cov}\{\boldsymbol{x}_n, \boldsymbol{y}_n \mid \boldsymbol{y}_{1:n-1}\}$
  - Perform the Kalman filter measurement update:

$$\begin{aligned}
\boldsymbol{K}_n &= \mathrm{Cov}\{\boldsymbol{x}_n, \boldsymbol{y}_n \mid \boldsymbol{y}_{1:n-1}\} \, \mathrm{Cov}\{\boldsymbol{y}_n \mid \boldsymbol{y}_{1:n-1}\}^{-1}, \\
\hat{\boldsymbol{x}}_{n|n} &= \hat{\boldsymbol{x}}_{n|n-1} + \boldsymbol{K}_n(\boldsymbol{y}_n - \mathrm{E}\{\boldsymbol{y}_n \mid \boldsymbol{y}_{1:n-1}\}), \\
\boldsymbol{P}_{n|n} &= \boldsymbol{P}_{n|n-1} - \boldsymbol{K}_n \, \mathrm{Cov}\{\boldsymbol{y}_n \mid \boldsymbol{y}_{1:n-1}\} \boldsymbol{K}_n^\mathsf{T}.
\end{aligned}$$

Intended Learning Outcomes

After this lecture, you will be able to:

- describe the basic idea of particle filtering,
- explain the three steps in particle filtering: simulation, weighting, resampling,
- identify the differences between Kalman filtering and particle filtering.

Discrete-Time Nonlinear State-Space Model

- Discrete-time nonlinear state-space model:

$$\begin{aligned} \boldsymbol{x}_n &= f(\boldsymbol{x}_{n-1}) + \boldsymbol{q}_n \\ \boldsymbol{y}_n &= g(\boldsymbol{x}_n) + \boldsymbol{r}_n \end{aligned}$$

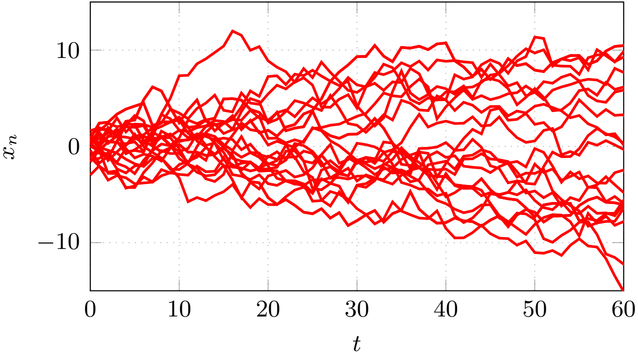
- Process noise: $\boldsymbol{q}_n \sim p(\boldsymbol{q}_n)$
- Measurement noise: $\boldsymbol{r}_n \sim p(\boldsymbol{r}_n)$
- Initial state: $\boldsymbol{x}_0 \sim p(\boldsymbol{x}_0)$

This is a stochastic process, each realization of the state sequence $\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n$ is different

Example: Random Walk Process (1/2)

- Dynamic model:

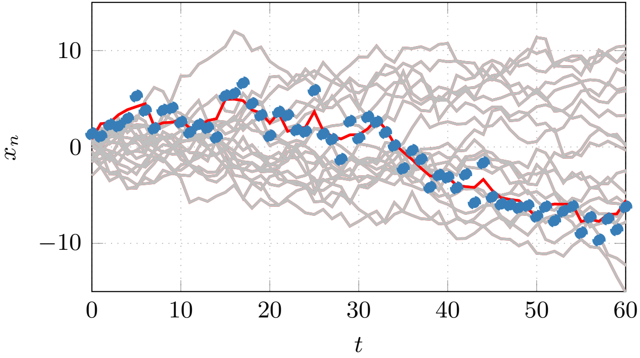
$$
\begin{aligned}
x_n &= x_{n-1} + q_n \\
x_0 &\sim \mathcal{N}(0, 1) \\
q_n &\sim \mathcal{N}(0, 1)
\end{aligned}
$$



Example: Random Walk Process (2/2)

- Only one realization of the process is observed
- Measurement model:

$$y_n = x_n + r_n$$
$$r_n \sim \mathcal{N}(0, 1)$$



Particle Filtering: Idea

Prediction

- Given: Simulated states $\boldsymbol{x}_{n-1}^{j}$ $(j = 1, \dots, J)$
- Simulate from t_{n-1} to t_n to obtain $\boldsymbol{x}_n^j$ $(j = 1, \dots, J)$

Measurement Update

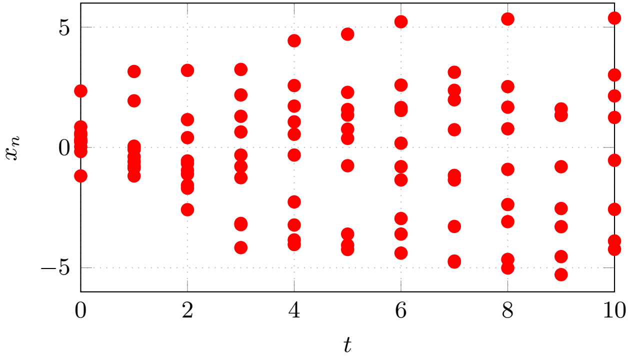
- Evaluate how well $\boldsymbol{x}_n^j$ explains $\boldsymbol{y}_n$ $(j = 1, \dots, J)$
- Assign a weight w_n^j to $\boldsymbol{x}_n^j$ $(j = 1, \dots, J)$

Prediction: Simulation

- Intuitive way: Use the dynamic model to simulate one time step

$$\boldsymbol{x}_n = f(\boldsymbol{x}_{n-1}) + \boldsymbol{q}_n$$

- Two step procedure:
  1. Sample $\boldsymbol{q}_n^j \sim p(\boldsymbol{q}_n)$,
  2. Calculate $\boldsymbol{x}_n^j = f(\boldsymbol{x}_{n-1}^j) + \boldsymbol{q}_n^j$.



Measurement Update: Importance Weights

- Weights w_n^j indicate the relevance of each sample
- Importance weights:
  - High weight w_n^j: Explains $\boldsymbol{y}_n$ well
  - Low weight w_n^j: Explains $\boldsymbol{y}_n$ poorly
  - Should sum to one:

$$\sum_{j=1}^{J} w_n^j = 1,$$

- Cost function gives low values for good estimates of $\boldsymbol{x}_n$

Measurement Update: Likelihood (1/2)

- Measurement model:

$$\boldsymbol{y}_n = g(\boldsymbol{x}_n) + \boldsymbol{r}_n$$
$$\boldsymbol{r}_n \sim p(\boldsymbol{r}_n)$$

- $\boldsymbol{r}_n$ is a random variable $\Rightarrow$ $\boldsymbol{y}_n$ is a random variable too
- $\boldsymbol{y}_n$ must have a probability density function (pdf)
- Given $\boldsymbol{x}_n$, the pdf for $\boldsymbol{y}_n$ is the same as for $\boldsymbol{r}_n$ but shifted by $g(\boldsymbol{x}_n)$
- The pdf for $\boldsymbol{y}_n$ given $\boldsymbol{x}_n$ is called the likelihood

$$\boldsymbol{y}_n \sim p(\boldsymbol{y}_n \mid \boldsymbol{x}_n)$$

Measurement Update: Likelihood (2/2)

- The likelihood is a suitable measure for the importance weights w_n^j
- The non-normalized weights are then:

$$\tilde{w}_n^j = p(\boldsymbol{y}_n \mid \boldsymbol{x}_n^j).$$

- Normalization:

$$w_n^j = \frac{\tilde{w}_n^j}{\sum_{i=1}^J \tilde{w}_n^i}.$$

Example: Gaussian Likelihood (1/2)

- Measurement model:

$$\boldsymbol{y}_n = g(\boldsymbol{x}_n) + \boldsymbol{r}_n$$

- The measurement noise is often (assumed) Gaussian:

$$p(\boldsymbol{r}_n) = \mathcal{N}(\boldsymbol{r}_n; 0, \boldsymbol{R}_n)$$

- Then, the likelihood is Gaussian too:

$$p(\boldsymbol{y}_n \mid \boldsymbol{x}_n) = \mathcal{N}(\boldsymbol{y}_n; g(\boldsymbol{x}_n), \boldsymbol{R}_n).$$

Example: Gaussian Likelihood (2/2)

- Example: Scalar case with

$$
\begin{aligned}
y_n &= g(x_n) + r_n \\
r_n &\sim \mathcal{N}(0, \sigma_r^2)
\end{aligned}
$$

Point Estimates

- Moments of the state can be calculated using weighted sums of the weighted samples
- Mean:

$$\hat{\boldsymbol{x}}_{n|n} = \sum_{j=1}^{J} w_n^j \boldsymbol{x}_n^j$$

- Covariance:

$$\boldsymbol{P}_{n|n} = \sum_{j=1}^{J} w_n^j (\boldsymbol{x}_n^j - \hat{\boldsymbol{x}}_{n|n})(\boldsymbol{x}_n^j - \hat{\boldsymbol{x}}_{n|n})^\mathsf{T}.$$

Summary: Sequential Sampling and Weighing

- Initialization: Sample J particles:

$$\boldsymbol{x}_0^j \sim p(\boldsymbol{x}_0)$$

- Prediction: Sample $\boldsymbol{q}_n^j$ and propagate particles:

$$\begin{aligned}\boldsymbol{q}_n^j &\sim p(\boldsymbol{q}_n) \\ \boldsymbol{x}_n^j &= f(\boldsymbol{x}_n^j) + \boldsymbol{q}_n^j\end{aligned}$$

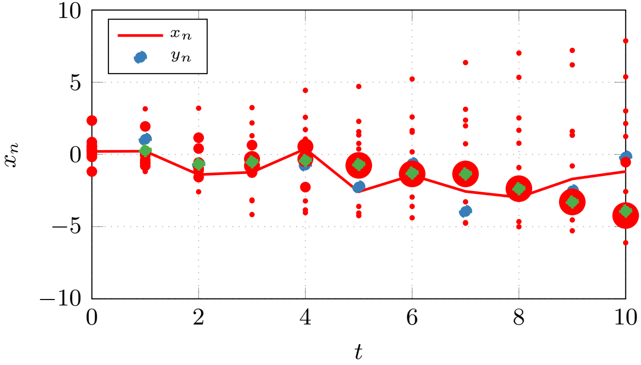
- Measurement update: Calculate and normalize the particle weights:

$$\begin{aligned}\tilde{w}_n^j &= p(\boldsymbol{y}_n \mid \boldsymbol{x}_n^j) \\ w_n^j &= \frac{w_n^j}{\sum_{i=1}^J \tilde{w}_n^i}\end{aligned}$$

Example: Random Walk Process

- State-space model:

$$x_n = x_{n-1} + q_n$$
$$y_n = x_n + r_n$$



Resampling (1/2)

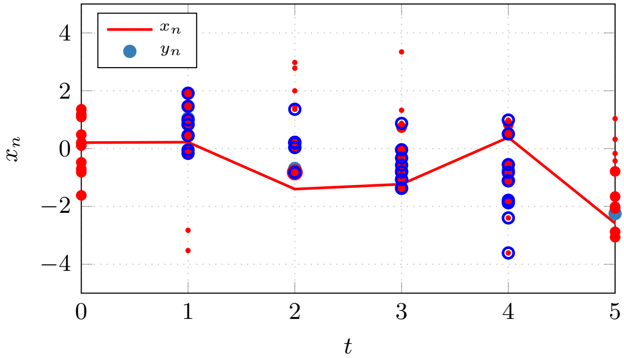
- Problem: The particles diverge after a few samples
- Resampling:
  - Remove samples with low weights
  - Replicate samples with high weights
  - Samples should be represented proportional to their weight:

$$\lfloor w_n^j J \rceil$$

- Equivalent interpretation

$$\Pr\{\tilde{\boldsymbol{x}}_n^i = \boldsymbol{x}_n^j\} = w_n^j,$$

Resampling (2/2)



Bootstrap Particle Filter

Algorithm 1 Bootstrap Particle Filter (Gaussian Noises)

Initialize: $\boldsymbol{x}_0^j \sim \mathcal{N}(\boldsymbol{m}_0, \boldsymbol{P}_0)$ $(j = 1, \dots, J)$
for $n = 1, 2, \dots$ **do**
    for $j = 1, 2, \dots, J$ **do**
        Sample: $\boldsymbol{q}_n^j \sim \mathcal{N}(0, \boldsymbol{Q})$
        Propagate the state: $\boldsymbol{x}_n^j = f(\boldsymbol{x}_{n-1}^j) + \boldsymbol{q}_n^j$
        Calculate the weights: $\tilde{w}_n^j = \mathcal{N}(\boldsymbol{y}_n; g(\boldsymbol{x}_n^j), \boldsymbol{R}_n)$
    end for
    Normalize the importance weights $(j = 1, \dots, J)$

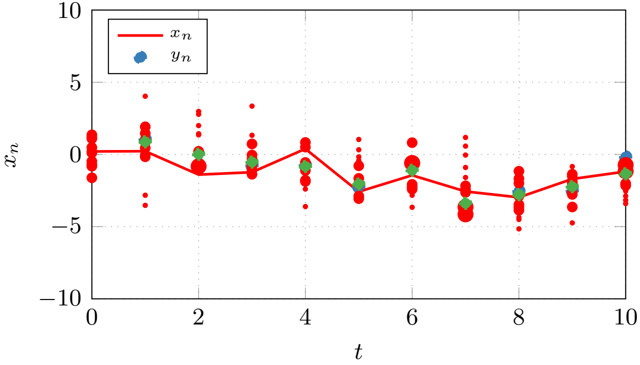
$$w_n^j = \frac{\tilde{w}_n^j}{\sum_{i=1}^{J} \tilde{w}_n^i}$$

    Calculate the mean $\hat{\boldsymbol{x}}_{n|n}$ and covariance $\boldsymbol{P}_{n|n}$
    Resample such that $\Pr\{\tilde{\boldsymbol{x}}_n^i = \boldsymbol{x}_n^j\} = w_n^j$
end for

Example: Random Walk

- State-space model:

$$x_n = x_{n-1} + q_n$$
$$y_n = x_n + r_n$$



Example: Object Tracking (1/3)

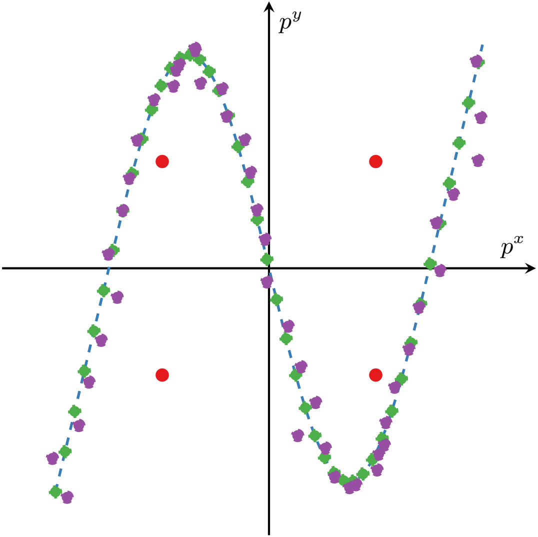
- Quasi-constant turn model:

$$\begin{bmatrix} \dot{p}^x(t) \\ \dot{p}^y(t) \\ \dot{v}(t) \\ \dot{\varphi}(t) \end{bmatrix} = \begin{bmatrix} v(t)\cos(\varphi(t)) \\ v(t)\sin(\varphi(t)) \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{w}(t)$$

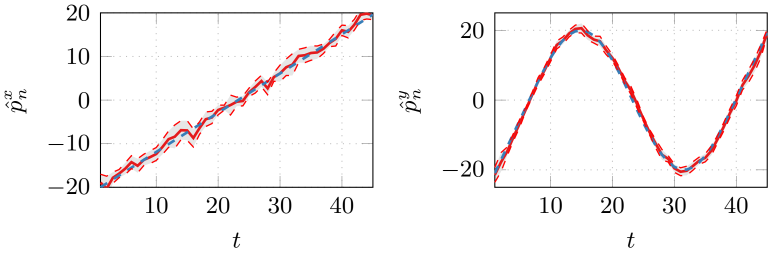
- Range (distance) measurements:

$$\boldsymbol{y}_n = \begin{bmatrix} |\boldsymbol{p}_n - \boldsymbol{p}_1^s| \\ |\boldsymbol{p}_n - \boldsymbol{p}_2^s| \\ \vdots \\ |\boldsymbol{p}_n - \boldsymbol{p}_K^s| \end{bmatrix} + \boldsymbol{r}_n$$

Example: Object Tracking (2/3)



Example: Object Tracking (3/3)



Summary

- The particle filter uses a set of random samples to estimate the state
- During prediction, the samples are propagated from t_{n-1} to t_n
- The bootstrap particle filter uses the dynamic model to propagate the samples
- The measurement update evaluates the likelihood to assign an importance weight to each sample
- Resampling is used to mitigate particle degeneracy
- Particle filtering is a universal approach equally applicable to linear and nonlinear system
- It can be shown that particle filters are asymptotically $(J \rightarrow \infty)$ optimal in many cases

Announcements

- Exam: Monday, December 10, 2018, 14.00–17.00 in TU6
- Allowed aids: 1 handwritten A4 sheet (not written by computer, not photocopied, etc.)
- Project Q&A session: Friday, November 30, 2018, 13.00–14.00 in F336 (Come prepared!)