Bayesian State Estimation via Deep Probabilistic Learning

Allen Umali

University of Southern California

June 2023

Overview

- Motivation
- Preliminaries
- Bayes Filter
- Approximations
    - Kalman Filter
    - Extended Kalman Filter
    - Monte Carlo Methods
        - Ensemble Kalman Filter
        - Particle Filter
- Illustrative example
    - Performance
    - Limitations
- Conditional GAN
- Summary

Motivation

- State Estimation is the problem of probabilistically inferring the state of a dynamical system by fusing model and sensor data when the true state of the system is hidden.
- In robotics applications, the robot's belief of its state or the state of the environment is critical for deciding/computing control action with state feedback
- Sensors are noisy and/or incomplete, so the robot must maintain some probabilistic belief of state.
- State-of-the-art state estimation techniques come with certain limitations.
- The topic of the dissertation is a novel state estimation method using a conditional generative adversarial network (cGAN) to investigate overcoming those limitations.

Probabilistic State Space Models

- **Linear, additive random noise**
  - Continuous

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{q}(t) \qquad (4.1)$$
$$\boldsymbol{y}(t) = \boldsymbol{H}\boldsymbol{x}(t) + \boldsymbol{D}\boldsymbol{u}(t) + \boldsymbol{r}(t) \qquad (4.2)$$

  - Discrete

$$\boldsymbol{x_k} = \bar{\boldsymbol{A}}\boldsymbol{x_{k-1}} + \bar{\boldsymbol{B}}\boldsymbol{u_{k-1}} + \boldsymbol{q_{k-1}} \qquad (4.3)$$
$$\boldsymbol{y_k} = \boldsymbol{H}\boldsymbol{x_k} + \boldsymbol{D}\boldsymbol{u_{k-1}} + \boldsymbol{r_k} \qquad (4.4)$$

- **Non-linear, non-additive random noise**
  - Continuous

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{q}(t)\ ) \qquad (4.5)$$
$$\boldsymbol{y} = \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{r}(t)) \qquad (4.6)$$

  - Discrete

$$\boldsymbol{x_k} = \boldsymbol{f}(\boldsymbol{x_{k-1}}, \boldsymbol{u_{k-1}}, \boldsymbol{q_{k-1}}) \qquad (4.7)$$
$$\boldsymbol{y_k} = \boldsymbol{g}(\boldsymbol{x_k}, \boldsymbol{u_{k-1}}, \boldsymbol{r_k}) \qquad (4.8)$$

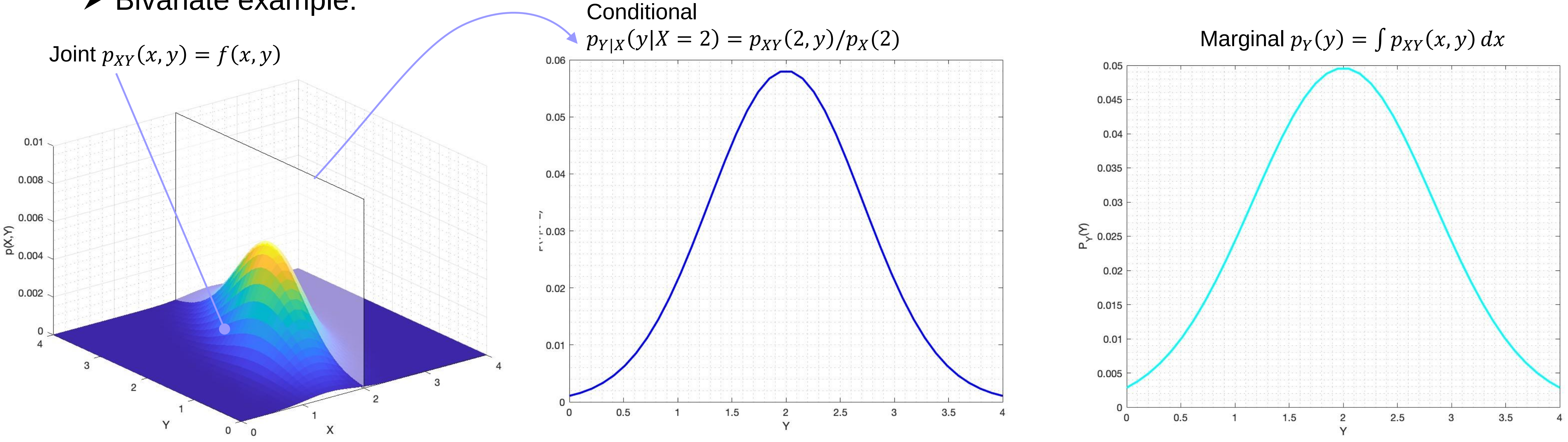
$\boldsymbol{x}$ = state vector, $\mathbb{R}^n$
$\boldsymbol{u}$ = input vector, $\mathbb{R}^p$
$\boldsymbol{y}$ = measurement vector, $\mathbb{R}^q$
$\boldsymbol{A}$ = State Matrix, $\mathbb{R}^{n\times n}$
$\boldsymbol{B}$ = Input Matrix, $\mathbb{R}^{n\times p}$
$\boldsymbol{H}$ = Output Matrix, $\mathbb{R}^{q\times n}$
$\boldsymbol{D}$ = Feedforward Matrix, $\mathbb{R}^{q\times p}$

$\boldsymbol{q}$ = process noise, $\mathbb{R}^n$
$\boldsymbol{r}$ = measurement noise, $\mathbb{R}^q$

Bayesian Filtering – Preliminaries

- Given a joint distribution $p_{XY}\,(\boldsymbol{x},\boldsymbol{y})$ of two random vectors $\boldsymbol{X}$ and $\boldsymbol{Y}$,
  - marginal distributions
    - $p_Y(\boldsymbol{y}) = \int p_{XY}(\boldsymbol{x},\boldsymbol{y})\, \boldsymbol{dx}$ (5.1)
    - $p_X(\boldsymbol{x}) = \int p_{XY}(\boldsymbol{x},\boldsymbol{y})\, \boldsymbol{dy}$ (5.2)
  - conditional distributions are given by
    - $p_{Y|X}\,(\boldsymbol{y}|\boldsymbol{x}) = p_{XY}(\boldsymbol{x},\boldsymbol{y})/p_X(\boldsymbol{x})$ (5.3)
    - $p_{X|Y}\,(\boldsymbol{x}|\boldsymbol{y}) = p_{XY}(\boldsymbol{x},\boldsymbol{y})/p_Y(\boldsymbol{y})$ (5.4)

  - Bivariate example:



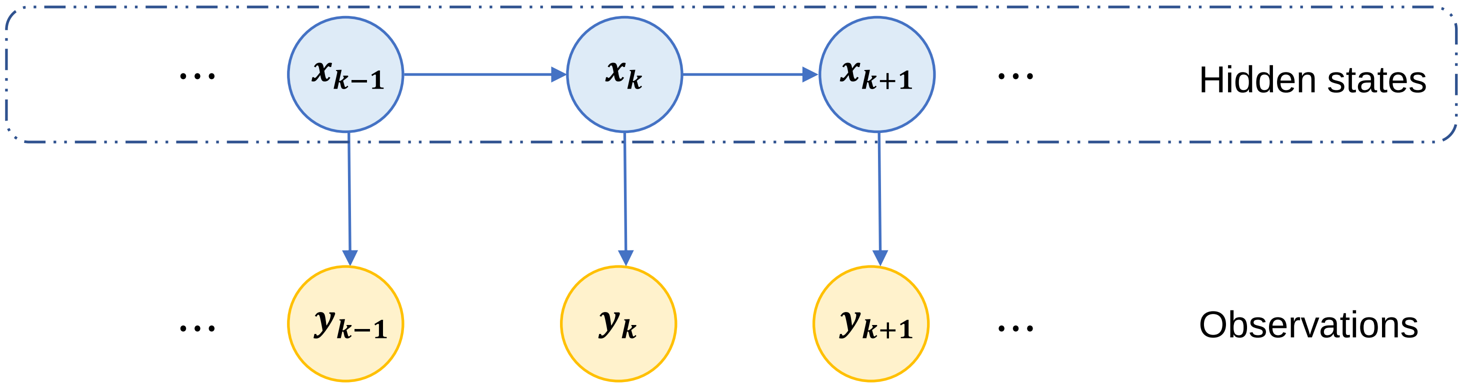
Bayesian Filtering – Markov and other assumptions

- Markov Model
  - Future independent of past given present (and vice versa)

$$p(\boldsymbol{x}_k|\boldsymbol{x}_{1:k-1}, \boldsymbol{y}_{1:k-1}) = p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) \tag{6.1}$$

- Measurements conditionally independent from previous states and measurements

$$p(\boldsymbol{y}_k|\boldsymbol{x}_{1:k}, \boldsymbol{y}_{1:k-1}) = p(\boldsymbol{y}_k|\boldsymbol{x}_k) \tag{6.2}$$



Bayesian Filtering

➢ State Estimation Problem

- Given a measurement sequence on the state of a dynamical system, $\boldsymbol{y_{1:k}}$, find the conditional probability distribution $p(\boldsymbol{x_k}|\boldsymbol{y_{1:k}})$, where $\boldsymbol{x_k}$ is the state at time *k.*

➢ Bayesian Filtering

- Converts state estimation to recursive problem to sequentially estimate $p(\boldsymbol{x_k}|\boldsymbol{y_{1:k}})$ assuming $p(\boldsymbol{x_{k-1}}|\boldsymbol{y_{1:k-1}})$ is known.
- Finds $p(\boldsymbol{x_k}|\boldsymbol{y_{1:k}})$ in two steps:

1. **Prediction**: Find conditional distribution $p(\boldsymbol{x_k}|\boldsymbol{y_{1:k-1}})$, where $\boldsymbol{x_k}$ is state at time *k*, and $\boldsymbol{y_{1:k-1}}$ is the sequence of observations up to the previous time step, $\boldsymbol{k-1}$
2. **Update**: Find conditional distribution $p(\boldsymbol{x_k}|\boldsymbol{y_{1:k}})$, where $\boldsymbol{x_k}$ is the state at time k and $\boldsymbol{y_{1:k}}$ is the sequence of observations up to the current time, $\boldsymbol{k}$

Bayesian Filtering – Prediction

- **Prediction step – Goal**: conditional distribution $p(\boldsymbol{x_k}|\boldsymbol{y_{1:k-1}})$.

- Given the distribution $p(\boldsymbol{x_{k-1}}|\boldsymbol{y_{1:k-1}})$
- From conditional (3.3)

$$p(\boldsymbol{x_k}|\boldsymbol{x_{k-1}},\boldsymbol{y_{1:k-1}}) = \frac{p(\boldsymbol{x_k},\boldsymbol{x_{k-1}}|\boldsymbol{y_{1:k-1}})}{p(\boldsymbol{x_{k-1}}|\boldsymbol{y_{1:k-1}})} \tag{8.1}$$

$$p(\boldsymbol{x_k},\boldsymbol{x_{k-1}}|\boldsymbol{y_{1:k-1}}) = p(\boldsymbol{x_k}|\boldsymbol{x_{k-1}},\boldsymbol{y_{1:k-1}})\, p(\boldsymbol{x_{k-1}}|\boldsymbol{y_{1:k-1}}) \tag{8.2}$$

- Integrating out $\boldsymbol{x_{k-1}}$ gives Chapman-Kolmogorov Equation:

$$p(\boldsymbol{x_k}|\boldsymbol{y_{1:k-1}}) = \int p(\boldsymbol{x_k}|\boldsymbol{x_{k-1}},\boldsymbol{y_{1:k-1}})\, p(\boldsymbol{x_{k-1}}|\boldsymbol{y_{1:k-1}})\, d\boldsymbol{x_{k-1}} \tag{8.3}$$

- Markov assumption gives:

$$\boxed{p(\boldsymbol{x_k}|\boldsymbol{y_{1:k-1}}) = \int p(\boldsymbol{x_k}|\boldsymbol{x_{k-1}})\, p(\boldsymbol{x_{k-1}}|\boldsymbol{y_{1:k-1}})\, d\boldsymbol{x_{k-1}}} \tag{8.4}$$

Bayesian Filtering – Update

- **Update step – Goal**: conditional distribution $p(\boldsymbol{x_k}|\boldsymbol{y_{1:k}})$
- From conditional (3.3)

$$p\left(\boldsymbol{x_k}|\boldsymbol{y_k},\boldsymbol{y_{1;k-1}}\right)=\frac{p(\boldsymbol{x_k},\boldsymbol{y_k}|\boldsymbol{y_{1:k-1}})}{p(\boldsymbol{y_k}|\boldsymbol{y_{1:k-1}})} \tag{9.1}$$

- From conditional (3.4) and measurement model (…)

$$p(\boldsymbol{y_k}|\boldsymbol{x_k},\boldsymbol{y_{1:k-1}})=\frac{p(\boldsymbol{x_k},\boldsymbol{y_k}|\boldsymbol{y_{1:k-1}})}{p(\boldsymbol{x_k}|\boldsymbol{y_{1:k-1}})} \tag{9.2}$$

- **Bayes Theorem**

$$p(\boldsymbol{x_k}|\boldsymbol{y_k},\boldsymbol{y_{1:k-1}})=\frac{p(\boldsymbol{y_k}|\boldsymbol{x_k},\boldsymbol{y_{1:k-1}})p(\boldsymbol{x_k}|\boldsymbol{y_{1:k-1}})}{Z_k} \tag{9.3}$$

- Recognizing $Z_k=\int p(\boldsymbol{y_k}|\boldsymbol{x_k},\boldsymbol{y_{1:k-1}})p(\boldsymbol{x_k}|\boldsymbol{y_{1:k-1}})d\boldsymbol{x_k}$

$$p(\boldsymbol{x_k}|\boldsymbol{y_{1:k}})=\frac{p(\boldsymbol{y_k}|\boldsymbol{x_k},\boldsymbol{y_{1:k-1}})p(\boldsymbol{x_k}|\boldsymbol{y_{1:k-1}})}{\int p(\boldsymbol{y_k}|\boldsymbol{x_k},\boldsymbol{y_{1:k-1}})p(\boldsymbol{x_k}|\boldsymbol{y_{1:k-1}})d\boldsymbol{x_k}} \tag{9.4}$$

- Conditional independence of measurements gives:

$$\boxed{p(\boldsymbol{x_k}|\boldsymbol{y_{1:k}})=\frac{p(\boldsymbol{y_k}|\boldsymbol{x_k})p(\boldsymbol{x_k}|\boldsymbol{y_{1:k-1}})}{\int p(\boldsymbol{y_k}|\boldsymbol{x_k})p(\boldsymbol{x_k}|\boldsymbol{y1:_{k-1}})d\boldsymbol{x_k}}} \tag{9.5}$$

Bayesian Filtering – Algorithm

- The goal of Bayesian Filter is to compute the posterior conditional distribution $p(\boldsymbol{x_k}|\boldsymbol{y_{1:k}})$

Algorithm

- for all time k
  - Start with posterior belief of system state at time $k-1$, $p(\boldsymbol{x_{k-1}}|\boldsymbol{y_{1:k-1}})$
  - Predict. Compute prior $p(\boldsymbol{x_k}|\boldsymbol{y_{1:k-1}})$
  - Get new measurement $\boldsymbol{y_k}$.
  - Update. Compute posterior $p(\boldsymbol{x_k}|\boldsymbol{y_{1:k}})$ incorporating measurement $\boldsymbol{y_k}$
  - Recursion. $p(\boldsymbol{x_k}|\boldsymbol{y_{1:k}})$ becomes posterior at $k-1$ for next time step k, $p(\boldsymbol{x_{k-1}}|\boldsymbol{y_{1:k-1}}) \leftarrow p(\boldsymbol{x_k}|\boldsymbol{y_{1:k}})$

Kalman Filter – special case

- Kalman Filter is a special case of the Bayesian Filter when:
  - Linear process, additive Gaussian noise
  - Linear measurement, additive Gaussian noise

Kalman Filter – Preliminaries

- We are interested in relationships between joint, marginal, conditional of a multivariate Gaussian distribution
- Let random vector $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)^T = \left((x_1, x_2, \cdots, x_m), (x_{m+1}, x_{m+2}, \cdots, x_n)\right)^T = (\boldsymbol{x}_1, \boldsymbol{x}_2)^T$
- Multivariate Gaussian distribution of $\boldsymbol{x}$ is given by:

$$p(\boldsymbol{x}_1, \boldsymbol{x}_2) = \frac{1}{\sqrt{(2\pi)^n |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}\left(\begin{bmatrix}\boldsymbol{x}_1\\\boldsymbol{x}_2\end{bmatrix} - \begin{bmatrix}\boldsymbol{\mu}_1\\\boldsymbol{\mu}_2\end{bmatrix}\right)^T \begin{bmatrix}\boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12}\\\boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22}\end{bmatrix}^{-1} \left(\begin{bmatrix}\boldsymbol{x}_1\\\boldsymbol{x}_2\end{bmatrix} - \begin{bmatrix}\boldsymbol{\mu}_1\\\boldsymbol{\mu}_2\end{bmatrix}\right)\right) \sim \mathcal{N}\left(\begin{pmatrix}\boldsymbol{\mu}_1\\\boldsymbol{\mu}_2\end{pmatrix}, \begin{pmatrix}\boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12}\\\boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22}\end{pmatrix}\right) \qquad (12.1)$$

where the vectors $\boldsymbol{\mu_i}$ and matrices $\boldsymbol{\Sigma_i}$ are the mean and covariance for each $\boldsymbol{x}_i$

- The marginals and conditionals are given by

$$p(\boldsymbol{x}_1) \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_{11}) \qquad (12.2)$$

$$p(\boldsymbol{x}_2) \sim \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_{22}) \qquad (12.3)$$

$$p(\boldsymbol{x}_1 | \boldsymbol{x}_2) \sim \mathcal{N}\left(\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}{\boldsymbol{\Sigma}_{22}}^{-1}(\boldsymbol{x}_2 - \boldsymbol{\mu}_2),\ \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}{\boldsymbol{\Sigma}_{22}}^{-1}\boldsymbol{\Sigma}_{21}\right) \qquad (12.4)$$

$$p(\boldsymbol{x}_2 | \boldsymbol{x}_1) \sim \mathcal{N}\left(\boldsymbol{\mu}_2 + \boldsymbol{\Sigma}_{21}{\boldsymbol{\Sigma}_{11}}^{-1}(\boldsymbol{x}_1 - \boldsymbol{\mu}_1),\ \boldsymbol{\Sigma}_{22} - \boldsymbol{\Sigma}_{21}{\boldsymbol{\Sigma}_{11}}^{-1}\boldsymbol{\Sigma}_{12}\right) \qquad (12.5)$$

- (derivations for 12.2 – 12.5 on slides 34-37)

Kalman Filter – Prediction derivation

- Assume that the previous step of the Bayesian filter is Gaussian:

$$p(\boldsymbol{x}_{k-1}|\boldsymbol{y}_{k-1}) \sim \mathcal{N}\left(\boldsymbol{\mu}_{k-1,k-1}, \boldsymbol{\Sigma}_{k-1,k-1}\right) \tag{13.1}$$

- Assume a Gaussian driven process

$$\boldsymbol{x_k} = \mathbf{A}\boldsymbol{x_{k-1}} + \boldsymbol{q_{k-1}} \tag{13.2}$$

with process noise centered at $\boldsymbol{\mu_q} = \mathbf{0}$

$$\boldsymbol{q_{k-1}} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{\boldsymbol{k-1}}) \tag{13.3}$$

then the conditional distribution $\boldsymbol{x}_k$ conditioned on $\boldsymbol{x}_{k-1}$ is given by:

$$p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) \sim \mathcal{N}(\mathbf{A}\boldsymbol{x_{k-1}}, \mathbf{Q}_{\boldsymbol{k-1}}) \tag{13.4}$$

- Then from (12.2, 12.5, 13.1, 13.4) the joint distribution $p(\boldsymbol{x}_{k-1}, \boldsymbol{x}_k|\boldsymbol{y}_{\mathbf{1}:k-1})$ is also Gaussian and is given by:

$$p(\boldsymbol{x}_{k-1}, \boldsymbol{x}_k|\boldsymbol{y}_{1:k-1}) = \sim\mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu}_{k-1,k-1} \\ \mathbf{A}\boldsymbol{\mu}_{k-1,k-1} \end{pmatrix}, \begin{pmatrix} \boldsymbol{\Sigma}_{k-1,k-1} & \boldsymbol{\Sigma}_{k-1,k-1}\mathbf{A}^T \\ \mathbf{A}\boldsymbol{\Sigma}_{k-1,k-1} & \mathbf{Q}_{\boldsymbol{k-1}} + \mathbf{A}\boldsymbol{\Sigma}_{k-1,k-1}\mathbf{A}^T \end{pmatrix}\right) \tag{13.5}$$

- And from (12.3) the conditional distribution $p(\boldsymbol{x}_k|\boldsymbol{y}_{\mathbf{1}:k-1})$ is given by:

$$p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1}) \sim \mathcal{N}\left(\mathbf{A}\boldsymbol{\mu}_{k-1,k-1}, \mathbf{A}\boldsymbol{\Sigma}_{k-1,k-1}\mathbf{A}^T + \mathbf{Q}_{k-1}\right) \triangleq \mathcal{N}\left(\boldsymbol{\mu}_{k,k-1}, \boldsymbol{\Sigma}_{k,k-1}\right) \tag{13.6}$$

$$\boxed{\begin{aligned} \boldsymbol{\mu}_{k,k-1} &= \mathbf{A}\boldsymbol{\mu}_{k-1,k-1} \\ \boldsymbol{\Sigma}_{k,k-1} &= \mathbf{A}\boldsymbol{\Sigma}_{k-1,k-1}\mathbf{A}^T + \mathbf{Q}_{k-1} \end{aligned}}$$

- (derivations on slides 40-42)

- From the previous prediction step, the distribution is given by

$$p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1}) \sim \mathcal{N}\left(\boldsymbol{\mu}_{k,k-1}, \boldsymbol{\Sigma}_{k,k-1}\right) \quad (14.1)$$

- Assume a Gaussian measurement model

$$\boldsymbol{y_k} = \mathbf{H}\boldsymbol{x_k} + \boldsymbol{r_k} \quad (14.2)$$

with noise centered at $\boldsymbol{\mu_r} = \mathbf{0}$

$$\boldsymbol{r_k} \sim \mathcal{N}(\mathbf{0}, \mathbf{R_k}) \quad (14.3)$$

then the conditional distribution of $\boldsymbol{y}_k$ conditioned on $\boldsymbol{x}_k$ is given by:

$$p(\boldsymbol{y}_k|\boldsymbol{x}_k) \sim \mathcal{N}(\mathbf{H}\boldsymbol{x_k}, \mathbf{R_k}) \quad (14.4)$$

- Then from (12.2, 12.5, 14.1, 14.4) the joint distribution $p(\boldsymbol{x}_k, \boldsymbol{y}_{\mathbf{1}:k})$ is also Gaussian and is given by:

$$p(\boldsymbol{x}_k, \boldsymbol{y}_k|\boldsymbol{y}_{1:k-\mathbf{1}}) = \sim\mathcal{N}\left(\begin{pmatrix}\boldsymbol{\mu}_{k,k-1}\\ \mathbf{H}\boldsymbol{\mu}_{k,k-1}\end{pmatrix}, \begin{pmatrix}\boldsymbol{\Sigma}_{k,k-1} & \boldsymbol{\Sigma}_{k,k-1}\mathbf{H}^T\\ \mathbf{H}\boldsymbol{\Sigma}_{k,k-1} & \mathbf{R_k} + \mathbf{H}\boldsymbol{\Sigma}_{k,k-1}\mathbf{H}^T\end{pmatrix}\right) \quad (14.5)$$

- And from (12.3) the conditional distribution $p(\boldsymbol{x}_k|\boldsymbol{y}_{\mathbf{1}:k-1})$ is given by:

$$p(\boldsymbol{x}_k|\boldsymbol{y}_k) \sim \mathcal{N}\left(\boldsymbol{\mu}_{k,k-1} + \boldsymbol{\Sigma}_{k,k-1}\mathbf{H}^T\left(\mathbf{R}_k + \mathbf{H}\boldsymbol{\Sigma}_{k,k-1}\mathbf{H}^T\right)^{-\mathbf{1}}\left(\boldsymbol{y}_k - \mathbf{H}\boldsymbol{\mu}_{k,k-1}\right),\ \ \boldsymbol{\Sigma}_{k,k-1} - \boldsymbol{\Sigma}_{k,k-1}\mathbf{H}^T\left(\mathbf{R}_k + \mathbf{H}\boldsymbol{\Sigma}_{k,k-1}\mathbf{H}^T\right)^{-\mathbf{1}}\mathbf{H}\boldsymbol{\Sigma}_{k,k-1}\right) \quad (14.6)$$

$$\boxed{\begin{aligned}\boldsymbol{\mu}_{k,k} &= \boldsymbol{\mu}_{k,k-1} + \boldsymbol{\Sigma}_{k,k-1}\mathbf{H}^T\left(\mathbf{R}_k + \mathbf{H}\boldsymbol{\Sigma}_{k,k-1}\mathbf{H}^T\right)^{-\mathbf{1}}\left(\boldsymbol{y}_k - \mathbf{H}\boldsymbol{\mu}_{k,k-1}\right) \quad (14.7)\\ \boldsymbol{\Sigma}_{k,k} &= \boldsymbol{\Sigma}_{k,k-1} - \boldsymbol{\Sigma}_{k,k-1}\mathbf{H}^T\left(\mathbf{R}_k + \mathbf{H}\boldsymbol{\Sigma}_{k,k-1}\mathbf{H}^T\right)^{-\mathbf{1}}\mathbf{H}\boldsymbol{\Sigma}_{k,k-1} \quad (14.8)\end{aligned}}$$

- (derivations on slides 43-45)

Kalman Filter – Algorithm

➢ The goal of Kalman Filter is to compute the expectation and covariance of the conditional distribution $p(\boldsymbol{x_k}|\boldsymbol{y_{1:k}})$

Algorithm

- for all time k
  - Start with posterior belief of system state at time $k-1$, $\boldsymbol{\mu}_{k-1,k-1}$
  - Predict. Compute expectation and covariance of $p(\boldsymbol{x_k}|\boldsymbol{y_{1:k-1}})$
    - $\boldsymbol{\mu}_{k,k-1} = \mathbf{A}\boldsymbol{\mu}_{k-1,k-1}$
    - $\boldsymbol{\Sigma}_{k,k-1} = \mathbf{A}\boldsymbol{\Sigma}_{k-1,k-1}\mathbf{A}^T + \mathbf{Q}_{k-1}$
  - Get new measurement $\boldsymbol{y_k}$
  - Update. Compute expectation and covariance of $p(\boldsymbol{x_k}|\boldsymbol{y_k})$ incorporating measurement $\boldsymbol{y_k}$
    - $\boldsymbol{\mu}_{k,k} = \boldsymbol{\mu}_{k,k-1} + \boldsymbol{\Sigma}_{k,k-1}\mathbf{H}^T\left(\mathbf{R}_k + \mathbf{H}\boldsymbol{\Sigma}_{k,k-1}\mathbf{H}^T\right)^{-\mathbf{1}}\left(\boldsymbol{y}_k - \mathbf{H}\boldsymbol{\mu}_{k,k-1}\right)$
    - $\boldsymbol{\Sigma}_{k,k} = \boldsymbol{\Sigma}_{k,k-1} - \boldsymbol{\Sigma}_{k,k-1}\mathbf{H}^T\left(\mathbf{R}_k + \mathbf{H}\boldsymbol{\Sigma}_{k,k-1}\mathbf{H}^T\right)^{-\mathbf{1}}\mathbf{H}\boldsymbol{\Sigma}_{k,k-1}$
  - Recursion. $p(\boldsymbol{x_k}|\boldsymbol{y_{1:k}})$ becomes posterior at $k-1$ for next time step k, i.e. $p(\boldsymbol{x_{k-1}}|\boldsymbol{y_{1:k-1}}) \leftarrow p(\boldsymbol{x_k}|\boldsymbol{y_{1:k}})$
    - $\boldsymbol{\mu}_{k-1,k-1} \leftarrow \boldsymbol{\mu}_{k,k}$
    - $\boldsymbol{\Sigma}_{k-1,k-1} \leftarrow \boldsymbol{\Sigma}_{k,k}$

Extended Kalman Filter

- Extended Kalman Filter for non-linear dynamics
- EKF is the same as KF, but linearizes the non-linear dynamics about the belief at each time step
- State matrix is linearized at each time step

$$\mathbf{f}(\boldsymbol{x}) = [f_1 \quad \cdots \quad f_n]^T \approx \mathbf{f}(\overline{\boldsymbol{x}}) + \mathbb{J}(\mathbf{f})\,\boldsymbol{\delta x} \tag{16.1}$$

- **Continuous**

$$\begin{bmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_n \end{bmatrix} \approx \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}_{\boldsymbol{x}=\overline{\boldsymbol{x}}} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \tag{16.2}$$

$$\dot{\boldsymbol{x}}_{\boldsymbol{k}}(t) = \widetilde{\boldsymbol{A}}\boldsymbol{x}_{\boldsymbol{k}}(t) + \boldsymbol{q}(t) \tag{16.3}$$

- **Discrete**

$$\dot{\boldsymbol{x}}(t) = \frac{\boldsymbol{x}_{\boldsymbol{k}} - \boldsymbol{x}_{\boldsymbol{k-1}}}{\boldsymbol{h}_{\boldsymbol{k}}} = \widetilde{\boldsymbol{A}}\boldsymbol{x}_{\boldsymbol{k-1}} \tag{16.4}$$

$$\boldsymbol{x}_{\boldsymbol{k}} = \widetilde{\boldsymbol{A}}\boldsymbol{h}_{\boldsymbol{k}}\boldsymbol{x}_{\boldsymbol{k-1}} + \boldsymbol{x}_{\boldsymbol{k-1}} = \left(\widetilde{\boldsymbol{A}}\boldsymbol{h}_{\boldsymbol{k}} + \mathbf{I}\right)\boldsymbol{x}_{\boldsymbol{k-1}} = \widehat{\boldsymbol{A}}\boldsymbol{x}_{\boldsymbol{k-1}} \tag{16.5}$$

$$\boldsymbol{x}_{\boldsymbol{k}} = \widehat{\boldsymbol{A}}\boldsymbol{x}_{\boldsymbol{k-1}} + \boldsymbol{q}_{\boldsymbol{k-1}} \tag{16.6}$$

- EKF algorithm and equations are identical to KF
  - Recursively Predict, Measure, Update

Ensemble Kalman Filter

- Monte Carlo Method
- Ensemble of n random variables that are possible realizations of the state
- Standard Kalman Filter equations applied to ensemble
- Not restricted to linear systems. Propagates each particle with non-linear dynamics
- Not restricted to Gaussian models, since Monte Carlo can be applied to any arbitrary distribution

Algorithm

- Randomly initialize state for n members $\boldsymbol{x}^{(i)}{}_{\mathbf{0}}, \ldots, \boldsymbol{x}^{(n)}{}_{\mathbf{0}}$
- **for** all time k do
  - **Prediction**
    - **for** $i = 1{:}n$
      - Sample $\boldsymbol{q}^{(i)}{}_{\boldsymbol{k-1}} \sim \boldsymbol{\mathcal{P}}_{\mathbf{Q}}$ then compute

$$\boldsymbol{x}^{(i)}{}_{\boldsymbol{k,k-1}} = \boldsymbol{f}\left(\boldsymbol{x}^{(i)}{}_{\boldsymbol{k-1,k-1}}, \boldsymbol{q}^{(i)}{}_{\boldsymbol{k-1}}\right)$$

$$\widehat{\boldsymbol{\Sigma}}_{k,k-1} = \frac{1}{n}\sum_{i=1}^{n}\left(\boldsymbol{x}^{(i)}{}_{\boldsymbol{k,k-1}} - \widehat{\boldsymbol{\mu}}_{\boldsymbol{k,k-1}}\right)^{T}\left(\boldsymbol{x}^{(i)}{}_{\boldsymbol{k,k-1}} - \widehat{\boldsymbol{\mu}}_{\boldsymbol{k,k-1}}\right)$$

  - **Update**
    - **for** $i = 1{:}n$

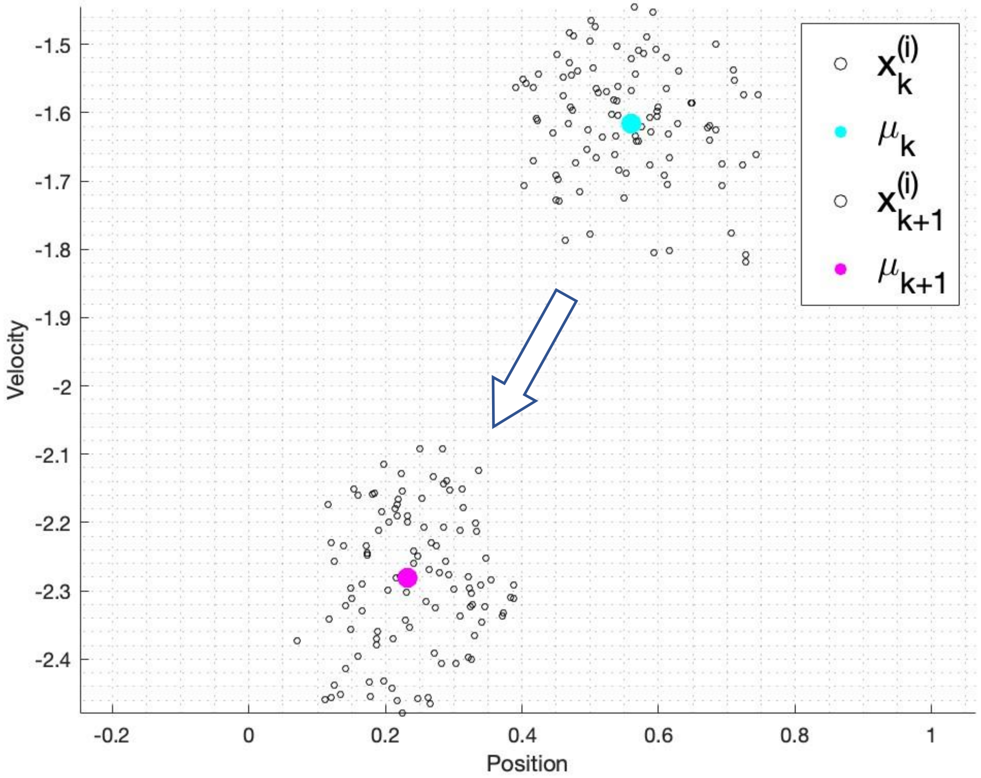
$$\boldsymbol{x}^{(i)}{}_{\boldsymbol{k,k}} = \boldsymbol{x}^{(i)}{}_{k,k-1} + \widehat{\boldsymbol{\Sigma}}_{k,k-1}\mathbf{H}^{T}\left(\mathbf{R}_{\mathbf{k}} + \mathbf{H}\widehat{\boldsymbol{\Sigma}}_{k,k-1}\mathbf{H}^{T}\right)^{\mathbf{-1}}\left(\boldsymbol{y}_{\boldsymbol{k}} - \boldsymbol{H}\boldsymbol{x}^{(i)}{}_{\boldsymbol{k,k-1}}\right)$$

$$\widehat{\boldsymbol{\mu}}_{\boldsymbol{k,k}} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}^{(i)}{}_{\boldsymbol{k,k}}, \qquad \widehat{\boldsymbol{\Sigma}}_{k,k} = \frac{1}{n}\sum_{i=1}^{n}\left(\boldsymbol{x}^{(i)}{}_{\boldsymbol{k,k}} - \widehat{\boldsymbol{\mu}}_{\boldsymbol{k,k}}\right)^{T}\left(\boldsymbol{x}^{(i)}{}_{\boldsymbol{k,k}} - \widehat{\boldsymbol{\mu}}_{\boldsymbol{k,k}}\right)$$

  - **Recursion,** $\boldsymbol{x}^{(i)}{}_{\boldsymbol{k-1,k-1}} \leftarrow \boldsymbol{x}^{(i)}{}_{\boldsymbol{k,k}}$

Particle Filter

- Monte Carlo Method
- Ensemble of n ‘particles’, random variables that are possible realizations of the state
- Sample approximation of the posterior distribution
- Not restricted to linear systems. Propagates each particle with non-linear dynamics
- Not restricted to Gaussian models, since Monte Carlo can be applied to any arbitrary distribution

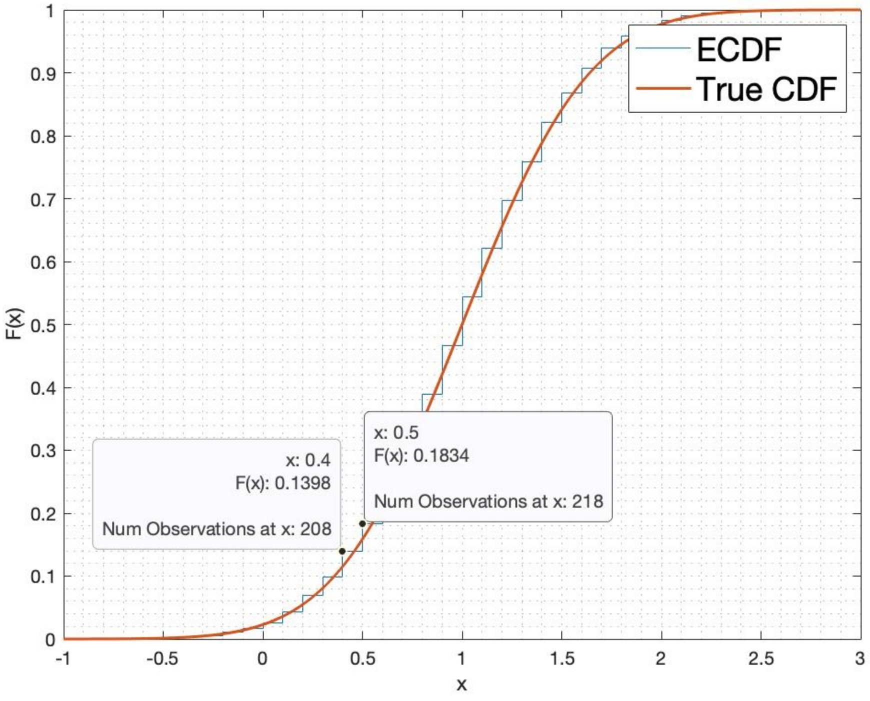


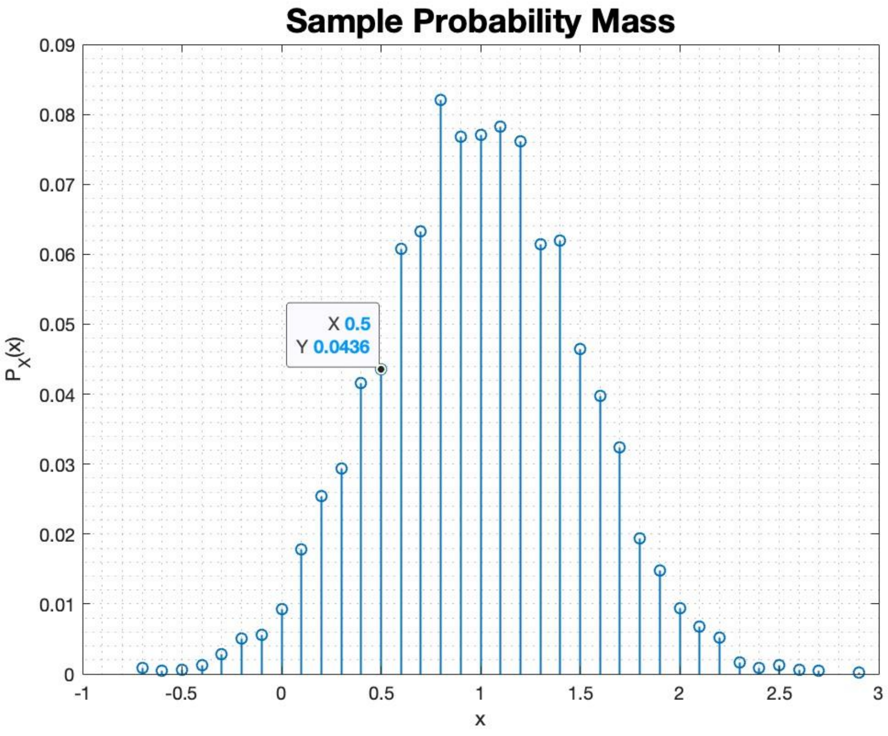
Particle Filter - preliminaries

- A sample distribution can be represented by a sum of weighted samples, where the weights correspond to the step sizes in the ECDF – that is, the fraction of samples from the sample set at a particular value of x.
- A sample approximation of a probability mass function can be given by

$$\hat{p}_X(x) = \sum_{i=1}^{n} \omega^{(i)} \delta(x - x^{(i)}) \quad (20.4)$$

- Example:





Particle Filter – Prediction derivation

- Bayes Filter prediction equation (8.4)

$$p(\boldsymbol{x_k}|\boldsymbol{y_{1:k-1}}) = \int p(\boldsymbol{x_k}|\boldsymbol{x_{k-1}})\, p(\boldsymbol{x_{k-1}}|\boldsymbol{y_{1:k-1}})\, d\boldsymbol{x_{k-1}} \tag{21.1}$$

- From (20.4), let us approximate the posterior at $k-1$:

$$\hat{p}(\boldsymbol{x_{k-1}}|\boldsymbol{y_{1:k-1}}) = \sum_{i=1}^{n} \omega_{k-1}^{(i)} \delta(\boldsymbol{x}_{k-1} - \boldsymbol{x}_{k-1}^{(i)}) \tag{21.2}$$

- From (21.1) and (21.2),

$$\hat{p}(\boldsymbol{x_k}|\boldsymbol{y_{1:k-1}}) = \sum_{i=1}^{n} \omega_{k-1}^{(i)} \int p(\boldsymbol{x_k}|\boldsymbol{x_{k-1}}) \delta(\boldsymbol{x}_{k-1} - \boldsymbol{x}_{k-1}^{(i)})\, d\boldsymbol{x_{k-1}} \tag{21.3}$$

- A property of Dirac Delta function:

$$\int_{\Omega} g(\boldsymbol{x})\, \delta(\boldsymbol{x} - \boldsymbol{q}) d\boldsymbol{x} = g(\boldsymbol{q}) \tag{21.4}$$

- Using (21.4) in (21.3),

$$\hat{p}(\boldsymbol{x_k}|\boldsymbol{y_{1:k-1}}) = \sum_{i=1}^{n} \omega_{k-1}^{(i)}\, p\left(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)}\right) \tag{21.5}$$

- This can also be written in terms of $\boldsymbol{x}_k$

$$\boxed{\hat{p}(\boldsymbol{x_k}|\boldsymbol{y_{1:k-1}}) = \sum_{i=1}^{n} \omega_{k-1}^{(i)}\, \delta(\boldsymbol{x_k} - \boldsymbol{x_k}^{(i)})} \tag{21.6}$$

Particle Filter – Update derivation

- Bayes Filter update equation (9.5)

$$p(\boldsymbol{x_k}|\boldsymbol{y_{1:k}}) = \frac{p(\boldsymbol{y_k}|\boldsymbol{x_k})p(\boldsymbol{x_k}|\boldsymbol{y_{1:k-1}})}{\int p(\boldsymbol{y_k}|\boldsymbol{x_k})p(\boldsymbol{x_k}|\boldsymbol{y_{k-1}})d\boldsymbol{x_k}} \quad (22.1)$$

- Substituting (21.6) into (22.1)

$$\hat{p}(\boldsymbol{x_k}|\boldsymbol{y_{1:k}}) = \frac{\sum_{i=1}^{n} \omega_{k-1}^{(i)}\, p\left(\boldsymbol{y_k}|\boldsymbol{x_k^{(i)}}\right)\delta(\boldsymbol{x_k} - \boldsymbol{x_k^{(i)}})}{\sum_{i=1}^{n} \omega_{k-1}^{(i)} \int p(\boldsymbol{y_k}|\boldsymbol{x_k})\delta(\boldsymbol{x_k} - \boldsymbol{x_k^{(i)}})d\boldsymbol{x_k}} = \sum_{i=1}^{n} \omega_k^{(i)}\, \delta(\boldsymbol{x_k} - \boldsymbol{x_k^{(i)}}) \quad (22.2)$$

- Then

$$\omega_k^{(i)} = \frac{\omega_{k-1}^{(i)} p\left(\boldsymbol{y_k}|\boldsymbol{x_k^{(i)}}\right)}{\sum_{i=1}^{n} \omega_{k-1}^{(i)} \int p(\boldsymbol{y_k}|\boldsymbol{x_k})\delta(\boldsymbol{x_k} - \boldsymbol{x_k^{(i)}})d\boldsymbol{x_k}} \quad (22.3)$$

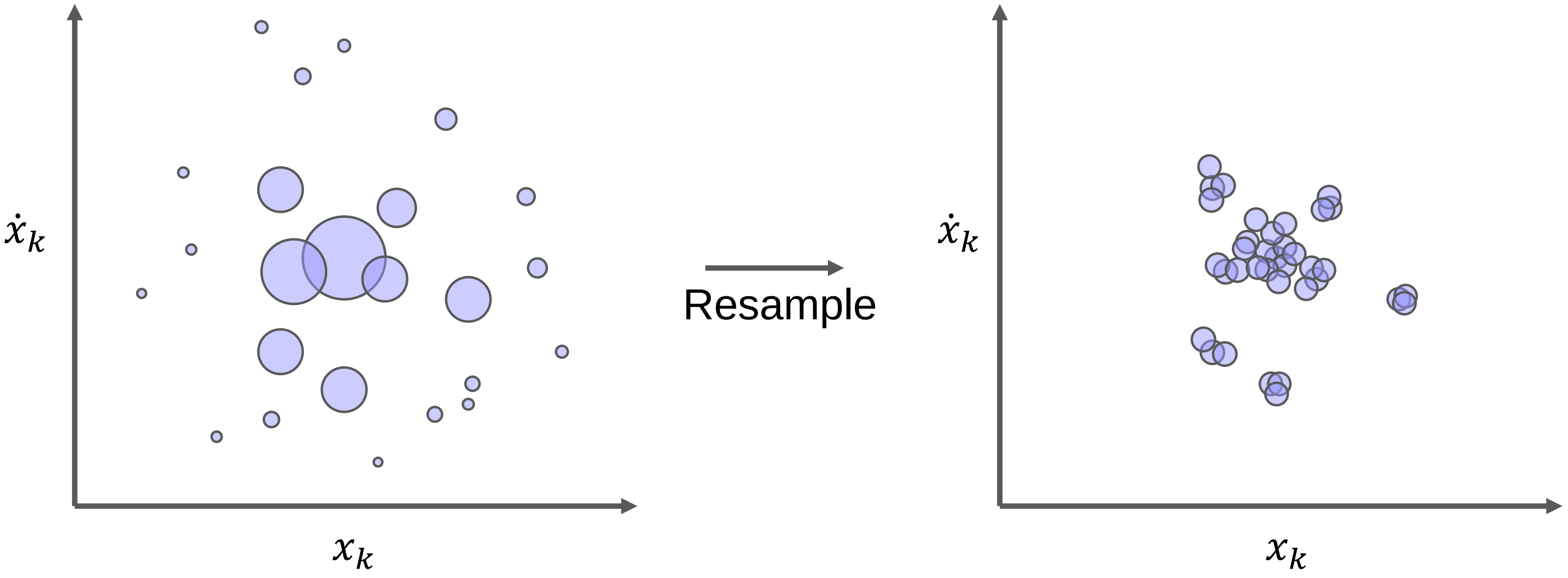
- From (20.4) into (22.4),

$$\boxed{\omega_k^{(i)} = \frac{\omega_{k-1}^{(i)} p\left(\boldsymbol{y_k}|\boldsymbol{x_k^{(i)}}\right)}{\sum_{i=1}^{n} \omega_{k-1}^{(i)}\, p\left(\boldsymbol{y_k}|\boldsymbol{x_k^{(i)}}\right)}} \quad (22.4)$$

$$\boxed{\hat{p}(\boldsymbol{x_k}|\boldsymbol{y_{1:k}}) = \sum_{i=1}^{n} \omega_k^{(i)}\, \delta(\boldsymbol{x_k} - \boldsymbol{x_k^{(i)}})} \quad (22.5)$$

Particle Filter – Resampling

- Algorithm so far is degenerate, since variance of weights increases with each time step
- Converges to a single particle with entire probability mass, $\omega^{(i)} = 1$, and all other particles are zero
- Resampling step – replace old set of particles with new set of particles, with number of each particular sample proportional to weight
  - Treat the posterior as the distribution of drawing a sample from the last set of particles



Particle Filter - algorithm

Algorithm

- Randomly initialize state for n particles. Initialize weights for n particles:

$$\{\omega^{(i)}{}_{\mathbf{0}}, \boldsymbol{x}^{(i)}{}_{\mathbf{0}} \,|i = 1, \dots, n\}, \qquad \omega^{(i)}{}_{\mathbf{0}} = 1/n$$

- for all time k
  - for $i = 1{:}n$
    - **Prediction**: sample $\boldsymbol{x}^{(i)}{}_{\boldsymbol{k},\boldsymbol{k-1}} \sim p\left(\boldsymbol{x}_k \middle| \boldsymbol{x}^{(i)}{}_{k-1}\right)$
  - Measurement: $\boldsymbol{y_k}$
  - for $i = 1{:}n$
    - **Update**: update weights, $\omega_k^{(i)} = \dfrac{\omega_{k-1}^{(i)} p\left(\boldsymbol{y_k} | \boldsymbol{x_k^{(i)}}\right)}{\sum_{i=1}^{n} \omega_{k-1}^{(i)} p\left(\boldsymbol{y_k} | \boldsymbol{x_k^{(i)}}\right)}$

update belief: $\mathbb{E}(\boldsymbol{x_k} | \boldsymbol{y_{1:k}}) \approx \sum_{i=1}^{n} \omega^{(i)} \overline{\boldsymbol{x}}^{(i)}$

  - Resample $\{\boldsymbol{x}^{(j)}{}_{\boldsymbol{k}} \,|j = 1, \dots, n\}$

$$\{\boldsymbol{x}^{(i)}{}_{\boldsymbol{k}} \,|i = 1, \dots, n\} \leftarrow \{\boldsymbol{x}^{(j)}{}_{\boldsymbol{k}} \,|j = 1, \dots, n\}$$

$$\omega^{(i)}{}_{\boldsymbol{k-1}} \leftarrow \omega^{(i)}{}_{\boldsymbol{k}}$$

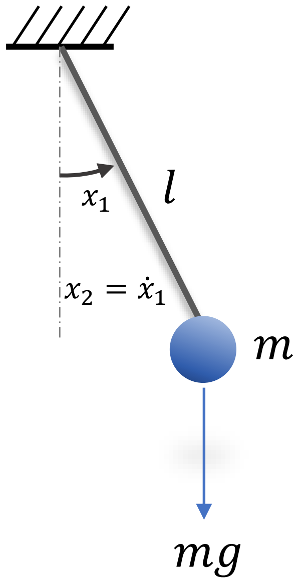
Example: Pendulum

- Non-linear Process Model

$$\dot{x} = \dot{x}_1 + q_1$$
$$\ddot{x} = \dot{x}_2 = -\frac{g}{l}\sin(x_1) + q_2$$

- Measurement Model

$$y_1 = x_1 + r_1$$
$$y_2 = x_2 + r_2$$

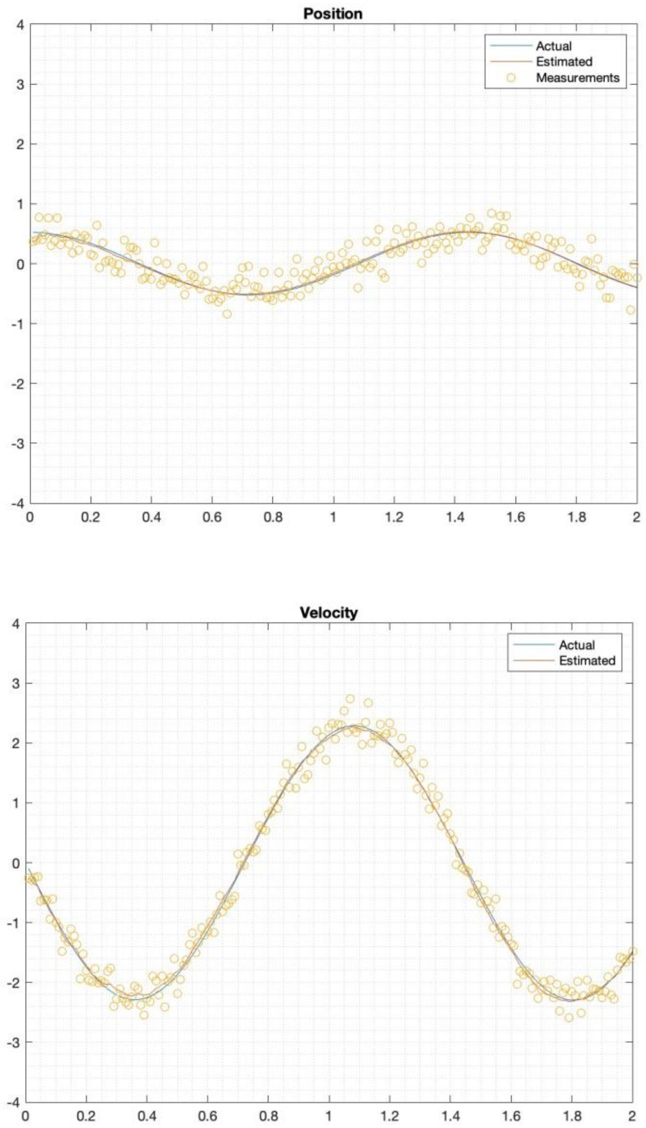


Advantages/Limitations

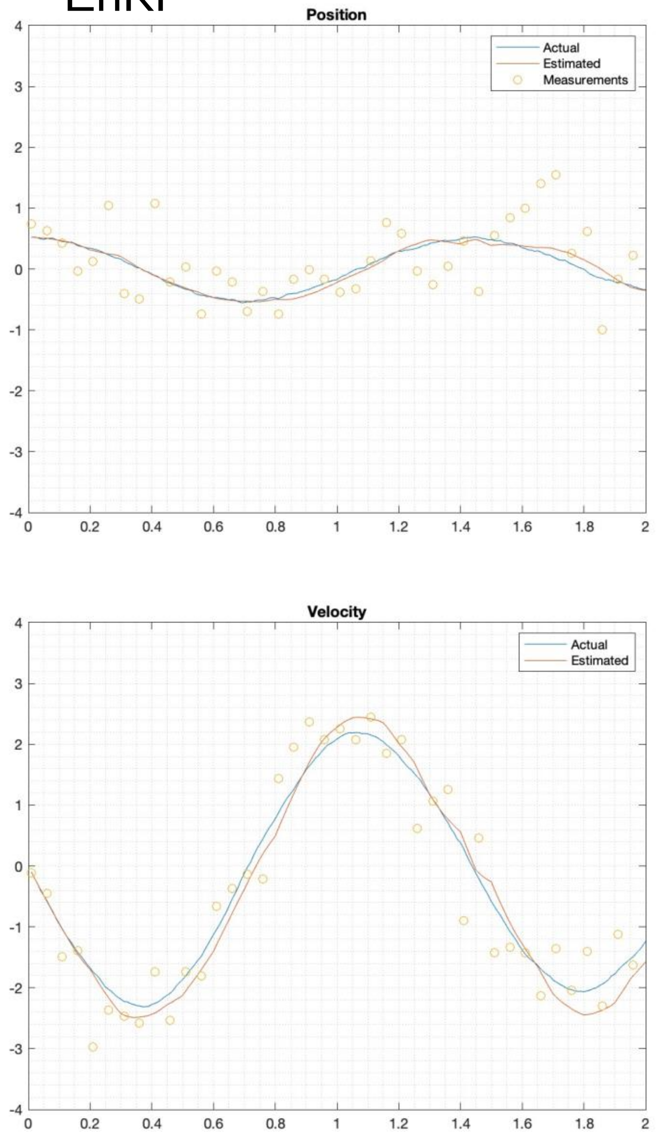
Approximation	Model	Noise	Solution	Computational Complexity
KF	Linear	Gaussian	Explicit	Scales with dimension (dominated by inverse matrix computation)
EKF	Non-linear	Gaussian	Explicit	Scales dimension (dominated by inverse matrix computation)
EnKF	Non-linear	Arbitrary	Sample Approx.	Scales with ensemble size.
PF	Non-linear	Arbitrary	Sample Approx.	Scales with ensemble size. No matrix inversions

Example: Pendulum

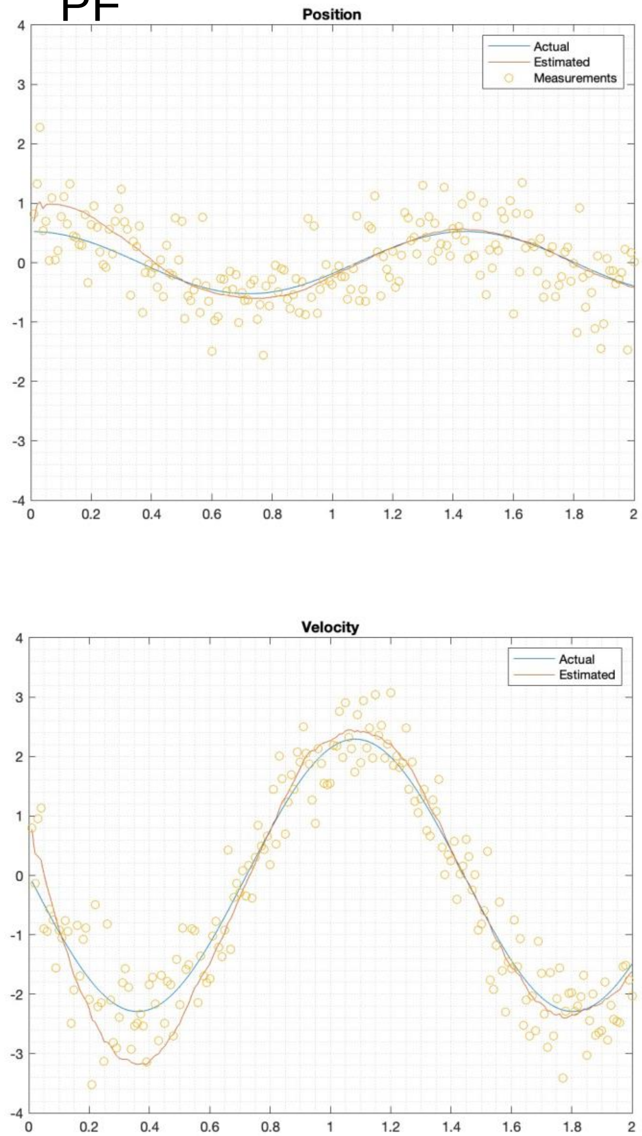
EKF



EnKF



PF



Deep Probabilistic Learning

- Deep probabilistic learning algorithms exist that approximate conditional distributions efficiently, i.e. CGAN and Diffusion Networks. The aim is to use these to learn/approximate the Bayes Filter update conditional distribution from which we can sample.
- Future work
  - Further develop CGAN or Diffusion Networks to approximate the Bayes Filter and investigate potential improvements to the state of the art
    - Non-linear dynamics
    - Arbitrary distributions
    - Incomplete sensing
    - Computational cost
  - Identify a particular robotic application for the problems of localization and pose estimation where CGAN may improve performance over state of the art

Citations

1. Katzfuss, Matthias, Stroud Jonathan R., Wikle Christopher K. "Understanding the Ensemble Kalman Filter" *The American Statistician 2016, Vol. 70, No. 4, 350-357*
2. Särkkä, Simo. "Lecture 3: Bayesian Filtering Equations and Kalman Filter"
3. Särkkä, Simo. "Lecture 6: Particle Filtering"
4. Rawlings, James B., "Model Predictive Control: Theory, Computation, and Design" (1st ed.) ISBN 978-0975937754
5. Jordan, M. "The Multivariate Gaussian" (PDF) PDF
6. Escudero, M. C., "Towards SMC: Using the Dirac-delta function in Sampling and Sequential Monte Carlo" PDF
7. Javier Murgoitio-Esandi, Deep Ray, Agnimitra Dasgupta, Assad A. Oberai "A Novel Conditional Wasserstein Generative Adversarial Network for Inverse Problems"

Back-up Slides

Multivariate Gaussian Marginalization

- Let random vector $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)^T = \left((x_1, x_2, \cdots, x_m), (x_{m+1}, x_{m+2}, \cdots, x_n)\right)^T = (\boldsymbol{x}_1, \boldsymbol{x}_2)^T$ (31.#)
- Multivariate Gaussian distribution of $\boldsymbol{x}$ is given by:

$$p(\boldsymbol{x}_1, \boldsymbol{x}_2) = \frac{1}{\sqrt{(2\pi)^n |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}\left(\begin{bmatrix}\boldsymbol{x}_1\\\boldsymbol{x}_2\end{bmatrix} - \begin{bmatrix}\boldsymbol{\mu}_1\\\boldsymbol{\mu}_2\end{bmatrix}\right)^T \begin{bmatrix}\boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12}\\\boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22}\end{bmatrix}^{-1} \left(\begin{bmatrix}\boldsymbol{x}_1\\\boldsymbol{x}_2\end{bmatrix} - \begin{bmatrix}\boldsymbol{\mu}_1\\\boldsymbol{\mu}_2\end{bmatrix}\right)\right) \sim \mathcal{N}\left(\begin{pmatrix}\boldsymbol{\mu}_1\\\boldsymbol{\mu}_2\end{pmatrix}, \begin{pmatrix}\boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12}\\\boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22}\end{pmatrix}\right) \quad (31.\#)$$

$$p(\boldsymbol{x}_1, \boldsymbol{x}_2) \propto \exp\left(-\frac{1}{2}\left(\boldsymbol{x}_1 - \boldsymbol{\mu}_1 - \boldsymbol{\Sigma}_{12}{\boldsymbol{\Sigma}_{22}}^{-1}(\boldsymbol{x}_2 - \boldsymbol{\mu}_2)\right)^T (\Sigma/\Sigma_{22})^{-1} \left(\boldsymbol{x}_1 - \boldsymbol{\mu}_1 - \boldsymbol{\Sigma}_{12}{\boldsymbol{\Sigma}_{22}}^{-1}(\boldsymbol{x}_2 - \boldsymbol{\mu}_2)\right)\right) \exp\left(-\frac{1}{2}(\boldsymbol{x}_2 - \boldsymbol{\mu}_2)^T{\boldsymbol{\Sigma}_{22}}^{-1}(\boldsymbol{x}_2 - \boldsymbol{\mu}_2)\right) \quad (31.\#)$$

$$p(\boldsymbol{x}_2) \propto \int p(\boldsymbol{x}_1, \boldsymbol{x}_2)\, d\boldsymbol{x}_1 \quad (31.\#)$$

$$\propto \exp\left(-\frac{1}{2}(\boldsymbol{x}_2 - \boldsymbol{\mu}_2)^T{\boldsymbol{\Sigma}_{22}}^{-1}(\boldsymbol{x}_2 - \boldsymbol{\mu}_2)\right) \int \exp\left(-\frac{1}{2}\left(\boldsymbol{x}_1 - \boldsymbol{\mu}_1 - \boldsymbol{\Sigma}_{12}{\boldsymbol{\Sigma}_{22}}^{-1}(\boldsymbol{x}_2 - \boldsymbol{\mu}_2)\right)^T (\Sigma/\Sigma_{22})^{-1} \left(\boldsymbol{x}_1 - \boldsymbol{\mu}_1 - \boldsymbol{\Sigma}_{12}{\boldsymbol{\Sigma}_{22}}^{-1}(\boldsymbol{x}_2 - \boldsymbol{\mu}_2)\right)\right) d\boldsymbol{x}_1 \quad (31.\#)$$

$$\boxed{p(\boldsymbol{x}_2) \propto \exp\left(-\frac{1}{2}(\boldsymbol{x}_2 - \boldsymbol{\mu}_2)^T{\boldsymbol{\Sigma}_{22}}^{-1}(\boldsymbol{x}_2 - \boldsymbol{\mu}_2)\right)} \quad (31.\#)$$

Multivariate Gaussian Conditioning

- Let random vector $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)^T = \left((x_1, x_2, \cdots, x_m), (x_{m+1}, x_{m+2}, \cdots, x_n)\right)^T = (\boldsymbol{x}_1, \boldsymbol{x}_2)^T$
- Multivariate Gaussian distribution of $\boldsymbol{x}$ is given by:

$$p(\boldsymbol{x}_1, \boldsymbol{x}_2) = \frac{1}{\sqrt{(2\pi)^n |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}\left(\begin{bmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\end{bmatrix} - \begin{bmatrix}\boldsymbol{\mu}_1\\ \boldsymbol{\mu}_2\end{bmatrix}\right)^T \begin{bmatrix}\boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12}\\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22}\end{bmatrix}^{-1} \left(\begin{bmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\end{bmatrix} - \begin{bmatrix}\boldsymbol{\mu}_1\\ \boldsymbol{\mu}_2\end{bmatrix}\right)\right) \quad (32.\#)$$

$$p(\boldsymbol{x}_1|\boldsymbol{x}_2) = \frac{p(\boldsymbol{x}_1, \boldsymbol{x}_2)}{p(\boldsymbol{x}_2)} \propto \exp\left(-\frac{1}{2}\left(\begin{bmatrix}\boldsymbol{x}_1 - \boldsymbol{\mu}_1\\ \boldsymbol{x}_2 - \boldsymbol{\mu}_2\end{bmatrix}\right)^T \begin{bmatrix}\boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12}\\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22}\end{bmatrix}^{-1} \left(\begin{bmatrix}\boldsymbol{x}_1 - \boldsymbol{\mu}_1\\ \boldsymbol{x}_2 - \boldsymbol{\mu}_2\end{bmatrix}\right) + (\boldsymbol{x}_2 - \boldsymbol{\mu}_2)^T \boldsymbol{\Sigma}_{22}{}^{-1} (\boldsymbol{x}_2 - \boldsymbol{\mu}_2)\right) \quad (32.\#)$$

Denote:

$$\begin{bmatrix}\boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12}\\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22}\end{bmatrix}^{-1} = \begin{bmatrix}\boldsymbol{\Sigma}^{11} & \boldsymbol{\Sigma}^{12}\\ \boldsymbol{\Sigma}^{21} & \boldsymbol{\Sigma}^{22}\end{bmatrix} \quad (32.\#)$$

$$p(\boldsymbol{x}_1|\boldsymbol{x}_2) \propto \exp\left(-\frac{1}{2}\left(\left(\begin{bmatrix}\boldsymbol{x}_1 - \boldsymbol{\mu}_1\\ \boldsymbol{x}_2 - \boldsymbol{\mu}_2\end{bmatrix}\right)^T \begin{bmatrix}\boldsymbol{\Sigma}^{11} & \boldsymbol{\Sigma}^{12}\\ \boldsymbol{\Sigma}^{21} & \boldsymbol{\Sigma}^{22}\end{bmatrix} \left(\begin{bmatrix}\boldsymbol{x}_1 - \boldsymbol{\mu}_1\\ \boldsymbol{x}_2 - \boldsymbol{\mu}_2\end{bmatrix}\right)\right) + \frac{1}{2}(\boldsymbol{x}_2 - \boldsymbol{\mu}_2)^T \boldsymbol{\Sigma}_{22}{}^{-1} (\boldsymbol{x}_2 - \boldsymbol{\mu}_2)\right) \quad (32.\#)$$

$$p(\boldsymbol{x}_1|\boldsymbol{x}_2) \propto \exp\left(-\frac{1}{2}\left(\left(\begin{bmatrix}\boldsymbol{x}_1 - \boldsymbol{\mu}_1\\ \boldsymbol{x}_2 - \boldsymbol{\mu}_2\end{bmatrix}\right)^T \begin{bmatrix}\boldsymbol{\Sigma}^{11}[\boldsymbol{x}_1 - \boldsymbol{\mu}_1] + \boldsymbol{\Sigma}^{12}[\boldsymbol{x}_2 - \boldsymbol{\mu}_2]\\ \boldsymbol{\Sigma}^{21}[\boldsymbol{x}_1 - \boldsymbol{\mu}_1] + \boldsymbol{\Sigma}^{22}[\boldsymbol{x}_2 - \boldsymbol{\mu}_2]\end{bmatrix}\right) + \frac{1}{2}(\boldsymbol{x}_2 - \boldsymbol{\mu}_2)^T \boldsymbol{\Sigma}_{22}{}^{-1} (\boldsymbol{x}_2 - \boldsymbol{\mu}_2)\right) \quad (32.\#)$$

$$p(\boldsymbol{x}_1|\boldsymbol{x}_2) \propto \exp\left(-\frac{1}{2}\left([\boldsymbol{x}_1 - \boldsymbol{\mu}_1]^T \boldsymbol{\Sigma}^{11}[\boldsymbol{x}_1 - \boldsymbol{\mu}_1] + [\boldsymbol{x}_2 - \boldsymbol{\mu}_2]^T \boldsymbol{\Sigma}^{21}[\boldsymbol{x}_1 - \boldsymbol{\mu}_1] + [\boldsymbol{x}_1 - \boldsymbol{\mu}_1]^T \boldsymbol{\Sigma}^{12}[\boldsymbol{x}_2 - \boldsymbol{\mu}_2] + [\boldsymbol{x}_2 - \boldsymbol{\mu}_2]^T \boldsymbol{\Sigma}^{22}[\boldsymbol{x}_2 - \boldsymbol{\mu}_2]\right) + \frac{1}{2}(\boldsymbol{x}_2 - \boldsymbol{\mu}_2)^T \boldsymbol{\Sigma}_{22}{}^{-1} (\boldsymbol{x}_2 - \boldsymbol{\mu}_2)\right) \quad (32.\#)$$

Multivariate Gaussian Conditioning

$$p(\boldsymbol{x}_1|\boldsymbol{x}_2) \propto \exp\left(-\frac{1}{2}\left([\boldsymbol{x}_1-\boldsymbol{\mu}_1]^T\boldsymbol{\Sigma}^{11}[\boldsymbol{x}_1-\boldsymbol{\mu}_1]+[\boldsymbol{x}_2-\boldsymbol{\mu}_2]^T\boldsymbol{\Sigma}^{21}[\boldsymbol{x}_1-\boldsymbol{\mu}_1]+[\boldsymbol{x}_1-\boldsymbol{\mu}_1]^T\boldsymbol{\Sigma}^{12}[\boldsymbol{x}_2-\boldsymbol{\mu}_2]+[\boldsymbol{x}_2-\boldsymbol{\mu}_2]^T\boldsymbol{\Sigma}^{22}[\boldsymbol{x}_2-\boldsymbol{\mu}_2]\right) + \frac{1}{2}(\boldsymbol{x}_2-\boldsymbol{\mu}_2)^T{\boldsymbol{\Sigma}_{22}}^{-1}(\boldsymbol{x}_2-\boldsymbol{\mu}_2)\right) \quad (33.\#)$$

$$\propto \exp\left(-\frac{1}{2}\left([\boldsymbol{x}_1-\boldsymbol{\mu}_1]^T\boldsymbol{\Sigma}^{11}[\boldsymbol{x}_1-\boldsymbol{\mu}_1]+2[\boldsymbol{x}_1-\boldsymbol{\mu}_1]^T\boldsymbol{\Sigma}^{12}[\boldsymbol{x}_2-\boldsymbol{\mu}_2]+[\boldsymbol{x}_2-\boldsymbol{\mu}_2]^T\boldsymbol{\Sigma}^{22}[\boldsymbol{x}_2-\boldsymbol{\mu}_2]\right)+\frac{1}{2}(\boldsymbol{x}_2-\boldsymbol{\mu}_2)^T{\boldsymbol{\Sigma}_{22}}^{-1}(\boldsymbol{x}_2-\boldsymbol{\mu}_2)\right) \quad (33.\#)$$

$$\begin{bmatrix}\boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12}\\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22}\end{bmatrix}^{-1} = \begin{bmatrix}\boldsymbol{\Sigma}^{11} & \boldsymbol{\Sigma}^{12}\\ \boldsymbol{\Sigma}^{21} & \boldsymbol{\Sigma}^{22}\end{bmatrix} = \begin{bmatrix}\left(\boldsymbol{\Sigma}_{11}-\boldsymbol{\Sigma}_{12}{\boldsymbol{\Sigma}_{22}}^{-1}\boldsymbol{\Sigma}_{21}\right)^{-1} & -\left(\boldsymbol{\Sigma}_{11}-\boldsymbol{\Sigma}_{12}{\boldsymbol{\Sigma}_{22}}^{-1}\boldsymbol{\Sigma}_{21}\right)^{-1}\boldsymbol{\Sigma}_{12}{\boldsymbol{\Sigma}_{22}}^{-1}\\ -{\boldsymbol{\Sigma}_{22}}^{-1}\boldsymbol{\Sigma}_{12}\left(\boldsymbol{\Sigma}_{11}-\boldsymbol{\Sigma}_{12}{\boldsymbol{\Sigma}_{22}}^{-1}\boldsymbol{\Sigma}_{21}\right)^{-1} & {\boldsymbol{\Sigma}_{22}}^{-1}+{\boldsymbol{\Sigma}_{22}}^{-1}\boldsymbol{\Sigma}_{12}\left(\boldsymbol{\Sigma}_{11}-\boldsymbol{\Sigma}_{12}{\boldsymbol{\Sigma}_{22}}^{-1}\boldsymbol{\Sigma}_{21}\right)^{-1}\boldsymbol{\Sigma}_{12}{\boldsymbol{\Sigma}_{22}}^{-1}\end{bmatrix} \quad (33.\#)$$

$$\propto \exp\left(-\frac{1}{2}\left([\boldsymbol{x}_1-\boldsymbol{\mu}_1]^T\left(\left(\boldsymbol{\Sigma}_{11}-\boldsymbol{\Sigma}_{12}{\boldsymbol{\Sigma}_{22}}^{-1}\boldsymbol{\Sigma}_{21}\right)^{-1}\right)[\boldsymbol{x}_1-\boldsymbol{\mu}_1]-2[\boldsymbol{x}_1-\boldsymbol{\mu}_1]^T\left(\left(\boldsymbol{\Sigma}_{11}-\boldsymbol{\Sigma}_{12}{\boldsymbol{\Sigma}_{22}}^{-1}\boldsymbol{\Sigma}_{21}\right)^{-1}\boldsymbol{\Sigma}_{12}{\boldsymbol{\Sigma}_{22}}^{-1}\right)[\boldsymbol{x}_2-\boldsymbol{\mu}_2] + [\boldsymbol{x}_2-\boldsymbol{\mu}_2]^T\left({\boldsymbol{\Sigma}_{22}}^{-1}+{\boldsymbol{\Sigma}_{22}}^{-1}\boldsymbol{\Sigma}_{12}\left(\boldsymbol{\Sigma}_{11}-\boldsymbol{\Sigma}_{12}{\boldsymbol{\Sigma}_{22}}^{-1}\boldsymbol{\Sigma}_{21}\right)^{-1}\boldsymbol{\Sigma}_{12}{\boldsymbol{\Sigma}_{22}}^{-1}\right)[\boldsymbol{x}_2-\boldsymbol{\mu}_2]\right)+\frac{1}{2}(\boldsymbol{x}_2-\boldsymbol{\mu}_2)^T{\boldsymbol{\Sigma}_{22}}^{-1}(\boldsymbol{x}_2-\boldsymbol{\mu}_2)\right) \quad (33.\#)$$

(re-writing but noting terms that drop out)

$$\propto \exp\left(\boxed{-\frac{1}{2}}\left([\boldsymbol{x}_1-\boldsymbol{\mu}_1]^T\left(\left(\boldsymbol{\Sigma}_{11}-\boldsymbol{\Sigma}_{12}{\boldsymbol{\Sigma}_{22}}^{-1}\boldsymbol{\Sigma}_{21}\right)^{-1}\right)[\boldsymbol{x}_1-\boldsymbol{\mu}_1]-2[\boldsymbol{x}_1-\boldsymbol{\mu}_1]^T\left(\left(\boldsymbol{\Sigma}_{11}-\boldsymbol{\Sigma}_{12}{\boldsymbol{\Sigma}_{22}}^{-1}\boldsymbol{\Sigma}_{21}\right)^{-1}\boldsymbol{\Sigma}_{12}{\boldsymbol{\Sigma}_{22}}^{-1}\right)[\boldsymbol{x}_2-\boldsymbol{\mu}_2] + \boxed{[\boldsymbol{x}_2-\boldsymbol{\mu}_2]^T\left({\boldsymbol{\Sigma}_{22}}^{-1}\right.}\left.+{\boldsymbol{\Sigma}_{22}}^{-1}\boldsymbol{\Sigma}_{12}\left(\boldsymbol{\Sigma}_{11}-\boldsymbol{\Sigma}_{12}{\boldsymbol{\Sigma}_{22}}^{-1}\boldsymbol{\Sigma}_{21}\right)^{-1}\boldsymbol{\Sigma}_{12}{\boldsymbol{\Sigma}_{22}}^{-1}\right)\boxed{[\boldsymbol{x}_2-\boldsymbol{\mu}_2]}\right)\boxed{+\frac{1}{2}(\boldsymbol{x}_2-\boldsymbol{\mu}_2)^T{\boldsymbol{\Sigma}_{22}}^{-1}(\boldsymbol{x}_2-\boldsymbol{\mu}_2)}\right) \quad (33.\#)$$

Multivariate Gaussian Conditioning

$$p(\boldsymbol{x}_1|\boldsymbol{x}_2) \propto \exp\left(-\frac{1}{2}\left([\boldsymbol{x}_1-\boldsymbol{\mu}_1]^T\left(\left(\boldsymbol{\Sigma}_{11}-\boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}{}^{-1}\boldsymbol{\Sigma}_{21}\right)^{-1}\right)[\boldsymbol{x}_1-\boldsymbol{\mu}_1] - 2[\boldsymbol{x}_1-\boldsymbol{\mu}_1]^T\left(\left(\boldsymbol{\Sigma}_{11}-\boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}{}^{-1}\boldsymbol{\Sigma}_{21}\right)^{-1}\boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}{}^{-1}\right)[\boldsymbol{x}_2-\boldsymbol{\mu}_2] + [\boldsymbol{x}_2-\boldsymbol{\mu}_2]^T\left(\boldsymbol{\Sigma}_{22}{}^{-1}\boldsymbol{\Sigma}_{12}\left(\boldsymbol{\Sigma}_{11}-\boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}{}^{-1}\boldsymbol{\Sigma}_{21}\right)^{-1}\boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}{}^{-1}\right)[\boldsymbol{x}_2-\boldsymbol{\mu}_2]\right)\right) \quad (34.\#)$$

Denote:

$$\mathbf{P}^{-1} = \left(\boldsymbol{\Sigma}_{11}-\boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}{}^{-1}\boldsymbol{\Sigma}_{21}\right)^{-1} \quad (34.\#)$$

$$\widetilde{\boldsymbol{x}_1} = [\boldsymbol{x}_1-\boldsymbol{\mu}_1] \quad (34.\#)$$

$$\widetilde{\boldsymbol{x}_2} = \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}{}^{-1}[\boldsymbol{x}_2-\boldsymbol{\mu}_2] \quad (34.\#)$$

$$p(\boldsymbol{x}_1|\boldsymbol{x}_2) \propto \exp\left(-\frac{1}{2}\left(\widetilde{\boldsymbol{x}_1}^T\mathbf{P}^{-1}\widetilde{\boldsymbol{x}_1} - 2\widetilde{\boldsymbol{x}_1}^T\mathbf{P}^{-1}\widetilde{\boldsymbol{x}_2} + \widetilde{\boldsymbol{x}_2}^T\mathbf{P}^{-1}\widetilde{\boldsymbol{x}_2}\right)\right) \quad (34.\#)$$

Note:

$$p(\boldsymbol{x}_1|\boldsymbol{x}_2) \propto \exp\left(-\frac{1}{2}\left(\widetilde{\boldsymbol{x}_1}-\widetilde{\boldsymbol{x}_2}\right)^T\mathbf{P}^{-1}\left(\widetilde{\boldsymbol{x}_1}-\widetilde{\boldsymbol{x}_2}\right)\right) \quad (34.\#)$$

$$p(\boldsymbol{x}_1|\boldsymbol{x}_2) \propto \exp\left(-\frac{1}{2}\left(\boldsymbol{x}_1-\left(\boldsymbol{\mu}_1+\boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}{}^{-1}[\boldsymbol{x}_2-\boldsymbol{\mu}_2]\right)\right)^T\left(\boldsymbol{\Sigma}_{11}-\boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}{}^{-1}\boldsymbol{\Sigma}_{21}\right)^{-1}\left(\boldsymbol{x}_1-\left(\boldsymbol{\mu}_1+\boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}{}^{-1}[\boldsymbol{x}_2-\boldsymbol{\mu}_2]\right)\right)\right) \quad (34.\#)$$

$$\boxed{\begin{aligned}\boldsymbol{\mu}_{x_1|x_2} &= \left(\boldsymbol{\mu}_1+\boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}{}^{-1}[\boldsymbol{x}_2-\boldsymbol{\mu}_2]\right) \quad (34.\#)\\ \mathbf{P}_{x_1|x_2} &= \left(\boldsymbol{\Sigma}_{11}-\boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}{}^{-1}\boldsymbol{\Sigma}_{21}\right) \quad (34.\#)\end{aligned}}$$

- **Update step – Goal**: conditional distribution $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})$
- From conditional (3.3)

$$p(\boldsymbol{x}_k|\boldsymbol{y}_k, \boldsymbol{y}_{1;k-1}) = \frac{p(\boldsymbol{x}_k, \boldsymbol{y}_k|\boldsymbol{y}_{1:k-1})}{p(\boldsymbol{y}_k|\boldsymbol{y}_{1:k-1})} \tag{35.1}$$

- From conditional (3.4) and measurement model (…)

$$p(\boldsymbol{y}_k|\boldsymbol{x}_k, \boldsymbol{y}_{1:k-1}) = \frac{p(\boldsymbol{x}_k, \boldsymbol{y}_k|\boldsymbol{y}_{1:k-1})}{p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1})} \tag{35.2}$$

- **Bayes Theorem**

$$p(\boldsymbol{x}_k|\boldsymbol{y}_k, \boldsymbol{y}_{1:k-1}) = \frac{p(\boldsymbol{y}_k|\boldsymbol{x}_k, \boldsymbol{y}_{1:k-1})p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1})}{p(\boldsymbol{y}_k|\boldsymbol{y}_{1:k-1})} \tag{35.3}$$

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

$$p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k}) = \frac{p(\boldsymbol{y}_k|\boldsymbol{x}_k, \boldsymbol{y}_{1:k-1})p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1})}{\int p(\boldsymbol{y}_k|\boldsymbol{x}_k, \boldsymbol{y}_{1:k-1})p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1})d\boldsymbol{x}_k} \tag{35.4}$$

- Conditional independence of measurements gives:

$$\boxed{p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k}) = \frac{p(\boldsymbol{y}_k|\boldsymbol{x}_k)p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1})}{\int p(\boldsymbol{y}_k|\boldsymbol{x}_k)p(\boldsymbol{x}_k|\boldsymbol{y}1_{:k-1})d\boldsymbol{x}_k}} \tag{35.5}$$

Bayesian Filtering Equations

- **Predict**: $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1})$

  $p(\boldsymbol{x}_{k-1}|\boldsymbol{y}_{1:k-1})$: posterior from previous time−step

  $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$: process model (dynamics)

  Prediction: $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1}) = \int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})p(\boldsymbol{x}_{k-1}|\boldsymbol{y}_{1:k-1})$

- **Measure**: $p(\boldsymbol{y}_k|\boldsymbol{x}_k)$

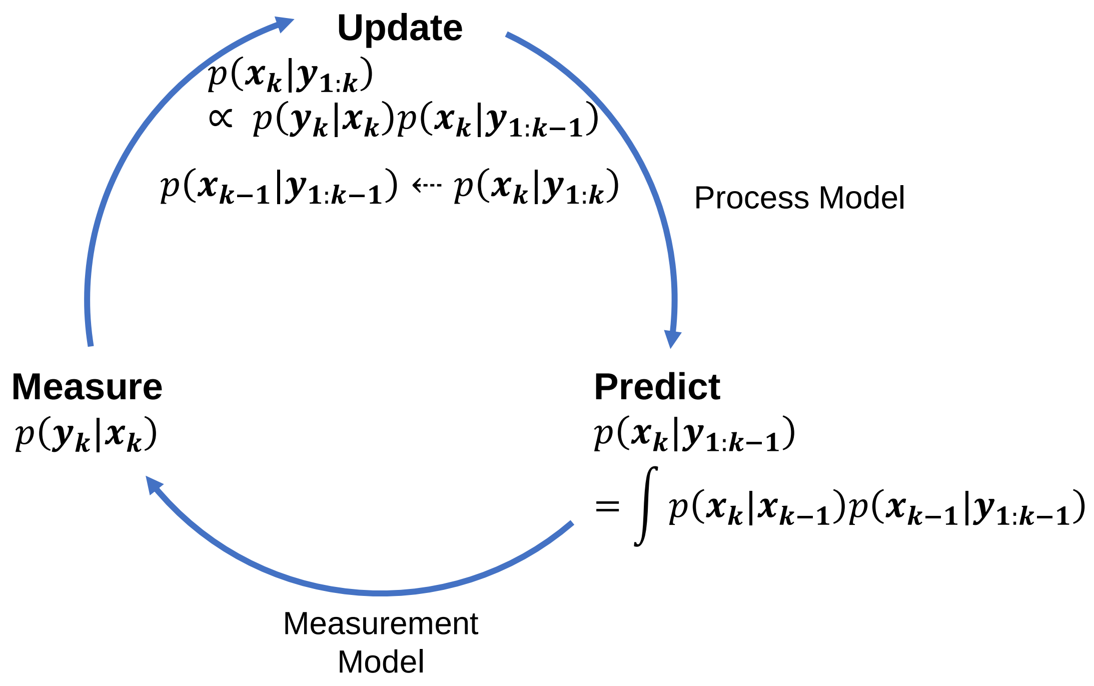
  - Measurement model is the likelihood. Given or assumed through sensor characterization

- **Update**: $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})$

  $p(\boldsymbol{y}_k|\boldsymbol{x}_{1:k})$: measurement model

  $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1})$: prediction (prior)

  Update: $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k}) = \dfrac{p(\boldsymbol{y}_k|\boldsymbol{x}_k)p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1})}{\int p(\boldsymbol{y}_k|\boldsymbol{x}_k)p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1})d\boldsymbol{x}_{k-1}}$



Kalman Filter – Preliminaries

- We are interested in relationships between joint, marginal, conditional of a multivariate Gaussian distribution

- Given a random vector $\boldsymbol{x} \in \mathbb{R}^n$
- Multivariate Gaussian distribution of $\boldsymbol{x}$ is given by:

$$p(\boldsymbol{x}) = f(x_1, x_2, \cdots, x_n) = \frac{1}{\sqrt{(2\pi)^n |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right) \tag{37.1}$$

- where the vector $\boldsymbol{\mu}$ is the mean and $\boldsymbol{\Sigma}$ is the covariance matrix
- $p(\boldsymbol{x})$ can also be expressed parametrically

$$p(\boldsymbol{x}) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{37.2}$$

Kalman Filter – Prediction derivation

- Assume a Gaussian driven process

$$\boldsymbol{x_k} = \mathbf{A}\boldsymbol{x_{k-1}} + \boldsymbol{q_{k-1}} \quad (38.1)$$

with process noise centered at $\boldsymbol{\mu}_{\boldsymbol{q}_{k-1}} = \mathbf{0}$

$$\boldsymbol{q_{k-1}} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{\boldsymbol{k-1}}) \quad (38.2)$$

then the conditional distribution $\boldsymbol{x}_k$ conditioned on $\boldsymbol{x}_{k-1}$ is given by:

$$p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) \sim \mathcal{N}(\mathbf{A}\boldsymbol{x_{k-1}}, \mathbf{Q}_{\boldsymbol{k-1}}) \quad (38.3)$$

and the marginal distribution is given by

$$p\left(\boldsymbol{x}_{k-1|k-1}\right) \sim \mathcal{N}\left(\boldsymbol{\mu}_{k-1|k-1}, \boldsymbol{\Sigma}_{k-1|k-1}\right) \quad (38.4)$$

- From 13.2, 13.5, and 14.3

$$\boldsymbol{\mu}_{k|k-1} = \mathbf{A}\boldsymbol{x_{k-1}} = \boldsymbol{\mu}_k + \boldsymbol{\Sigma}_{k-1,k}\boldsymbol{\Sigma}_{k-1,k-1}{}^{-1}(\boldsymbol{x}_{k-1} - \boldsymbol{\mu}_{k-1}) \quad (38.5)$$

$$\mathbf{A}\boldsymbol{x_{k-1}} = \boldsymbol{\mu}_k + \boldsymbol{\Sigma}_{k-1,k}\boldsymbol{\Sigma}_{k-1,k-1}{}^{-1}\boldsymbol{x}_{k-1} - \boldsymbol{\Sigma}_{k-1,k}\boldsymbol{\Sigma}_{k-1,k-1}{}^{-1}\boldsymbol{\mu}_{k-1} \quad (38.6)$$

$$\boldsymbol{\mu}_k - \boldsymbol{\Sigma}_{k-1,k}\boldsymbol{\Sigma}_{k-1,k-1}{}^{-1}\boldsymbol{\mu}_{k-1} = \mathbf{0} \quad (38.7)$$

$$\mathbf{A}\boldsymbol{x_{k-1}} = \boldsymbol{\Sigma}_{k-1,k}\boldsymbol{\Sigma}_{k-1,k-1}{}^{-1}\boldsymbol{x}_{k-1} \quad (38.8)$$

$$\mathbf{A} = \boldsymbol{\Sigma}_{k-1,k}\boldsymbol{\Sigma}_{k-1,k-1}{}^{-1} \quad (38.9)$$

$$\boldsymbol{\Sigma}_{k-1,k} = \mathbf{A}\boldsymbol{\Sigma}_{k-1,k-1} = \boldsymbol{\Sigma}_{k,k-1}{}^{T} \quad (38.10)$$

- Assume a Gaussian driven process

$$\boldsymbol{x_k} = \mathbf{A}\boldsymbol{x_{k-1}} + \boldsymbol{q_{k-1}} \tag{14.1}$$

with process noise centered at $\boldsymbol{\mu_q} = \mathbf{0}$

$$\boldsymbol{q_{k-1}} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{\boldsymbol{k-1}}) \tag{14.2}$$

then the conditional distribution $\boldsymbol{x}_k$ conditioned on $\boldsymbol{x}_{k-1}$ is given by:

$$p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) \sim \mathcal{N}(\mathbf{A}\boldsymbol{x_{k-1}}, \mathbf{Q}_{\boldsymbol{k-1}}) \tag{14.3}$$

and the marginal distribution is given by

$$p(\boldsymbol{x}_{k-1}) \sim \mathcal{N}\left(\boldsymbol{\mu}_{k-1}, \boldsymbol{\Sigma}_{k-1,k-1}\right) \tag{14.4}$$

- From 13.2, 13.5, and 14.3

$$\boldsymbol{\Sigma}_{k|k-1} = \mathbf{Q}_{\boldsymbol{k-1}} = \quad \boldsymbol{\Sigma}_{kk} - \boldsymbol{\Sigma}_{k-1,k}\boldsymbol{\Sigma}_{k-1,k-1}{}^{-1}\boldsymbol{\Sigma}_{k,k-1} \tag{39.1}$$

$$\mathbf{Q}_{\boldsymbol{k-1}} + \boldsymbol{\Sigma}_{k-1,k}\boldsymbol{\Sigma}_{k-1,k-1}{}^{-1}\boldsymbol{\Sigma}_{k,k-1} = \quad \boldsymbol{\Sigma}_{kk} \tag{39.2}$$

- From 14.10 and 15.2

$$\mathbf{Q}_{\boldsymbol{k-1}} + \mathbf{A}\boldsymbol{\Sigma}_{k-1,k-1}\boldsymbol{\Sigma}_{k-1,k-1}{}^{-1}\boldsymbol{\Sigma}_{k-1,k-1}\mathbf{A}^T = \quad \boldsymbol{\Sigma}_{kk} \tag{39.3}$$

$$\mathbf{Q}_{\boldsymbol{k-1}} + \mathbf{A}\boldsymbol{\Sigma}_{k-1,k-1}\mathbf{A}^T = \quad \boldsymbol{\Sigma}_{kk} \tag{39.4}$$

Kalman Filter – Prediction derivation

- From 13.1, 13.2, 13.5, 14.4, 14.10, and 15.4, the joint distribution of $\boldsymbol{x}_k$ and $\boldsymbol{x}_{k-1}$ is given by

$$p(\boldsymbol{x}_{k-1}, \boldsymbol{x}_k) = \sim\mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu}_{k-1} \\ \mathbf{A}\boldsymbol{\mu}_{k-1} \end{pmatrix}, \begin{pmatrix} \boldsymbol{\Sigma}_{k-1,k-1} & \boldsymbol{\Sigma}_{k-1,k-1}\mathbf{A}^T \\ \mathbf{A}\boldsymbol{\Sigma}_{k-1,k-1} & \mathbf{Q}_{\boldsymbol{k-1}} + \mathbf{A}\boldsymbol{\Sigma}_{k-1,k-1}\mathbf{A}^T \end{pmatrix}\right) \tag{40.1}$$

- Then from 13.3

$$p(\boldsymbol{x}_k) \sim \mathcal{N}\left(\mathbf{A}\boldsymbol{\mu}_{k-1}, \mathbf{A}\boldsymbol{\Sigma}_{k-1,k-1}\mathbf{A}^T + \mathbf{Q}_{\boldsymbol{k-1}}\right) \tag{40.2}$$

- $p(\boldsymbol{x}_k)$ is marginal with respect to joint $p(\boldsymbol{x}_{k-1}, \boldsymbol{x}_k)$, but all are conditioned on $\boldsymbol{y}_{1:k-1}$

$$\boxed{p(\boldsymbol{x}_k) = p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1}) \sim \mathcal{N}\left(\mathbf{A}\boldsymbol{\mu}_{k-1}, \mathbf{A}\boldsymbol{\Sigma}_{k-1,k-1}\mathbf{A}^T + \mathbf{Q}_{\boldsymbol{k-1}}\right)} \tag{40.3}$$

Kalman Filter – Update derivation

- Assume a Gaussian measurement model

$$\boldsymbol{y_k} = \boldsymbol{H}\boldsymbol{x_k} + \boldsymbol{r_k} \tag{41.1}$$

with noise centered at $\boldsymbol{\mu_r} = \mathbf{0}$

$$\boldsymbol{r_k} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{\boldsymbol{k}}) \tag{41.2}$$

then the conditional distribution $\boldsymbol{y}_k$ conditioned on $\boldsymbol{x}_k$ is given by:

$$p(\boldsymbol{y}_k | \boldsymbol{x}_k) \sim \mathcal{N}(\mathbf{H}\boldsymbol{x_k}, \mathbf{R_k}) \tag{41.3}$$

and the marginal distribution is given by

$$p(\boldsymbol{x}_k) \sim \mathcal{N}\left(\boldsymbol{\mu}_{x_k}, \boldsymbol{\Sigma}_{k,k}\right) \tag{41.4}$$

- From 13.2, 13.5, and 18.3

$$\boldsymbol{\mu}_{y_k|x_k} = \mathbf{H}\boldsymbol{x}_k = \boldsymbol{\mu}_{y_k} + \boldsymbol{\Sigma}_{y_k,x_k}\boldsymbol{\Sigma}_{x_k,x_k}{}^{-1}\left(\boldsymbol{x}_k - \boldsymbol{\mu}_{x_k}\right) \tag{41.5}$$

$$\mathbf{H}\boldsymbol{x}_k = \boldsymbol{\mu}_{y_k} + \boldsymbol{\Sigma}_{y_k,x_k}\boldsymbol{\Sigma}_{x_k,x_k}{}^{-1}\boldsymbol{x}_k - \boldsymbol{\Sigma}_{y_k,x_k}\boldsymbol{\Sigma}_{x_k,x_k}{}^{-1}\boldsymbol{\mu}_{x_k} \tag{41.6}$$

$$\boldsymbol{\mu}_{y_k} - \boldsymbol{\Sigma}_{y_k,x_k}\boldsymbol{\Sigma}_{x_k,x_k}{}^{-1}\boldsymbol{\mu}_{x_k} = \mathbf{0} \tag{41.7}$$

$$\mathbf{H}\boldsymbol{x}_k = \boldsymbol{\Sigma}_{y_k,x_k}\boldsymbol{\Sigma}_{x_k,x_k}{}^{-1}\boldsymbol{x}_k \tag{41.8}$$

$$\mathbf{H} = \boldsymbol{\Sigma}_{y_k,x_k}\boldsymbol{\Sigma}_{x_k,x_k}{}^{-1} \tag{41.9}$$

$$\boldsymbol{\Sigma}_{y_k,x_k} = \mathbf{H}\boldsymbol{\Sigma}_{x_k,x_k} = \boldsymbol{\Sigma}_{x_k,y_k}{}^{T} \tag{41.10}$$

Kalman Filter – Update derivation

- Assume a Gaussian measurement model

$$\boldsymbol{y_k} = \boldsymbol{H}\boldsymbol{x_k} + \boldsymbol{r_k} \tag{17.1}$$

with noise centered at $\boldsymbol{\mu_r} = \mathbf{0}$

$$\boldsymbol{r_k} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{\boldsymbol{k}}) \tag{17.2}$$

then the conditional distribution $\boldsymbol{y}_k$ conditioned on $\boldsymbol{x}_k$ is given by:

$$p(\boldsymbol{y}_k | \boldsymbol{x}_k) \sim \mathcal{N}(\mathbf{H}\boldsymbol{x_k}, \mathbf{R_k}) \tag{17.3}$$

and the marginal distribution is given by

$$p(\boldsymbol{x}_k) \sim \mathcal{N}\left(\boldsymbol{\mu}_{x_k}, \boldsymbol{\Sigma}_{k,k}\right) \tag{17.4}$$

- From 13.2, 13.5, and 17.3

$$\boldsymbol{\Sigma}_{y_k|x_k} = \mathbf{R_k} = \boldsymbol{\Sigma}_{y_k,y_k} - \boldsymbol{\Sigma}_{y_k,x_k}\boldsymbol{\Sigma}_{x_k,x_k}{}^{-1}\boldsymbol{\Sigma}_{x_k,y_k} \tag{42.1}$$

$$\mathbf{R_k} + \boldsymbol{\Sigma}_{y_k,x_k}\boldsymbol{\Sigma}_{x_k,x_k}{}^{-1}\boldsymbol{\Sigma}_{x_k,y_k} = \boldsymbol{\Sigma}_{y_k,y_k} \tag{42.2}$$

- From 17.10 and 18.2

$$\mathbf{R_k} + \mathbf{H}\boldsymbol{\Sigma}_{x_k,x_k}\boldsymbol{\Sigma}_{x_k,x_k}{}^{-1}\boldsymbol{\Sigma}_{x_k,x_k}\mathbf{H}^T = \boldsymbol{\Sigma}_{y_k,y_k} \tag{42.3}$$

$$\boldsymbol{\Sigma}_{y_k,y_k} = \mathbf{R_k} + \mathbf{H}\boldsymbol{\Sigma}_{x_k,x_k}\mathbf{H}^T \tag{42.4}$$

Kalman Filter – Update derivation

- From 13.1, 13.2, 13.5, 17.4, 17.10, and 18.4, the joint distribution of $\boldsymbol{x}_k$ and $\boldsymbol{y}_k$ is given by

$$p(\boldsymbol{x}_k, \boldsymbol{y}_k) = \sim\mathcal{N}\left(\begin{pmatrix}\boldsymbol{\mu}_k \\ \boldsymbol{H}\boldsymbol{\mu}_k\end{pmatrix}, \begin{pmatrix}\boldsymbol{\Sigma}_{x_k,x_k} & \boldsymbol{\Sigma}_{x_k,x_k}\mathbf{H}^T \\ \mathbf{H}\boldsymbol{\Sigma}_{x_k,x_k} & \mathbf{R_k} + \mathbf{H}\boldsymbol{\Sigma}_{x_k,x_k}\mathbf{H}^T\end{pmatrix}\right) \tag{43.1}$$

- Then from 13.4

$$\boxed{p(\boldsymbol{x}_k|\boldsymbol{y}_k)\sim\mathcal{N}\left(\boldsymbol{\mu}_k+\boldsymbol{\Sigma}_{x_k,x_k}\mathbf{H}^T\left(\mathbf{R_k} + \mathbf{H}\boldsymbol{\Sigma}_{x_k,x_k}\mathbf{H}^T\right)^{-1}(\boldsymbol{y}_k - \boldsymbol{H}\boldsymbol{\mu}_k),\quad \boldsymbol{\Sigma}_{x_k,x_k} - \boldsymbol{\Sigma}_{x_k,x_k}\mathbf{H}^T\left(\mathbf{R_k} + \mathbf{H}\boldsymbol{\Sigma}_{x_k,x_k}\mathbf{H}^T\right)^{-1}\mathbf{H}\boldsymbol{\Sigma}_{x_k,x_k}\right)} \tag{43.2}$$

Particle Filter - preliminaries

- **Goal**: sample approximation $\hat{p}(\boldsymbol{x_k}|\boldsymbol{y_{1:k-1}})$ and $\hat{p}(\boldsymbol{x_k}|\boldsymbol{y_{1:k}})$. Represent these by a set of weighted samples
- Consider a univariate empirical cumulative distribution function constructed from sampling.
  - series of piece-wise step functions
  - Value of eCDF represents the fraction of sampled observations less than or equal to a particular sample value
- We can approximate the true cumulative distribution function with the eCDF as:

$$\hat{F}_N(x) = \frac{1}{n}\sum_{i=1}^{n} u_{x^{(i)}}(x) \tag{44.1}$$

$$u_{x^{(i)}} = \begin{cases} 1, & x \geq x_i \\ 0, & x < x_i \end{cases} \quad \text{(Heaviside Step)} \tag{44.2}$$

- An approximation of the probability mass function is the derivative of the eCDF:

$$\hat{p}_X(x) = \frac{d\hat{F}_x(x)}{dx} = \frac{1}{n}\sum_{i=1}^{n} \delta(x - x_i) \tag{44.3}$$

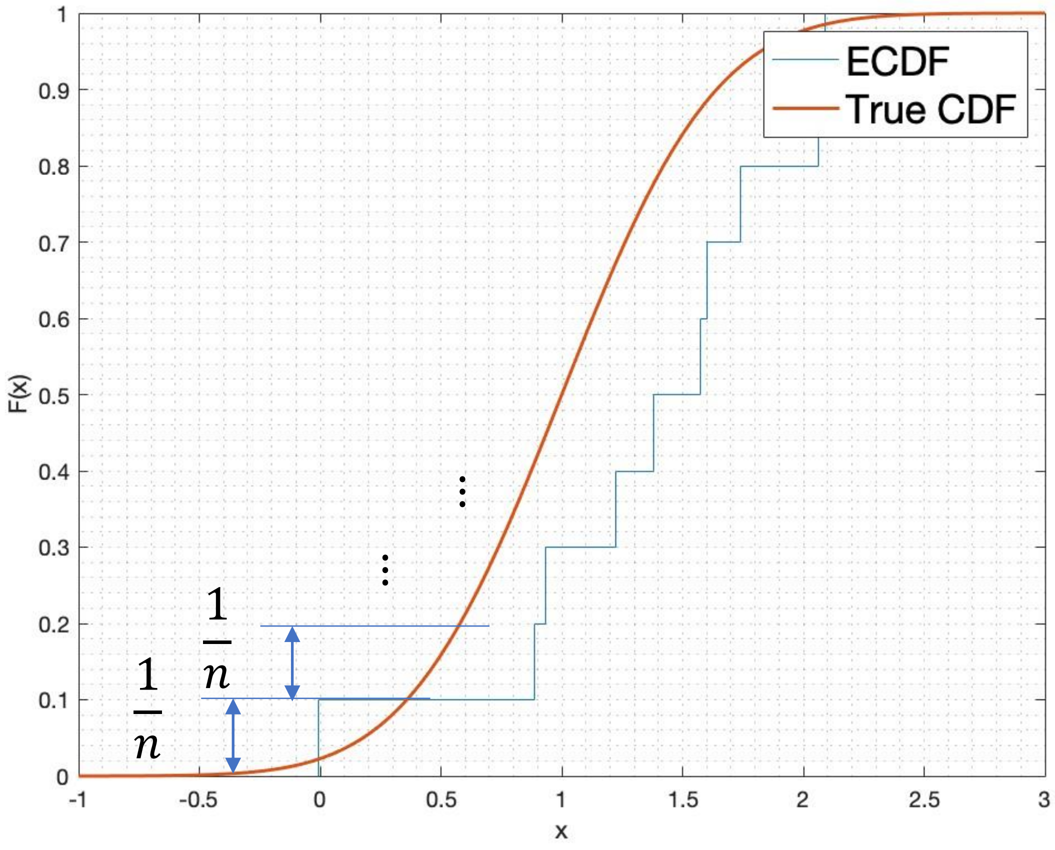
- The derivative $\frac{du_{x_i}(x)}{dx} = +\infty$. We define the derivative by the Dirac delta function $\frac{du_{x_i}(x)}{dx} = \delta(x - x_i)$

$$\delta(x - x_i) = \begin{cases} \text{not defined}, & x = x_i \\ 0, & \forall x \neq x_i \end{cases} \quad \text{(Dirac Delta)} \tag{44.4}$$

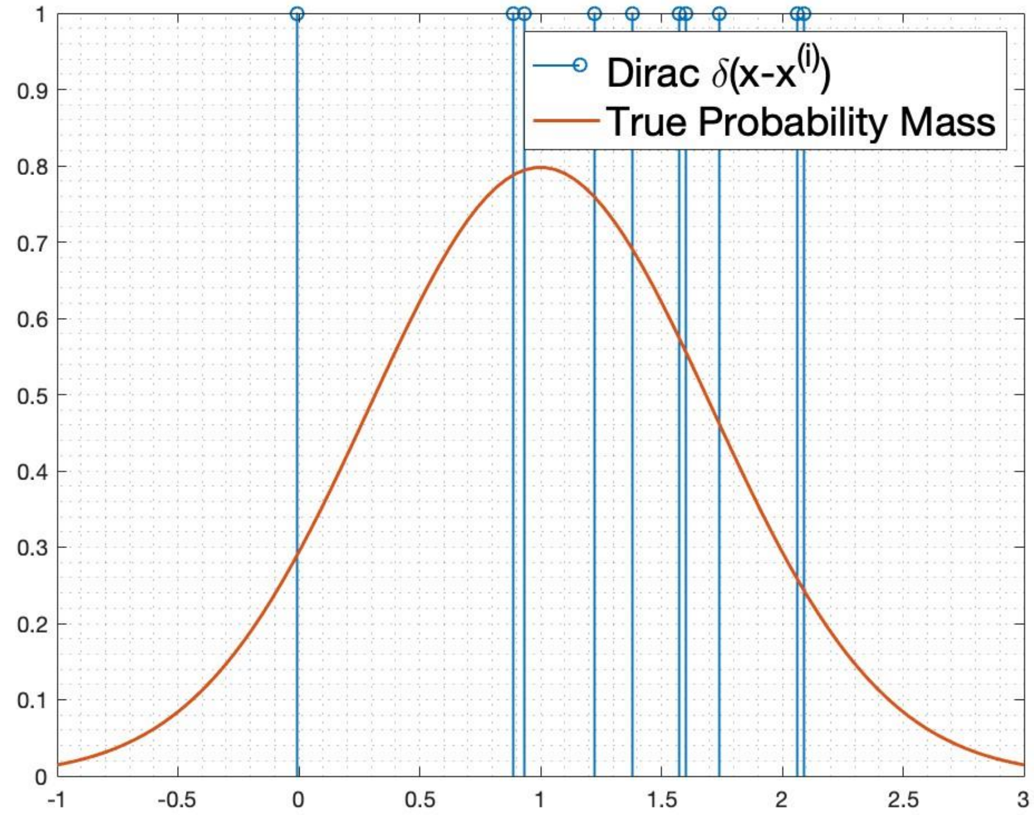
Particle Filter - preliminaries

- Example sample distribution, each sample weighted by $\frac{1}{n}$

Sample Cumulative Distribution



Sample Probability Mass



$x^{(i)} =$
-0.0087
0.8871
0.9351
1.2251
1.3784
1.5735
1.6021
1.7398
2.0607
2.0890

Particle Filter - preliminaries

- **Goal**: sample approximation $\hat{p}(\boldsymbol{x_k}|\boldsymbol{y_{k-1}})$ and $\hat{p}(\boldsymbol{x_k}|\boldsymbol{y_k})$. Represent these by a set of weighted samples
- The particle filter forms a weighted sample with probability mass function given by generalizing equation (…)

$$\hat{p}_x(x) = \frac{d\hat{F}_x(x)}{dx} = \omega^{(i)} \sum_{i=1}^{n} \delta(x - x^{(i)}) \quad (46.1)$$

- We seek expectations of the target conditional distributions:

$\boldsymbol{x}^{(i)}$ is sampled from $\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{y})$

$$\mathbb{E}(\boldsymbol{g}(\boldsymbol{x})|\boldsymbol{y}) = \int \boldsymbol{g}(\boldsymbol{x})\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{y})\boldsymbol{dx} \approx \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{g}(\boldsymbol{x}^{(i)}) \quad (46.2)$$

- For arbitrary distributions $p(\boldsymbol{x_k}|\boldsymbol{y_k})$ may be difficult or impossible to sample. Instead, we can sample from an importance distribution and weigh the samples proportional to the ratio of the target and proposal distribution. This is importance sampling.

$$\mathbb{E}(\boldsymbol{g}(\boldsymbol{x})|\boldsymbol{y}) = \int \boldsymbol{g}(\boldsymbol{x})\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{y})dx = \int \boldsymbol{g}(\boldsymbol{x})\frac{\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{y})}{\boldsymbol{\pi}(\boldsymbol{x}|\boldsymbol{y})}\boldsymbol{\pi}(\boldsymbol{x}|\boldsymbol{y})dx \quad (46.3)$$

$\overline{\boldsymbol{x}}^{(i)}$ is sampled from $\boldsymbol{\pi}(\boldsymbol{x}|\boldsymbol{y})$

$$\mathbb{E}(\boldsymbol{g}(\boldsymbol{x})|\boldsymbol{y}) \approx \sum_{i=1}^{n} \omega^{(i)}\boldsymbol{g}(\overline{\boldsymbol{x}}^{(i)}) \quad (46.4)$$

Particle Filter – prediction

- **Goal**: sample approximation $\hat{p}(\boldsymbol{x_k}|\boldsymbol{y_{k-1}})$ and $\hat{p}(\boldsymbol{x_k}|\boldsymbol{y_k})$. Represent these by a set of weighted samples
- Propagate each particle with process model
- Prediction weights:

$$\omega_{k,k-1}^{(i)} = \frac{1}{n}$$

- In practice, these weights are re-initialized with equal weighting at the start of each recursion and carried to the update step without change. Weights are only updated to compute expectation of posterior $p(\boldsymbol{x_k}|\boldsymbol{y_k})$
- Intuition is that without new likelihood information, each particle is equally weighted
- Not necessary to compute expectation and covariance at prediction step

Particle Filter – update

- **Goal**: sample approximation $\hat{p}(\boldsymbol{x_k}|\boldsymbol{y_{k-1}})$ and $\hat{p}(\boldsymbol{x_k}|\boldsymbol{y_k})$. Represent these by a set of weighted samples

- From (25.4), the un-normalized weights are

$$\omega^{*(i)} = \frac{1}{n}\frac{\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{y})}{\boldsymbol{\pi}(\boldsymbol{x}|\boldsymbol{y})}$$

- Assume we can't sample $\hat{p}(\boldsymbol{x_k}|\boldsymbol{y_{1:k}})$

- Considering entire sequence, the full posterior:

$$p(\boldsymbol{x_{0:k}}|\boldsymbol{y_{1:k}}) \propto p(\boldsymbol{y_k}|\boldsymbol{x_{0:k}}, \boldsymbol{y_{1:k-1}})p(\boldsymbol{x_{0:k}}|\boldsymbol{y_{1:k-1}})$$ (Bayesian Filter Update)

$$p(\boldsymbol{x_{0:k}}|\boldsymbol{y_{1:k}}) \propto p(\boldsymbol{y_k}|\boldsymbol{x_k})p(\boldsymbol{x_{0:k}}|\boldsymbol{y_{1:k-1}})$$ (Markov)

$$p(\boldsymbol{x_{0:k}}|\boldsymbol{y_{1:k}}) \propto p(\boldsymbol{y_k}|\boldsymbol{x_k})p(\boldsymbol{x_k}|\boldsymbol{x_{0:k-1}}, \boldsymbol{y_{1:k-1}})p(\boldsymbol{x_{0:k-1}}|\boldsymbol{y_{1:k-1}})$$

$$p(\boldsymbol{x_{0:k}}|\boldsymbol{y_{1:k}}) \propto p(\boldsymbol{y_k}|\boldsymbol{x_k})p(\boldsymbol{x_k}|\boldsymbol{x_{k-1}})p(\boldsymbol{x_{0:k-1}}|\boldsymbol{y_{1:k-1}})$$ (Markov)

Particle Filter – update

- **Goal**: sample approximation $\hat{p}(\boldsymbol{x_k}|\boldsymbol{y_{k-1}})$ and $\hat{p}(\boldsymbol{x_k}|\boldsymbol{y_k})$. Represent these by a set of weighted samples

- Update weights:

  - Choose importance distribution $\pi(\boldsymbol{x_{0:k}}|\boldsymbol{y_{1:k}}) = \pi(\boldsymbol{x_k}|\boldsymbol{x_{0:k-1}}, \boldsymbol{y_{1:k}})\pi(\boldsymbol{x_{0:k-1}}|\boldsymbol{y_{1:k-1}})$

$$\omega^{*(i)} = \frac{1}{n}\frac{p\left(\boldsymbol{x}^{(i)}{}_{0:k}\middle|\boldsymbol{y}_{1:k}\right)}{\pi\left(\boldsymbol{x}^{(i)}{}_{0:k}\middle|\boldsymbol{y}_{1:k}\right)} \propto \frac{1}{n}\frac{p\left(\boldsymbol{y}_k\middle|\boldsymbol{x}^{(i)}{}_k\right)p\left(\boldsymbol{x}^{(i)}{}_k\middle|\boldsymbol{x}^{(i)}{}_{k-1}\right)}{\pi(\boldsymbol{x_k}|\boldsymbol{x_{0:k-1}}, \boldsymbol{y_{1:k}})}\frac{p\left(\boldsymbol{x}^{(i)}{}_{0:k-1}\middle|\boldsymbol{y}_{1:k-1}\right)}{\pi(\boldsymbol{x_{0:k-1}}|\boldsymbol{y_{1:k-1}})}$$

$$\omega^{*(i)}_{k|k} \propto \left(\frac{1}{n}\right)\frac{p\left(\boldsymbol{y}_k\middle|\boldsymbol{x}^{(i)}{}_k\right)p\left(\boldsymbol{x}^{(i)}{}_k\middle|\boldsymbol{x}^{(i)}{}_{k-1}\right)}{\pi(\boldsymbol{x_k}|\boldsymbol{x}_{0:k-1}, \boldsymbol{y}_{1:k})}\,\omega^{*(i)}_{k-1|k-1}$$

Particle Filter – update

- **Goal**: sample approximation $\hat{p}(\boldsymbol{x_k}|\boldsymbol{y_{k-1}})$ and $\hat{p}(\boldsymbol{x_k}|\boldsymbol{y_k})$. Represent these by a set of weighted samples

- The normalized weights are then:

$$\omega_{k|k}^{(i)} = \frac{\omega^{*(i)}}{\sum \omega^{*(i)}}$$

- Expectation and covariance can then be computed:

$\boldsymbol{x}^{(i)}$ is sampled from $\pi(\boldsymbol{x_k}|\boldsymbol{x}_{0:k-1}, \boldsymbol{y}_{1:k})$

$$\mathbb{E}(\boldsymbol{x_k}|\boldsymbol{y_k}) \approx \sum_{i=1}^{n} \omega^{(i)} \overline{\boldsymbol{x}}^{(i)}$$

$$Cov(\boldsymbol{x_k}|\boldsymbol{y_k})$$

- Where samples $\overline{\boldsymbol{x}}^{(i)}$ are drawn from the importance distribution $\pi(\boldsymbol{x_k}|\boldsymbol{x}_{0:k-1}, \boldsymbol{y}_{1:k})$

- We can choose the importance distribution $\pi(\boldsymbol{x_k}|\boldsymbol{x}_{0:k-1}, \boldsymbol{y}_{1:k})$

  - Simple choice is the Bootstrap Particle Filter: $\pi(\boldsymbol{x_k}|\boldsymbol{x}_{0:k-1}, \boldsymbol{y}_{1:k}) = p\left(\boldsymbol{x}^{(i)}{}_k|\boldsymbol{x}^{(i)}{}_{k-1}\right)$

$$\omega_{k|k}^{(i)} = \eta p\left(\boldsymbol{y}_k|\boldsymbol{x}^{(i)}{}_k\right)$$

Example: Pendulum

- Non-linear Dynamics

$$\dot{x} = \frac{dx}{dt} = f_1$$
$$\ddot{x} = -\frac{g}{l}\sin(x) = f_2$$

- 1st-order Taylor Series

$$\mathbf{f}(\boldsymbol{x}) \approx \mathbf{f}(\bar{\boldsymbol{x}}) + \nabla\mathbf{f}\ \boldsymbol{\delta x}$$

$$\begin{bmatrix}\dot{x}\\ \ddot{x}\end{bmatrix} = \begin{bmatrix}\frac{dx}{dt}\\ -\frac{g}{l}\sin(x)\end{bmatrix}_{x=\bar{x}} + \begin{bmatrix}\frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial \dot{x}}\\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial \dot{x}}\end{bmatrix}_{x=\bar{\boldsymbol{x}}} \begin{bmatrix}(x-\bar{x})\\ (\dot{x}-\bar{\dot{x}})\end{bmatrix}$$

- Linearized Dynamics
  - Continuous

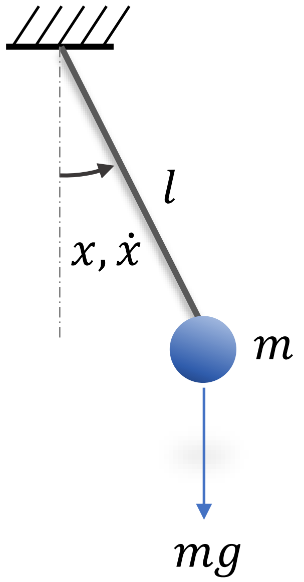
$$\begin{bmatrix}x(\dot{t})\\ x(\ddot{t})\end{bmatrix} \approx \begin{bmatrix}0 & 1\\ -\frac{g}{l}\cos(\bar{x}) & 0\end{bmatrix}\begin{bmatrix}x(t)\\ x(\dot{t})\end{bmatrix} + \boldsymbol{q}(t)$$

  - Discretized

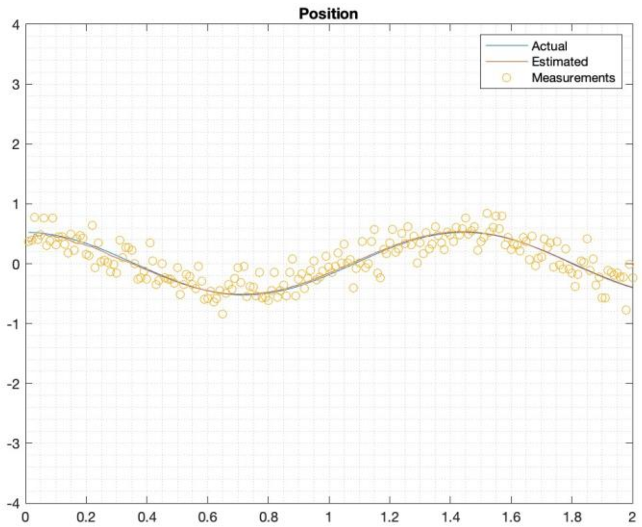
$$\begin{bmatrix}x_k\\ \dot{x}_k\end{bmatrix} \approx \begin{bmatrix}1 & \Delta t\\ -\frac{g}{l}\cos(\bar{x})\,\Delta t & 1\end{bmatrix}\begin{bmatrix}x_{k-1}\\ \dot{x}_{k-1}\end{bmatrix} + \boldsymbol{q}_{k-1}$$

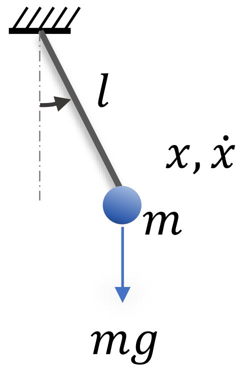
- Measurement Model

$$\boldsymbol{y_k} = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}x_{k-1}\\ \dot{x}_{k-1}\end{bmatrix} + \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\boldsymbol{r_k}$$

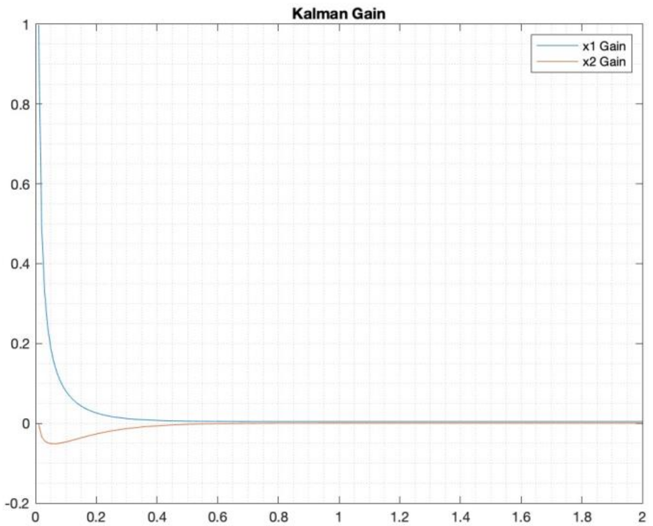


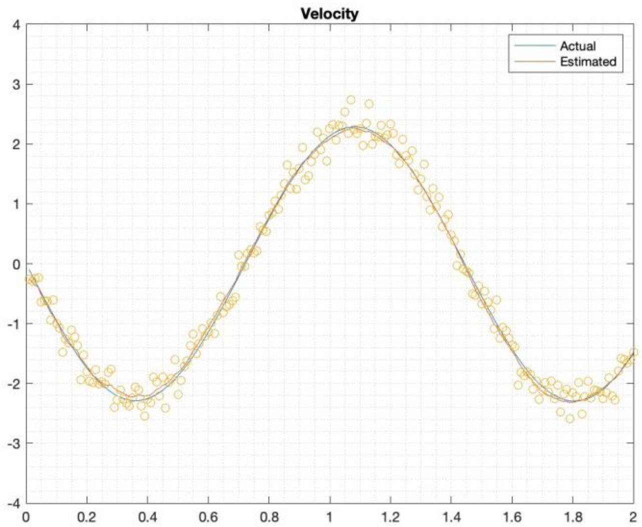
- Advantages
  - Computes mean and covariance working with distributions explicitly
- Limitations
  - Linearized processes
  - Additive, Gaussian distributions
  - Computational cost for high dimension





Time step, Δt	0.01	s
Length, l	0.5	m
Process variance	0.1	(rad, rad/s)
Measurement variance	0.2	(rad, rad/s)
Measurement period	0.01	s





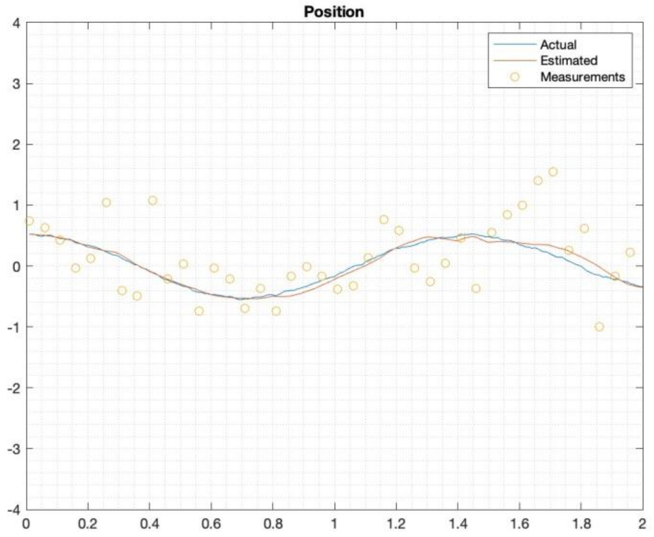
$$\begin{bmatrix} x_k \\ \dot{x}_k \end{bmatrix} \approx \begin{bmatrix} 1 & \Delta t \\ -\frac{g}{l}\cos(\bar{x})\,\Delta t & 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ \dot{x}_{k-1} \end{bmatrix} + \boldsymbol{q}_{k-1}$$

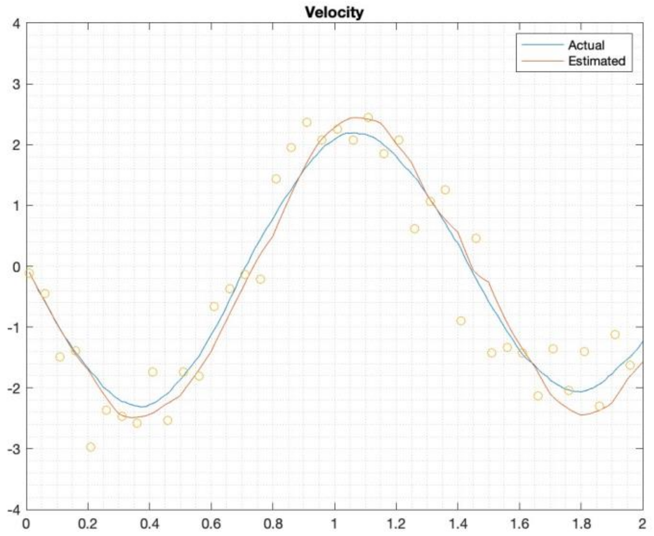
$$\boldsymbol{y_k} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ \dot{x}_{k-1} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{r_k}$$

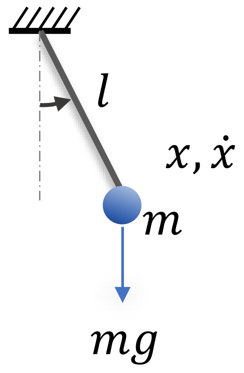
Example: Pendulum – Ensemble Kalman Filter

- Advantages
  - Non-linear dynamics
  - Arbitrary distributions

- Limitations
  - Trade computational cost (ensemble size) for accuracy
  - Sample covariance estimate is often error-prone







Time step, h_k	0.01	s
Length, l	0.5	m
Process variance	0.1	(rad, rad/s)
Measurement variance	0.2	(rad, rad/s)
Measurement period	0.05	s
Ensemble size	20	

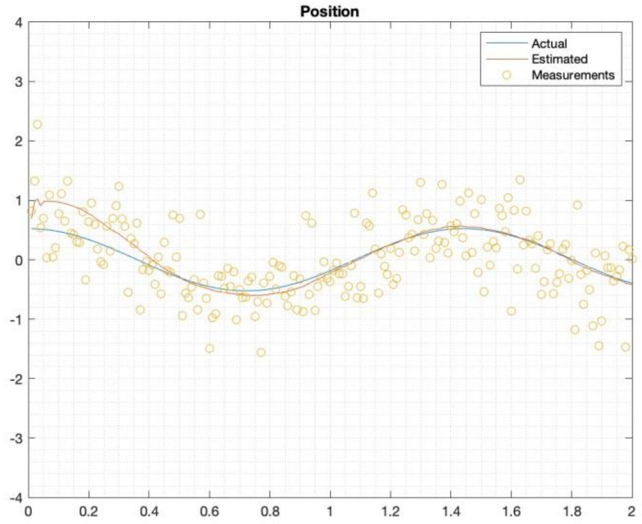
$$\ddot{x} = -\frac{g}{l}\sin(x)$$

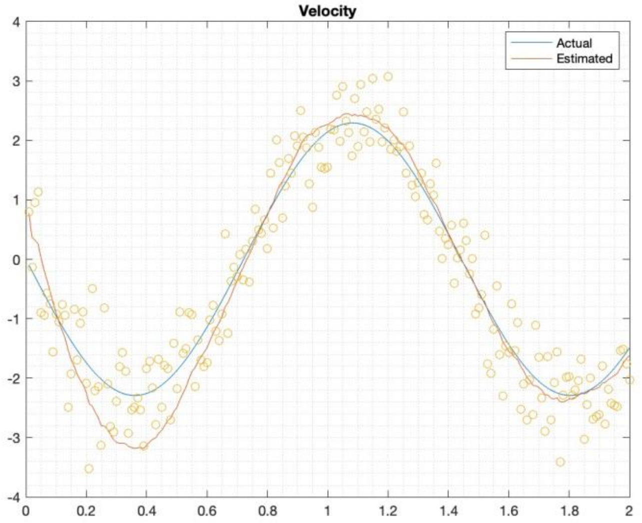
- $(k) =$
- $\dot{x}_2(k) = -\frac{g}{l}\sin(x_1)$
- $\boldsymbol{y_k} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ \dot{x}_{k-1} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{r_k}$

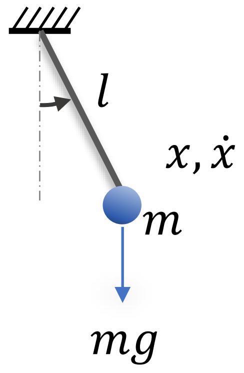
Example: Pendulum – Particle Filter

- Advantages
  - Non-linear dynamics
  - Arbitrary distributions

- Limitations
  - Trade computational cost (ensemble size) for accuracy







Time step, h_k	0.01	s
Length, l	0.5	m
Process variance	0.1	(rad, rad/s)
Measurement variance	0.2	(rad, rad/s)
Measurement period	0.01	s
Ensemble size		

$$\ddot{x} = -\frac{g}{l}\sin(x)$$

- $x_2 = \dot{x}_1$
- $\dot{x}_2 = -\frac{g}{l}\sin(x_1)$
- $\boldsymbol{y_k} = \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\begin{bmatrix}x_{k-1}\\\dot{x}_{k-1}\end{bmatrix} + \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\boldsymbol{r_k}$

Particle Filter - algorithm

Prediction:
sample $\boldsymbol{x}^{(\boldsymbol{i})}{}_{\boldsymbol{k}|\boldsymbol{k}-\mathbf{1}} \sim p\left(\boldsymbol{x}^{(i)}{}_k \middle| \boldsymbol{x}^{(i)}{}_{k-1}\right)$

Update:
update weights, $\omega^{*(i)} \propto p\left(\boldsymbol{y}_k \middle| \boldsymbol{x}^{(i)}{}_k\right)$

normalize: $\omega^{(i)} = \frac{\omega^{*(i)}}{\sum \omega^{*(i)}}$

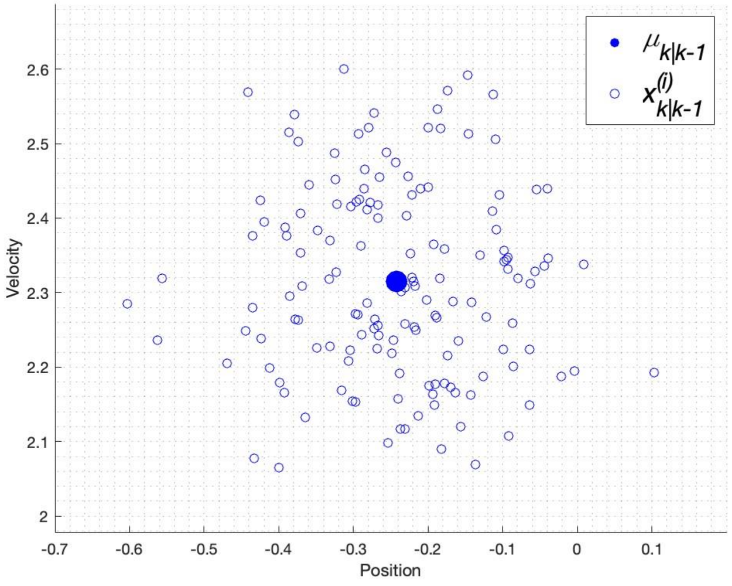
update belief:

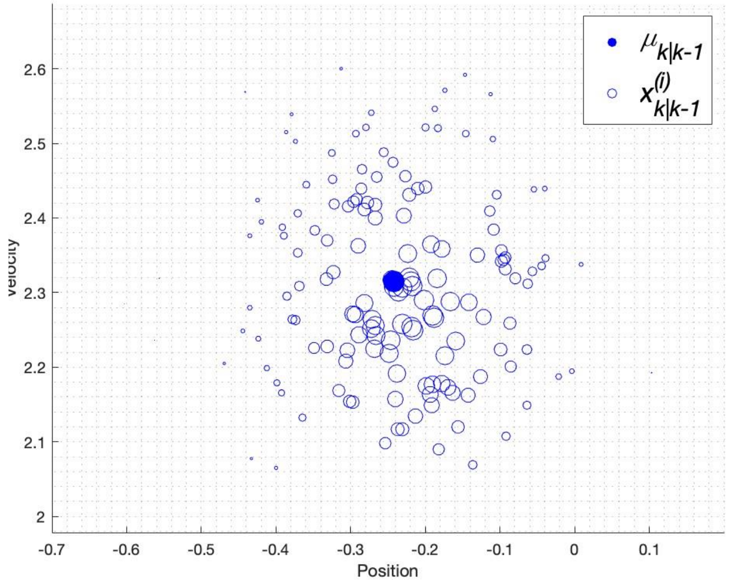
$$\mathbb{E}(\boldsymbol{x_k}|\boldsymbol{y_k}) \approx \sum_{i=1}^{n} \omega^{(i)} \overline{\boldsymbol{x}}^{(i)}$$

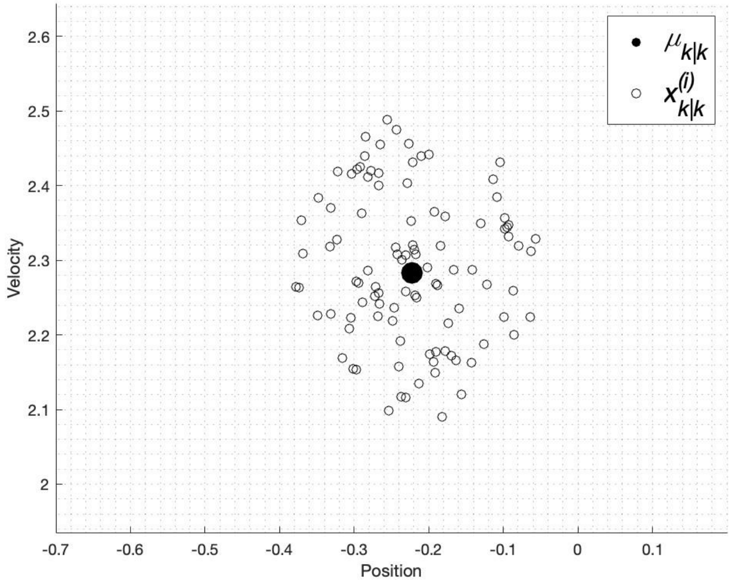
Resample:

Create new particle set:

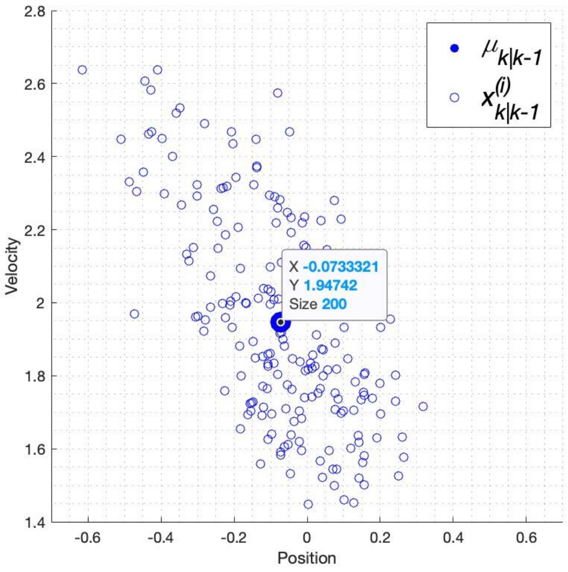
$$\left\{\boldsymbol{\omega}^{(j)} = \frac{1}{n}, \boldsymbol{x}^{(j)}{}_{\boldsymbol{k}} \mid j = 1, \dots, n\right\}$$

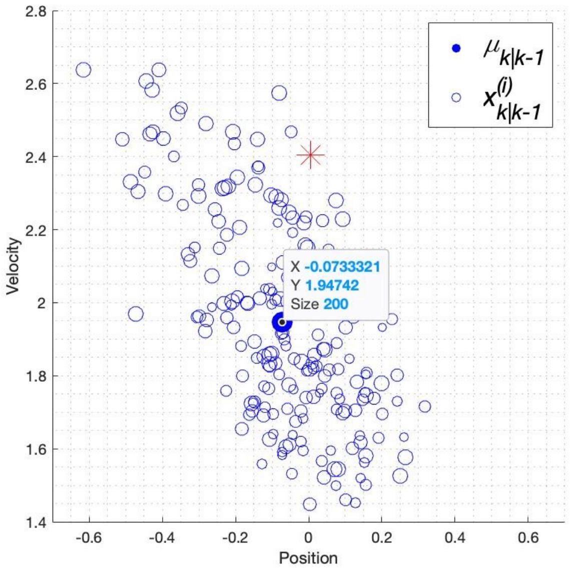


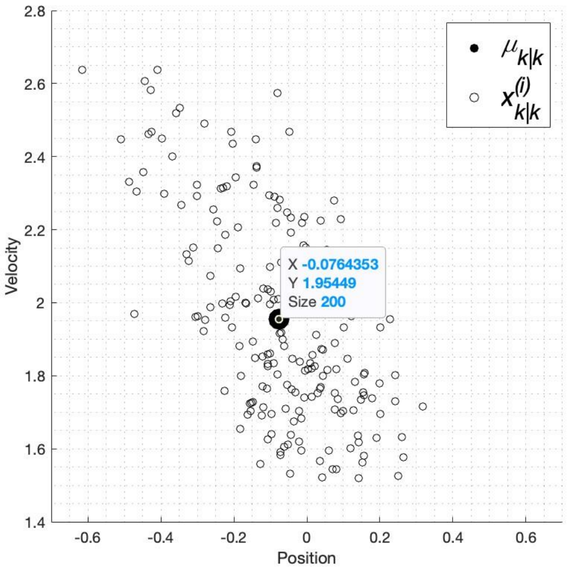




From illustrative example given in previous slides







Conditional GAN

- **Given**
  - Sample set of ground truth, paired state and measurement as training data

$$\{\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}\}_{i=1}^{n}$$

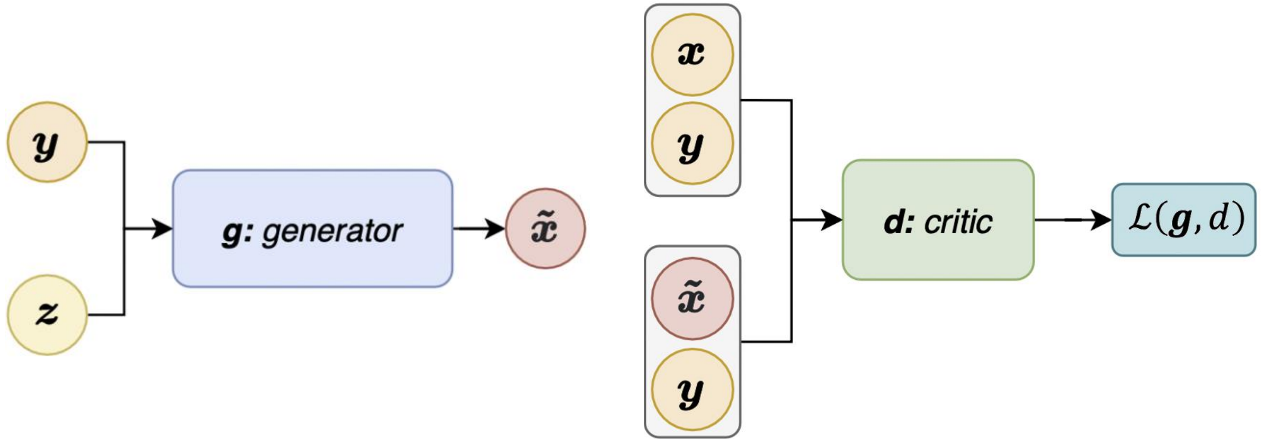
  - Measurement model

$$\boldsymbol{y} = \mathcal{F}(\boldsymbol{x}; \boldsymbol{\eta})$$

- **Objective:**
  - CGAN will learn the conditional distribution from which we can sample

$$\widetilde{\boldsymbol{x}} = \mathcal{F}^{-1}(\boldsymbol{y}) \approx \widetilde{\boldsymbol{p}}(\boldsymbol{x}_{\boldsymbol{k}} | \boldsymbol{y}_{\boldsymbol{1}:\boldsymbol{k}})$$

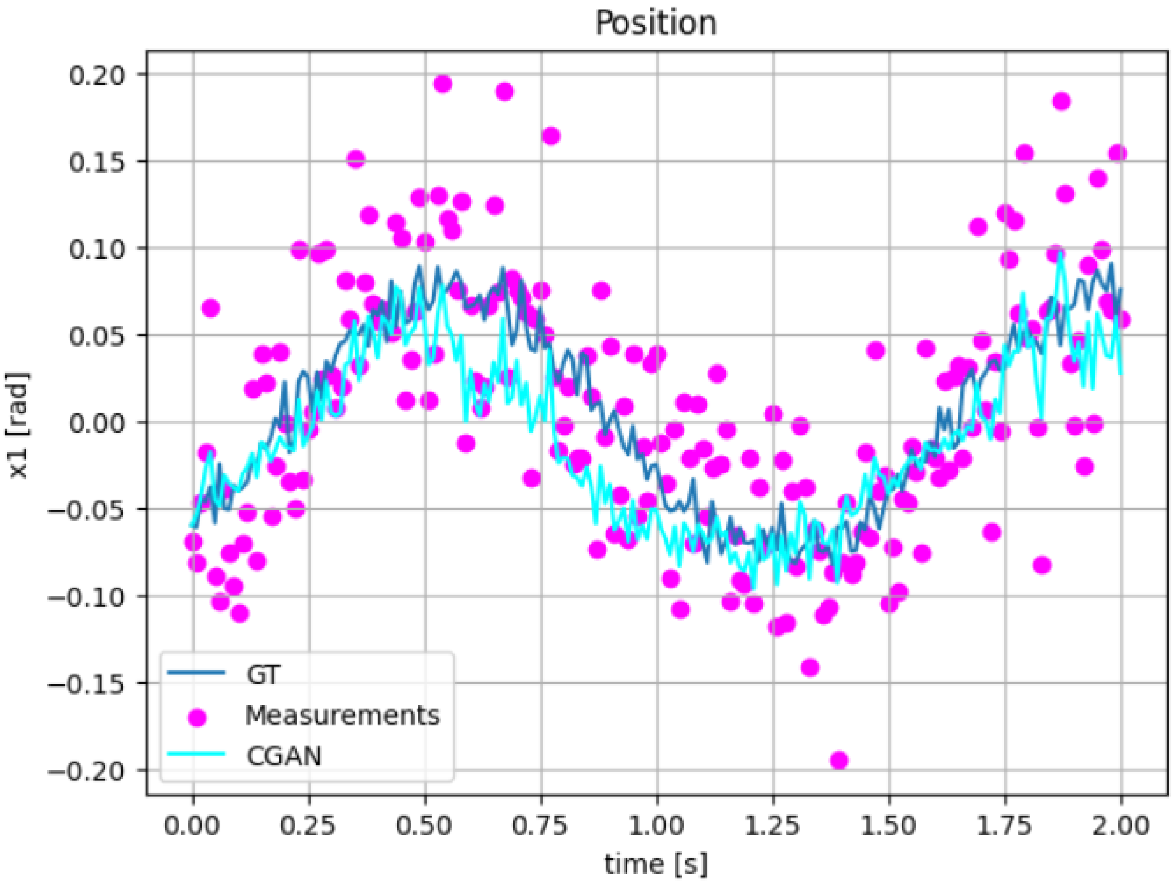
$$\boldsymbol{x} \sim \boldsymbol{\mu}_{\boldsymbol{X}|\boldsymbol{Y}}$$

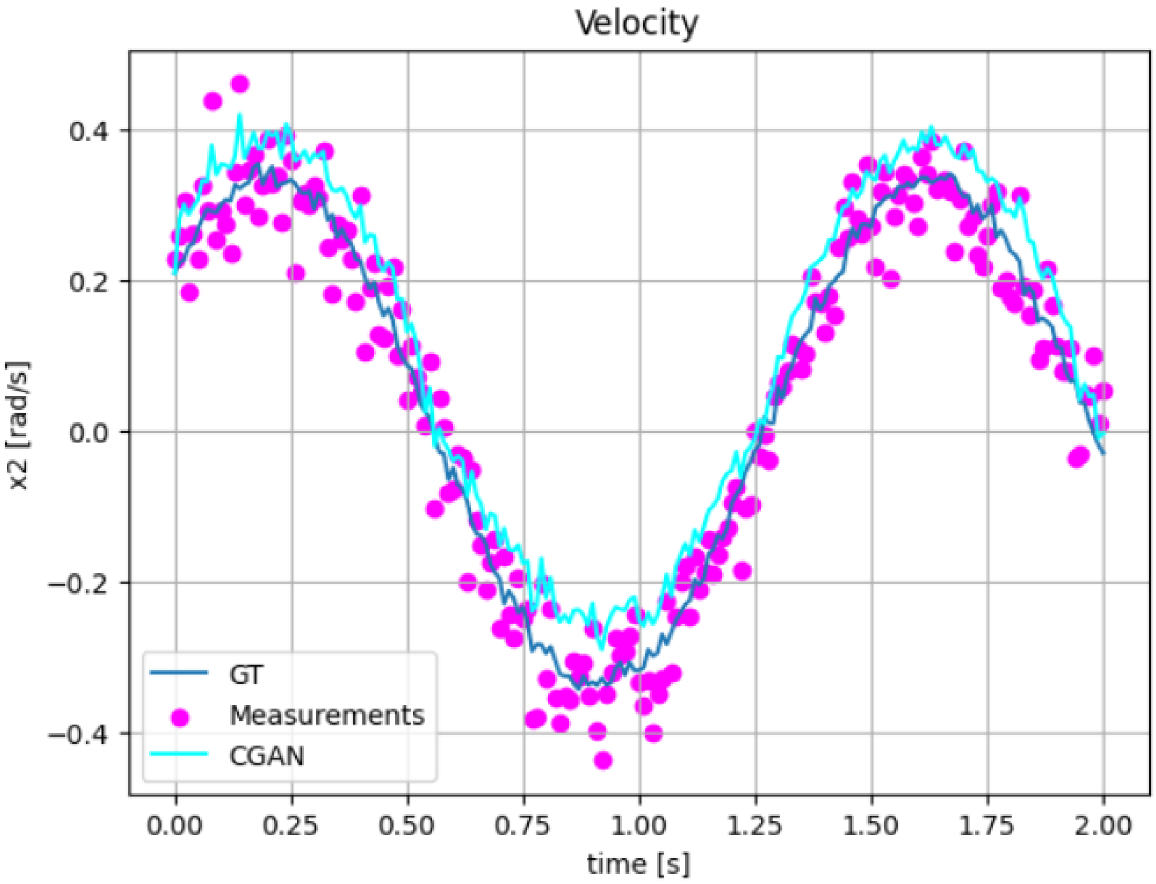


Generator network **Critic network**

Example: Pendulum - CGAN

- Results





Example: Pendulum - CGAN

- Results