Sequential Data II

CMPT 419/726
Mo Chen
SFU Computing Science
9/3/2020

Outline

- Goal: Review filtering, and consider the continuous state case

- Motivational Application: Localization
- Bayes' Filter
- Kalman Filter
- Extended Kalman Filter
- Particle Filter

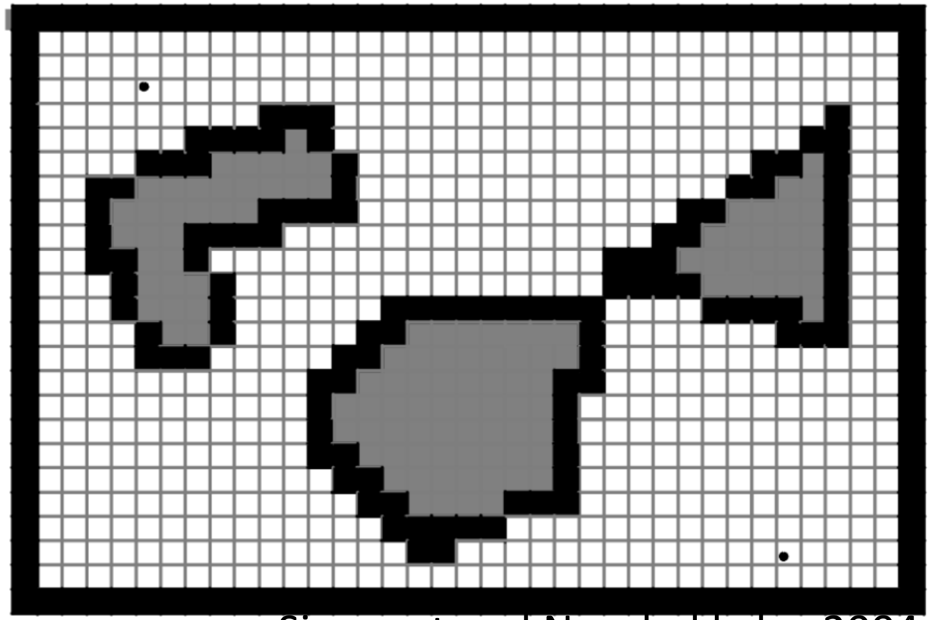
Outline

- Goal: Review filtering, and consider the continuous state case

- **Motivational Application: Localization**
- Bayes’ Filter
- Kalman Filter
- Extended Kalman Filter
- Particle Filter

Localization

- Assume a map is given: $m = \{m_1, m_2, \dots, m_N\}$
  - Location based: each m_i represents a specific location and whether it's occupied (eg. Occupancy grid)



Siegwart and Nourbakhshs, 2004

  - Feature based: each m_i contains the location and properties of a feature (eg. lighthouses, GPS)

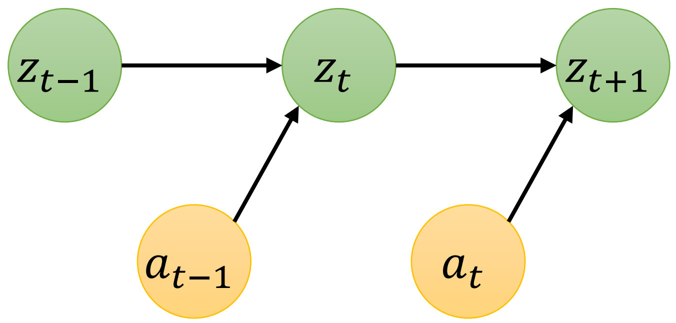
Localization

- Assume a map is given: $m = \{m_1, m_2, \dots, m_N\}$
  - Location based: each m_i represents a specific location and whether it's occupied (eg. Occupancy grid)
  - Feature based: each m_i contains the location and properties of a feature (eg. Topological map)

- Robot maintains and updates its belief about where it is with respect to the map
  - Position belief is updated based on sensor data
  - Position belief is a probability distribution

Robot-Environment Interaction: Definitions

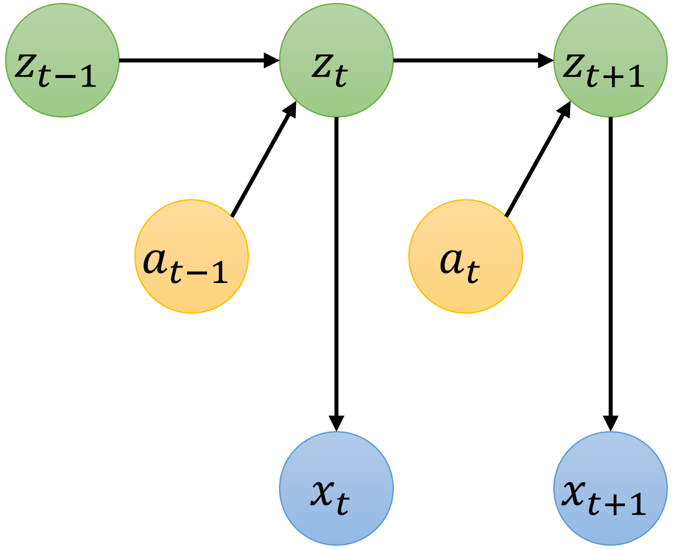
- State z_t: includes the environment (eg. objects, features)
  - Assume the state z_t is complete / the Markov property
- Control data a_t
  - Usually decreases robot's knowledge
- Probabilistic model of state evolution
  - Transition probabilities
  - System dynamics
  - $p(z_t | z_{t-1}, a_{t-1})$



Robot-Environment Interaction: Definitions

- Measurement data x_t
  - Increases robot's knowledge

- Measurement equation:
  - $p(x_t|z_t)$



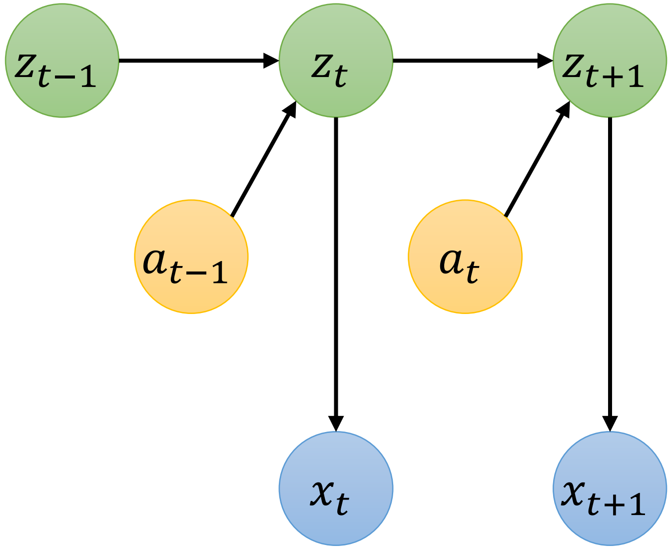
Prediction and Belief Distributions

- Prediction distribution:
  - Robot's prediction of the state before making an observation

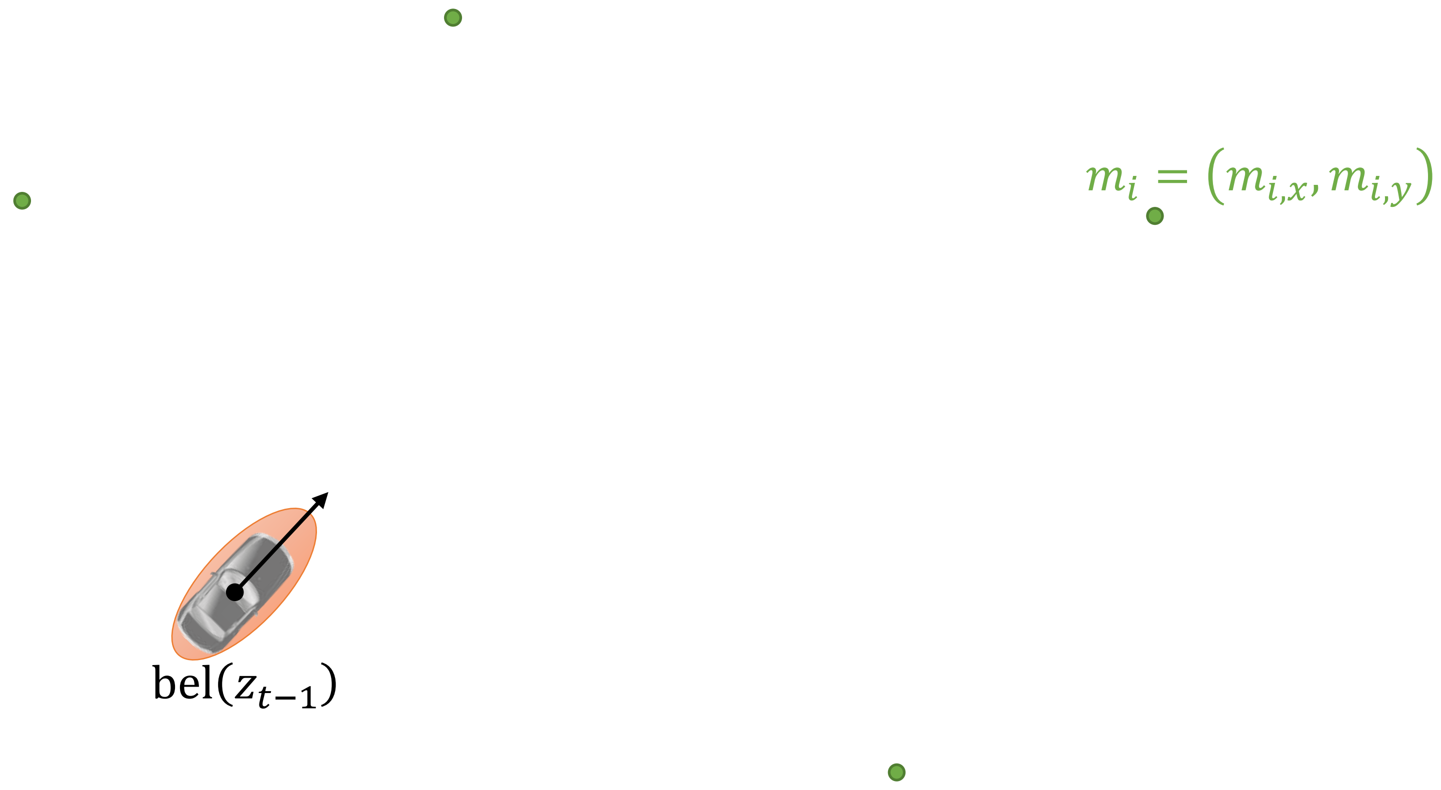
$$\overline{\mathrm{bel}}(z_t) \coloneqq p(z_t | x_{1:t-1}, a_{1:t-1})$$

- Belief distribution:
  - Robot's internal knowledge about the state

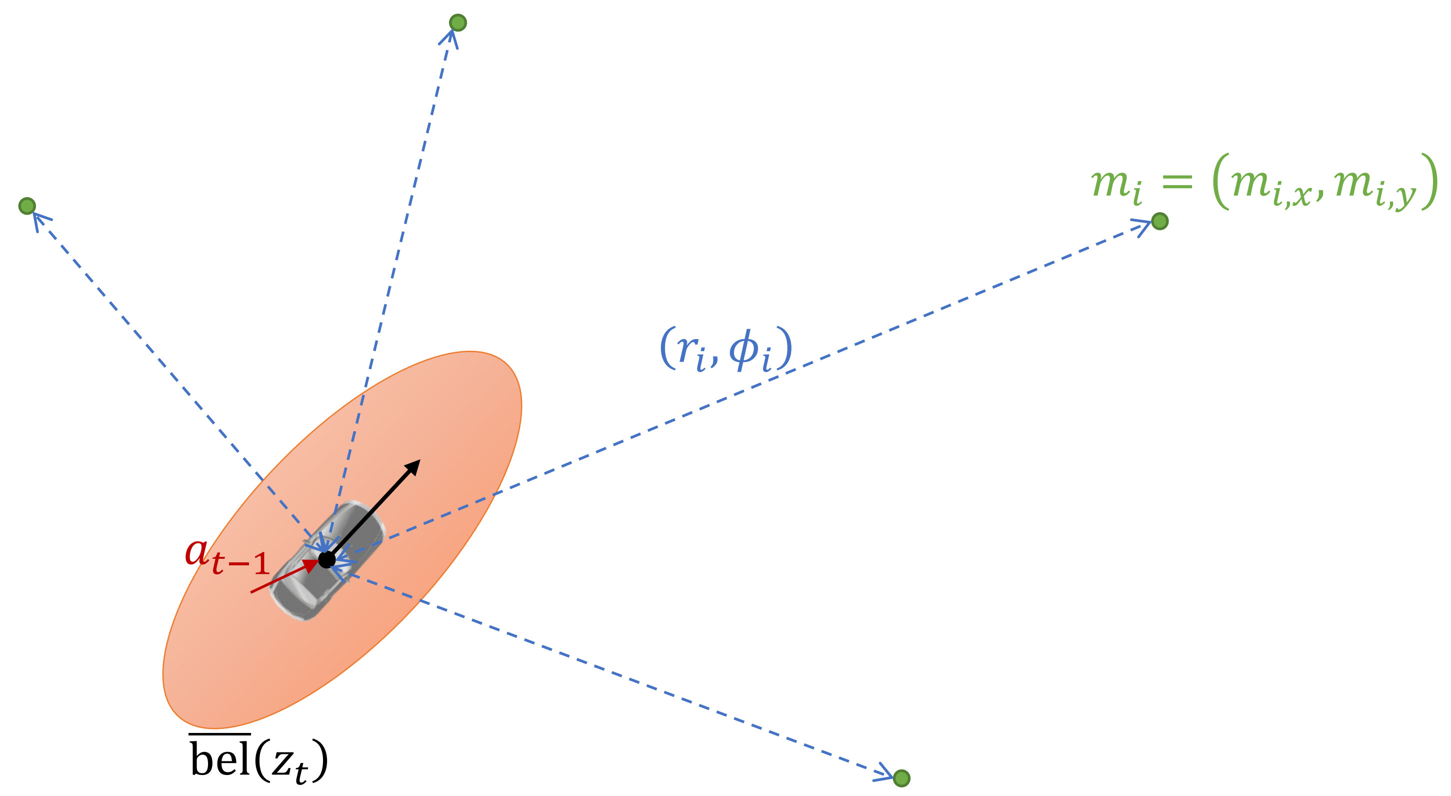
$$\mathrm{bel}(z_t) \coloneqq p(z_t | x_{1:t}, a_{1:t-1})$$



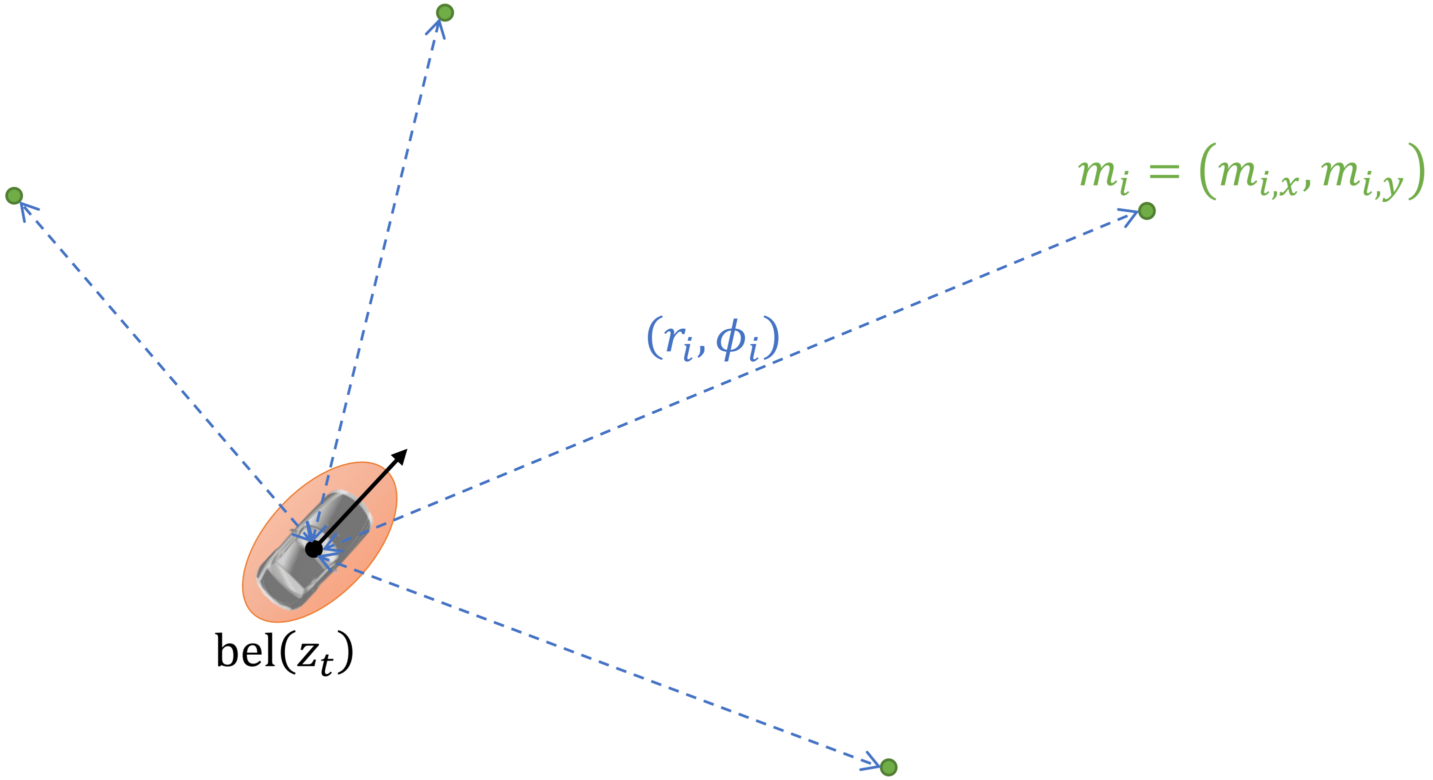
Localization



Localization



Localization



Outline

- Goal: Review filtering, and consider the continuous state case

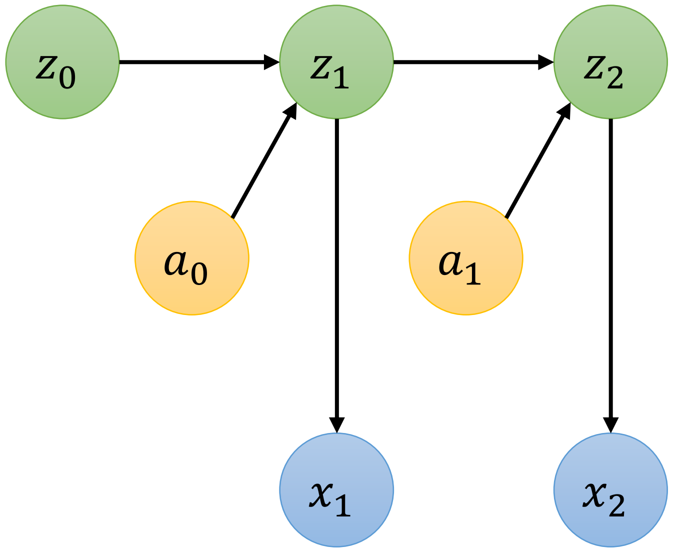
- Motivational Application: Localization
- **Bayes' Filter**
- Kalman Filter
- Extended Kalman Filter
- Particle Filter

Bayes Filter (Continuous)

- Robot and environment have state z_0
  - Initialize $\text{bel}(z_0)$ (eg. to be uniform or dirac distribution)

- From z_0, choose an action a_0 → robot moves to z_1
  1. Predict the next state by computing $\overline{\text{bel}}(z_1)$ using dynamics $p(z_t | z_{t-1}, a_{t-1})$
  2. Make an observation x_1, and use it to compute $\text{bel}(z_1)$

- Repeat for $z_2, z_3, \ldots$



Bayes Filter (Continuous)

- Robot and environment have state z_0
  - Initialize $\text{bel}(z_0)$ (eg. to be uniform or dirac distribution)

- From z_0, choose an action a_0 → robot moves to z_1
  1. Predict the next state by computing $\overline{\text{bel}}(z_1)$ using dynamics $p(z_t|z_{t-1}, a_{t-1})$
  2. Make an observation x_1, and use it to compute $\text{bel}(z_1)$

- Repeat for $z_2, z_3, \ldots$

- Bayes' filter algorithm:

Input: $\text{bel}(z_{t-1}), a_{t-1}, x_t$

Output: $\text{bel}(z_t)$

For every z_t,

Perform prediction:

$$\overline{\text{bel}}(z_t) = \int p(z_t|a_{t-1}, z_{t-1})\text{bel}(z_{t-1})dz_{t-1}$$

Perform measurement update:

$$\text{bel}(z_t) = \eta p(x_t|z_t)\overline{\text{bel}}(z_t)$$

Return $\text{bel}(z_t)$

Bayes Filter (Continuous)

$$\overline{\mathrm{bel}}(x_t) = p(z_t|x_{1:t-1}, a_{1:t-1})$$
$$= \int p(z_t|z_{t-1}, x_{1:t-1}, a_{1:t-1})p(z_{t-1}|x_{1:t-1}, a_{1:t-1})dz_{t-1}$$

Theorem of total probability

$$p(y) = \int p(x,y)dx = \int p(y|x)p(x)dx$$

- Bayes' filter algorithm:

Input: $\mathrm{bel}(z_{t-1}), a_{t-1}, x_t$

Output: $\mathrm{bel}(z_t)$

For every z_t,

Perform prediction:

$$\overline{\mathrm{bel}}(z_t) = \int p(z_t|a_{t-1}, z_{t-1})\mathrm{bel}(z_{t-1})dz_{t-1}$$

Perform measurement update:

$$\mathrm{bel}(z_t) = \eta p(x_t|z_t)\overline{\mathrm{bel}}(z_t)$$

Return $\mathrm{bel}(z_t)$

Bayes Filter (Continuous)

$$\overline{\text{bel}}(x_t) = p(z_t|x_{1:t-1}, a_{1:t-1})$$
$$= \int p(z_t|z_{t-1}, x_{1:t-1}, a_{1:t-1})p(z_{t-1}|x_{1:t-1}, a_{1:t-1})dz_{t-1}$$
$$= \int p(z_t|z_{t-1}, a_{t-1})p(z_{t-1}|x_{1:t-1}, a_{1:t-1})dz_{t-1}$$

Markov assumption

- Bayes' filter algorithm:

Input: $\text{bel}(z_{t-1}), a_{t-1}, x_t$

Output: $\text{bel}(z_t)$

For every z_t,

Perform prediction:

$$\overline{\text{bel}}(z_t) = \int p(z_t|a_{t-1}, z_{t-1})\text{bel}(z_{t-1})dz_{t-1}$$

Perform measurement update:

$$\text{bel}(z_t) = \eta p(x_t|z_t)\overline{\text{bel}}(z_t)$$

Return $\text{bel}(z_t)$

Bayes Filter (Continuous)

$$\overline{\text{bel}}(x_t) = p(z_t|x_{1:t-1}, a_{1:t-1})$$
$$= \int p(z_t|z_{t-1}, x_{1:t-1}, a_{1:t-1})p(z_{t-1}|x_{1:t-1}, a_{1:t-1})dz_{t-1}$$
$$= \int p(z_t|z_{t-1}, a_{t-1})p(z_{t-1}|x_{1:t-1}, a_{1:t-1})dz_{t-1}$$
$$= \int p(z_t|z_{t-1}, a_{t-1})p(z_{t-1}|x_{1:t-1}, a_{1:t-2})dz_{t-1}$$

a_{t-1} does not affect probability of z_{t-1}

- Bayes' filter algorithm:

Input: $\text{bel}(z_{t-1}), a_{t-1}, x_t$
Output: $\text{bel}(z_t)$
For every z_t,
    Perform prediction:
    $\overline{\text{bel}}(z_t) = \int p(z_t|a_{t-1}, z_{t-1})\text{bel}(z_{t-1})dz_{t-1}$
    Perform measurement update:
    $\text{bel}(z_t) = \eta p(x_t|z_t)\overline{\text{bel}}(z_t)$
Return $\text{bel}(z_t)$

Bayes Filter (Continuous)

$$\overline{\mathrm{bel}}(x_t) = p(z_t|x_{1:t-1}, a_{1:t-1})$$
$$= \int p(z_t|z_{t-1}, x_{1:t-1}, a_{1:t-1})p(z_{t-1}|x_{1:t-1}, a_{1:t-1})dz_{t-1}$$
$$= \int p(z_t|z_{t-1}, a_{t-1})p(z_{t-1}|x_{1:t-1}, a_{1:t-1})dz_{t-1}$$
$$= \int p(z_t|z_{t-1}, a_{t-1})p(z_{t-1}|x_{1:t-1}, a_{1:t-2})dz_{t-1}$$
$$= \int p(z_t|z_{t-1}, a_{t-1})\mathrm{bel}(z_{t-1})dz_{t-1}$$

- Bayes' filter algorithm:

Input: $\mathrm{bel}(z_{t-1}), a_{t-1}, x_t$

Output: $\mathrm{bel}(z_t)$

For every z_t,

Perform prediction:

$$\overline{\mathrm{bel}}(z_t) = \int p(z_t|a_{t-1}, z_{t-1})\mathrm{bel}(z_{t-1})dz_{t-1}$$

Perform measurement update:

$$\mathrm{bel}(z_t) = \eta p(x_t|z_t)\overline{\mathrm{bel}}(z_t)$$

Return $\mathrm{bel}(z_t)$

Bayes Filter (Continuous)

$$\overline{\text{bel}}(x_t) = p(z_t|x_{1:t-1}, a_{1:t-1})$$
$$= \int p(z_t|z_{t-1}, x_{1:t-1}, a_{1:t-1})p(z_{t-1}|x_{1:t-1}, a_{1:t-1})dz_{t-1}$$
$$= \int p(z_t|z_{t-1}, a_{t-1})p(z_{t-1}|x_{1:t-1}, a_{1:t-1})dz_{t-1}$$
$$= \int p(z_t|z_{t-1}, a_{t-1})p(z_{t-1}|x_{1:t-1}, a_{1:t-2})dz_{t-1}$$
$$= \int p(z_t|z_{t-1}, a_{t-1})\text{bel}(z_{t-1})dz_{t-1}$$

$$\text{bel}(z_t) = p(z_t|x_{1:t}, a_{1:t-1})$$
$$= \frac{p(x_t|z_t, x_{1:t-1}, a_{1:t-1})p(z_t|x_{1:t-1}, a_{1:t-1})}{p(x_t|x_{1:t-1}, a_{1:t-1})}$$

Bayes' rule

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)}$$

- Bayes' filter algorithm:

Input: $\text{bel}(z_{t-1}), a_{t-1}, x_t$

Output: $\text{bel}(z_t)$

For every z_t,

Perform prediction:

$$\overline{\text{bel}}(z_t) = \int p(z_t|a_{t-1}, z_{t-1})\text{bel}(z_{t-1})dz_{t-1}$$

Perform measurement update:

$$\text{bel}(z_t) = \eta p(x_t|z_t)\overline{\text{bel}}(z_t)$$

Return $\text{bel}(z_t)$

Bayes Filter (Continuous)

$$\overline{\text{bel}}(x_t) = p(z_t|x_{1:t-1}, a_{1:t-1})$$
$$= \int p(z_t|z_{t-1}, x_{1:t-1}, a_{1:t-1})p(z_{t-1}|x_{1:t-1}, a_{1:t-1})dz_{t-1}$$
$$= \int p(z_t|z_{t-1}, a_{t-1})p(z_{t-1}|x_{1:t-1}, a_{1:t-1})dz_{t-1}$$
$$= \int p(z_t|z_{t-1}, a_{t-1})p(z_{t-1}|x_{1:t-1}, a_{1:t-2})dz_{t-1}$$
$$= \int p(z_t|z_{t-1}, a_{t-1})\text{bel}(z_{t-1})dz_{t-1}$$

$$\text{bel}(z_t) = p(z_t|x_{1:t}, a_{1:t-1})$$
$$= \frac{p(x_t|z_t, x_{1:t-1}, a_{1:t-1})p(z_t|x_{1:t-1}, a_{1:t-1})}{p(x_t|x_{1:t-1}, a_{1:t-1})}$$
$$= \eta p(x_t|z_t)p(z_t|x_{1:t-1}, a_{1:t-1})$$

Markov property

- Bayes' filter algorithm:

Input: $\text{bel}(z_{t-1}), a_{t-1}, x_t$

Output: $\text{bel}(z_t)$

For every z_t,

Perform prediction:

$$\overline{\text{bel}}(z_t) = \int p(z_t|a_{t-1}, z_{t-1})\text{bel}(z_{t-1})dz_{t-1}$$

Perform measurement update:

$$\text{bel}(z_t) = \eta p(x_t|z_t)\overline{\text{bel}}(z_t)$$

Return $\text{bel}(z_t)$

Bayes Filter (Continuous)

$$\overline{\text{bel}}(x_t) = p(z_t|x_{1:t-1}, a_{1:t-1})$$
$$= \int p(z_t|z_{t-1}, x_{1:t-1}, a_{1:t-1})p(z_{t-1}|x_{1:t-1}, a_{1:t-1})dz_{t-1}$$
$$= \int p(z_t|z_{t-1}, a_{t-1})p(z_{t-1}|x_{1:t-1}, a_{1:t-1})dz_{t-1}$$
$$= \int p(z_t|z_{t-1}, a_{t-1})p(z_{t-1}|x_{1:t-1}, a_{1:t-2})dz_{t-1}$$
$$= \int p(z_t|z_{t-1}, a_{t-1})\text{bel}(z_{t-1})dz_{t-1}$$

$$\text{bel}(z_t) = p(z_t|x_{1:t}, a_{1:t-1})$$
$$= \frac{p(x_t|z_t, x_{1:t-1}, a_{1:t-1})p(z_t|x_{1:t-1}, a_{1:t-1})}{p(x_t|x_{1:t-1}, a_{1:t-1})}$$
$$= \eta p(x_t|z_t)p(z_t|x_{1:t-1}, a_{1:t-1})$$
$$= \eta p(x_t|z_t)\overline{\text{bel}}(z_t)$$

- Bayes' filter algorithm:

Input: $\text{bel}(z_{t-1}), a_{t-1}, x_t$
Output: $\text{bel}(z_t)$
For every z_t,
  Perform prediction:
  $\overline{\text{bel}}(z_t) = \int p(z_t|z_{t-1}, a_{t-1})\text{bel}(z_{t-1})dz_{t-1}$
  Perform measurement update:
  $\text{bel}(z_t) = \eta p(x_t|z_t)\overline{\text{bel}}(z_t)$
Return $\text{bel}(z_t)$

Bayes Filter

- Continuous state space: Closed-form $\mathrm{bel}(z_t)$ is unlikely. Need discretization and interpolation
- Can discretize z_t, but still must iterate through every z_t
    - Recall if z_t has K possible values, each prediction and measurement update is $O(K^2)$
    - Number of states is exponential in state space dimension
    - If there are n state variables (e.g. x-position, y-position, θ heading), and we have M discrete points per variable, then $K = M^n$

Bayes Filter

- Solution: exploit structure and/or make assumptions
- Parametric filters: assume a form for distributions
- Non-parametric filters: represent distributions using samples

Parametric and Non-parametric Filters

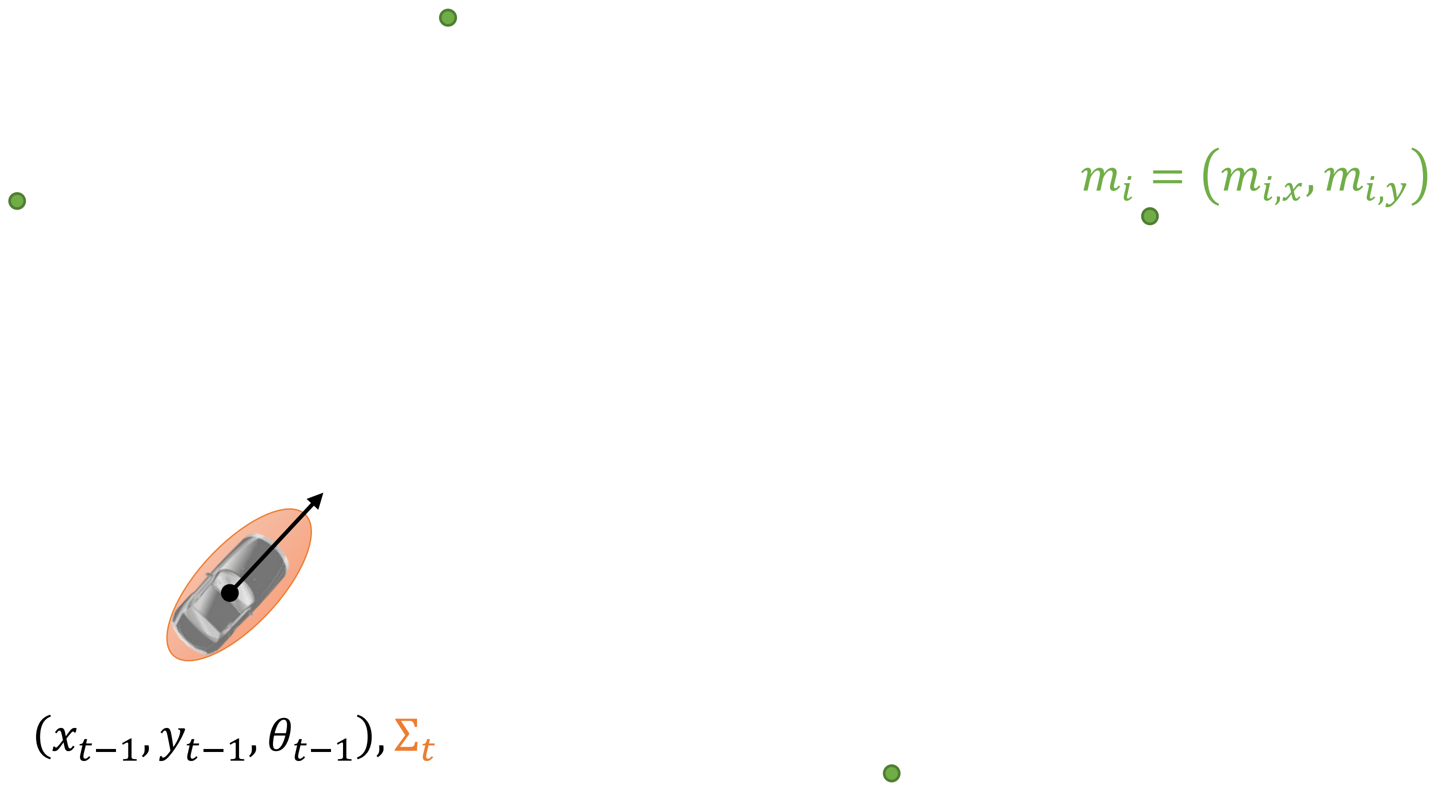
- Kalman Filter
  - Parametric filter for linear systems and measurement models
- Extended Kalman Filter
  - Extension to nonlinear systems and measurement models
- Unscented Kalman Filter
  - (Somewhat) non-parametric filter
- Particle Filter
  - Non-parametric filter

Outline

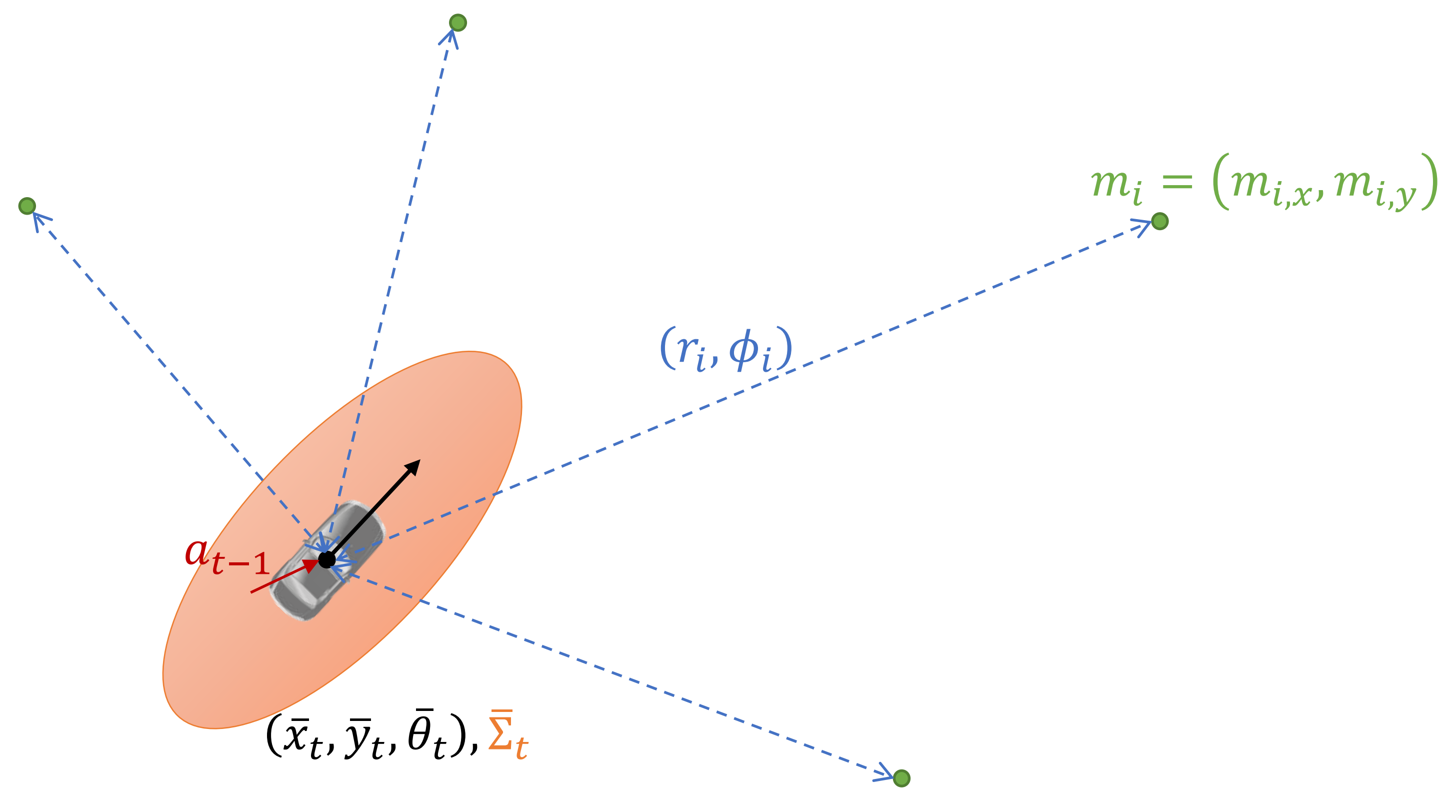
- Goal: Review filtering, and consider the continuous state case

- Motivational Application: Localization
- Bayes' Filter
- **Kalman Filter**
- Extended Kalman Filter
- Particle Filter

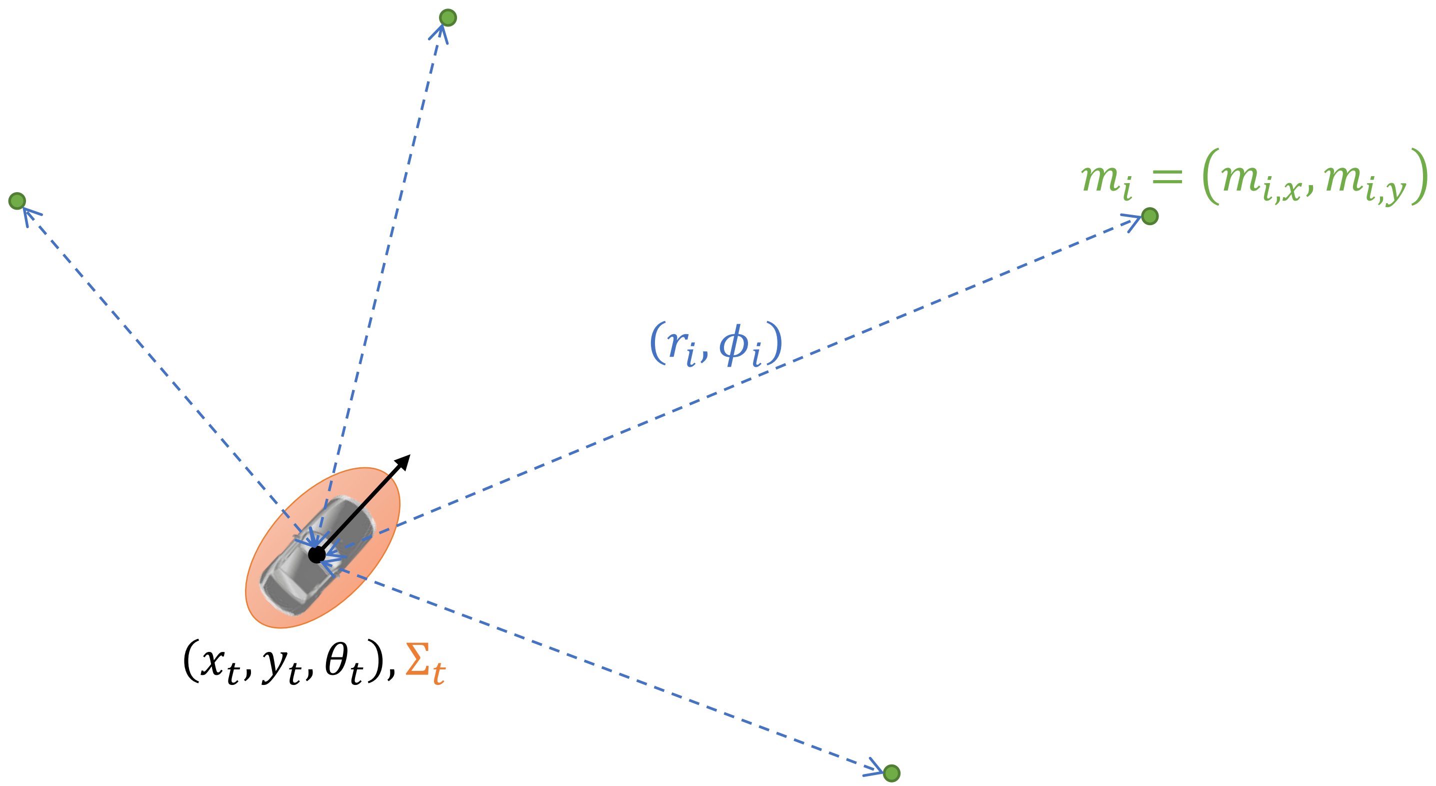
Localization



Localization



Localization



Kalman Filter

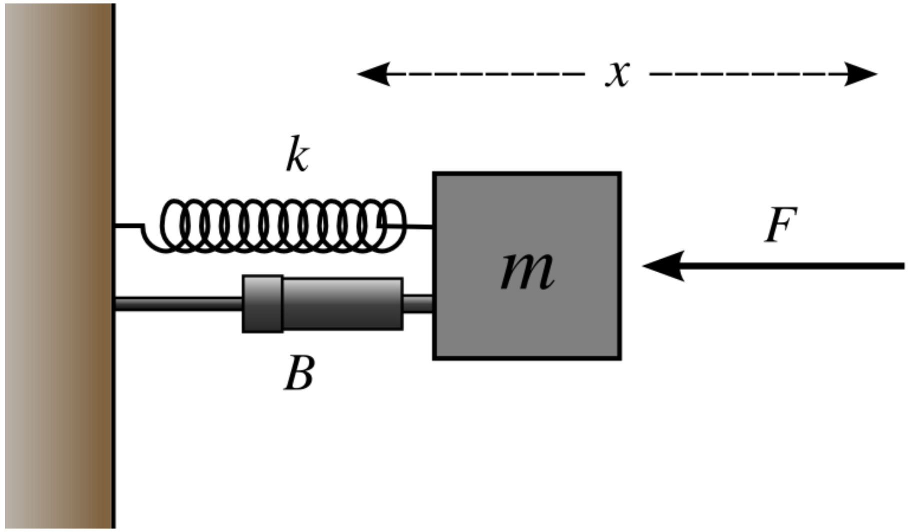
- Bayes filter with additional assumptions

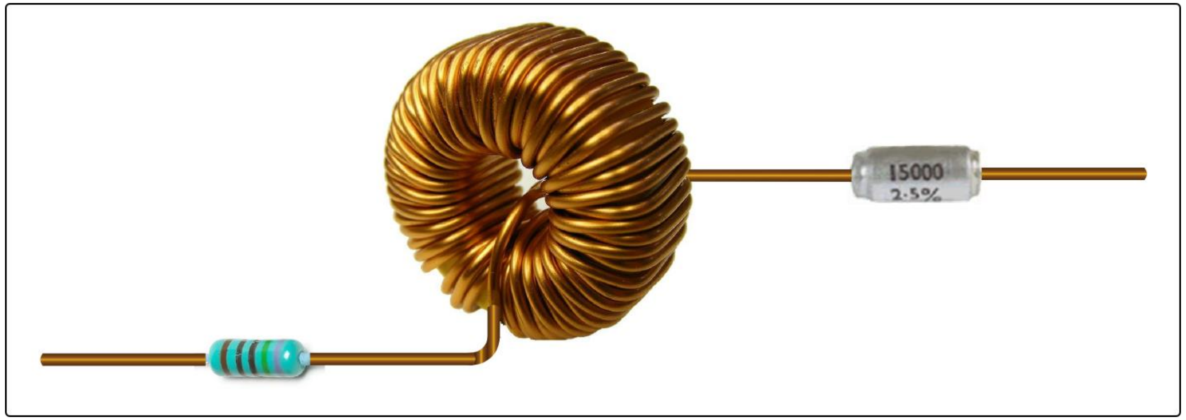
1. Initial Gaussian belief
   - $\mathrm{bel}(z_0) \sim N(\mu_0, \Sigma_0)$
   - Approximates single-modal distributions well

2. Linear system dynamics (transition model) with Gaussian noise
   - $z_t = Az_{t-1} + Ba_{t-1} + \epsilon_t$
   - Noise is independent $\epsilon_t \sim N(0, R_t)$

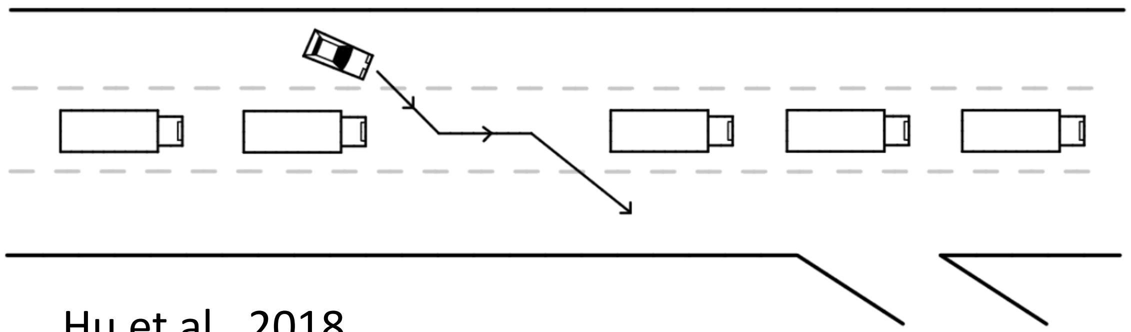
3. Linear measurement model
   - $x_t = C_t z_t + \delta_t, \delta_t \sim N(0, Q_t)$

Linear Systems





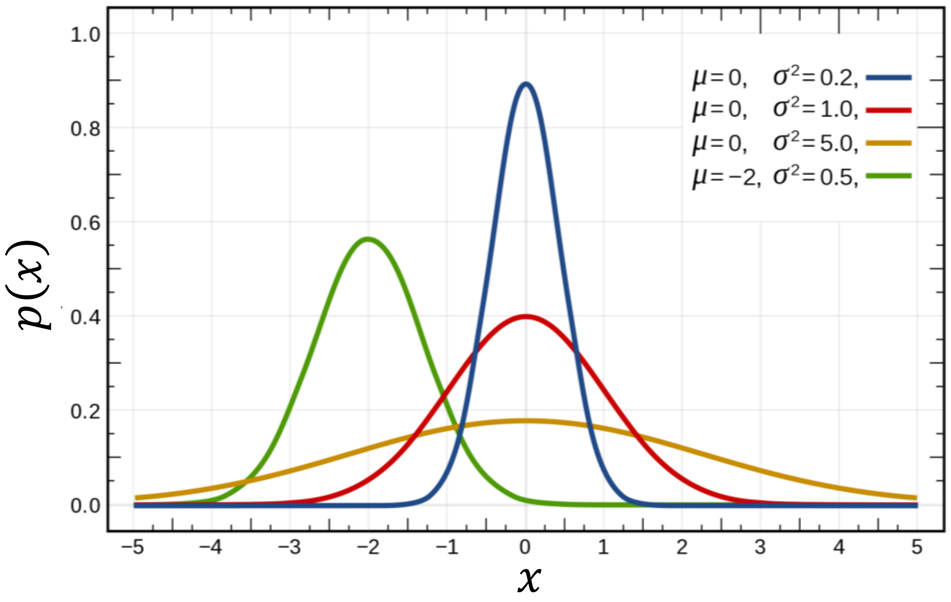
Linear Systems



Hu et al., 2018

(If flying near hover, and slowly)
Bouffard, 2012

Gaussian Distributions

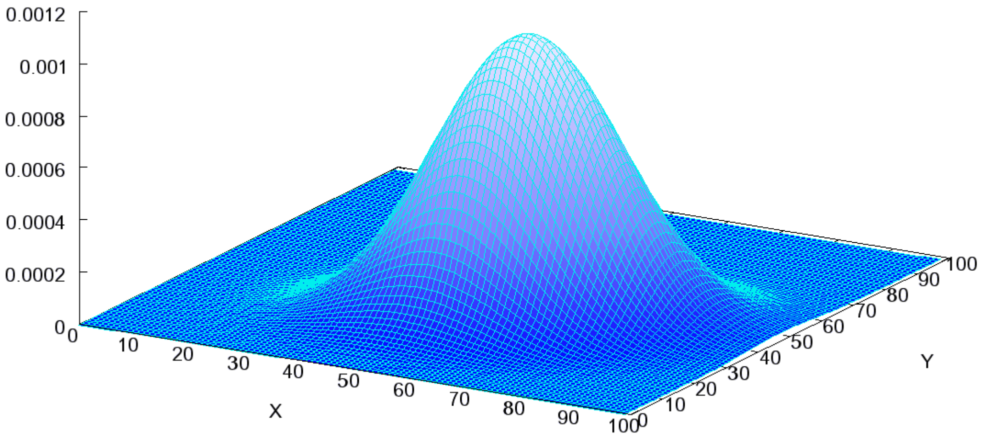


- Probability density function, scalar case:

$$p(x) = (2\pi\sigma^2)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\right) \sim N(\mu, \sigma^2)$$

- Probability density function, vector case:

$$p(x) = \det(2\pi\Sigma)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right) \sim N(\mu, \Sigma)$$



Key Properties Needed

- If $X \sim N(\mu, \Sigma)$, and $Y = AX + b$, then
$$Y \sim N(A\mu + b, A\Sigma A^\top)$$

- If $X_1 \sim N(\mu_1, \Sigma_1), X_2 \sim N(\mu_2, \Sigma_2)$, and $Y = X_1 + X_2$, then
$$Y \sim N(\mu_1 + \mu_2, \Sigma_1 + \Sigma_2)$$

- Product of Gaussian probability distribution functions is also Gaussian
    - More complicated expression/derivation

Result of Assumptions and Gaussian Distribution Properties

1. Gaussian initial belief:

$$\mathrm{bel}(z_0) = p(z_0) = \det(2\pi\Sigma_0)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(z_0 - \mu_0)^\top \Sigma_0^{-1}(z_0 - \mu_0)\right)$$

2. Linear dynamics $z_t = Az_{t-1} + Ba_{t-1} + \epsilon_t$, $\epsilon_t \sim N(0, R_t)$ implies

$$p(z_t | z_{t-1}, a_{t-1}) = \det(2\pi R_t)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(z_t - Az_{t-1} - Ba_{t-1})^\top R_t^{-1}(z_t - Az_{t-1} - Ba_{t-1})\right)$$

3. Linear measurement model $x_t = C_t z_t + \delta_t, \delta_t \sim N(0, Q_t)$ implies

$$p(x_t | z_t) = \det(2\pi Q_t)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x_t - C_t z_t)^\top Q_t^{-1}(x_t - C_t z_t)\right)$$

- Result: Posterior belief $\mathrm{bel}(z_t)$ is Gaussian for all t!
  - Start with $\mathrm{bel}(z_0) \sim N(\mu_0, \Sigma_0)$, obtain $\mathrm{bel}(z_t) \sim N(\mu_t, \Sigma_t)$ from $\mathrm{bel}(z_{t-1}) \sim N(\mu_{t-1}, \Sigma_{t-1})$
  - Only the parameters μ_t and Σ_t need to be updated to capture distribution over all z_t

Kalman Filter

- Bayes' filter algorithm:

Input: $\text{bel}(z_{t-1}), a_{t-1}, x_t$
Output: $\text{bel}(z_t)$
For every z_t,

Perform prediction:

$\overline{\text{bel}}(z_t) = \int p(z_t|z_{t-1}, a_{t-1})\text{bel}(z_{t-1})dz_{t-1}$

Perform measurement update:

$\text{bel}(z_t) = \eta p(x_t|z_t)\overline{\text{bel}}(z_t)$

Return $\text{bel}(z_t)$

- Kalman filter algorithm:

Input: $\mu_{t-1}, \Sigma_{t-1}, u_{t-1}, x_t$
Output: μ_t, Σ_t
Perform prediction:

Perform measurement update:

Return μ_t, Σ_t

Key Properties of Gaussian Distributions

- If $X \sim N(\mu, \Sigma)$, and $Y = AX + b$, then

$$Y \sim N(A\mu + b, A\Sigma A^\top)$$

- If $X_1 \sim N(\mu_1, \Sigma_1), X_2 \sim N(\mu_2, \Sigma_2)$, and $Y = X_1 + X_2$, then

$$Y \sim N(\mu_1 + \mu_2, \Sigma_1 + \Sigma_2)$$

- Linear dynamics: $z_t = Az_{t-1} + Ba_{t-1} + \epsilon_t$, $\epsilon_t \sim N(0, R_t)$
- If $z_{t-1} \sim N(\mu_{t-1}, \Sigma_{t-1})$ then $z_t \sim (\bar{\mu}_t, \bar{\Sigma}_t)$, where
  - $\bar{\mu}_t = A\mu_{t-1} + Ba_{t-1}$
  - $\bar{\Sigma}_t = A\Sigma_{t-1}A^\top + R_t$

Kalman Filter

- Bayes' filter algorithm:

Input: $\text{bel}(z_{t-1}), a_{t-1}, x_t$
Output: $\text{bel}(z_t)$
For every z_t,

Perform prediction:

$\overline{\text{bel}}(z_t) = \int p(z_t|z_{t-1}, a_{t-1})\text{bel}(z_{t-1})dz_{t-1}$

Perform measurement update:

$\text{bel}(z_t) = \eta p(x_t|z_t)\overline{\text{bel}}(z_t)$

Return $\text{bel}(z_t)$

- Kalmn filter algorithm:

Input: $\mu_{t-1}, \Sigma_{t-1}, a_{t-1}, x_t$
Output: μ_t, Σ_t
Perform prediction:

$$\bar{\mu}_t = A\mu_{t-1} + Ba_{t-1}$$
$$\bar{\Sigma}_t = A\Sigma_{t-1}A^\top + R_t$$

Perform measurement update:

Return μ_t, Σ_t

Key Property of Gaussian Distributions

- Product of Gaussian probability distribution functions are also Gaussian random variables
  - More complicated expression/derivation

- Linear measurement model
  - $x_t = C_t z_t + \delta_t, \delta_t \sim N(0, Q_t)$
- Measurement update: $\underbrace{\mathrm{bel}(z_t)}_{\text{Gaussian } N(\mu_t, \Sigma_t)} = \underset{\text{constant}}{\eta} \overbrace{p(x_t|z_t)}^{\text{Gaussian } N(Cz_t, Q_t)} \overbrace{\overline{\mathrm{bel}}(z_t)}^{\text{Gaussian } N(\bar{\mu}_t, \bar{\Sigma}_t)}$

  - $K_t = \bar{\Sigma}_t C_t^\top (C_t \bar{\Sigma}_t C_t^\top + Q_t)^{-1}$
  - $\mu_t = \bar{\mu}_t + K_t(x_t - C_t \bar{\mu}_t)$
  - $\Sigma_t = (I - K_t C_t)\bar{\Sigma}_t$

Kalman Filter

- Bayes' filter algorithm:

Input: $\mathrm{bel}(z_{t-1}), a_{t-1}, x_t$
Output: $\mathrm{bel}(z_t)$
For every z_t,

Perform prediction:

$$\overline{\mathrm{bel}}(z_t) = \int p(z_t|z_{t-1}, a_{t-1})\mathrm{bel}(z_{t-1})dz_{t-1}$$

Perform measurement update:

$$\mathrm{bel}(z_t) = \eta p(x_t|z_t)\overline{\mathrm{bel}}(z_t)$$

Return $\mathrm{bel}(z_t)$

- Kalmn filter algorithm:

Input: $\mu_{t-1}, \Sigma_{t-1}, a_{t-1}, x_t$
Output: μ_t, Σ_t
Perform prediction:

$$\bar{\mu}_t = A\mu_{t-1} + Ba_{t-1}$$
$$\bar{\Sigma}_t = A\Sigma_{t-1}A^\top + R_t$$

Perform measurement update:

$$K_t = \bar{\Sigma}_t C_t^\top (C_t \bar{\Sigma}_t C_t^\top + Q_t)^{-1}$$
$$\mu_t = \bar{\mu}_t + K_t(x_t - C_t\bar{\mu}_t)$$
$$\Sigma_t = (I - K_t C_t)\bar{\Sigma}_t$$

Return μ_t, Σ_t

Kalman Filter: Discussion

- "**Kalman gain**":
  - $K_t = \bar{\Sigma}_t C_t^\top (C_t \bar{\Sigma}_t C_t^\top + Q_t)^{-1}$

- Update mean: $\mu_t = \bar{\mu}_t + K_t(x_t - C_t\bar{\mu}_t)$
  - $K_t(x_t - C_t\bar{\mu}_t)$ term compares actual x_t and predicted measurement $C_t\bar{\mu}_t$
  - $x_t - C_t\bar{\mu}_t$ is called "**innovation**"

  - $K_t \approx 0$ → observation is not useful (eg. $Q_t \to \infty$ or $\bar{\Sigma}_t = 0$)
  - $K_t \approx C_t^{-1}$ → prediction is not useful (eg. $\bar{\Sigma}_t \to \infty$)

Kalman Filter: Discussion

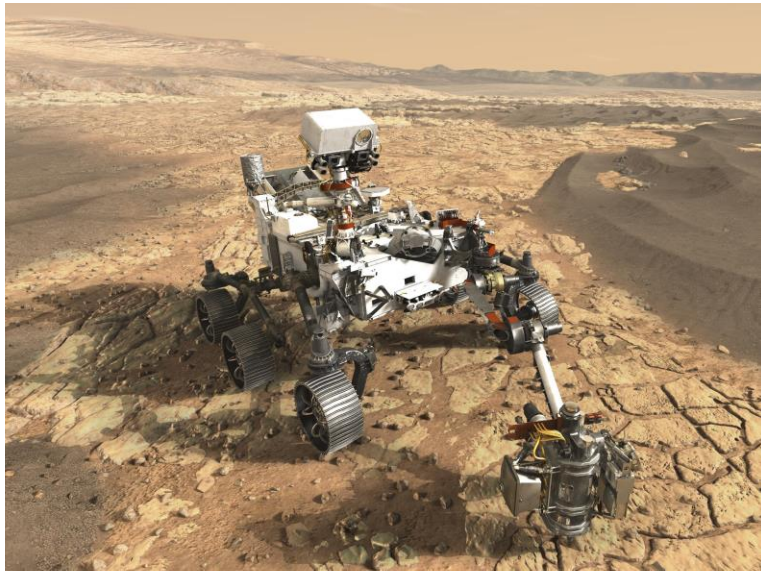
Possible advantages

- Only $O(n^2)$ parameters to update
  - μ has $O(n)$ parameters
  - Σ has $O(n^2)$ parameters
  - Bayes filter has $O(M^n)$
- Closed form update formulas
  - Bayes filter requires numerical integration

Possible disadvantages

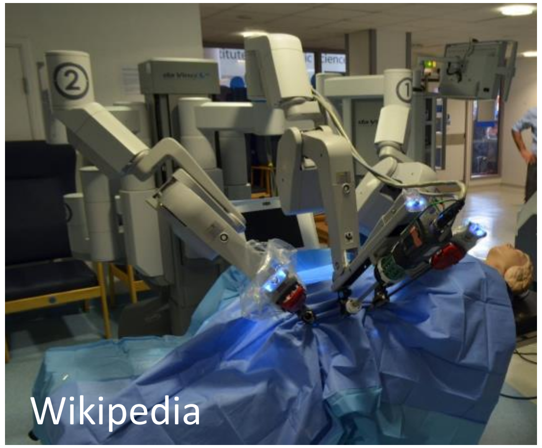
- Linear system dynamics / transition model
  - Most systems are nonlinear
- Gaussian distribution assumption
  - Only unimodal situations can be considered

Nonlinear Systems



- Almost all systems are nonlinear


Amazon.com Inc.


Wikipedia

Extended Kalman Filter

- Addresses the linear dynamics assumption

$$z_t = g(z_{t-1}, a_{t-1}) + \epsilon_t, \epsilon_t \sim N(0, R_t)$$
$$x_t = h(z_t) + \delta_t, \delta_t \sim N(0, Q_t)$$

- Linearize the nonlinear maps

$$g(z_{t-1}, a_{t-1}) \approx g(\mu_{t-1}, a_{t-1}) + \nabla g(\mu_{t-1}, a_{t-1})(z_{t-1} - \mu_{t-1})$$
$$h(z_t) \approx h(\bar{\mu}_t) + \nabla h(\bar{\mu}_t)(z_t - \mu_t)$$

  - Compatible with non-linear systems and nonlinear measurement models
  - Gaussian initial belief implies Gaussian belief for all time

EKF algorithm

- Kalmn filter algorithm:
  - $z_t = Az_{t-1} + Ba_{t-1} + \epsilon_t, \epsilon_t \sim N(0, R_t)$
  - $x_t = C_t z_t + \delta_t, \delta_t \sim N(0, Q_t)$

Input: $\mu_{t-1}, \Sigma_{t-1}, a_{t-1}, x_t$

Output: μ_t, Σ_t

Perform prediction:

$$\bar{\mu}_t = A\mu_{t-1} + Ba_{t-1}$$
$$\bar{\Sigma}_t = A\Sigma_{t-1}A^\top + R_t$$

Perform measurement update:

$$K_t = \bar{\Sigma}_t C_t^\top (C_t \bar{\Sigma}_t C_t^\top + Q_t)^{-1}$$
$$\mu_t = \bar{\mu}_t + K_t(x_t - C_t\bar{\mu}_t)$$
$$\Sigma_t = (I - K_t C_t)\bar{\Sigma}_t$$

Return μ_t, Σ_t

- Extended Kalman filter algorithm:
  - $z_t = g(z_{t-1}, u_{t-1}) + \epsilon_t, \epsilon_t \sim N(0, R_t)$
  - $x_t = h(z_t) + \delta_t, \delta_t \sim N(0, Q_t)$

Input: $\mu_{t-1}, \Sigma_{t-1}, u_{t-1}, x_t$

Output: μ_t, Σ_t

Perform prediction:

Perform measurement update:

Return μ_t, Σ_t

EKF Prediction

- Linear dynamics
  - $z_t = Az_{t-1} + Ba_{t-1} + \epsilon_t, \epsilon_t \sim N(0, R_t)$
- Nonlinear dynamics
  - $z_t = g(z_{t-1}, a_{t-1}) + \epsilon_t, \epsilon_t \sim N(0, R_t)$
- Linearized dynamics
  - $z_t \approx g(\mu_{t-1}, a_{t-1}) + G_t(z_{t-1} - \mu_{t-1}),$ $G_t := \nabla g(\mu_{t-1}, a_{t-1})$
- Kalman filter prediction
  - $\bar{\mu}_t = A\mu_{t-1} + Ba_{t-1}$
  - $\bar{\Sigma}_t = A\Sigma_{t-1}A^\top + R_t$
- EFK Prediction
  - $\bar{\mu}_t = g(\mu_{t-1}, a_{t-1})$
  - $\bar{\Sigma}_t = G_t\Sigma_{t-1}G_t^\top + R_t$

EKF algorithm

- Kalmn filter algorithm:
  - $x_t = Ax_{t-1} + Bu_{t-1} + \epsilon_t, \epsilon_t \sim N(0, R_t)$
  - $z_t = C_t x_t + \delta_t, \delta_t \sim N(0, Q_t)$

Input: $\mu_{t-1}, \Sigma_{t-1}, u_{t-1}, z_t$

Output: μ_t, Σ_t

Perform prediction:

$$\bar{\mu}_t = A\mu_{t-1} + Bu_{t-1}$$
$$\bar{\Sigma}_t = A\Sigma_{t-1}A^\top + R_t$$

Perform measurement update:

$$K_t = \bar{\Sigma}_t C_t^\top (C_t \bar{\Sigma}_t C_t^\top + Q_t)^{-1}$$
$$\mu_t = \bar{\mu}_t + K_t(z_t - C_t\bar{\mu}_t)$$
$$\Sigma_t = (I - K_t C_t)\bar{\Sigma}_t$$

Return μ_t, Σ_t

- Extended Kalman filter algorithm:
  - $z_t = g(z_{t-1}, a_{t-1}) + \epsilon_t, \epsilon_t \sim N(0, R_t)$
  - $x_t = h(z_t) + \delta_t, \delta_t \sim N(0, Q_t)$
  - Linearization: $G_t = \nabla g(\mu_{t-1}, a_{t-1}), H_t = \nabla h(\bar{\mu}_t)$

Input: $\mu_{t-1}, \Sigma_{t-1}, a_{t-1}, x_t$

Output: μ_t, Σ_t

Perform prediction:

$$\bar{\mu}_t = g(\mu_{t-1}, a_{t-1})$$
$$\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^\top + R_t$$

Perform measurement update:

Return μ_t, Σ_t

EKF Measurement Updates

- Linear measurement model
  - $x_t = C_t z_t + \delta_t, \delta_t \sim N(0, Q_t)$
- Kalman filter measurement update
  - $K_t = \bar{\Sigma}_t C_t^\top (C_t \bar{\Sigma}_t C_t^\top + Q_t)^{-1}$
  - $\mu_t = \bar{\mu}_t + K_t(x_t - C_t \bar{\mu}_t)$
  - $\Sigma_t = (I - K_t C_t)\bar{\Sigma}_t$

- Nonlinear measurement model
  - $x_t = h(z_t) + \delta_t, \delta_t \sim N(0, Q_t)$
- Linearized measurement model
  - $h(z_t) \approx h(\bar{\mu}_t) + H_t(z_t - \bar{\mu}_t),$ $H_t \coloneqq \nabla h(\bar{\mu}_t)$
- EFK measurement update
  - $K_t = \bar{\Sigma}_t H_t^\top (H_t \bar{\Sigma}_t H_t^\top + Q_t)^{-1}$
  - $\mu_t = \bar{\mu}_t + K_t\big(x_t - h(\bar{\mu}_t)\big)$
  - $\Sigma_t = (I - K_t H_t)\bar{\Sigma}_t$

EKF algorithm

- Kalmn filter algorithm:
  - $z_t = Az_{t-1} + Ba_{t-1} + \epsilon_t, \epsilon_t \sim N(0, R_t)$
  - $x_t = C_t z_t + \delta_t, \delta_t \sim N(0, Q_t)$

Input: $\mu_{t-1}, \Sigma_{t-1}, a_{t-1}, x_t$

Output: μ_t, Σ_t

Perform prediction:

$$\bar{\mu}_t = A\mu_{t-1} + Ba_{t-1}$$
$$\bar{\Sigma}_t = A\Sigma_{t-1}A^\top + R_t$$

Perform measurement update:

$$K_t = \bar{\Sigma}_t C_t^\top (C_t \bar{\Sigma}_t C_t^\top + Q_t)^{-1}$$
$$\mu_t = \bar{\mu}_t + K_t(x_t - C_t \bar{\mu}_t)$$
$$\Sigma_t = (I - K_t C_t)\bar{\Sigma}_t$$

Return μ_t, Σ_t

- Extended Kalman filter algorithm:
  - $z_t = g(z_{t-1}, a_{t-1}) + \epsilon_t, \epsilon_t \sim N(0, R_t)$
  - $x_t = h(z_t) + \delta_t, \delta_t \sim N(0, Q_t)$
  - Linearization: $G_t = \nabla g(\mu_{t-1}, a_{t-1}), H_t = \nabla h(\bar{\mu}_t)$

Input: $\mu_{t-1}, \Sigma_{t-1}, a_{t-1}, x_t$

Output: μ_t, Σ_t

Perform prediction:

$$\bar{\mu}_t = g(\mu_{t-1}, a_{t-1})$$
$$\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^\top + R_t$$

Perform measurement update:

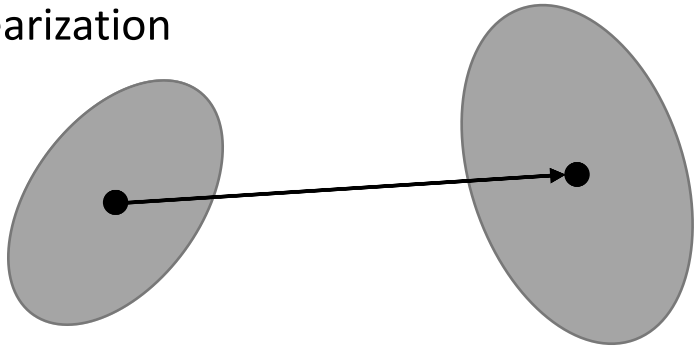
$$K_t = \bar{\Sigma}_t H_t^\top (H_t \bar{\Sigma}_t H_t^\top + Q_t)^{-1}$$
$$\mu_t = \bar{\mu}_t + K_t\big(x_t - h(\bar{\mu}_t)\big)$$
$$\Sigma_t = (I - K_t H_t)\bar{\Sigma}_t$$

Return μ_t, Σ_t

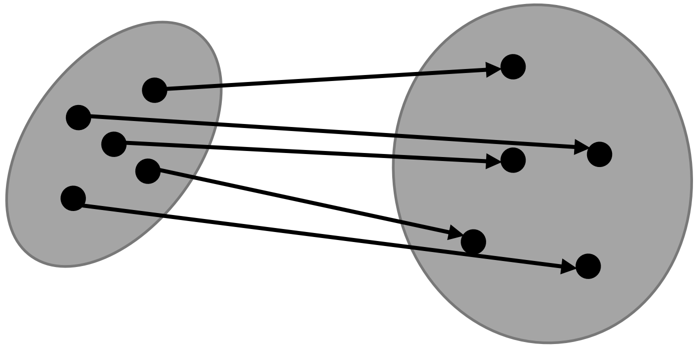
Unscented Kalman Filter

- Takes full knowledge of nonlinear dynamics
  - No linearization
  - Represents distributions using "Sigma points"
  - Transforms sigma points using nonlinear dynamics
- Approximates distribution using sigma points
  - Best fit Gaussian distribution given weights

EKF: transform Gaussian distributions using linearization



UKF: transforms sigma points and fits Gaussian distributions



Particle Filter

- Non-parametric filter
- Probability distributions $\mathrm{bel}(z_{t-1})$ directly represented by samples

$$\mathcal{Z}_{t-1} = \left\{z_{t-1}^{[i]}\right\}_{i=1}^{M}$$

- Prediction step: sample using dynamics
  - $\bar{z}_t^{[i]} \sim p\left(z_t \middle| a_{t-1}, z_{t-1}^{[i]}\right)$
- Measurement update step: weighted resampling based on measurements
  - Select M new particles from $\left\{\bar{z}_t^{[i]}\right\}$ with probability $\propto \ w_t^{[i]} = p\left(x_t \middle| z_t^{[i]}\right)$

Particle Filter

- Bayes' filter algorithm:

Input: $\mathrm{bel}(z_{t-1}), a_{t-1}, x_t$
Output: $\mathrm{bel}(z_t)$
For every z_t,

Perform prediction:

$\overline{\mathrm{bel}}(z_t) = \int p(z_t | a_{t-1}, z_{t-1}) \mathrm{bel}(z_{t-1}) dz_{t-1}$

Perform measurement update:

$\mathrm{bel}(z_t) = \eta p(x_t | z_t) \overline{\mathrm{bel}}(z_t)$

Return $\mathrm{bel}(z_t)$

- Particle filter algorithm:
  - Represent $\mathrm{bel}(z_t)$ with M samples

Input: $\mathcal{Z}_{t-1}, a_{t-1}, x_t$
Output: $\mathcal{Z}_t$
Perform prediction:

Draw $\bar{z}_t^{[i]} \sim p\left(z_t \middle| a_{t-1}, z_{t-1}^{[i]}\right), i = 1, \dots, M \rightarrow \bar{\mathcal{Z}}_t = \left\{z\bar{x}_t^{[i]}\right\}_{i=1}^{M}$

Perform measurement update:

Compute weights $w_t^{[i]} = p\left(x_t \middle| \bar{z}_t^{[i]}\right), i = 1, .., M$

Resample M times from $\bar{\mathcal{Z}}_t \rightarrow \mathcal{Z}_t$

- Each time, draw $\bar{z}_t^{[i]}$ with probability $\frac{w_t^{[i]}}{\sum_i w_t^{[i]}}$

Return $\mathcal{Z}_t$