Kalman Filter

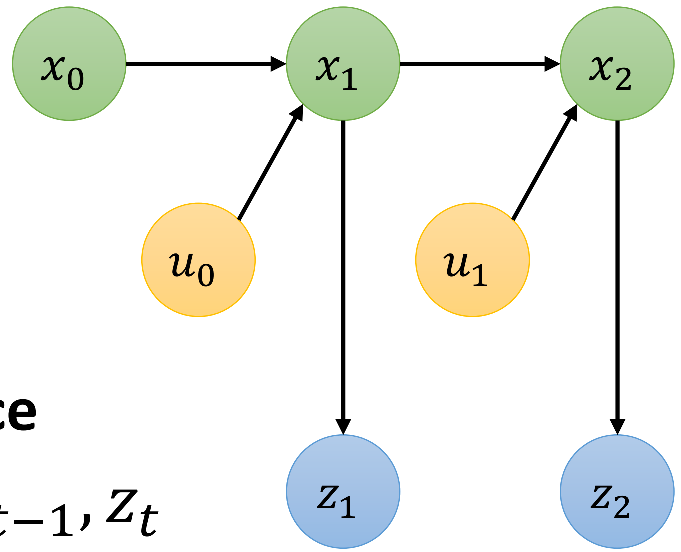
CMPT 882

Mar. 27

Outline

- Kalman Filter
  - Parametric filter for linear systems and measurement models
- Extended Kalman Filter
  - Extension to nonlinear systems and measurement models
- Unscented Kalman Filter
  - Non-parametric filter

Bayes filter



Continuous state space

Input: $\mathrm{bel}(x_{t-1}), u_{t-1}, z_t$

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

For every x_t,

Perform prediction:

$$\overline{\mathrm{bel}}(x_t) = \int p(x_t | u_{t-1}, x_{t-1}) \mathrm{bel}(x_{t-1}) dx_{t-1}$$

Perform measurement update:

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

Return $\mathrm{bel}(x_t)$

Discrete state space

Input: $\{p_{k,t-1}\}, u_{t-1}, z_t$

Output: $\{p_{k,t}\}$

For every k,

Perform prediction:

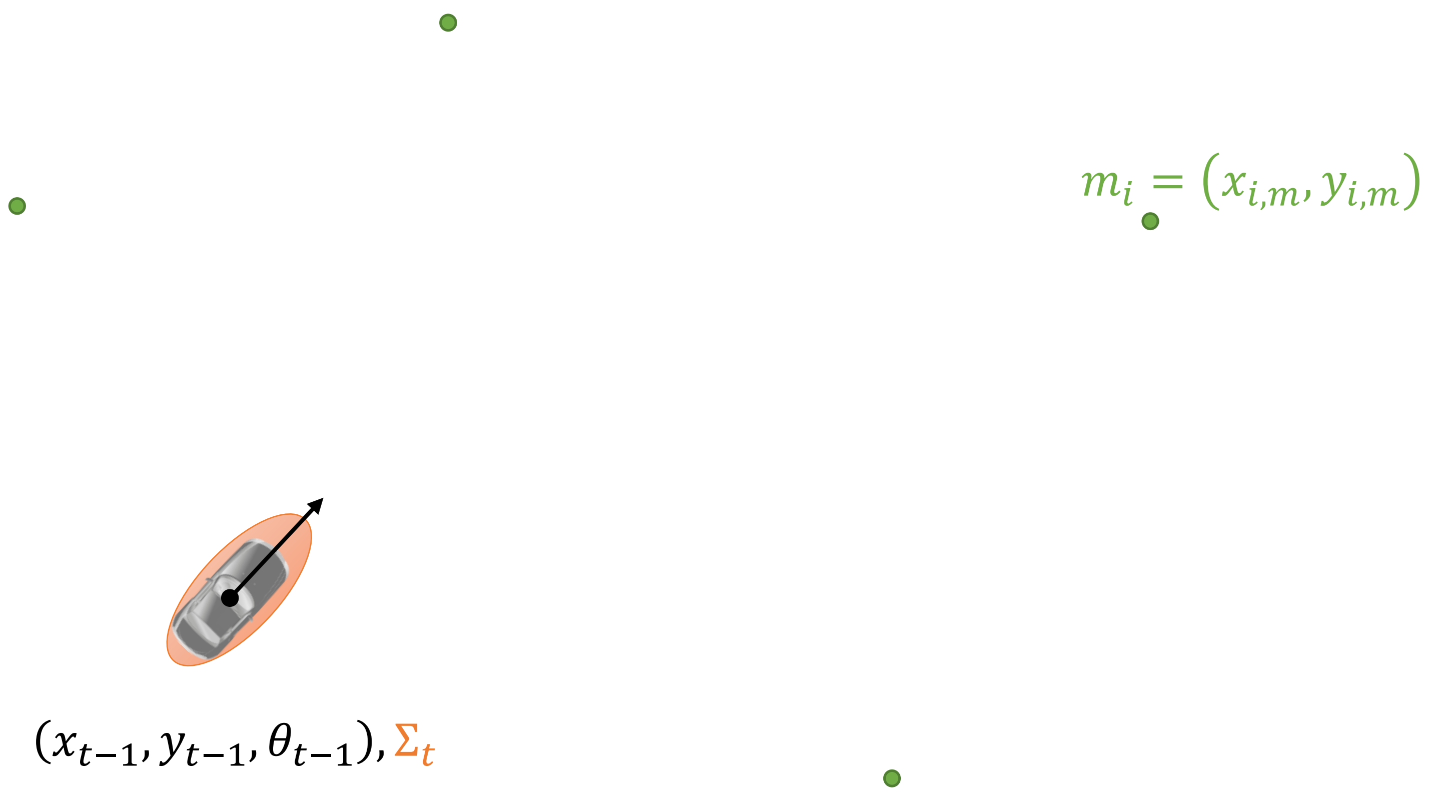
$$\bar{p}_{k,t} = \sum p(x_t | u_{t-1}, x_{t-1}) p_{k,t-1}$$

Perform measurement update:

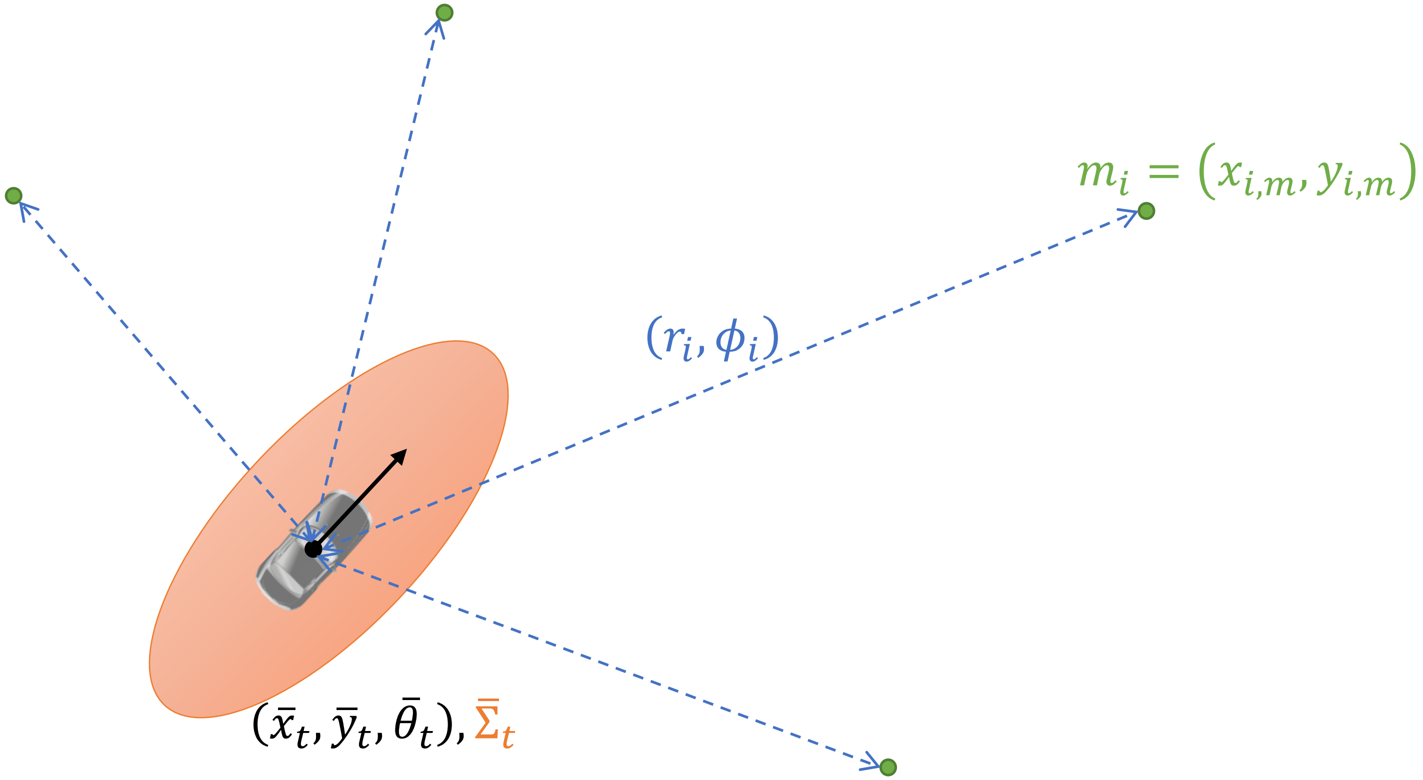
$$\mathrm{bel}(x_t) = \eta p(z_t | x_t) \bar{p}_{k,t}$$

Return $\{p_{k,t}\}$

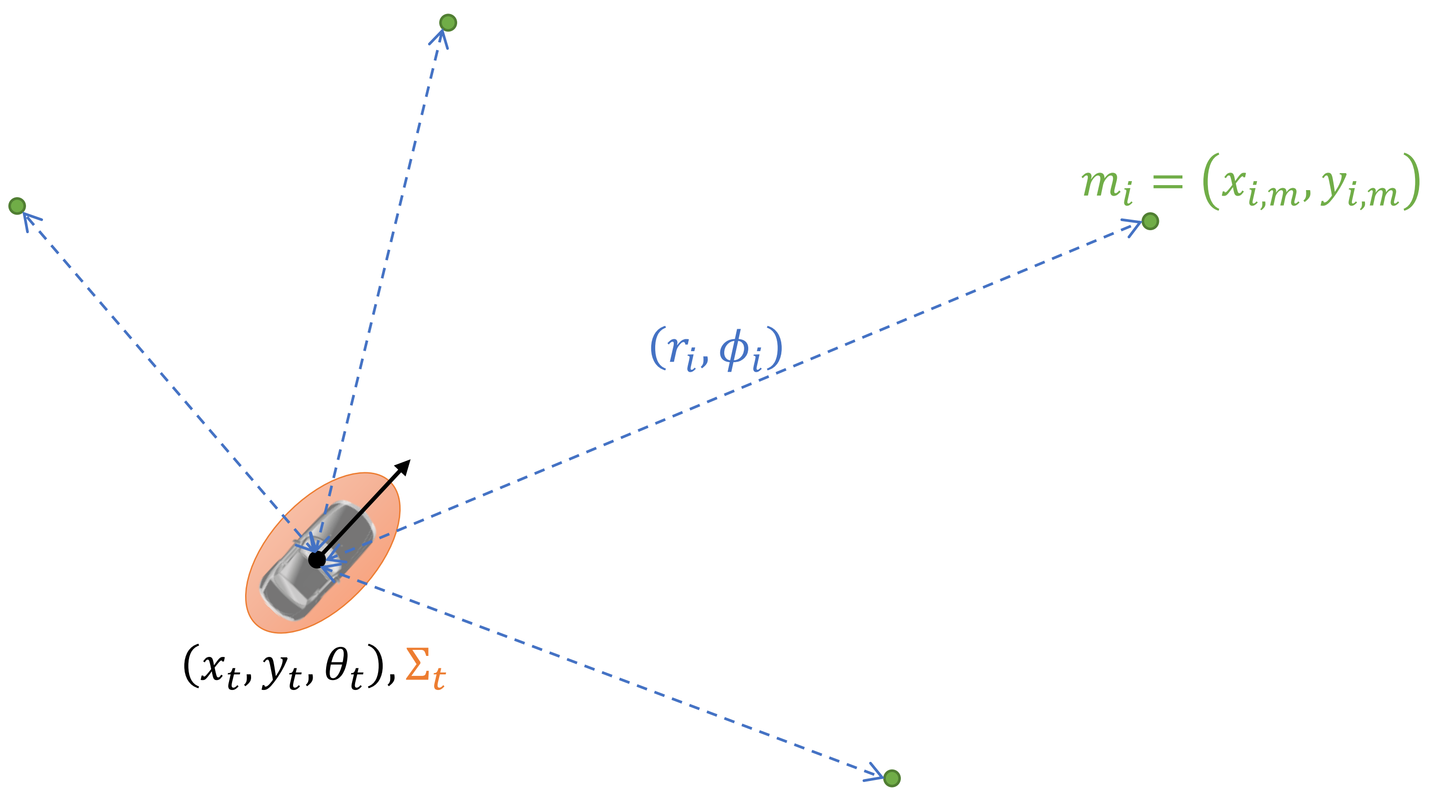
Localization



Localization



Localization



Bayes filter

- Continuous state space: Closed-form $\mathrm{bel}(x_t)$ is unlikely. Need discretization and interpolation

- Must iterate through every x_t or every k
  - Number of states is exponential in state space dimension

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

Kalman Filter

- Bayes filter with additional assumptions

- Initial Gaussian belief
  - $\mathrm{bel}(x_0) \sim N(\mu_0, \Sigma_0)$
  - Approximates single-modal distributions well

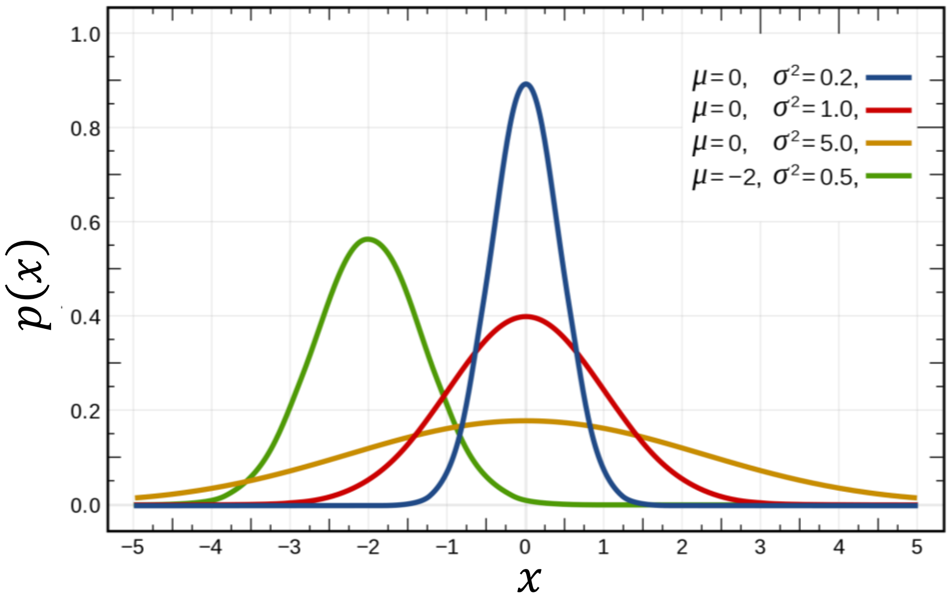
- Linear system dynamics with Gaussian noise
  - $x_t = Ax_{t-1} + Bu_{t-1} + \epsilon_t$
  - Noise is independent $\epsilon_t \sim N(0, R_t)$

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

Gaussian Distributions

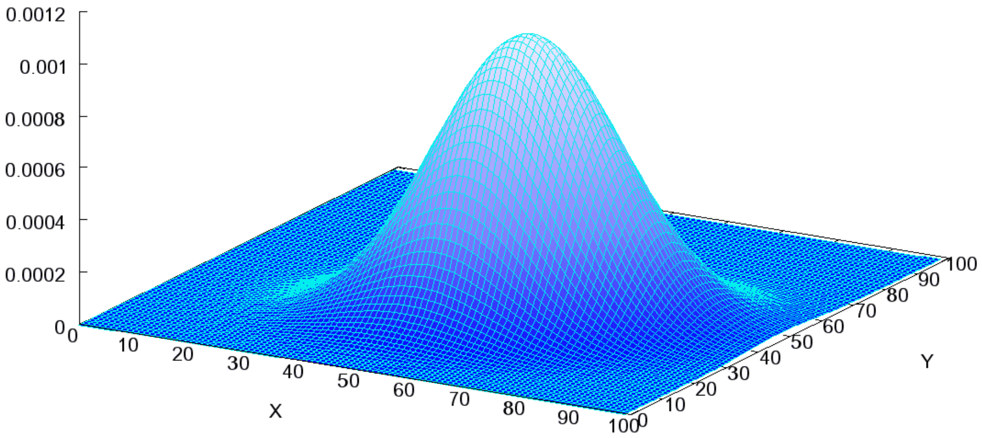
- Probability density function, scalar case:

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



- 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 are also Gaussian random variables
  - More complicated expression/derivation

Result of Assumptions and Gaussian Distribution Properties

1. Gaussian initial belief:

$$\mathrm{bel}(x_0) = p(x_0) = \det(2\pi\Sigma_0)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x_0 - \mu_0)^\top \Sigma_0^{-1}(x_0 - \mu_0)\right)$$

2. Linear dynamics $x_t = Ax_{t-1} + Bu_{t-1} + \epsilon_t$, $\epsilon_t \sim N(0, R_t)$ implies

$$p(x_t | x_{t-1}, u_{t-1}) = \det(2\pi R_t)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x_t - Ax_{t-1} - Bu_{t-1})^\top R_t^{-1}(x_t - Ax_{t-1} - Bu_{t-1})\right)$$

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

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

- Result: Posterior belief $\mathrm{bel}(x_t)$ is Gaussian for all t!
  - Start with $\mathrm{bel}(x_0) \sim N(\mu_0, \Sigma_0)$, obtain $\mathrm{bel}(x_t) \sim N(\mu_t, \Sigma_t)$ from $\mathrm{bel}(x_{t-1}) \sim N(\mu_{t-1}, \Sigma_{t-1})$
  - Only the parameters μ_t and Σ_t need to be updated to capture distribution over all x_t

Kalman Filter

- Bayes' filter algorithm:

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

Perform prediction:
$\overline{\mathrm{bel}}(x_t) = \int p(x_t|u_{t-1}, x_{t-1})\mathrm{bel}(x_{t-1})dx_{t-1}$

Perform measurement update:

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

Return $\mathrm{bel}(x_t)$

- Kalmn filter algorithm:

Input: $\mu_{t-1}, \Sigma_{t-1}, u_{t-1}, z_t$
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: $x_t = Ax_{t-1} + Bu_{t-1} + \epsilon_t, \epsilon_t \sim N(0, R_t)$
- If $x_{t-1} \sim N(\mu_{t-1}, \Sigma_{t-1})$ then $x_t \sim (\bar{\mu}_t, \bar{\Sigma}_t)$, where
  - $\bar{\mu}_t = A\mu_{t-1} + Bu_{t-1}$
  - $\bar{\Sigma}_t = A\Sigma_{t-1}A^\top + R_t$

Kalman Filter

- Bayes' filter algorithm:

Input: $\text{bel}(x_{t-1}), u_{t-1}, z_t$

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

For every x_t,

Perform prediction:

$$\overline{\text{bel}}(x_t) = \int p(x_t|u_{t-1}, x_{t-1})\text{bel}(x_{t-1})dx_{t-1}$$

Perform measurement update:

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

Return $\text{bel}(x_t)$

- Kalmn filter algorithm:

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:

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

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

Kalman Filter

- Bayes' filter algorithm:

Input: $\mathrm{bel}(x_{t-1}), u_{t-1}, z_t$

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

For every x_t,

Perform prediction:

$$\overline{\mathrm{bel}}(x_t) = \int p(x_t | u_{t-1}, x_{t-1}) \mathrm{bel}(x_{t-1}) dx_{t-1}$$

Perform measurement update:

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

Return $\mathrm{bel}(x_t)$

- Kalmn filter algorithm:

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

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: $\mu_t = \bar{\mu}_t + K_t(z_t - C_t \bar{\mu}_t)$
  - $K_t(z_t - C_t \bar{\mu}_t)$ term compares actual z_t and predicted measurement $C_t \bar{\mu}_t$
  - $z_t - C_t \bar{\mu}_t$ 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 $O(N^n)$
- Closed form update formulas
  - Bayes filter requires numerical integration

Possible disadvantages

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

Extended Kalman Filter

- Addresses the linear dynamics assumption

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

- Linearize the nonlinear maps

$$g(x_{t-1}, u_{t-1}) \approx g(\mu_{t-1}, u_{t-1}) + \nabla g(\mu_{t-1}, u_{t-1})(x_{t-1} - \mu_{t-1})$$
$$h(x_t) \approx h(\bar{\mu}_t) + \nabla h(\bar{\mu}_t)(x_t - \mu_t)$$

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

EKF algorithm

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

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

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

Output: μ_t, Σ_t

Perform prediction:

Perform measurement update:

Return μ_t, Σ_t

EKF Prediction

- Linear dynamics
  - $x_t = Ax_{t-1} + Bu_{t-1} + \epsilon_t, \epsilon_t \sim N(0, R_t)$
- Nonlinear dynamics
  - $x_t = g(x_{t-1}, u_{t-1}) + \epsilon_t, \epsilon_t \sim N(0, R_t)$
- Linearized dynamics
  - $x_t \approx g(\mu_{t-1}, u_{t-1}) + G_t(x_{t-1} - \mu_{t-1}),$ $G_t \coloneqq \nabla g(\mu_{t-1}, u_{t-1})$
- Kalman filter prediction
  - $\bar{\mu}_t = A\mu_{t-1} + Bu_{t-1}$
  - $\bar{\Sigma}_t = A\Sigma_{t-1}A^\top + R_t$
- EFK Prediction
  - $\bar{\mu}_t = g(\mu_{t-1}, u_{t-1})$
  - $\bar{\Sigma}_t = G_t\Sigma_{t-1}G_t^\top + R_t$

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

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

Output: μ_t, Σ_t

Perform prediction:

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

Perform measurement update:

Return μ_t, Σ_t

EKF Measurement Updates

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

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

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

Output: μ_t, Σ_t

Perform prediction:

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

Perform measurement update:

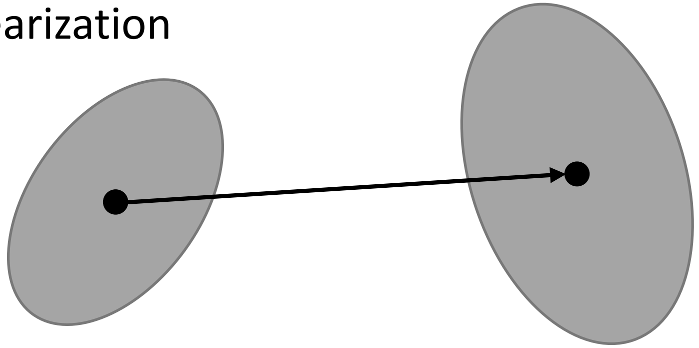
$$K_t = \bar{\Sigma}_t H_t^\top (H_t \bar{\Sigma}_t H_t^\top + Q_t)^{-1}$$
$$\mu_t = \bar{\mu}_t + K_t\big(z_t - h(\bar{\mu}_t)\big)$$
$$\Sigma_t = (I - K_t H_t)\bar{\Sigma}_t$$

Return μ_t, Σ_t

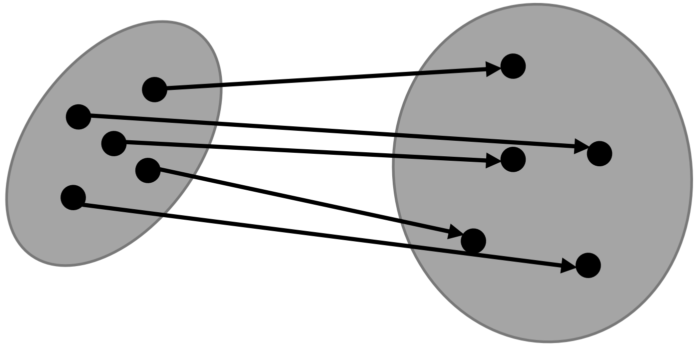
Unscented Kalman Filter

- 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 $\mathrm{bel}(x_{t-1})$ directly represented by samples

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

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

Particle Filter

- Bayes' filter algorithm:

Input: $\mathrm{bel}(x_{t-1}), u_{t-1}, z_t$
Output: $\mathrm{bel}(x_t)$
For every x_t,
    Perform prediction:
    $\overline{\mathrm{bel}}(x_t) = \int p(x_t | u_{t-1}, x_{t-1}) \mathrm{bel}(x_{t-1}) dx_{t-1}$
    Perform measurement update:
    $\mathrm{bel}(x_t) = \eta p(z_t | x_t) \overline{\mathrm{bel}}(x_t)$

Return $\mathrm{bel}(x_t)$

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

Input: $\mathcal{X}_{t-1}, u_{t-1}, z_t$
Output: $\mathcal{X}_t$
Perform prediction:
    Draw $\bar{x}_t^{[i]} \sim p\left(x_t \middle| u_{t-1}, x_{t-1}^{[i]}\right), i = 1, \ldots, M \to \bar{\mathcal{X}}_t = \{\bar{x}_t^{[i]}\}_{i=1}^{M}$
Perform measurement update:
    Compute weights $w_t^{[i]} = p\left(z_t \middle| \bar{x}_t^{[i]}\right), i = 1, .., M$
    Resample M times from $\bar{\mathcal{X}}_t \to \mathcal{X}_t$
    - Each time, draw $\bar{x}_t^{[i]}$ with probability $\frac{w_t^{[i]}}{\sum_i w_t^{[i]}}$

Return $\mathcal{X}_t$