L08. EKF-UKF LOCALIZATION & INTRO TO PARTICLE FILTERS

NA568 Mobile Robotics: Methods & Algorithms
Localization

“Using sensory information to locate the robot in its environment is the most fundamental problem to providing a mobile robot with autonomous capabilities.” [Cox ’91]

- **Given**
  - Map of the environment.
  - Sequence of sensor measurements.

- **Wanted**
  - Estimate of the robot’s position.

- **Problem classes**
  - Position tracking
  - Global localization
  - Kidnapped robot problem (recovery)
Landmark-based Localization
\begin{align*}
\mathbf{x}_t &= \begin{bmatrix} x_t & y_t & \theta_t \end{bmatrix}^T \\
\mathbf{u}_t &= \begin{bmatrix} v_t & \omega_t \end{bmatrix}^T
\end{align*}

Velocity-based process model

Ensure that angular differences lie within \([-\pi, \pi]\)

If known correspondence, Can strike these
**EKF_localization** \((\mu_{t-1}, \Sigma_{t-1}, u_t, z_t, m)\):

**Prediction:**

\[
G_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial x_{t-1}} = \begin{pmatrix}
\frac{\partial x'}{\partial \mu_{t-1,x}} & \frac{\partial x'}{\partial \mu_{t-1,y}} & \frac{\partial x'}{\partial \mu_{t-1,\theta}} \\
\frac{\partial y'}{\partial \mu_{t-1,x}} & \frac{\partial y'}{\partial \mu_{t-1,y}} & \frac{\partial y'}{\partial \mu_{t-1,\theta}} \\
\frac{\partial \theta'}{\partial \mu_{t-1,x}} & \frac{\partial \theta'}{\partial \mu_{t-1,y}} & \frac{\partial \theta'}{\partial \mu_{t-1,\theta}}
\end{pmatrix}
\]

Jacobian of \(g\) w.r.t location

\[
V_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial u_t} = \begin{pmatrix}
\frac{\partial x'}{\partial v_t} & \frac{\partial x'}{\partial \omega_t} \\
\frac{\partial y'}{\partial v_t} & \frac{\partial y'}{\partial \omega_t} \\
\frac{\partial \theta'}{\partial v_t} & \frac{\partial \theta'}{\partial \omega_t}
\end{pmatrix}
\]

Jacobian of \(g\) w.r.t control

\[
M_t = \begin{pmatrix}
\alpha_1 v_t^2 + \alpha_2 \omega_t^2 & 0 \\
0 & \alpha_3 v_t^2 + \alpha_4 \omega_t^2
\end{pmatrix}
\]

Motion noise

\[
\bar{\mu}_t = g(u_t, \mu_{t-1})
\]

Predicted mean

\[
\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + V_t M_t V_t^T
\]

Predicted covariance
EKF\_localization (\( \mu_{t-1}, \Sigma_{t-1}, u_t, z_t, m \)):

**Correction:**

\[
\hat{z}_t = \begin{cases} 
\sqrt{(m_x - \bar{\mu}_{t,x})^2 + (m_y - \bar{\mu}_{t,y})^2} \\
\text{atan}2(m_y - \bar{\mu}_{t,y}, m_x - \bar{\mu}_{t,x}) - \bar{\mu}_{t,0}
\end{cases}
\]

- Predicted measurement mean

\[
H_t = \frac{\partial h(\bar{\mu}_t, m)}{\partial x_t} = \begin{pmatrix} 
\frac{\partial r_t}{\partial \bar{\mu}_{t,x}} & \frac{\partial r_t}{\partial \bar{\mu}_{t,y}} & \frac{\partial r_t}{\partial \bar{\mu}_{t,0}} \\
\frac{\partial \phi_t}{\partial \bar{\mu}_{t,x}} & \frac{\partial \phi_t}{\partial \bar{\mu}_{t,y}} & \frac{\partial \phi_t}{\partial \bar{\mu}_{t,0}}
\end{pmatrix}
\]

- Jacobian of h w.r.t location

\[
Q_t = \begin{pmatrix} 
\sigma_r^2 & 0 \\
0 & \sigma_\phi^2
\end{pmatrix}
\]

- Measurement covariance

\[
S_t = H_t \bar{\Sigma}_t H_t^T + Q_t
\]

- Innovation covariance

\[
K_t = \bar{\Sigma}_t H_t^T S_t^{-1}
\]

- Kalman gain

\[
\mu_t = \bar{\mu}_t + K_t(z_t - \hat{z}_t)
\]

- Updated mean

\[
\Sigma_t = (I - K_t H_t) \bar{\Sigma}_t
\]

- Updated covariance
EKF Motion Prediction

\[
M_t = \begin{bmatrix}
\alpha_1 v_t^2 + \alpha_2 w_t^2 & 0 \\
0 & \alpha_3 v_t^2 + \alpha_4 w_t^2
\end{bmatrix}
\]

\(\alpha_1, \alpha_4 = < 10\%, 10\% >\)

\(\alpha_1, \alpha_4 = < 10\%, 30\% >\)

\(\alpha_1, \alpha_4 = < 30\%, 10\% >\)

\(\alpha_1, \alpha_4 = < 30\%, 30\% >\)
EKF Correction Step
Estimation Sequence (1): Accurate Landmark Sensor
Estimation Sequence (2): Less Accurate Landmark Sensor
Comparison to GroundTruth
Velocity-based process model

Augmented state allows for non-additive process noise models

Take care with mean of circular quantities

Take care with difference of circular quantities

1: Algorithm UKF_localization(μ_{t-1}, Σ_{t-1}, u_t, z_t, m):
   Generate augmented mean and covariance
   2: \( M_t = \begin{pmatrix} α_1 v_t^2 + α_2 ω_t^2 & 0 \\ 0 & α_3 v_t^2 + α_4 ω_t^2 \end{pmatrix} \)
   3: \( Q_t = \begin{pmatrix} σ_r^2 & 0 \\ 0 & σ_φ^2 \end{pmatrix} \)
   4: \( μ^{a-1}_t = (μ^T_{t-1} (0 0)^T (0 0)^T)^T \)
   5: \( Σ^{a}_{t-1} = \begin{pmatrix} Σ_{t-1} & 0 & 0 \\ 0 & M_t & 0 \\ 0 & 0 & Q_t \end{pmatrix} \)
   Generate sigma points
   6: \( X^{a}_{t-1} = (μ^{a}_{t-1} μ^{a}_{t-1} + γ √Σ^{a}_{t-1} μ^{a}_{t-1} - γ √Σ^{a}_{t-1}) \)
   Pass sigma points through motion model and compute Gaussian statistics
   7: \( \bar{X}^x_t = g(u_t + \bar{X}^u_{t-1}, \bar{X}^x_{t-1}) \)
   8: \( \bar{μ}_t = ∑_{i=0}^{2L} w_i^{(m)} \bar{X}^x_{i,t} \)
   9: \( \bar{Σ}_t = ∑_{i=0}^{2L} w_i^{(c)} (\bar{X}^x_{i,t} - \bar{μ}_t)(\bar{X}^x_{i,t} - \bar{μ}_t)^T \)
   Predict observations at sigma points and compute Gaussian statistics
   10: \( \bar{Z}_t = h(\bar{X}^x_t) + \bar{X}^z_t \)
   11: \( \bar{z}_t = ∑_{i=0}^{2L} w_i^{(m)} \bar{Z}_{i,t} \)
   12: \( S_t = ∑_{i=0}^{2L} w_i^{(c)} (\bar{Z}_{i,t} - \bar{z}_t)(\bar{Z}_{i,t} - \bar{z}_t)^T \)
   13: \( Σ^{x,z}_t = ∑_{i=0}^{2L} w_i^{(c)} (\bar{X}^x_{i,t} - \bar{μ}_t)(\bar{Z}_{i,t} - \bar{z}_t)^T \)
   Update mean and covariance
   14: \( K_t = Σ^{x,z}_t S_t^{-1} \)
   15: \( μ_t = \bar{μ}_t + K_t(z_t - \bar{z}_t) \)
   16: \( Σ_t = Σ_t - K_t S_t K_t^T \)
   17: \( p_{zt} = det (2πS_t)^{-\frac{1}{2}} exp \{ -\frac{1}{2} (z_t - \bar{z}_t)^T S_t^{-1} (z_t - \bar{z}_t) \} \)
   18: return μ_t, Σ_t, p_{zt}
**UKF_localization** \((\mu_{t-1}, \Sigma_{t-1}, u_t, z_t, m)\):

**Prediction:**

\[
M_t = \begin{pmatrix}
\alpha_1 v_t^2 + \alpha_2 \omega_i^2 & 0 \\
0 & \alpha_3 v_t^2 + \alpha_4 \omega_i^2
\end{pmatrix}
\]

Motion noise

\[
Q_t = \begin{pmatrix}
\sigma_r^2 & 0 \\
0 & \sigma_r^2
\end{pmatrix}
\]

Measurement noise

\[
\mu_{t-1}^a = \begin{pmatrix}
\mu_{t-1}^T \\
0 \\
0
\end{pmatrix}
\]

Augmented state mean

\[
\Sigma_{t-1}^a = \begin{pmatrix}
\Sigma_{t-1} & 0 & 0 \\
0 & M_t & 0 \\
0 & 0 & Q_t
\end{pmatrix}
\]

Augmented covariance

\[
\chi_{t-1}^a = \begin{pmatrix}
\mu_{t-1}^a \\
\mu_{t-1}^a + \gamma \sqrt{\Sigma_{t-1}^a} \\
\mu_{t-1}^a - \gamma \sqrt{\Sigma_{t-1}^a}
\end{pmatrix}
\]

Sigma points

\[
\overline{x}_x^t = g(u_t + \chi_t^u, \chi_{t-1}^x)
\]

Prediction of sigma points

\[
\overline{\mu}_t = \sum_{i=0}^{2L} w_m^i \, \overline{x}_{i,t}^x
\]

Predicted mean

\[
\overline{\Sigma}_t = \sum_{i=0}^{2L} w_c^i \left( \overline{x}_{i,t}^x - \overline{\mu}_t \right) \left( \overline{x}_{i,t}^x - \overline{\mu}_t \right)^T
\]

Predicted covariance
UKF\_localization (µ_{t-1}, \Sigma_{t-1}, u_t, z_t, m):

Correction:

\[ \bar{Z}_t = h(\bar{x}_t^x) + \chi_t^z \]  
Measurement sigma points

\[ \hat{z}_t = \sum_{i=0}^{2L} w^i_w \bar{Z}_{i,t} \]  
Predicted measurement mean

\[ S_t = \sum_{i=0}^{2L} w^i_w (\bar{Z}_{i,t} - \hat{z}_t)(\bar{Z}_{i,t} - \hat{z}_t)^T \]  
Pred. measurement covariance

\[ \Sigma^{x,z}_t = \sum_{i=0}^{2L} w^i_w (\bar{x}_{i,t}^x - \bar{\mu}_t)(\bar{Z}_{i,t} - \hat{z}_t)^T \]  
Cross-covariance

\[ K_t = \Sigma^{x,z}_t S_t^{-1} \]  
Kalman gain

\[ \mu_t = \bar{\mu}_t + K_t(z_t - \hat{Z}_t) \]  
Updated mean

\[ \Sigma_t = \Sigma_t - K_t S_t K_t^T \]  
Updated covariance
UKF Motion Prediction

\[ M_t = \begin{bmatrix} \alpha_1 v_t^2 + \alpha_2 w_t^2 & 0 \\ 0 & \alpha_3 v_t^2 + \alpha_4 w_t^2 \end{bmatrix} \]

\( \alpha_1, \alpha_4 = <10\%, 10\%> \)

\( \alpha_1, \alpha_4 = <10\%, 30\%> \)

\( \alpha_1, \alpha_4 = <30\%, 10\%> \)

\( \alpha_1, \alpha_4 = <30\%, 30\%> \)
UKF Observation Prediction

\( \sigma_r \) large
\( \sigma_\phi \) small

\( \sigma_r \) small
\( \sigma_\phi \) large
UKF Correction Step
Estimation Sequence

EKF

UKF
Prediction Quality

EKF

UKF
Particle Filters

- Different approach to state estimation
- Instead of *parametric* description of state (and uncertainty), use a set of *state samples*. The distribution of these *particles* represents the posterior distribution.
- Can represent arbitrary PDFs, not just Gaussians.
Sequential Monte Carlo (SMC): Brief History

- Basic idea of SMC around since the 1950’s
- Explored through 60’s and 70’s, but largely overlooked and ignored
  - 1) modest computational power at the time
  - 2) vanilla SIS leads to degeneracy over time

- The major contribution to development of the SMC method was the inclusion of the resampling step [Neil Gordon et al, 1993]
Roots of SMC are in MC Integration

- Let $I$ be the result of a multivariate integral

$$I = \int g(x) \, dx$$

- Thought experiment
  - Imagine discretizing and evaluating $I$ numerically. What is the complexity?
Suppose we can factorize $g(x)$

$$g(x) = f(x)\pi(x)$$

Such that $\pi(x)$ can be interpreted as a pdf

$$\pi(x) \geq 0 \text{ and } \int \pi(x) dx = 1$$

Draw $N>>1$ i.i.d. samples from $\pi(x)$, then

$$I = \int f(x)\pi(x) dx = E_\pi[f(x)] \approx I_N = \frac{1}{N} \sum_{i=1}^{N} f(x^i)$$
MC Integration continued…

- If the samples $x^i$ are i.i.d., then $I_N$ is the unbiased estimate of the integral $I$.
- According to the law of large numbers $I_N$ will almost surely converge to $I$.
- If the variance of $f(x)$ is finite, i.e.

$$\sigma^2 = \int (f(x) - I)^2 \pi(x) \, dx$$

then the CLT holds and the estimation error converges in distribution:

$$\lim_{N \to \infty} \sqrt{N} (I_N - I) \sim \mathcal{N}(0, \sigma^2)$$
MC Integration …

- Punch line
  - The error $e = I_N - I$ is on the order of $O(N^{-\frac{1}{2}})$
  - i.e. the rate of convergence is independent of the dimension of the integrand $n_x$!
Example

- Integrate \( g(x) = \sin(x)\cos(y) \) over domain \( 0<x<\pi \) and \( 0<y<\pi/2 \)

\[
I = \int_0^{\pi/2} \int_0^\pi \sin(x)\cos(y)\,dx\,dy = -\cos(x)|_0^\pi \sin(y)|_0^{\pi/2} = 2
\]
Continued…

- **Equivalent MC Integral**
  
  \[ g(x) \equiv f(x)\pi(x) = \left( \frac{\pi^2}{2} \sin(x) \cos(y) \right) \left( \frac{1}{\pi^2} \right) \]

- **Draw** \( x^i \) i.i.d. samples from \( \pi(x) \) and compute \( f(x^i) \), compute mean of \( f(x) \) for \( N=10, 100, 1000, 10000, \ldots, 10^7 \)

- **Expect std of error**
  
  \[ \text{std}(e) = \frac{\sqrt{2.0881}}{\sqrt{N}} \]

  \[ \sigma^2 = \int \left( \frac{\pi^2}{2} \sin(x) \cos(y) - 2 \right)^2 \pi(x)dx = \frac{\pi^4}{16} - 4 \approx 2.0881 \]
It works!
This technique can be applied in a Bayesian framework where $\pi(x)$ is the sought after posterior density.
Next Lecture

- Particle Filters for Bayesian Estimation
- PF Localization