Robot Mapping

Unscented Kalman Filter

Cyrill Stachniss

KF, EKF and UKF

- Kalman filter requires linear models
- EKF linearizes via Taylor expansion

Is there a better way to linearize?

Unscented Transform

Unscented Kalman Filter (UKF)

Taylor Approximation (EKF)

Linearization of the non-linear function through Taylor expansion

Unscented Transform

Compute a set of (so-called) sigma points
Unscented Transform

Transform each sigma point through the non-linear function

Unscented Transform Overview

- Compute a set of sigma points
- Each sigma points has a weight
- Transform the point through the non-linear function
- Compute a Gaussian from weighted points

- Avoids to linearize around the mean as the EKF does

Sigma Points

- How to choose the sigma points?
- How to set the weights?
**Sigma Points Properties**

- How to choose the sigma points?
- How to set the weights?
- Select $\mathcal{X}^{[i]}, w^{[i]}$ so that:
  \[
  \sum_i w^{[i]} = 1 \\
  \mu = \sum_i w^{[i]} \mathcal{X}^{[i]} \\
  \Sigma = \sum_i w^{[i]} (\mathcal{X}^{[i]} - \mu)(\mathcal{X}^{[i]} - \mu)^T \\
  \]
- There is no unique solution for $\mathcal{X}^{[i]}, w^{[i]}$

**Sigma Points**

- Choosing the sigma points
  \[
  \mathcal{X}^{[0]} = \mu \\
  \mathcal{X}^{[i]} = \mu + \left(\sqrt{n + \lambda}\right) \Sigma_i \quad \text{for } i = 1, \ldots, n \\
  \mathcal{X}^{[i]} = \mu - \left(\sqrt{n + \lambda}\right) \Sigma_{i-n} \quad \text{for } i = n + 1, \ldots, 2n
  \]

**Matrix Square Root**

- Defined as $S$ with $\Sigma = SS^T$
- Computed via diagonalization
  \[
  \Sigma = VDV^{-1} \\
  = V \begin{pmatrix}
  d_{11} & \ldots & 0 \\
  0 & \ddots & 0 \\
  0 & \ldots & d_{nn}
  \end{pmatrix} V^{-1} \\
  = V \left(\sqrt{d_{11}} \ldots 0 \right) V^{-1}
  \]

- $d_{ii}$: scaling parameter
- $n$: dimensionality
- $V$: matrix square root
- $\mu$: column vectors
Matrix Square Root

- Thus, we can define
  \[
  S = V \begin{pmatrix}
  \sqrt{d_{11}} & \cdots & 0 \\
  0 & \ddots & 0 \\
  0 & \cdots & \sqrt{d_{nn}} \\
  \end{pmatrix} V^{-1}
  \]

- so that
  \[
  SS = (VD^{1/2}V^{-1})(VD^{1/2}V^{-1}) = VDV^{-1} = \Sigma
  \]

Cholesky Matrix Square Root

- Alternative definition of the matrix square root
  \[
  L with \Sigma = LL^T
  \]

- Result of the Cholesky decomposition

- Numerically stable solution

- Often used in UKF implementations

- \(L\) and \(\Sigma\) have the same Eigenvectors

Sigma Points and Eigenvectors

- Sigma point can but do not have to lie on the main axes of \(\Sigma\)

- \[
  \chi^{[i]} = \mu + (\sqrt{(n+\lambda)\Sigma})_i \quad for \ i = 1, \ldots, n
  \]

- \[
  \chi^{[i]} = \mu - (\sqrt{(n+\lambda)\Sigma})_{i-n} \quad for \ i = n+1, \ldots, 2n
  \]
**Sigma Point Weights**

- Weight sigma points for computing the mean

\[ w_m^{[0]} = \frac{\lambda}{n + \lambda} \]

\[ w_c^{[0]} = w_m^{[0]} + (1 - \alpha^2 + \beta) \]

\[ w_m^{[i]} = w_c^{[i]} = \frac{1}{2(n + \lambda)} \text{ for } i = 1, \ldots, 2n \]

- for computing the covariance

**Recover the Gaussian**

- Compute Gaussian from weighted and transformed points

\[ \mu' = \sum_{i=0}^{2n} w_m^{[i]} g(\mathbf{x}^{[i]}) \]

\[ \Sigma' = \sum_{i=0}^{2n} w_c^{[i]} (g(\mathbf{x}^{[i]}) - \mu')(g(\mathbf{x}^{[i]}) - \mu')^T \]

**Example**

![Example Diagram](image)

**Examples**

\[ g((x, y)^T) = \left( \begin{array}{c} x + 1 \\ y + 1 \end{array} \right)^T \]

\[ g((x, y)^T) = \left( \begin{array}{c} 1 + x + \sin(2x) + \cos(y) \\ 2 + 0.2y \end{array} \right)^T \]
Unscented Transform Summary

- Sigma points
  \[ \chi^{[0]} = \mu \]
  \[ \chi^{[i]} = \mu + \left( \sqrt{(n + \lambda) \Sigma} \right)_i \text{ for } i = 1, \ldots, n \]
  \[ \chi^{[i]} = \mu - \left( \sqrt{(n + \lambda) \Sigma} \right)_{i-n} \text{ for } i = n + 1, \ldots, 2n \]

- Weights
  \[ w^{[0]}_m = \frac{\lambda}{n + \lambda} \]
  \[ w^{[0]}_c = w^{[0]}_m + (1 - \alpha^2 + \beta) \]
  \[ w^{[i]}_m = w^{[i]}_c = \frac{1}{2(n + \lambda)} \text{ for } i = 1, \ldots, 2n \]

UT Parameters

- Free parameters as there is no unique solution
- Scales Unscented Transform uses
  \[ \kappa \geq 0 \] Influence how far the sigma points are away from the mean
  \[ \alpha \in (0, 1] \]
  \[ \lambda = \alpha^2 (n + \kappa) - n \]
  \[ \beta = 2 \] Optimal choice for Gaussians

Examples

- \( \kappa = 3, \alpha = 0.01 \)
- \( \kappa = 3, \alpha = 0.1 \)
- \( \kappa = 3, \alpha = 0.25 \)
- \( \kappa = 3, \alpha = 0.75 \)
- \( \kappa = 3, \alpha = 0.25 \)
- \( \kappa = 10, \alpha = 0.25 \)
- \( \kappa = 50, \alpha = 0.25 \)
- \( \kappa = 120, \alpha = 0.25 \)
EKF Algorithm

1: \texttt{Extended Kalman filter}(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t):

2: \quad \bar{\mu}_t = g(u_t, \mu_{t-1})
3: \quad \bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_t
4: \quad K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1}
5: \quad \mu_t = \bar{\mu}_t + K_t (z_t - h(\bar{\mu}_t))
6: \quad \Sigma_t = (I - K_t H_t) \bar{\Sigma}_t
7: \quad \text{return } \mu_t, \Sigma_t

EKF to UKF – Prediction

1: \texttt{Extended Kalman filter}(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t):

2: \quad \bar{\mu}_t = \text{replace this by sigma point propagation of the motion}
3: \quad \bar{\Sigma}_t = \text{replace this by sigma point propagation of the motion}
4: \quad K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1}
5: \quad \mu_t = \bar{\mu}_t + K_t (z_t - h(\bar{\mu}_t))
6: \quad \Sigma_t = (I - K_t H_t) \bar{\Sigma}_t
7: \quad \text{return } \mu_t, \Sigma_t

UKF Algorithm – Prediction

1: \texttt{Unscented Kalman filter}(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t):

2: \quad \chi_{t-1} = (\mu_{t-1}, \mu_{t-1} + \gamma \sqrt{\Sigma_{t-1}}, \mu_{t-1} - \gamma \sqrt{\Sigma_{t-1}})
3: \quad \chi_t = g(u_t, \chi_{t-1})
4: \quad \bar{\mu}_t = \sum_{i=0}^{2n} w_i^{[i]} \chi_t^{[i]}\
5: \quad \bar{\Sigma}_t = \sum_{i=0}^{2n} w_i^{[i]} (\chi_t^{[i]} - \bar{\mu}_t) (\chi_t^{[i]} - \bar{\mu}_t)^T + R_t

EKF to UKF – Correction

1: \texttt{Extended Kalman filter}(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t):

2: \quad \bar{\mu}_t = \text{replace this by sigma point propagation of the motion}
3: \quad \bar{\Sigma}_t = \text{replace this by sigma point propagation of the motion}
4: \quad \mu_t = \bar{\mu}_t + K_t (z_t - \hat{z}_t)
5: \quad \Sigma_t = \bar{\Sigma}_t - K_t \hat{S} \hat{K}_t^T
6: \quad \Sigma_t = \bar{\Sigma}_t - K_t \hat{S} \hat{K}_t^T
7: \quad \text{return } \mu_t, \Sigma_t
UKF Algorithm – Correction (1)

6: \( \hat{x}_t = (\bar{\mu}_t \ \bar{\mu}_t + \gamma \sqrt{\hat{\Sigma}_t} \ \bar{\mu}_t - \gamma \sqrt{\hat{\Sigma}_t}) \)
7: \( \hat{z}_t = h(\hat{x}_t) \)
8: \( \hat{S}_t = \sum_{i=0}^{2n} w_i^{[i]} (\hat{z}_t - \hat{z}_t)(\hat{z}_t - \hat{z}_t)^T + Q_t \)
9: \( \hat{\Sigma}_{t,z} = \sum_{i=0}^{2n} w_i^{[i]} (\hat{x}_t - \bar{\mu}_t)(\hat{z}_t - \hat{z}_t)^T \)
10: \( K_t = \hat{\Sigma}_{t,z} \hat{S}_t^{-1} \)
11: return \( \mu_t, \Sigma_t \)

UKF Algorithm – Correction (2)

6: \( \hat{x}_t = (\bar{\mu}_t \ \bar{\mu}_t + \gamma \sqrt{\hat{\Sigma}_t} \ \bar{\mu}_t - \gamma \sqrt{\hat{\Sigma}_t}) \)
7: \( \hat{z}_t = h(\hat{x}_t) \)
8: \( \hat{S}_t = \sum_{i=0}^{2n} w_i^{[i]} (\hat{z}_t - \hat{z}_t)(\hat{z}_t - \hat{z}_t)^T + Q_t \)
9: \( \hat{\Sigma}_{t,z} = \sum_{i=0}^{2n} w_i^{[i]} (\hat{x}_t - \bar{\mu}_t)(\hat{z}_t - \hat{z}_t)^T \)
10: \( K_t = \hat{\Sigma}_{t,z} \hat{S}_t^{-1} \)
11: return \( \mu_t, \Sigma_t \)
From EKF to UKF – Computing the Covariance

\[
\Sigma_t = (I - K_t H_t) \Sigma_t \nabla_t - K_t H_t \Sigma_t = \Sigma_t - K_t \left( \Sigma_{x,z} \right)^T \nabla_t - K_t \left( K_t S_t \right)^T \nabla_t = \Sigma_t - K_t S_t^T K_t^T \nabla_t = \Sigma_t - K_t S_t K_t^T
\]
UKF vs. EKF

- Same results as EKF for linear models
- Better approximation than EKF for non-linear models
- Differences often “somewhat small”
- No Jacobians needed for the UKF
- Same complexity class
- Slightly slower than the EKF
- Still requires Gaussian distributions

UT/UKF Summary

- Unscented transforms as an alternative to linearization
- UT is a better approximation than Taylor expansion
- UT uses sigma point propagation
- Free parameters in UT
- UKF uses the UT in the prediction and correction step

Literature

Unscented Transform and UKF

- Thrun et al.: “Probabilistic Robotics”, Chapter 3.4
- “A New Extension of the Kalman Filter to Nonlinear Systems” by Julier and Uhlmann, 1995
- “Dynamische Zustandsschätzung” by Fränken, 2006, pages 31-34