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 point has a weight
- Transform the point through the non-linear function
- Compute a Gaussian from weighted points
- Avoids to linearize around the mean as Taylor expansion (and 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 $\chi^{[i]}$, $w^{[i]}$ so that:
  \[ \sum_i w^{[i]} = 1 \]
  \[ \mu = \sum_i w^{[i]} \chi^{[i]} \]
  \[ \Sigma = \sum_i w^{[i]} (\chi^{[i]} - \mu)(\chi^{[i]} - \mu)^T \]
- There is no unique solution for $\chi^{[i]}, w^{[i]}$

Sigma Points

- Choosing the sigma points
  \[ \chi^{[0]} = \mu \]
  First sigma point is the mean

Sigma Points

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

Matrix Square Root

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

\[ L \text{ with } \Sigma = LL^T \]

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 \( \Sigma \)

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

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

Sigma Points Example
**Sigma Point Weights**

- Weight sigma points for computing the mean

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

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

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

for computing the covariance

**Recover the Gaussian**

- Compute Gaussian from weighted and transformed points

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

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

**Example**

**Examples**

\[
g((x, y)^T) = \begin{pmatrix} x + 1 \\ y + 1 \end{pmatrix}^T
\]

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

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

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

UT Parameters

- Free parameters as there is no unique solution
- Scaled Unscented Transform suggests

\[
\kappa \geq 0 \quad \text{Influence how far the sigma points are away from the mean} \\
\alpha \in (0, 1] \\
\lambda = \alpha^2(n + \kappa) - n \\
\beta = 2 \quad \text{Optimal choice for Gaussians}
\]
**EKF Algorithm**

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

---

**UKF Algorithm – Prediction**

1: Unscented Kalman filter($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$):
2: $\chi_{t-1} = (\mu_{t-1} + \sqrt{(n + \lambda)\Sigma_{t-1}}, \mu_{t-1} - \sqrt{(n + \lambda)\Sigma_{t-1}})$
3: $\chi_t^* = g(u_t, \chi_{t-1})$
4: $\tilde{\mu}_t = \sum_{i=0}^{2n} w_i^{[i]} \chi_t^{*, [i]}$
5: $\tilde{\Sigma}_t = \sum_{i=0}^{2n} w_i^{[i]} (\chi_t^{*, [i]} - \tilde{\mu}_t)(\chi_t^{*, [i]} - \tilde{\mu}_t)^T + R_t$

---

**EKF to UKF – Prediction**

1: Extended Kalman filter($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$):
2: $\tilde{\mu}_t = $ replace this by sigma point propagation of the motion
3: $\tilde{\Sigma}_t = $ propagation of the motion
4: $K_t = \tilde{\Sigma}_t H_t^T (H_t \tilde{\Sigma}_t H_t^T + Q_t)^{-1}$
5: $\mu_t = \tilde{\mu}_t + K_t(z_t - h(\tilde{\mu}_t))$
6: $\Sigma_t = (I - K_t H_t) \tilde{\Sigma}_t$
7: return $\mu_t, \Sigma_t$

---

**EKF to UKF – Correction**

1: Extended Kalman filter($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$):
2: $\tilde{\mu}_t = $ replace this by sigma point propagation for the motion
3: $\tilde{\Sigma}_t = $ propagation of the motion
4: $\tilde{\mu}_t = \tilde{\mu}_t + K_t(z_t - \bar{z}_t)$
5: $\Sigma_t = \tilde{\Sigma}_t - K_t S_t K_t^T$
6: return $\mu_t, \Sigma_t$
UKF Algorithm – Correction (1)

6: \( \dot{X}_t = (\dot{\mu}_t, \dot{\mu}_t + \sqrt{(n + \lambda)}\Sigma_t) \quad \dot{\mu}_t - \sqrt{(n + \lambda)}\Sigma_t \)
7: \( \dot{Z}_t = h(\dot{X}_t) \)
8: \( \dot{z}_t = \sum_{i=0}^{2n} w_m[i] \dot{Z}_t[i] \)
9: \( S_t = \sum_{i=0}^{2n} w_m[i] (\dot{Z}_t[i] - \dot{z}_t)(\dot{Z}_t[i] - \dot{z}_t)^T + Q_t \)
10: \( \Sigma_t^{x,x} = \sum_{i=0}^{2n} w_m[i] (X_t[i] - \tilde{\mu}_t)(\tilde{Z}_t[i] - \tilde{z}_t)^T \)
11: \( K_t = \Sigma_t^{x,x} S_t^{-1} \)

UKF Algorithm – Correction (2)

6: \( \dot{X}_t = (\dot{\mu}_t, \dot{\mu}_t + \sqrt{(n + \lambda)}\Sigma_t) \quad \dot{\mu}_t - \sqrt{(n + \lambda)}\Sigma_t \)
7: \( \dot{Z}_t = h(\dot{X}_t) \)
8: \( \dot{z}_t = \sum_{i=0}^{2n} w_m[i] \dot{Z}_t[i] \)
9: \( S_t = \sum_{i=0}^{2n} w_m[i] (\dot{Z}_t[i] - \dot{z}_t)(\dot{Z}_t[i] - \dot{z}_t)^T + Q_t \)
10: \( \Sigma_t^{x,x} = \sum_{i=0}^{2n} w_m[i] (X_t[i] - \tilde{\mu}_t)(\tilde{Z}_t[i] - \tilde{z}_t)^T \)
11: \( K_t = \Sigma_t^{x,x} S_t^{-1} \)
12: \( \mu_t = \tilde{\mu}_t + K_t(\tilde{z}_t - \tilde{z}_t) \)
13: \( \Sigma_t = \Sigma_t - K_t S_t K_t^T \)
14: return \( \mu_t, \Sigma_t \)
From EKF to UKF – Computing the Covariance

\[
\Sigma_t = (I - K_t H_t) \tilde{\Sigma}_t \\
= \tilde{\Sigma}_t - K_t H_t \tilde{\Sigma}_t \\
= \tilde{\Sigma}_t - K_t (\tilde{\Sigma}_{x,z})^T \\
= \tilde{\Sigma}_t - K_t (\tilde{\Sigma}_{x,z} S_t^{-1} S_t)^T \\
= \tilde{\Sigma}_t - K_t (K_t S_t)^T \\
= \tilde{\Sigma}_t - K_t S_t^T K_t^T \\
= \tilde{\Sigma}_t - K_t S_t K_t^T
\]

UKF vs. EKF (Small Covariance)

UKF vs. EKF – Banana Shape

EKF approximation

UKF approximation
**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

**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 restricted to Gaussian distributions

**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