Probabilistic Fundamentals in Robotics

Gaussian Filters

Basilio Bona
DAUIN – Politecnico di Torino

Course Outline

- Basic mathematical framework
- Probabilistic models of mobile robots
- Mobile robot localization problem
- Robotic mapping
- Probabilistic planning and control

Reference textbook
http://www.probabilistic-robotics.org/
Basic mathematical framework

- Recursive state estimation
  - Basic concepts in probability
  - Robot environment
  - Bayes filters
- Gaussian filters (parametric filters)
  - Kalman filter
  - Extended Kalman Filter
  - Unscented Kalman filter
  - Information filter
- Nonparametric filters
  - Histogram filter
  - Particle filter

Introduction

- Gaussian filters are different implementations of Bayes filters for continuous spaces, with specific assumptions on probability distributions
- Beliefs are represented by multi-variate normal distributions
Multi-variate Gaussian distribution

\[ p(\mathbf{x}) = \det((2\pi \Sigma)^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu) \right\} \]

Examples

Bi-dimensional Gaussian with conditional probabilities

Mixture of Gaussians
Covariance matrix

The covariance matrix $\Sigma$ has the following properties

- $\Sigma = \Sigma^T \in \mathbb{R}^{n \times n}$. Square and symmetric
- $x^T \Sigma x > 0$: positive definite; $x^T \Sigma x = 0$ only for $x = 0$
- number of free elements $n(n - 1)/2 = (n^2 - n)/2$
- elements have dimension $[x]^2$

Kalman filter (1)

- KF computes the belief for continuous states governed by linear dynamic state equations
- Beliefs are expressed by normal distributions $N(x; \mu, \Sigma)$
- KF is not applicable to discrete or hybrid state space systems
Kalman filter (2)

State transition probability \( p(x_t|u_t, x_{t-1}) \) is a linear function of the states and controls

\[
x_t = A_t x_{t-1} + B_t u_t + w_t
\]

with \( x_t \in \mathbb{R}^n, A_t \in \mathbb{R}^{n \times n}, u_t \in \mathbb{R}^m, B_t \in \mathbb{R}^{n \times m} \)

\( w_t \) is a zero mean Gaussian random variable distributed as \( N(w_t, 0, R_t) \)

\[
p(x_t|u_t, x_{t-1}) = \det(2\pi R_t)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(x_t - A_t x_{t-1} - B_t u_t)^\top R_t^{-1}(x_t - A_t x_{t-1} - B_t u_t)\right\}
\]

Kalman filter (3)

Measurement probability \( p(z_t|x_t) \) is a linear function of the states

\[
z_t = C_t x_t + v_t
\]

with \( z_t \in \mathbb{R}^q, C_t \in \mathbb{R}^{q \times n} \). \( v_t \) is a zero mean Gaussian random variable \( N(v_t, 0, Q_t) \)

\[
p(z_t|x_t) = \det(2\pi Q_t)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(z_t - C_t x_t)^\top Q_t^{-1}(z_t - C_t x_t)\right\}
\]
Kalman filter (4)

Initial conditions must be also normally distributed
\[
bel(x_0) = p(x_0) = \exp\left\{-\frac{1}{2}(x_0 - \mu_0)^T \Sigma_0^{-1}(x_0 - \mu_0)\right\}
\]

These three conditions ensure that the posterior \( bel(x_t) \) at each instant of time is always normally distributed.

Kalman filter algorithm (1)

```
Kalman_Filter(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)

1. \hat{\mu}_t = A_t \mu_{t-1} + B_t u_t
2. \hat{\Sigma}_t = A_t \Sigma_{t-1} A^T + R_t
3. K_t = \hat{\Sigma}_t C^T_t (C_t \hat{\Sigma}_t C^T_t + Q_t)^{-1}
4. \mu_t = \hat{\mu}_t + K_t (z_t - C_t \hat{\mu}_t)
5. \Sigma_t = (I - K_t C_t) \hat{\Sigma}_t
6. return (\mu_t, \Sigma_t)
```
Block diagram

Kalman filter algorithm (2)
Kalman filter example

From Kalman filter to extended Kalman filter

- Kalman filter is based on linearity assumptions
- Gaussian random variables are expressed by means and covariance matrices of normal distributions
- Gaussian distributions are transformed into Gaussian distributions

\[
(\mu_{t-1}, \Sigma_{t-1}) \Rightarrow (\mu_t, \Sigma_t)
\]

- Kalman filter is optimal
- Kalman filter is efficient

If \( k \) is the measurement number and \( n \) is the state number, \( O(k^{2.376} + n^2) \)
Linear transformation of Gaussians

![Linear transformation diagram]

Extended Kalman Filter (EKF)

- When the linearity assumptions do not hold (as in robot motion models or orientation models) a closed form solution of the predicted belief does not exists

\[
x_t = g(x_{t-1}, u_t) + w_t \quad \text{Nonlinear state & measurement equations}
\]

\[
z_t = h(x_t) + v_t
\]

- Extended Kalman Filter (EKF) approximates the nonlinear transformations with a linear one
- Linearization is performed around the most likely value: i.e., the mean value
Approximating Gaussian uses mean and covariance of the Montecarlo generated distribution

Basilio Bona
EKF Example

Approximating Gaussian:
the normal distribution built using mean and covariance of the true nonlinear distributions

EKF Gaussian:
the normal distribution built using mean and covariance of the true nonlinear distributions

EKF linearization

Taylor expansion

\[ G_t := \frac{\partial g(x_{t-1}, u_t)}{\partial x_{t-1}} \bigg|_{x_{t-1}=\mu_{t-1}} \]
 State Jacobian

\[ H_t := \frac{\partial h(x_t)}{\partial x_t} \bigg|_{x_{t-1}=\mu_t} \]
 Measurement Jacobian

\[ x_t \approx g(\mu_{t-1}, u_t) + G_t(x_{t-1} - \mu_{t-1}) + w_t \]

\[ z_t \approx h(\mu_t) + H_t(x_t - \mu_t) + v_t \]

Depends only on the mean
EKF algorithm

\[
\text{EKF\_Kalman\_Filter}(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)
\]

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

BF

KF vs EKF

\[
\text{Kalman\_Filter}(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)
\]

1. \( \bar{\mu}_t = A_t \mu_{t-1} + B_t u_t \)
2. \( \bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t \)
3. \( K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1} \)
4. \( \mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t) \)
5. \( \Sigma_t = (I - K_t C_t) \bar{\Sigma}_t \)
6. return \((\mu_t, \Sigma_t)\)

\[
\text{EKF\_Kalman\_Filter}(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)
\]

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

- EKF is a very popular tool for state estimation in robotics
- It has the same time complexity of the KF
- It is robust and simple
- Limitations: rarely state and measurement functions are linear.
- Goodness of linear approximation depends on:
  - Degree of uncertainty
  - Degree of nonlinearity
- When using EKF the uncertainty must be kept small as much as possible

Uncertainty

More uncertain

Less uncertain
Uncertainty

Nonlinearity
Nonlinearity

Example: EKF Localization within a sensor infrastructure

Mobile Robot can acquire odometric measurements and distance information from sensors in known positions

Fixed sensors (deployed in known positions inside the environment)

True position of the mobile robot

KF estimate (time zero)
**Example: EKF Localization within a sensor infrastructure**

**STEP 1:**
- Acquire odometry

---

**Example: EKF Localization within a sensor infrastructure**

**STEP 1:**
- Acquire odometry
- Filter Prediction
Example: EKF Localization within a sensor infrastructure

**STEP 1:**
- Acquire odometry
- Filter Prediction
- Acquire meas.

Luca Carlone – Politecnico di Torino

---

Example: EKF Localization within a sensor infrastructure

**STEP 1:**
- Acquire odometry
- Filter Prediction
- Acquire meas.
- Filter Update

Luca Carlone – Politecnico di Torino
Example: EKF Localization within a sensor infrastructure

STEP 1:
- Acquire odometry
- Filter Prediction
- Acquire meas.
- Filter Update

STEP 2:
- Acquire odometry
Example: EKF Localization within a sensor infrastructure

**STEP 1:**
- Acquire odometry
- Filter Prediction
- Acquire meas.
- Filter Update

**STEP 2:**
- Acquire odometry
- Filter Prediction
- Acquire meas.
Example: EKF Localization within a sensor infrastructure

**STEP 1:**
- Acquire odometry
- Filter Prediction
- Acquire meas.
- Filter Update

**STEP 2:**
- Acquire odometry
- Filter Prediction
- Acquire meas.
- Filter Update

Luca Carlone – Politecnico di Torino

---

**Unscented Kalman Filter (UKF)**

- UKF performs a stochastic linearization based on a *weighted statistical linear regression*
- A deterministic sampling technique (the *unscented transform*) is used to pick a minimal set of sample points (*sigma points*) around the mean value of the normal pdf
- The sigma points are propagated through the nonlinear functions, and then used to compute the mean and covariance of the transformed distribution
- This approach
  - removes the need to explicitly compute Jacobians, which for complex functions can be difficult to calculate
  - produces a more accurate estimate of the posterior distribution

Basilio Bona
Sigma Points

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

where \( \left[ \sqrt{(n + \lambda) \Sigma} \right]_i \) is the \( i \)-th column of the matrix \( \sqrt{n + \lambda \Sigma^{1/2}} \)

UKF

Weights and Parameters
Each sigma point depends on numerical parameters

\[ \gamma = \sqrt{n + \lambda} \]
\[ w^0_m = \frac{\lambda}{\gamma^2} \]
\[ w^0_c = \frac{\lambda}{\gamma^2} + (1 - \alpha^2 + \beta) \]
\[ w^i_m = w^i_c = \frac{1}{2\gamma^2} \quad \forall i = 1, \ldots, 2n \]
\[ \lambda = \alpha^2(n + \kappa) - n \]

\( \alpha \) and \( \kappa \) are scaling parameters that determine how far the sigma points are from the mean. \( \beta \) can be chosen to encode additional knowledge about distribution. When distribution is Gaussian \( \beta = 2 \).
UKF

\[ \mathbf{y}^{[i]} = g(\mathbf{x}^{[i]}) \]

\[ \mu' = \sum_{i=0}^{2n} w_m \mathbf{y}^{[i]} \]

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

UKF Algorithm – part a)

Unscented_Kalman_Filter(\( \mu_{t-1}, \Sigma_{t-1}, u_t, z_t \))

1. \( \mathbf{x}_{t-1} = (\mu_{t-1}, \mu_{t-1} + \gamma \sqrt{\Sigma_{t-1}}, \mu_{t-1} - \gamma \sqrt{\Sigma_{t-1}}) \)
2. \( \tilde{\mathbf{x}}_{t-1}^s = g(\mathbf{x}_{t-1}, u_t) \)
3. \( \tilde{\mu} = \sum_{i=0}^{2n} w_m \tilde{\mathbf{x}}_{t-1}^s[i] \)
4. \( \tilde{\Sigma}_t = \sum_{i=0}^{2n} w_c^i \left( \tilde{\mathbf{x}}_{t-1}^s[i] - \tilde{\mu} \right) \left( \tilde{\mathbf{x}}_{t-1}^s[i] - \tilde{\mu} \right)^T + R_t \)
5. \( \tilde{\mathbf{x}}_t = (\tilde{\mu}_t, \tilde{\mu}_t + \gamma \sqrt{\Sigma_t}, \tilde{\mu}_t - \gamma \sqrt{\Sigma_t}) \)
6. \( \tilde{z}_t = h(\tilde{\mathbf{x}}_t) \)
7. \( \hat{z}_t = \sum_{i=0}^{2n} w_m \hat{Z}_t \)
UKF Algorithm – part b)

Unscented_Kalman_Filter(\(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t\))

1. \(S_t = \sum_{i=0}^{2n} w_i (\hat{z}_t - \hat{z}_t)(\hat{z}_t - \hat{z}_t)^\top + Q_t\)

2. \(\Sigma_{x,z} = \sum_{i=0}^{2n} w_i (x_t - \bar{\mu})(\hat{z}_t - \hat{z}_t)^\top\)

3. \(K_t = \Sigma_{x,z} S_t^{-1}\)

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

5. \(\Sigma_t = \Sigma_t - K_t S_t K_t^\top\)

6. return \((\mu_t, \Sigma_t)\)

Cross covariance

EKF vs UKF

Basilio Bona
EKF vs UKF

KF – EKF – UKF

\[ \mu_t = \bar{\mu}_t + K_t(z_t - C_t \bar{\mu}_t) \]
\[ \Sigma_t = \bar{\Sigma}_t - K_t C_t \bar{\Sigma}_t \]

\[ \mu_t = \bar{\mu}_t + K_t(z_t - h_t(\bar{\mu}_t)) \]
\[ \Sigma_t = \bar{\Sigma}_t - K_t H_t \bar{\Sigma}_t \]

\[ \mu_t = \bar{\mu}_t + K_t(z_t - \hat{z}_t) \]
\[ \Sigma_t = \bar{\Sigma}_t - K_t S_t K_t^T \]
**Information filters**

Belief is represented by Gaussians

Moments parameterization

- KF – EKF – UKF

- Mean: \( \mu = \Omega^{-1} \xi \)
- Covariance: \( \Sigma = \Omega^{-1} \)

Canonical parameterization

- IF – EIF

- Information vector: \( \xi = \Sigma^{-1} \mu \)
- Information matrix: \( \Omega = \Sigma^{-1} \)

**Multivariate normal distribution**

\[
p(x) = \det(2\pi \Sigma)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu)\right\} \\
= \det(2\pi \Sigma)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2} x^T \Sigma^{-1} x + x^T \Sigma^{-1} \mu - \frac{1}{2} \mu^T \Sigma^{-1} \mu\right\} \\
= \det(2\pi \Sigma)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2} \mu^T \Sigma^{-1} \mu\right\} \exp\left\{-\frac{1}{2} x^T \Sigma^{-1} x + x^T \Sigma^{-1} \mu\right\} \\
= \eta \exp\left\{-\frac{1}{2} x^T \Omega x + x^T \xi\right\}
\]
Mahalanobis distance

\[- \log p(x) = - \log \eta + \frac{1}{2} x^T \Omega x - x^T \xi\]

\[= \frac{1}{2} x^T \Omega x - x^T \xi + \text{const.}\]

Mahalanobis distance

\[\mu = \Omega^{-1}\]

IF algorithm

Information Filter \((\xi_{t-1}, \Omega_{t-1}, u_t, z_t)\)

1. \(\bar{\Omega}_t = (A_t \Omega_{t-1}^{-1} A_t^T + R_t)^{-1}\)
2. \(\bar{\xi}_t = \Omega_{t}(A_t \Omega_{t-1}^{-1} \xi_{t-1} + B_t u_t)\)
3. \(\Omega_t = C_t^T Q_t^{-1} C_t + \bar{\Omega}_t\)
4. \(\xi_t = C_t^T Q_t^{-1} z_t + \bar{\xi}_t\)
5. return \((\mu_t, \Sigma_t)\)
**IF vs KF**

**IF**
- Prediction step requires two matrix inversion
  - $O(n^2.4)$
- Measurements update is additive
  - $O(n^2)$

**KF**
- Prediction step is additive
  - $O(n^2)$
- Measurements update requires matrix inversion
  - $O(n^{2.4})$

---

**Extended information filter – EIF**

- It is similar to EKF and applies when state and measurement equations are nonlinear
- Jacobians $G$ and $H$ replace $A$, $B$ and $C$ matrices

**Extended Information Filter** ($\xi_{t-1}, \Omega_{t-1}, u_t, z_t$)

1. $\mu_{t-1} = \Omega_{t-1}^{-1} \xi_{t-1}$  \textbf{State estimate}
2. $\bar{\mu}_t = g(\mu_{t-1}, u_t)$
3. $\Omega_t = (G_t \Omega_{t-1}^{-1} G_t^T + R_t)^{-1}$
4. $\tilde{\xi}_t = \tilde{\Omega}_t \bar{\mu}_t$
5. $\Omega_t = \Omega_t + H_t^T Q_t^{-1} H_t$
6. $\xi_t = \tilde{\xi}_t + H_t^T Q_t^{-1} [z_t - h(\bar{\mu}_t) + H_t \bar{\mu}_t]$
7. return $(\mu_t, \Sigma_t)$
Practical considerations

- **IF advantages over KF:**
  - Simpler global uncertainty representation: set $\Omega = 0$
  - Numerically more stable (in many but not all robotics applications)
  - Integrates information in simpler way
  - Is naturally fit for multi-robot problems (decentralized data integration $\Rightarrow$ Bayes rule $\Rightarrow$ logarithmic form $\Rightarrow$ addition of terms $\Rightarrow$ arbitrary order)

- **IF limitations:**
  - A state estimation is required (inversion of a matrix)
  - Other matrix inversions are necessary (not required for EKF)
  - Computationally inferior to EKF for high-dim state spaces

Final comments

- In many problems the interaction between state variable is local $\Rightarrow$ structure on $\Omega$ $\Rightarrow$ sparseness of $\Omega$ but not of $\Sigma$
- Information filters as graphs: sparse information matrix = sparse graph
- Such graphs are known as *Gaussian Markov random fields*
Thank you. Any question?