Mobile Robotics, Localization: Kalman Filter

Material based on the book Probabilistic Robotics (Thrun, Burgard, Fox) [PR]; Chapter 3.2, 3.3, 7.1-7.4

Part of the material is based on lectures from Cyrill Stachniss

Summary

- Introduction to Kalman filter
- Kalman Filter [Chapter 3.2]
- Extended Kalman Filter [Chapter 3.3]
- Markov Localization [Chapter 7.1-7.3]
- EKF localization [Chapter 7.4, partly]

Intro to Kalman filter

- Realization of Bayes Filter for the **Gaussian linear** case
- Incorporate controls in the **prediction step**
- Exploit observations in the **correction step**
- Optimal approach for Gaussian linear systems

Bayes Filter for state estimation

- **General** recursive approach to state estimation
- Prediction Step:

$$\overline{Bel(x_t)} = \int P(x_t|u_t, x_{t-1})Bel(x_{t-1})dx_{t-1}$$

- Correction Step:

$$Bel(x_t) = \eta P(z_t|x_t)\overline{Bel(x_t)}$$

Kalman Filter

- Concrete realization of Bayes Filter
- Everything is Gaussian

$$P(x) = det(2\pi\Sigma)^{\frac{1}{2}} exp(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu))$$

- x and μ vectors of n components, Σ is the covariance matrix (nxn)
- Optimal solution for linear models and Gaussian distribution

Update of Gaussian Belief based on motions and observations

- Key ingredients: Marginalization and Conditioning
  - Given $x = (x_a, x_b)^T$ and $P(x) = \mathcal{N}$
  - Marginals are Gaussians: $P(x_a) = \mathcal{N}$ and $P(x_b) = \mathcal{N}$
  - Conditionals are Gaussians: $P(x_a|x_b) = \mathcal{N}$ and $P(x_b|x_a) = \mathcal{N}$
- Linear models for motions and observations
  - Linear models: $f(x) = Ax + b$
  - A Gaussian probability transformed through a linear function stays Gaussian
- Motion and observation models for Kalman filter
  - $x_t = A_t x_{t-1} + B_t u_t + \epsilon_t$ **Motion model**
  - $z_t = C_t x_t + \delta_t$ **Observation model**

Main components for the Kalman filter

◊ A_t matrix $(n \times n)$ describing state evolution from x_{t-1} to x_t **without** controls

◊ B_t matrix $(n \times l)$ describing state evolution from x_{t-1} to x_t due to **control command** (no noise)

◊ C_t matrix $(k \times n)$ mapping state x_t to **observation** z_t (no noise)

◊ ϵ_t and δ_t Gaussian noise for state evolution and measurement. Assumed to be independent from each other with zero mean and covariance R_t and Q_t respectively.

Linear motion and observation models

- Motion model under Gaussian noise

$$P(x_t|u_t, x_{t-1}) = det(2\pi R_t)^{-\frac{1}{2}} exp\left(-\frac{1}{2}(x_t - A_t x_{t-1} - B_t u_t)^T R_t^{-1}(x_t - A_t x_{t-1} - B_t u_t)\right)$$

- Observation model under Gaussian noise

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

Belief update for Gaussian distributions

- Given an initial Gaussian belief, the belief stays Gaussian

$$Bel(x_t) = \eta P(z_t|x_t)\overline{Bel(x_t)}$$

  - we know $P(z_t|x_t)$ is Gaussian
  - if $\overline{Bel(x_t)}$ is a Gaussian then $Bel(x_t)$ is also Gaussian

$$\overline{Bel(x_t)} = \int P(x_t|u_t, x_{t-1})Bel(x_{t-1})dx_{t-1}$$

  - we know $P(x_t|u_t, x_{t-1})$ is Gaussian
  - we **can** assume $Bel(x_{t-1})$ is a Gaussian (initial belief $Bel(x_0)$ is Gaussian)
  - we can **show** that $\overline{Bel(x_t)}$ is Gaussian
- As a result everything stays Gaussian

Representing and Updating Gaussian Distributions

- A (multi-variate) Gaussian distribution is completely determined by:
  - the mean μ_t
  - the covariance matrix Σ_t
- To determine μ_t and Σ_t we exploit the following properties
  - Product of two Gaussians stays Gaussian
  - Linear transformations of a Gaussian results in a Gaussian
  - Marginal and conditional distribution of a Gaussian is still a Gaussian
  - Methods to compute mean and covariance of marginal and conditional distributions for Gaussians
  - ...

The Kalman Filter Algorithm

Algorithm Kalman_filter($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$)**:**

$$\bar{\mu}_t = A_t\, \mu_{t-1} + B_t\, u_t$$

$$\bar{\Sigma}_t = A_t\, \Sigma_{t-1}\, A_t^T + R_t$$

$$K_t = \bar{\Sigma}_t\, C_t^T (C_t\, \bar{\Sigma}_t\, C_t^T + Q_t)^{-1}$$

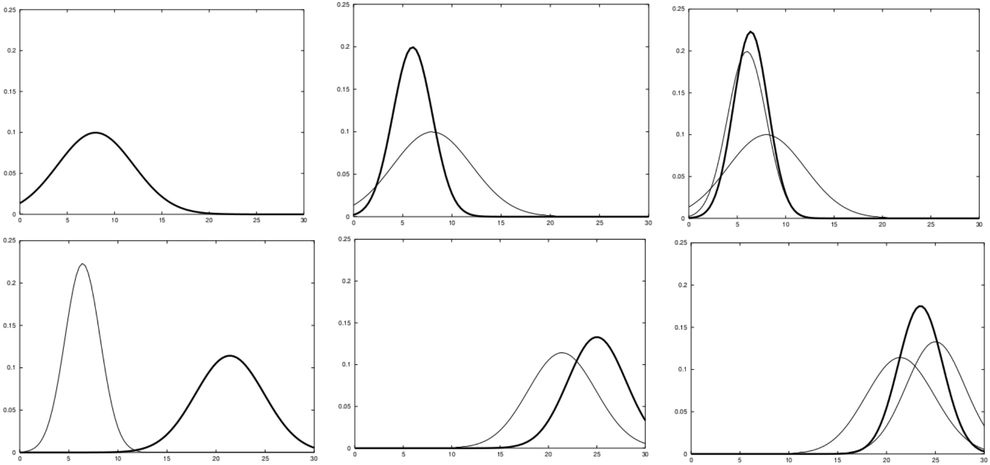
$$\mu_t = \bar{\mu}_t + K_t (z_t - C_t\, \bar{\mu}_t)$$

$$\Sigma_t = (I - K_t\, C_t)\, \bar{\Sigma}_t$$

return μ_t, Σ_t

Kalman Filter Algorithm, source [PR]

1D Kalman filter example, source [PR]



Source [PR]

Kalman filter assumptions

- Gaussian distributions and noise
- Linear motion and observation models
  - $x_t = A_t x_{t-1} + B_t u_t + \epsilon_t$ **Motion model**
  - $z_t = C_t x_t + \delta_t$ **Observation model**
- Under these assumptions Kalman Filter is the best estimator for the state
- What can we do if assumptions do not hold

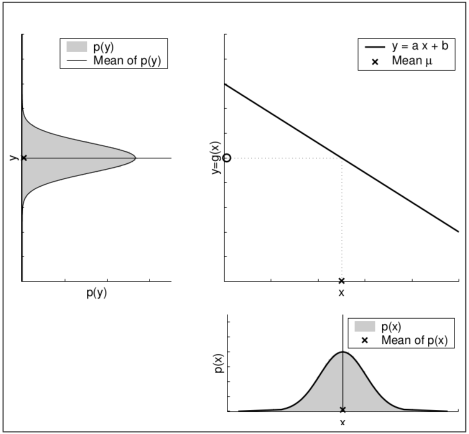
Non-linear dynamic systems

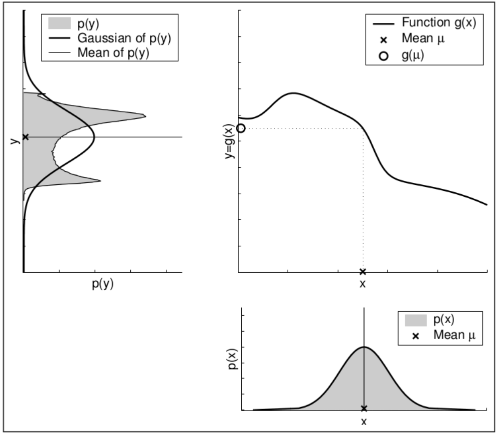
- $x_t = A_t x_{t-1} + B_t u_t + \epsilon_t$ **Motion model**
- $z_t = C_t x_t + \delta_t$ **Observation model**

◇ General motion and observation models

- $x_t = g(x_{t-1}, u_t) + \epsilon_t$ **General Motion model**
- $z_t = h(x_t) + \delta_t$ **General Observation model**

Linearity assumption: illustration





Source [PR]

Local linearization: EKF and first order Taylor expansion

- **key idea**: assume our function is linear around the mean
- prediction:

$$g(u_t, x_{t-1}) \approx g(u_t, \mu_{t-1}) + \frac{\partial g(u_t, \mu_{t-1})}{\partial x_{t-1}}(x_{t-1} - \mu_{t-1})$$

- correction:

$$h(x_t) \approx h(\overline{\mu}_t) + \frac{\partial h(\overline{\mu}_t)}{\partial x_t}(x_t - \overline{\mu}_t)$$

Reminder: Jacobian

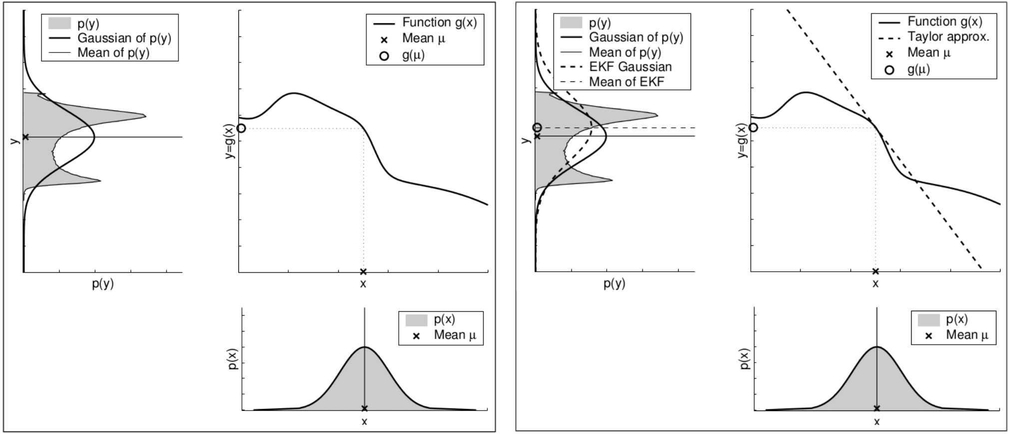
- Partial derivative in the multi-dimensional case is a Jacobian
- Given a vector valued function m and a vector x of n components

$$f(x) = \begin{pmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_m(x) \end{pmatrix}$$

- The Jacobian is a $m \times n$ function composed of the partial derivatives:

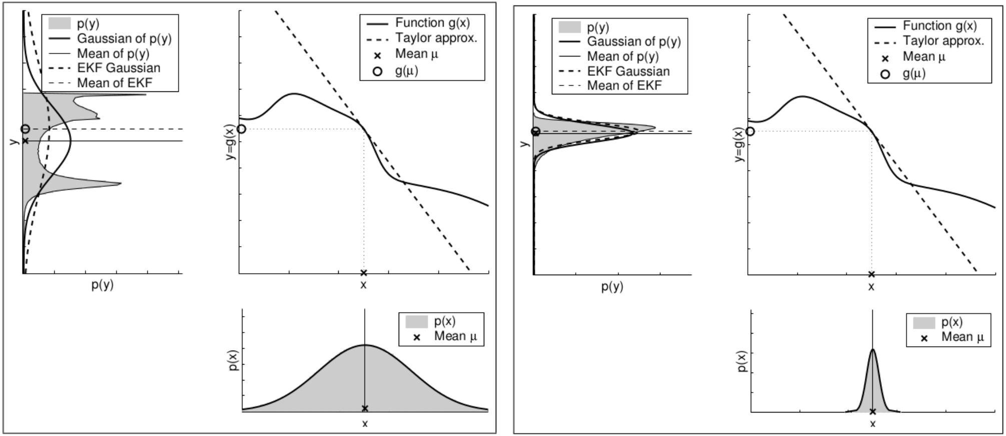
$$F_x = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \dots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \dots & \frac{\partial f_m}{\partial x_n} \end{pmatrix}$$

Linearization: illustration



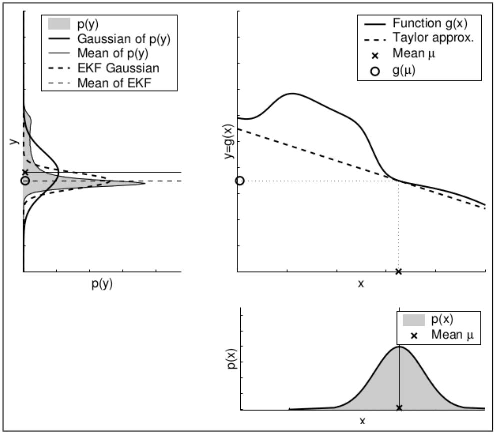
Source [PR]

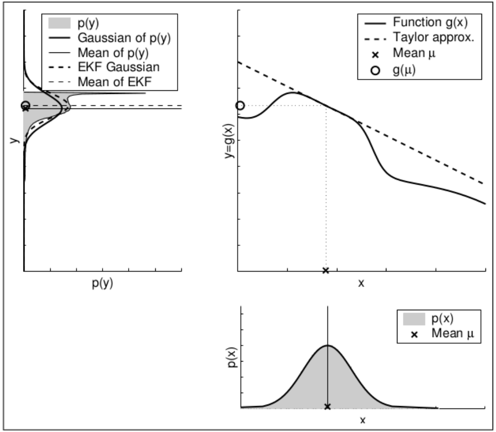
Effect of variance on linearization: illustration



Source [PR]

Effect of local linearity: illustration





Source [PR]

Linearized motion and observation models

◇ Linearized Motion model under Gaussian noise

$$P(x_t|u_t, x_{t-1}) \approx det(2\pi R_t)^{-\frac{1}{2}}$$
$$exp\left(-\frac{1}{2}(x_t - g(u_t, \mu_{t-1}) - G_t(x_{t-1} - \mu_{t-1}))^T R_t^{-1}(x_t - g(u_t, \mu_{t-1}) - G_t(x_{t-1} - \mu_{t-1}))\right)$$

◇ Observation model under Gaussian noise

$$P(z_t|x_t) = det(2\pi Q_t)^{-\frac{1}{2}}$$
$$exp\left(-\frac{1}{2}(z_t - h(\overline{\mu}_t) - H_t(x_t - \overline{\mu}_t))^T Q_t^{-1}(z_t - h(\overline{\mu}_t) - H_t(x_t - \overline{\mu}_t))\right)$$

The Extended Kalman Filter Algorithm

Algorithm Extended_Kalman_filter($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$)**:**

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

$\bar{\Sigma}_t = G_t \, \Sigma_{t-1} \, G_t^T + R_t$

$K_t = \bar{\Sigma}_t \, H_t^T (H_t \, \bar{\Sigma}_t \, H_t^T + Q_t)^{-1}$

$\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))$

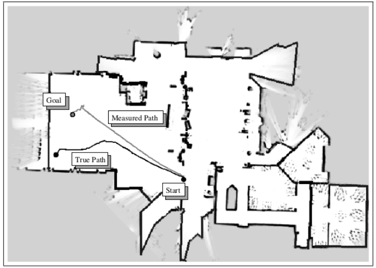
$\Sigma_t = (I - K_t \, H_t) \, \bar{\Sigma}_t$

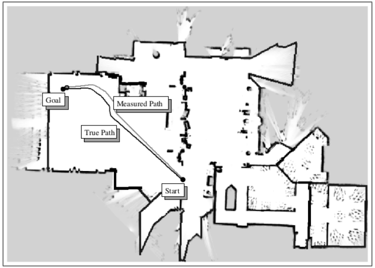
return μ_t, Σ_t

Extended Kalman Filter Algorithm, source [PR]

Mobile Robot Localization

- ♢ **Goal**: find robot's pose with respect to the map
- ♢ Map: feature based, location based (e.g., occupancy grid maps)
- ♢ Taxonomy of localization problems
  - Local vs. Global (kidnapped robot)
  - Static vs. Dynamic Environments
  - Passive vs. Active
  - Single vs. Multi Robot

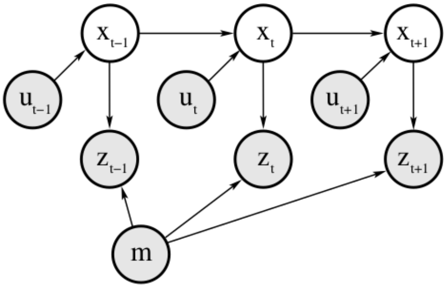




Passive vs. Active Localization, source [PR], courtesy of N. Roy

Markov Localization

Position Estimation



Algorithm Markov_localization($bel(x_{t-1}), u_t, z_t, m$)**:**
  for all x_t *do*
    $\overline{bel}(x_t) = \int p(x_t \mid u_t, x_{t-1}, m)\ bel(x_{t-1})\ dx$
    $bel(x_t) = \eta\ p(z_t \mid x_t, m)\ \overline{bel}(x_t)$
  endfor
  return $bel(x_t)$

Graphical model and Markov Localization Algorithm, source [PR]

Markov Localization: illustration

1: **Algorithm Markov_localization**($bel(x_{t-1}), u_t, z_t, m$)**:**

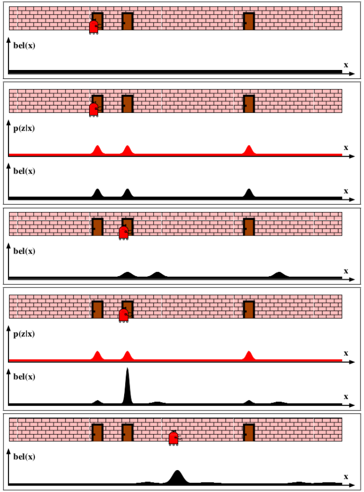
2: *for all* x_t *do*

3: $\overline{bel}(x_t) = \int p(x_t \mid u_t, x_{t-1}, m)\ bel(x_{t-1})\ dx$

4: $bel(x_t) = \eta\ p(z_t \mid x_t, m)\ \overline{bel}(x_t)$

5: *endfor*

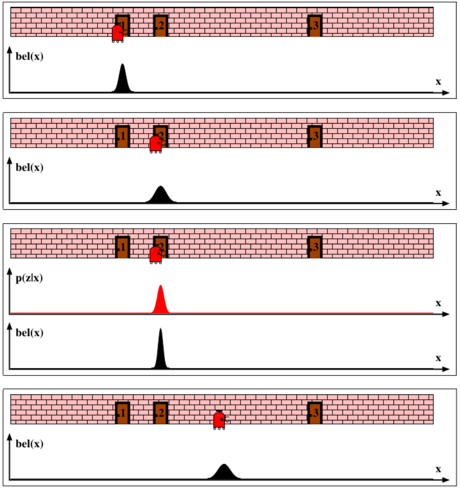
6: *return* $bel(x_t)$



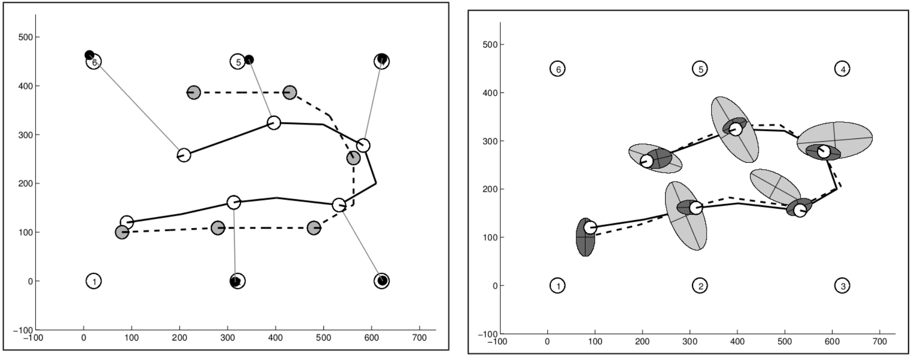
Markov Localization algorithm and Illustration of Markov Localization, source [PR]

EKF Localization for feature-based maps

- Special case of Markov Localization
  - $Bel(x_t) = \mathcal{N}(\mu_t, \Sigma_t)$
  - Range and bearing of nearby features $z_t = \{z_t^1, z_t^2, \dots\}$
  - Feature are uniquely identifiable

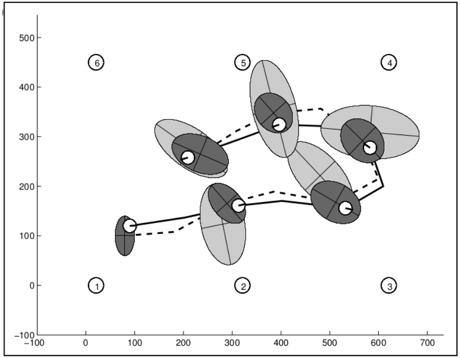


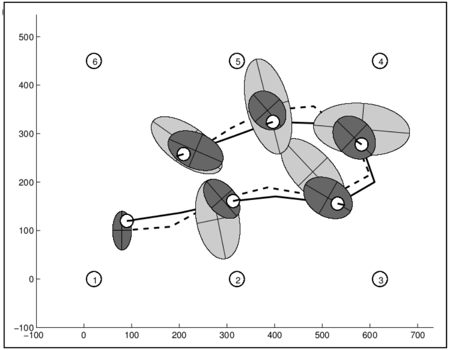
The Extended Kalman Filter: example, accurate landmark sensor



Left, problem setting; Right, EKF estimation; Source [PR]

The Extended Kalman Filter: example, less accurate landmark sensor





Left, EKF estimation with accurate sensor; Right, EKF estimation with less accurate sensor; Source [PR]

Extension and alternatives to EKF localization

- Localization with **unknown** correspondences
  - ML estimation for correspondence between measurements and landmarks
  - Multiple Hypothesis Tracking, use mixture of Gaussian to represent belief for different associations
- Unscented Kalman Filter
  - uses a set of points (sigma points) to represent belief
  - passes points through the non-linear function
- Information Filter
  - different (canonical) parametrization of KF, information vector and matrix
  - Dual of KF, some operations are much faster and viceversa

Summary

- ◇ EKF: extension of Kalman filter
- ◇ Straighforward way to handle non-linearities
- ◇ Works well in practice for moderate non-linearities and small uncertainties (i.e., good sensors)
- ◇ Localization has many different aspects
- ◇ Markov Localization: Bayes filter approach for pose estimation (map is given)
- ◇ EKF, special case of Markov Localization
- ◇ Many extensions to deal with specific issues (e.g., unknown correspondences, MHT, ...)