Mobile Robots

Localization and SLAM Examples

Lecture 9

To solve robot localization or SLAM problem, we need two models:

- *State transition model* (or *motion model*) for the robot which can be described in terms of a probability distribution on state transitions

$$p(s_t|s_{t-1}, u_t),$$

  the state transition is assumed to be a **Markov process** in which the next state s_t depends only on the immediately preceding state s_{t-1} and the applied control u_t, and is independent of both the observations and the map.
- *Observation model* (or *sensor model*) describes the probability of making an observation z_t when the vehicle location s_t and the map m are known

$$p(z_t|s_t, m)$$

- In the SLAM problem we do not know the robot location, neither do we know the environment.

SLAM Problem Formulation

We use Bayes rule to transform these mathematical relationships into a form where we can recover probability distributions over those latent variables from the measured data (recursive formulation):

$$Bel(s_t, m) = \eta \, p(z_t \mid s_t, m) \iint p(s_t, m \mid s_{t-1}, m, u_{t-1}) Bel(s_{t-1}, m) ds_{t-1} dm,$$

where $Bel()$ denotes "belief" distribution, and η is a normalization factor.

Usually, a static environment is assumed

$$\underbrace{Bel(s_t, m)}_{\text{Estimate}} = \eta \; \underbrace{p(z_t \mid s_t, m)}_{\text{Sensor Model}} \int \underbrace{p(s_t \mid s_{t-1}, u_{t-1})}_{\text{Motion Model}} Bel(s_{t-1}, m) ds_{t-1}$$

Odometry Motion Model I

- Odometry information is obtained by integrating wheel encoder data. Integrated pose estimation is available in periodic time intervals.
- We consider odometry data as control signals.
- At time t the pose of the robot is modeled by the random variable s_t. The robot odometry estimates this pose.
- However, due to drift and slippage there is no fixed coordinate transformation between the coordinates used by the robot's internal odometry and the physical world coordinates.
- In the time interval $(t, t-1]$, the robot advances from a pose s_{t-1} to pose s_t.
- The odometry reports back to us a related advance from $\bar{s}_{t-1} = [\bar{x}, \bar{y}, \bar{\theta}]^\mathsf{T}$ to $\bar{s}_t = [\bar{x}', \bar{y}', \bar{\theta}']^\mathsf{T}$.
- The bar indicates that these are odometry measurements embedded in a robot internal coordinate whose relation to the global world coordinate system is unknown.
- The motion information u_t is given by the pair $u_t = [\bar{s}_{t-1}, \bar{s}_t]^\mathsf{T}$.

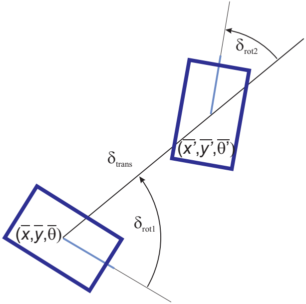
Odometry Motion Model II

- To extract relative odometry, u_t is transformed into a sequence of three steps: the initial rotation δ_{rot1}, followed by the translation δ_{trans}, and the second rotation δ_{rot2}.

$$\delta_{rot1} = \mathrm{atan2}(\bar{y}' - \bar{y},\ \bar{x}' - \bar{x}) - \bar{\theta} \tag{1}$$

$$\delta_{trans} = \sqrt{(\bar{x} - \bar{x}')^2 + (\bar{y} - \bar{y}')^2} \tag{2}$$

$$\delta_{rot2} = \bar{\theta}' - \bar{\theta} - \delta_{rot1} \tag{3}$$



- Each pair $(\bar{s}_{t-1}, \bar{s}_t)$ has a unique parameter vector $[\delta_{rot1}, \delta_{trans}, \delta_{rot2}]^{\mathsf{T}}$, and these parameters form a sufficient statistics of the relative motion encoded by the odometry.
- To model the motion error, we assume that the "true" values of the rotation $\hat{\delta}_{rot1}, \hat{\delta}_{rot2}$ and translation $\hat{\delta}_{trans}$ are obtained from the measured ones and additive independent noise ε with zero mean and variance σ^2 (white Gaussian noise):

$$
\begin{aligned}
\hat{\delta}_{rot1} &= \delta_{rot1} + \varepsilon_{rot1}, & \varepsilon_{rot1} &\sim \mathcal{N}(0, \sigma^2_{rot1}) \\
\hat{\delta}_{trans} &= \delta_{trans} + \varepsilon_{trans}, & \varepsilon_{trans} &\sim \mathcal{N}(0, \sigma^2_{trans}) \\
\hat{\delta}_{rot2} &= \delta_{rot2} + \varepsilon_{rot1}, & \varepsilon_{rot2} &\sim \mathcal{N}(0, \sigma^2_{rot2})
\end{aligned}
$$

Odometry Motion Model IV

- Standard deviations σ are approximated as follows:

$$\begin{aligned}\sigma_{rot1} &= \alpha_1|\delta_{rot1}| + \alpha_2|\delta_{trans}|\\ \sigma_{trans} &= \alpha_3|\delta_{trans}| + \alpha_4(|\delta_{rot1}| + |\delta_{rot2}|)\\ \sigma_{rot2} &= \alpha_1|\delta_{rot2}| + \alpha_2|\delta_{trans}|,\end{aligned}$$

  where $\alpha_1, \ldots \alpha_4$ are weights.
- The true pose s_t is obtained from s_{t-1} by the initial rotation with angle $\hat{\delta}_{rot1}$, followed by the translation with distance $\hat{\delta}_{trans}$ and the second rotation with angle $\hat{\delta}_{rot2}$:

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} \hat{\delta}_{trans}\ \cos(\theta + \hat{\delta}_{rot1}) \\ \hat{\delta}_{trans}\ \sin(\theta + \hat{\delta}_{rot1}) \\ \hat{\delta}_{rot1} + \hat{\delta}_{rot2} \end{pmatrix} \tag{4}$$

- Assuming, that s_t is the true final pose, the error in odometry can be defined as

$$\delta_{rot1} - \hat{\hat{\delta}}_{rot1}$$
$$\delta_{trans} - \hat{\hat{\delta}}_{trans}$$
$$\delta_{rot2} - \hat{\hat{\delta}}_{rot2}$$

- The probability of these errors is given by

$$P_1 = \varepsilon_{rot1}(\delta_{rot1} - \hat{\hat{\delta}}_{rot1}) \tag{5}$$
$$P_2 = \varepsilon_{trans}(\delta_{trans} - \hat{\hat{\delta}}_{trans}) \tag{6}$$
$$P_3 = \varepsilon_{rot2}(\delta_{rot2} - \hat{\hat{\delta}}_{rot2}) \tag{7}$$

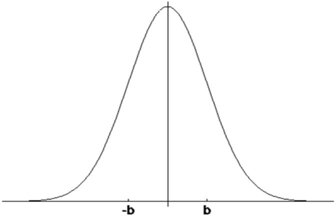
- Since the errors are assumed to be independent, the joint error probability is the product: $P = P_1 \cdot P_2 \cdot P_3$.

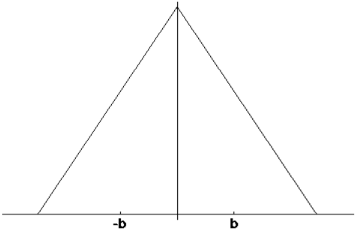
1. Normal distribution with zero mean and variance b^2

$$\varepsilon_{b^2}(x) = \frac{1}{\sqrt{2\pi\ b}}\ exp^{-\frac{x^2}{2b^2}} \tag{8}$$

2. Triangular distribution

$$\varepsilon_{b^2}(x) = \max(0, \frac{1}{\sqrt{6}b} - \frac{|x|}{6b^2}) \tag{9}$$





A posteriori probability for odometry based motion model

Algorithm **motion_model_odometry**(s_t, u_t, s_{t-1})

$\delta_{rot1} = \text{atan2}(\bar{y}' - \bar{y},\ \bar{x}' - \bar{x}) - \bar{\theta}$

$\delta_{trans} = \sqrt{(\bar{x} - \bar{x}')^2 + (\bar{y} - \bar{y}')^2}$

$\delta_{rot2} = \bar{\theta}' - \bar{\theta} - \delta_{rot1}$

$\hat{\delta}_{rot1} = \text{atan2}(y' - y,\ x' - x) - \theta$

$\hat{\delta}_{trans} = \sqrt{(x - x')^2 + (y - y')^2}$

$\hat{\delta}_{rot2} = \theta' - \theta - \hat{\delta}_{rot1}$

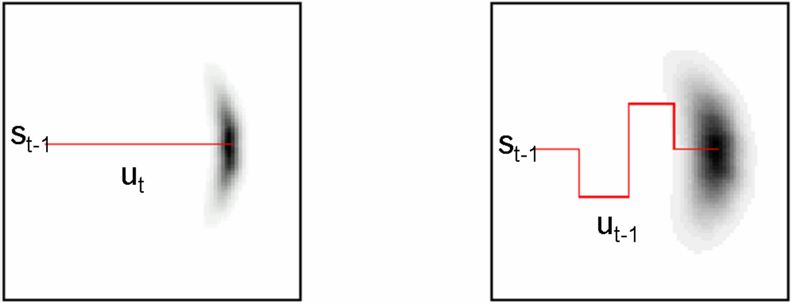
$p_1 = \text{prob}(\delta_{rot1} - \hat{\delta}_{rot1}, \alpha_1|\delta_{rot1}| + \alpha_2|\delta_{trans}|)$

$p_2 = \text{prob}(\delta_{trans} - \hat{\delta}_{trans}, \alpha_3|\delta_{trans}| + \alpha_4(|\delta_{rot1}| + |\delta_{rot2}|))$

$p_3 = \text{prob}(\delta_{rot2} - \hat{\delta}_{rot2}, \alpha_1|\delta_{rot2}| + \alpha_2|\delta_{trans}|)$

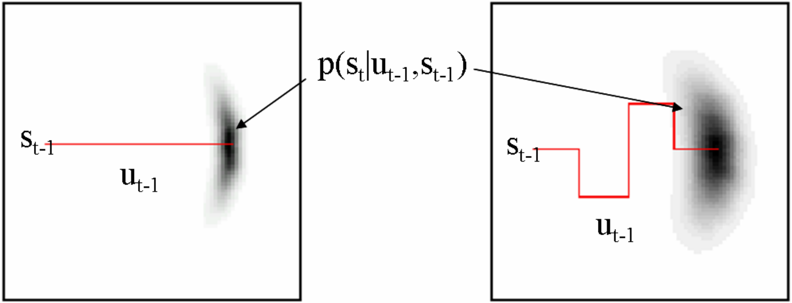
return $p_1 \cdot p_2 \cdot p_3$

Typical Motion Model



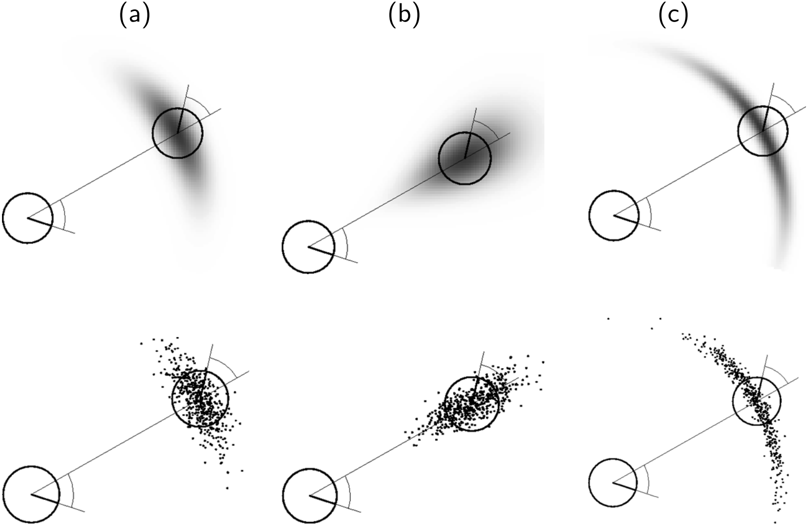
Typical banana-shaped distributions obtained for 2D-projection of 3D posterior.

Typical Motion Model



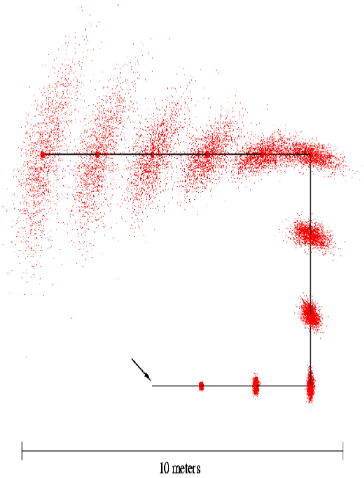
Typical banana-shaped distributions obtained for 2D-projection of 3D posterior.

Examples of Odometry Motion Models



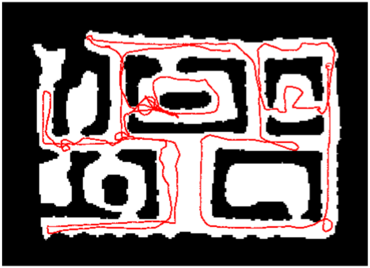
Different noise parameter settings α_1 to α_4: (a) a typical model; (b), (c) unusually large translational and rotational errors, respectively.

Odometry Model

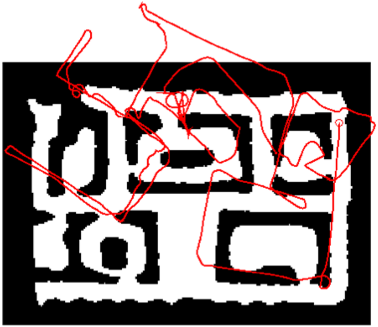


Sampling approximation of the position belief

Odometry Quality



Real path (red line)



Path obtained from the odometry

The central task is to determine $P(z_t \mid s_t, m)$, i.e., the probability of a measurement z_t given that the robot is at pose s_t.
Once again we, use Bayes filter

$$Bel(s_t, m) = \eta \underbrace{P(z_t \mid s_t, m)}_{\text{sensor model}} \int \underbrace{P(s_t \mid s_{t-1}, u_t)}_{\text{motion model}} Bel(s_{t-1}, m) ds_{t-1}$$

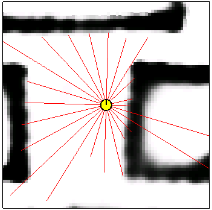
Beam-based Sensor Model. Scan z_t consists of K measurements

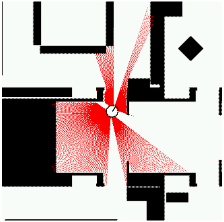
$$z_t = \{z_t^1, \ldots, z_t^K\},$$

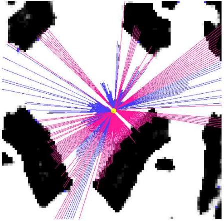
z_t^k refers to an individual measurement.
Individual measurements are independent given the robot position:

$$P(z_t \mid s_t, m) = \prod_{k=1}^{K} P(z_t^k \mid s_t, m) \tag{10}$$

Beam Models of Range Finders





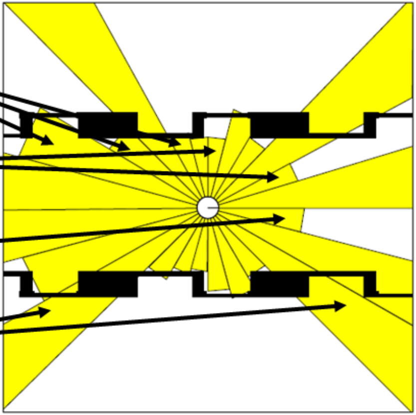


Typical Measurement Errors of Range Finders:

1. *Measurement noise* – this error arises from the limited resolution, atmospheric effects on measurement signal, etc.
2. *Unexpected objects* – environments of mobile robots are dynamic while maps are static (example of moving objects are people).
3. *Failures* – a typical result is a max-range measurement due to specular reflections, light-absorbing objects, etc.
4. *Random measurements* – entirely unexplainable measurements, e.g. sonars often generate phantom readings when they bounce off walls.

Typical Measurement Errors of an Range Measurements

1. Beams reflected by obstacles
2. Beams reflected by persons / caused by crosstalk
3. Random measurements
4. Maximum range measurements



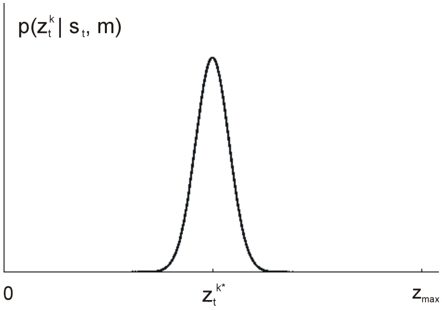
1. *Measurement noise*
   In practice, the measured values are limited to the interval $[0, z_{max}]$, where z_{max} denotes the maximum sensor range and the expected value of the measurement is z_t^{k*}. The measurement probability P_{hit}, is given by

$$P_{hit}(z_t^k \mid s_t, m) = \begin{cases} \frac{\eta}{\sqrt{2\pi\sigma_{hit}^2}} \, e^{\frac{-(z_t^k - z_t^{k*})^2}{2\sigma_{hit}^2}} & \text{if } 0 \leqslant z_t^k \leqslant z_{max} \\ 0 & \text{otherwise,} \end{cases} \tag{11}$$

where normalizer η evaluates to

$$\eta = \left(\int_0^{z_{max}} \frac{1}{\sqrt{2\pi \; \sigma_{hit}}} \, e^{\frac{-(x - z_t^{k*})^2}{2\sigma_{hit}^2}} \, dx \right)^{-1} \tag{12}$$

Beam Model of Range Finders



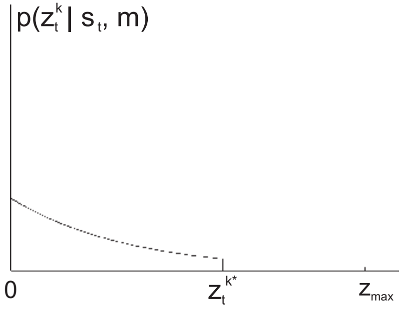
Gaussian distribution P_{hit}

2. *Unexpected objects*
The likelihood of sensing unexpected objects decreases with range. The probability is described by an exponential distribution

$$P_{short}(z_t^k \mid s, m) = \begin{cases} \eta\,\lambda\,e^{-\lambda z_t^k} & \text{if } 0 \leqslant z_t^k \leqslant z_t^{k*} \\ 0 & \text{otherwise,} \end{cases} \tag{13}$$

where the parameter λ is an intrinsic parameter of the measurement model. The value of η can be derived as:

$$\eta = \frac{1}{\int_0^{z_t^{k*}} \lambda\,e^{-\lambda x} dx} = \frac{1}{1 - e^{-\lambda z_t^{k*}}} \tag{14}$$



Exponential distribution P_{short}

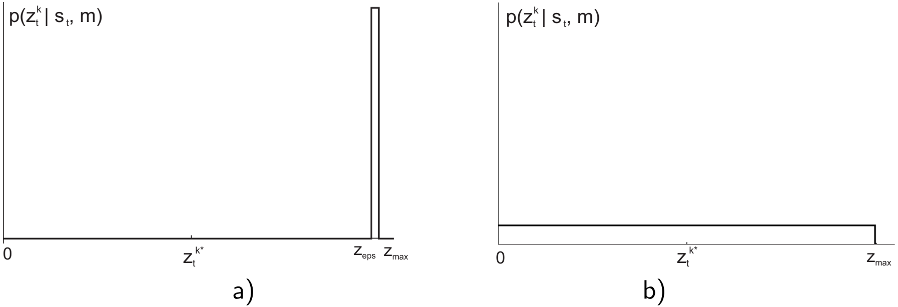
3. *Failures*
   Objects are missed altogether, the sensor returns its maximum allowable value z_{max}. We model this as very narrow uniform distribution centered at z_{max}

$$P_{max}(z_t^k \mid s, m) = \begin{cases} \frac{1}{z_{eps}} & \text{if } z_{max} - z_{eps} \leqslant z_t^k \leqslant z_{max} \\ 0 & \text{otherwise,} \end{cases} \tag{15}$$

   where z_{eps} is the value close to 0 ($z_{eps} \to 0$).

4. *Random measurements*
   Such measurements are modeled using uniform distribution spread over the entire sensor measurement range $[0;\ z_{max}]$:

$$P_{rand}(z_t^k \mid s, m) = \begin{cases} \frac{1}{z_{max}} & \text{if } 0 \leqslant z_t^k \leqslant z_{max} \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

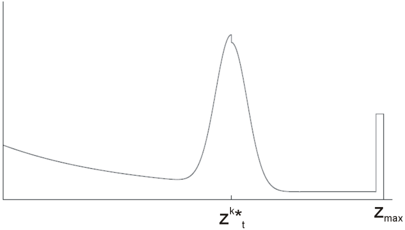


Uniform distributions: a) Failures – max-range error; b) Random measurements

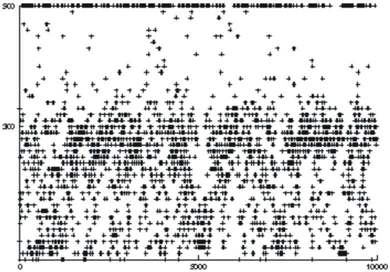
These four different distributions are mixed by a weighted average:

$$P(z_t^k \mid s_t, m) = \begin{pmatrix} \alpha_{hit} \\ \alpha_{short} \\ \alpha_{max} \\ \alpha_{rand} \end{pmatrix}^{\mathsf{T}} \cdot \begin{pmatrix} P_{hit}(z_t^k \mid s_t, m) \\ P_{short}(z_t^k \mid s_t, m) \\ P_{max}(z_t^k \mid s_t, m) \\ P_{rand}(z_t^k \mid s_t, m) \end{pmatrix}, \qquad (17)$$

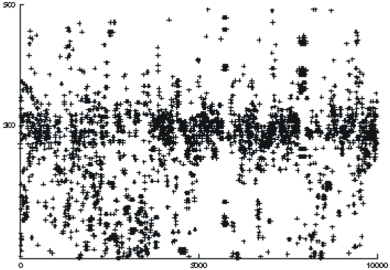
where $\alpha_{hit} + \alpha_{short} + \alpha_{max} + \alpha_{rand} = 1$.



„Pseudo-density" of a typical mixture distribution $P(z_t^k \mid s_t, m)$



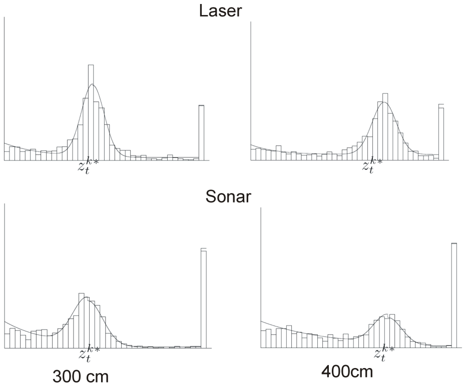
Sonar



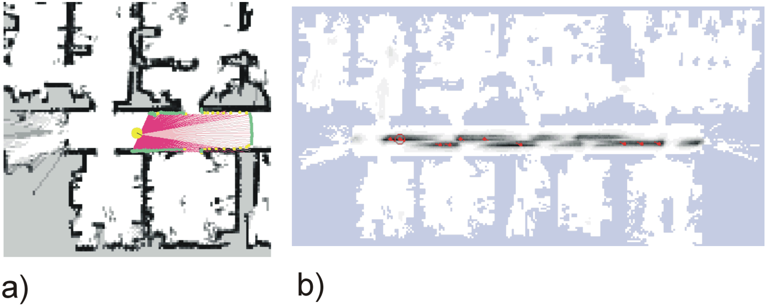
Laser

Typical data (10 000 measurements) obtained with a sonar and a laser-range finder for a "true" range of 300 cm and a maximum range of 500 cm.

Discrete Approximation



- The smaller the range the more accurate measurement.
- The relatively high likelihood of short and random measurements:
  - \+ this model is less susceptible to unmodeled systematic perturbations, such as people who block the robot's path for long periods of time.
  - – it reduces the information in each sensor reading, since the difference in likelihood between a hit and a random measurement is small.



Probabilistic model of perception:
(a) Laser range scan, project into a previously acquired map;
(b) The likelihood $P(z_t|s_t, m)$ evaluated for all s_t (shown in gray). The darker a position, the larger $P(z_t|s_t, m)$.

Summary of Sensor Models

- Explicitly modeling uncertainty in sensing is key to robustness.
- In many cases, good models can be found by the following approach:
  1. Determine parametric model of noise free measurement.
  2. Analyze sources of noise.
  3. Add adequate noise to parameters (eventually mix in densities for noise).
  4. Learn (and verify) parameters by fitting model to data.
  5. Likelihood of measurement is given by "probabilistically comparing" the actual with the expected measurement.
- This holds for motion models as well.
- It is extremely important to be aware of the underlying assumptions!
- In choosing the right model, it is important to trade off physical realism with properties that might be desirable for an algorithm using these models.
- As a rule of thumb, the more accurate a model, the better.

Discrete Localization using Bayes Filter

Belief distribution of the robot pose x_t

$$Bel(x_t) = \eta^{-1} \overbrace{p(z_t \,|\, x_t)}^{\text{Sensor Model}} \sum_{x_{t-1}} \overbrace{p(x_t \,|\, x_{t-1}, u_t)}^{\text{Motion Model}} Bel(x_{t-1})$$

Two main steps in Bayes Filter:

1. Prediction update

$$Bel'(x_t) = p(x_t \,|\, x_{t-1}, u_t) * Bel(x_{t-1}) = \sum_{x_{t-1}} p(x_t \,|\, x_{t-1}, u_t)\, Bel(x_{t-1})$$

2. Correction (measurement update)

$$Bel(x_t) = \eta^{-1}\, p(z_t \,|\, x_t)\, Bel'(x_t)$$

Let $d = \{u_1, z_1, u_2, z_2, \ldots, u_k, z_k\}$.
Algorithm *Discrete_Bayes_Filter*$(Bel(x), d)$:

```
η = 0
IF d is an action data u THEN
    FOR ∀x_t DO
        Bel'(x_t) = Σ_{x_{t-1}} p(x_t | x_{t-1}, u_t) bel(x_{t-1})
ELSE IF d is a measurement data z THEN
    FOR ∀x DO
        Bel(x_t) = p(z_t | x_t) Bel'(x_t)
        η = η + Bel(x_t)
    FOR ∀x_t DO
        Bel(x_t) = η^{-1} Bel(x_t)
RETURN Bel(x_t)
```

Discrete 1D Localization – Example

We consider a robot moving in a one dimensional environment. Let us discretize the configuration space into 5 cells. We need to solve 1D discrete localization problem (Fig. 1a). Robot, at time t, can be only in one discrete position $x = 1, 2, 3, 4, 5$.

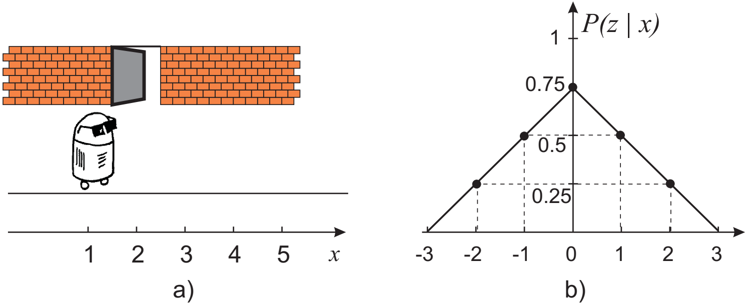


Figure: 1D localization: a) robot states, b) sensor probabilistic model

Discrete 1D Localization – Example

- Robot can move right ($u = 1$) or left ($u = -1$).
- If the control command ($u = 1$ or $u = -1$) is sent, robot can move one unit or stay in place.
- Assume that state $x = 5$ precedes/follows state $x = 1$ and vice versa (a loop).
- Probabilistic motion model is the following:

$$P(x+1|x, u=1) = \frac{3}{4}, \qquad P(x|x, u=1) = \frac{1}{4};$$

$$P(x-1|x, u=-1) = \frac{3}{4}, \qquad P(x|x, u=-1) = \frac{1}{4}$$

- The door is located at cell $x = 2$, that robot can detect it (*measurement* z).
- Robot can detect the door, only if it closer than 3 units (the probabilistic sensor model is shown in Fig. 1b).
- We assume that initial robot position is unknown.
- Robot moves and takes measurements. The following data sequence has been acquired: $d = \{u = 1, z, u = 1, z\}$, which means that at $t = 1$ the robot executed control command $u = 1$, at $t = 2$ performed door measurement z, etc.

Problem:

a) Calculate $Bel_t(x)$ for $t = 1, \ldots, 4$ using Discrete Bayes Filter algorithm.

b) What is the most probable final robot position?

Discrete 1D Localization – Solution I

$t = 0$: $Bel_0(x = i) = \frac{1}{5}$ dla $i = 1, \ldots, 5$ – discrete uniform distribution

$t = 1$: $u_1 = 1$, $Bel_t(x) = \sum_{x'=1}^{5} P(x|x', u) \cdot Bel_{t-1}(x')$ ← prediction step

$$Bel'_1(1) = P(1|1,1)Bel_0(1) + P(1|5,1)Bel_0(5) = \frac{1}{4} \cdot \frac{1}{5} + \frac{3}{4} \cdot \frac{1}{5} = \frac{1}{5}$$

$$Bel'_1(2) = P(2|1,1)Bel_0(1) + P(2|2,1)Bel_0(2) = \frac{3}{4} \cdot \frac{1}{5} + \frac{1}{4} \cdot \frac{1}{5} = \frac{1}{5}$$

$$Bel'_1(3) = P(3|2,1)Bel_0(2) + P(3|3,1)Bel_0(3) = \frac{3}{4} \cdot \frac{1}{5} + \frac{1}{4} \cdot \frac{1}{5} = \frac{1}{5}$$

$$Bel'_1(4) = P(4|3,1)Bel_0(3) + P(4|4,1)Bel_0(4) = \frac{3}{4} \cdot \frac{1}{5} + \frac{1}{4} \cdot \frac{1}{5} = \frac{1}{5}$$

$$Bel'_1(5) = P(5|4,1)Bel_0(4) + P(5|5,1)Bel_0(5) = \frac{3}{4} \cdot \frac{1}{5} + \frac{1}{4} \cdot \frac{1}{5} = \frac{1}{5}$$

Discrete 1D Localization – Solution II

$t = 2$: z_2, $\eta = 0$, $Bel_t(x) = P(z_t|x) \cdot Bel'_{t-1}(x)$ ← correction step

$$Bel_2(1) = P(z_2|1)Bel_1(1) = \frac{1}{2} \cdot \frac{1}{5} = \frac{1}{10}; \quad \eta = \eta + Bel_2(1) = \frac{1}{10}$$

$$Bel_2(2) = P(z_2|2)Bel_1(2) = \frac{3}{4} \cdot \frac{1}{5} = \frac{3}{20}; \quad \eta = \eta + Bel_2(2) = \frac{1}{4}$$

$$Bel_2(3) = P(z_2|3)Bel_1(3) = \frac{1}{2} \cdot \frac{1}{5} = \frac{1}{10}; \quad \eta = \eta + Bel_2(3) = \frac{7}{20}$$

$$Bel_2(4) = P(z_2|4)Bel_1(4) = \frac{1}{4} \cdot \frac{1}{5} = \frac{1}{20}; \quad \eta = \eta + Bel_2(4) = \frac{8}{20}$$

$$Bel_2(5) = P(z_2|5)Bel_1(5) = \frac{1}{4} \cdot \frac{1}{5} = \frac{1}{20}; \quad \eta = \eta + Bel_2(5) = \frac{9}{20}$$

Normalization: $Bel_2(i) = Bel_2(i)/\eta$ dla $i = 1, \ldots, 5$

$$Bel_2(1) = \frac{1}{10} \cdot \frac{20}{9} = \frac{2}{9}; \qquad Bel_2(2) = \frac{3}{20} \cdot \frac{20}{9} = \frac{3}{9};$$

$$Bel_2(3) = \frac{1}{10} \cdot \frac{20}{9} = \frac{2}{9}; \qquad Bel_2(4) = \frac{1}{20} \cdot \frac{20}{9} = \frac{1}{9};$$

$$Bel_2(5) = \frac{1}{20} \cdot \frac{20}{9} = \frac{1}{9}$$

$t = 3$: $u_3 = 1,\ Bel_t(x) = \sum_{x'=1}^{5} P(x|x',u) \cdot Bel_{t-1}(x')$ ← prediction step

$$Bel'_3(1) = P(1|1,1)Bel_2(1) + P(1|5,1)Bel_2(5) = \frac{1}{4} \cdot \frac{2}{9} + \frac{3}{4} \cdot \frac{1}{9} = \frac{5}{36}$$

$$Bel'_3(2) = P(2|1,1)Bel_2(1) + P(2|2,1)Bel_2(2) = \frac{3}{4} \cdot \frac{2}{9} + \frac{1}{4} \cdot \frac{3}{9} = \frac{9}{36}$$

$$Bel'_3(3) = P(3|2,1)Bel_2(2) + P(3|3,1)Bel_2(3) = \frac{3}{4} \cdot \frac{3}{9} + \frac{1}{4} \cdot \frac{2}{9} = \frac{11}{36}$$

$$Bel'_3(4) = P(4|3,1)Bel_2(3) + P(4|4,1)Bel_2(4) = \frac{3}{4} \cdot \frac{2}{9} + \frac{1}{4} \cdot \frac{1}{9} = \frac{7}{36}$$

$$Bel'_3(5) = P(5|4,1)Bel_2(4) + P(5|5,1)Bel_2(5) = \frac{3}{4} \cdot \frac{1}{9} + \frac{1}{4} \cdot \frac{1}{9} = \frac{4}{36}$$

Discrete 1D Localization – Solution IV

$t = 4$: $z_4,\ \eta = 0,\ Bel_t(x) = P(z_t|x) \cdot Bel'_{t-1}(x)$ $\leftarrow$ correction step

$Bel_4(1) = P(z_4|1)Bel_3(1) = \frac{1}{2} \cdot \frac{5}{36} = \frac{5}{72}; \quad \eta = \eta + Bel_4(1) = \frac{5}{72}$

$Bel_4(2) = P(z_4|2)Bel_3(2) = \frac{3}{4} \cdot \frac{9}{36} = \frac{27}{144}; \quad \eta = \eta + Bel_4(2) = \frac{37}{144}$

$Bel_4(3) = P(z_4|3)Bel_3(3) = \frac{1}{2} \cdot \frac{11}{36} = \frac{11}{72}; \quad \eta = \eta + Bel_4(3) = \frac{59}{144}$

$Bel_4(4) = P(z_4|4)Bel_3(4) = \frac{1}{4} \cdot \frac{7}{36} = \frac{7}{144}; \quad \eta = \eta + Bel_4(4) = \frac{66}{144}$

$Bel_4(5) = P(z_4|5)Bel_3(5) = \frac{1}{4} \cdot \frac{4}{36} = \frac{4}{144}; \quad \eta = \eta + Bel_4(5) = \frac{70}{144}$

Normalization: $Bel_4(i) = Bel_4(i)/\eta$ dla $i = 1, \ldots, 5$

$Bel_2(1) = \frac{5}{72} \cdot \frac{144}{70} = \frac{10}{70}; \quad Bel_2(2) = \frac{27}{144} \cdot \frac{144}{70} = \mathbf{\frac{27}{70}};$

$Bel_2(3) = \frac{11}{72} \cdot \frac{144}{70} = \frac{22}{70}; \quad Bel_2(4) = \frac{7}{144} \cdot \frac{144}{70} = \frac{7}{70};$

$Bel_2(5) = \frac{4}{144} \cdot \frac{144}{70} = \frac{4}{70}.$

The most likely robot final position at $t = 4$ is $x = 2$.

Discrete Kalman Filter

- Given the linear dynamical system:

$$s_k = A_k s_{k-1} + B_k u_k + w_k \text{ – state transition (action) model}$$
$$z_k = H_k s_k + v_k \text{ – observation model,}$$

where:
s_k – a state vector at time k (unknown),
u_k – a vector of control inputs, (known),
z_k – a measurement vector (known, measured)
A_k – state transition matrix, B_k – output matrix, H_k – observation matrix; (all matrices are known).

- Random variables representing the process w_k and measurement noise v_k are zero-mean white Gaussians with probability distributions:

$$p(w_k) \sim \mathcal{N}(0, Q_k),$$
$$p(v_k) \sim \mathcal{N}(0, R_k),$$

where Q_k i R_k are known covariance matrices.

- Kalman Filter recursively computes estimates of state s_k, which is evolving according to the process and observation models.

The filter has two distinct stages:

1. **Prediction** – projects the current state and error covariance estimates ahead in time to obtain the *a priori* estimates for the next time step.
2. **Correction** – incorporates a new measurement into the *a priori* estimate to obtain an improved *a posteriori* estimate.

The filter estimates the process state at some time and then obtains feedback in the form of (noisy) measurements.

$$\hat{s}_k = A_k \hat{s}_{k-1} + B_k u_k + w_k$$
$$z_k = H_k \hat{s}_k + v_k,$$

where $\hat{s}_k$ state estimate at time k.

State estimate:

- $\hat{s}_k^-$ – *a priori* state estimate at step k s_k given knowledge of the process prior to step k.
- $\hat{s}_k$ – *a posteriori* state estimate at step k given measurement z_k.

We can then define *a priori* and *a posteriori* estimate errors as:

$$\varepsilon_k^- = s_k - \hat{s}_k^-$$
$$\varepsilon_k = s_k - \hat{s}_k,$$

The *a priori* and *a posteriori* estimate error covariance:

$$P_k^- = E[\varepsilon_k^- \varepsilon_k^{-T}] \text{ – \textit{a priori}}$$
$$P_k = E[\varepsilon_k \varepsilon_k^\mathsf{T}] \text{ – \textit{a posteriori}},$$

where $E[\cdot]$ is the expectation.

Discrete Kalman Filter Algorithm I

1. Prediction: Use the state transition model to obtain *a priori* state estimate: $\hat{s}_k^-$ and *a priori* estimate error covariance P_k^-

$$\hat{s}_k^- = A_k \hat{s}_{k-1} + B_k u_k$$
$$P_k^- = A_k P_{k-1} A_k^\mathsf{T} + Q_k$$

2. Correction: The vector λ_k is called the measurement *innovation* or the *residual* and S_k is the covariance of the innovation.

$$\lambda_k = z_k^r - z_k = z_k^r - H_k \hat{s}_k^-$$
$$S_k = H_k P_k^- H_k^\mathsf{T} + R_k,$$

where z_k^r is the actual measurement, and z_k is the predicted measurement.

The residual reflects the discrepancy between the predicted measurement and the actual measurement.

Discrete Kalman Filter Algorithm II

The task during the correction step is to compute the Kalman gain K_k and the error covariance P_k:

$$
\begin{aligned}
K_k &= P_k^- H_k^\mathsf{T} S_k^{-1} \\
\hat{s}_k &= \hat{s}_k^- + K_k \lambda_k \\
P_k &= (I - K_k H_k) P_k^-
\end{aligned}
$$

After each prediction-correction pair, the process is repeated with the previous *a posteriori* estimates used to project or predict the new *a priori* estimates.

1D Kalman Filter

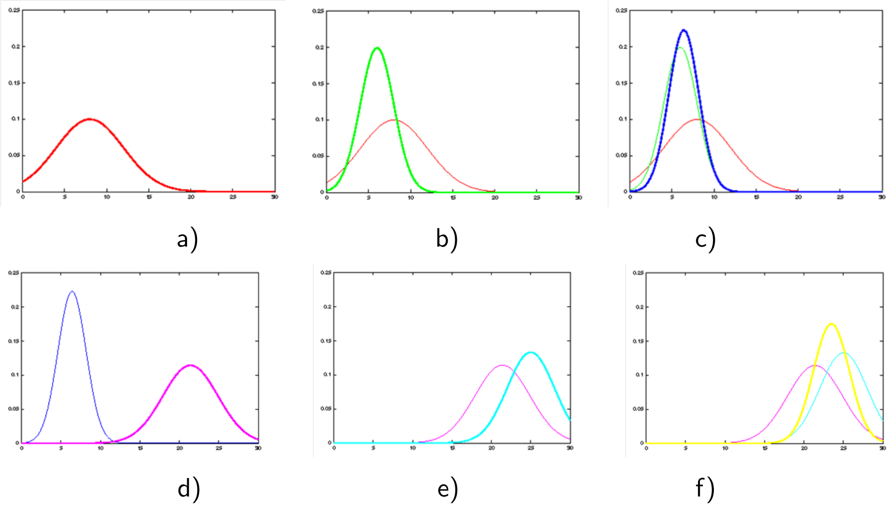


Illustration of Kalman filters: a) initial belief, b) a measurement with the associated uncertainty, c) after integrating the measurement into the belief, d) after motion to the right(which introduces uncertainty), e) a new measurement, f) the resulting belief

The Extended Kalman Filter (EKF) I

State transitions and measurements are rarely linear in practice.
In EKF the state probability and measurement probability a given by nonlinear functions:

$$
\begin{aligned}
s_k &= f(s_{k-1}, u_k, w_k) \\
z_k &= h(s_k, v_k)
\end{aligned}
$$

In practice, of course one does not know the individual values of the noise at each time step

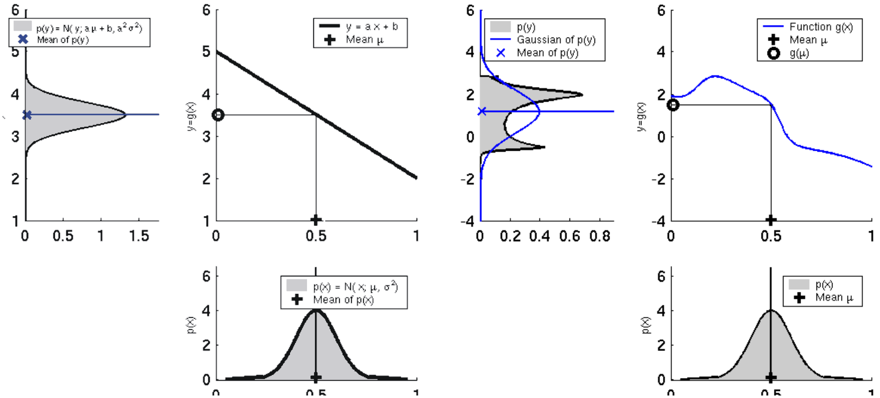
$$
\begin{aligned}
\hat{s}_k^- &= f(\hat{s}_{k-1}, u_k, 0) \\
z_k^- &= h(\hat{s}_k, 0)
\end{aligned}
$$

The EKF is simply an *ad hoc* state estimator that only approximates the optimality of Bayes' rule by linearization.

We can linearize the estimation (via Taylor series expansion) around the current estimate using the partial derivatives of the process and measurement functions to compute estimates:

$$s_k \approx \hat{s}_k^- + A_k(s_{k-1} - \hat{s}_{k-1}) + W_k w_k$$
$$z_k \approx z_k^- + H_k(s_k - \hat{s}_k^-) + V_k v_k$$

where
$A_{[i,j]} = \dfrac{\partial f_{[i]}}{\partial s_{[j]}}(\hat{s}_{k-1}, u_k, 0)$, $W_{[i,j]} = \dfrac{\partial f_{[i]}}{\partial w_{[j]}}(\hat{s}_{k-1}, u_k, 0)$, $H_{[i,j]} = \dfrac{\partial h_{[i]}}{\partial s_{[j]}}(\hat{s}_k, 0)$,
$V_{[i,j]} = \dfrac{\partial h_{[i]}}{\partial v_{[j]}}(\hat{s}_k, 0)$.



a) b)

(a) Linear, (b) nonlinear transformation. The random variable is passed through the linear and nonlinear function

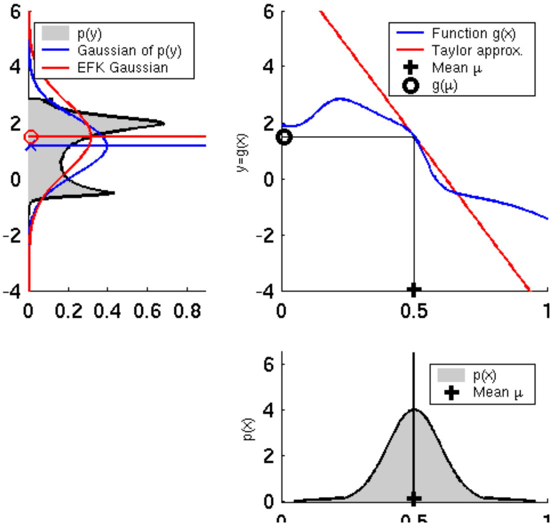
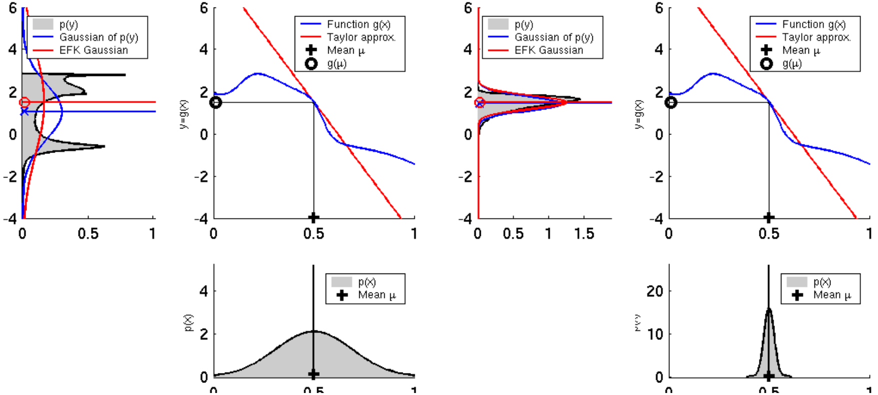


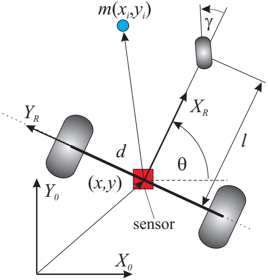
Illustration of linearisation applied by the EKF. The linearisation incurs an approximation error



a) b)

Both Gaussians have the same mean and are passed through the same nonlinear function. The higher uncertainty produces a more distorted density

EKF SLAM – A Single Landmark Example



Tricycle robot

Deterministic kinematic model:

$$\begin{cases} \dot{x} = v\cos\theta \\ \dot{y} = v\sin\theta \\ \dot{\theta} = \dfrac{v\tan\gamma}{l} \end{cases},$$

where v is a linear velocity of the robot.

Nonlinear probabilistic model:

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} x_k + \Delta t\, v_k \cos\theta \\ y_k + \Delta t\, v_k \sin\theta \\ \theta_k + \frac{\Delta t v_k \tan\gamma}{l} \end{bmatrix} + w_k,$$

where w_k is a process noise.

EKF SLAM – A Single Landmark Example

Recall that in the SLAM algorithm, landmarks are assumed to be stationary:
$m_{k+1} = m_k = [x_{1,k}, y_{1,k}]^\mathsf{T} = [x_1, y_1]^\mathsf{T}$.
The state transition model:

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \\ x_{1,k+1} \\ y_{1,k+1} \end{bmatrix} = \begin{bmatrix} x_k + \Delta t\, v_k \cos\theta \\ y_k + \Delta t\, v_k \sin\theta \\ \theta_k + \frac{\Delta t\, v_k \tan\gamma}{l} \\ x_{1,k} \\ y_{1,k} \end{bmatrix} + \begin{bmatrix} w_{x,k} \\ w_{y,k} \\ w_{\theta,k} \\ 0 \\ 0 \end{bmatrix}$$

The observation model:

$$z_k = h(s_k) + \nu_k$$

The sensor returns the range r_k and bearing α_k to a landmark 1. Thus, the observation model is

$$r_k = \sqrt{(x_1 - x_k)^2 + (y_1 - y_k)^2} + \nu_{r,k}$$
$$\alpha_k = \operatorname{atan2}(y_1 - y_k,\, x_1 - x_k) - \theta_k + \nu_{\alpha,k}$$

EKF SLAM – A Single Landmark Example

Prediction:

$$\hat{s}_k^- = f(\hat{s}_{k-1},\, u_k,\, 0)$$
$$P_k^- = A_k P_{k-1} A_k^\mathsf{T} + W_k Q_k W_k^\mathsf{T},$$

where $A_k = \frac{\partial f}{\partial s}$ is the Jacobi matrix:

$$A_k = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial \theta} & \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial y_1} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial \theta} & \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial y_1} \\ \frac{\partial f_3}{\partial x} & \frac{\partial f_3}{\partial y} & \frac{\partial f_3}{\partial \theta} & \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial y_1} \\ \frac{\partial f_4}{\partial x} & \frac{\partial f_4}{\partial y} & \frac{\partial f_4}{\partial \theta} & \frac{\partial f_4}{\partial x_1} & \frac{\partial f_4}{\partial y_1} \\ \frac{\partial f_5}{\partial x} & \frac{\partial f_5}{\partial y} & \frac{\partial f_5}{\partial \theta} & \frac{\partial f_5}{\partial x_1} & \frac{\partial f_5}{\partial y_1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\Delta t\, v_k \sin\theta & 0 & 0 \\ 0 & 1 & \Delta t\, v_k \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

EKF SLAM – A Single Landmark Example

Prediction:

$$\hat{s}_k^- = f(\hat{s}_{k-1},\, u_k,\, 0)$$
$$P_k^- = A_k P_{k-1} A_k^{\mathsf{T}} + W_k Q_k W_k^{\mathsf{T}},$$

where $W_k = \frac{\partial f}{\partial w}$ is the Jacobi matrix:

$$W_k = \begin{bmatrix} \frac{\partial f_1}{\partial w_x} & \frac{\partial f_1}{\partial w_y} & \frac{\partial f_1}{\partial w_\theta} & \frac{\partial f_1}{\partial w_{x1}} & \frac{\partial f_1}{\partial w_{y1}} \\ \frac{\partial f_2}{\partial w_x} & \frac{\partial f_2}{\partial w_y} & \frac{\partial f_2}{\partial w_\theta} & \frac{\partial f_2}{\partial w_{x1}} & \frac{\partial f_2}{\partial w_{y1}} \\ \frac{\partial f_3}{\partial w_x} & \frac{\partial f_3}{\partial w_y} & \frac{\partial f_3}{\partial w_\theta} & \frac{\partial f_3}{\partial w_{x1}} & \frac{\partial f_3}{\partial w_{y1}} \\ \frac{\partial f_4}{\partial w_x} & \frac{\partial f_4}{\partial w_y} & \frac{\partial f_4}{\partial w_\theta} & \frac{\partial f_4}{\partial w_{x1}} & \frac{\partial f_4}{\partial w_{y1}} \\ \frac{\partial f_5}{\partial w_x} & \frac{\partial f_5}{\partial w_y} & \frac{\partial f_5}{\partial w_\theta} & \frac{\partial f_5}{\partial w_{x1}} & \frac{\partial f_5}{\partial w_{y1}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

EKF SLAM – A Single Landmark Example

The observation model:

$$z_k = \begin{bmatrix} r_k \\ \alpha_k \end{bmatrix} = \begin{bmatrix} \sqrt{(x_1 - \hat{x}_k^-)^2 + (y_1 - \hat{y}_k^-)^2} \\ \text{atan2}(y_1 - \hat{y}_k^-, x_1 - \hat{x}_k^-) - \hat{\theta}_k^- \end{bmatrix} + \nu_k,$$

where $H_k = \frac{\partial h}{\partial w}$ is the Jacobi matrix:

$$H_k = \begin{bmatrix} \frac{\partial h_1}{\partial x} & \frac{\partial h_1}{\partial y} & \frac{\partial h_1}{\partial \theta} & \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial y_1} \\ \frac{\partial h_2}{\partial x} & \frac{\partial h_2}{\partial y} & \frac{\partial h_2}{\partial \theta} & \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial y_1} \end{bmatrix} = \begin{bmatrix} \frac{x - x_1}{r} & \frac{y - y_1}{r} & 0 & \frac{x_1 - x}{r} & \frac{y_1 - y}{r} \\ \frac{y_1 - y}{r^2} & \frac{x_1 - x}{r^2} & -1 & \frac{y - y_1}{r^2} & \frac{x - x_1}{r^2} \end{bmatrix},$$

where $r = \sqrt{(x_1 - x_k)^2 + (y_1 - y_k)^2}$.

The observation model:

$$z_k = \begin{bmatrix} r_k \\ \alpha_k \end{bmatrix} = \begin{bmatrix} \sqrt{(x_1 - \hat{x}_k^-)^2 + (y_1 - \hat{y}_k^-)^2} \\ \operatorname{atan2}(y_1 - \hat{y}_k^-, x_1 - \hat{x}_k^-) - \hat{\theta}_k^- \end{bmatrix} + \nu_k,$$

where $V_k = \frac{\partial h}{\partial \nu}$ is the Jacobi matrix:

$$V_k = \begin{bmatrix} \frac{\partial h_1}{\partial \nu_r} & \frac{\partial h_1}{\partial \nu_\theta} \\ \frac{\partial h_2}{\partial \nu_r} & \frac{\partial h_2}{\partial \nu_\theta} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

The Kalman gain:

$$K_k = P_k^- H_k^\mathsf{T}(H_k P_k^- H_k^\mathsf{T} + R_k)^{-1} = \frac{P_k^- H_k^\mathsf{T}}{H_k P_k^- H_k^\mathsf{T} + R_k}$$

Correction:

$$\hat{s}_k = \hat{s}_k^- + K_k(z_k - h(\hat{s}_k^-, 0))$$
$$P_k = (I - K_k H_k) P_k^-$$

There is only one landmark and it is incorporated into the model from the start.
z_k is 10 fabricated measurements of range and bearing to landmark 1 at point (3, 4).