Localization

What is it?

“The problem of determining the pose of a robot relative to a given map of the environment.”

Thrun, S., Burgard, W., Fox, D. (2005). *Probabilistic robotics*. Cambridge, Mass.: MIT Press.

Taxonomy of problems

- **Position tracking**: accommodate local uncertainty of a robot with known initial pose;
- **Global localization**: localizing a robot from scratch.
  - robot knows that it doesn't know where it is.
- **Kidnapping** a well-localized robot is secretly teleported somewhere else without being told—it is the hardest of the three localization problems.
  - the robot might believe it knows where it is while it does not.

Why challenging?
Sensors

- Sensor noise
  - reduces the useful information content of sensor readings.
    - E.g. Illumination condition for cameras
    - Solution: multiple readings, temporal or multisensor fusion

Why challenging?
Sensors

- Perceptual aliasing
  - many-to-one mapping from environmental states to the robot’s perceptual input
  - human-sized maze - extremely difficult to solve without landmarks or clues
    - without visual uniqueness, human localization competence degrades rapidly.

Why challenging?
Effectors

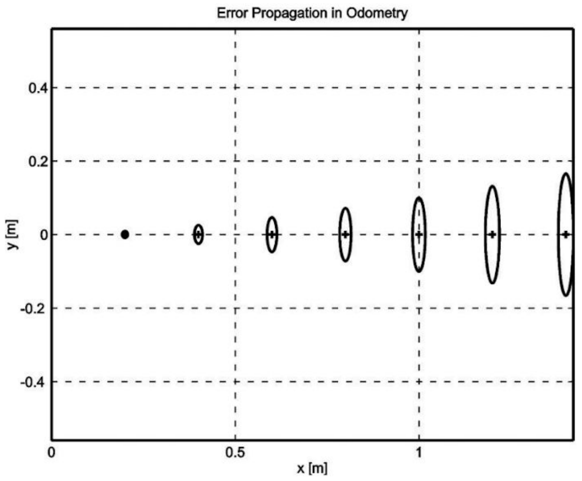
- Error in motion = error in odometry
    - inability to estimate position over time from kinematics and dynamics.
- Source → incomplete model of the environment
    - floor profile
        - Variation in the contact point of the wheel;
        - Unequal floor contact (slipping, nonplanar surface, etc.)
    - actual wheel behaviour
        - Misalignment of the wheels (deterministic);
        - Uncertainty in the wheel diameter and in particular unequal wheel diameter (deterministic);
    - a human pushes the robot.
- physical motion ≠ intended motion ≠ proprioceptive sensor estimates of motion.

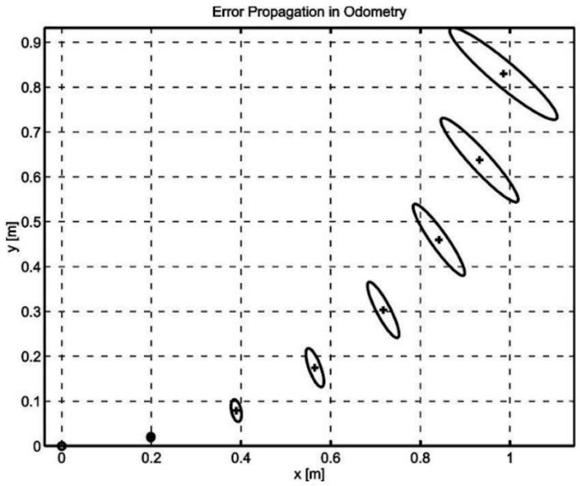
Why challenging?
Effectors

- Range error: integrated path length of the robot’s movement
  - → sum of the wheel movements
- Turn error: similar to range error, but for turns
  - → difference of the wheel motions
- Drift error: difference in the error of the wheels leads to an error in the robot’s angular orientation

Error model
Effectors

- Establish an error model for odometric accuracy and see how the errors propagate over time.





Probabilistic robotics

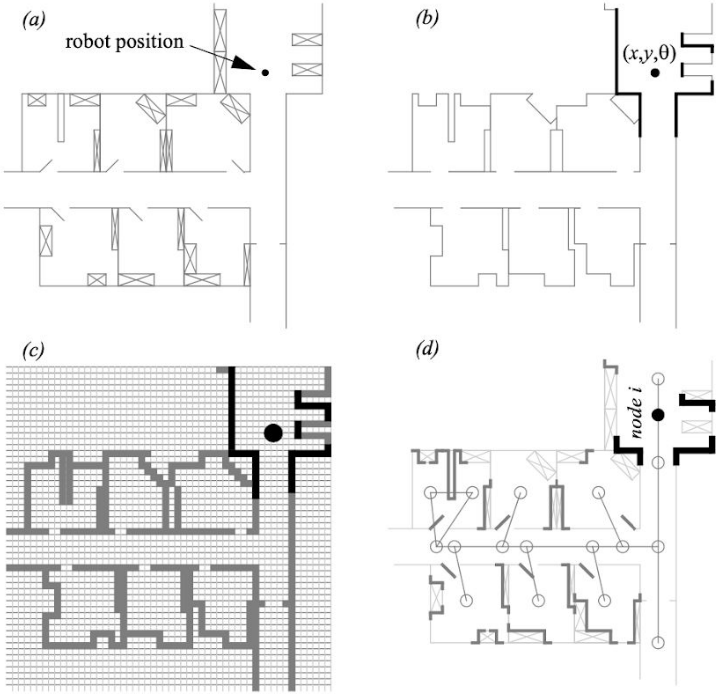
Uncertainty → Probability Theory

Belief representation

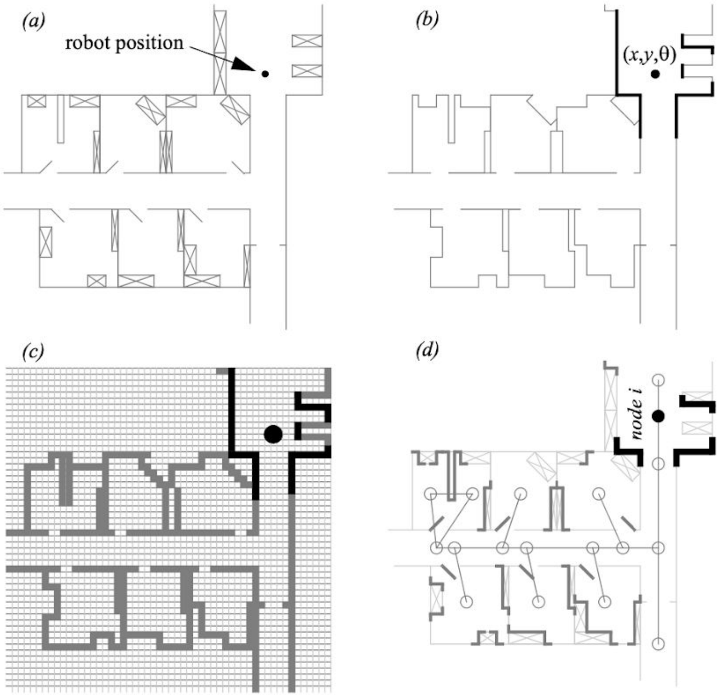
- The robot must have a representation of its belief regarding its position on the map.
- Design questions
  - Single unique position?
  - Set of possible positions?
  - If multiple possible positions are expressed in a single belief, how are those multiple positions ranked, if at all?

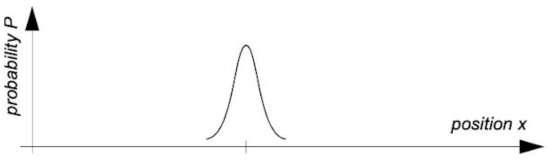
Belief representation

Maps...



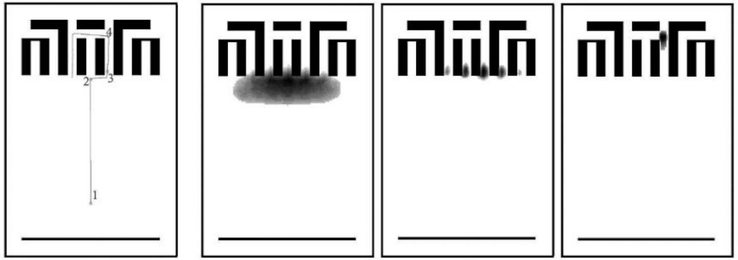
Belief representation
Single- hypothesis belief
Examples





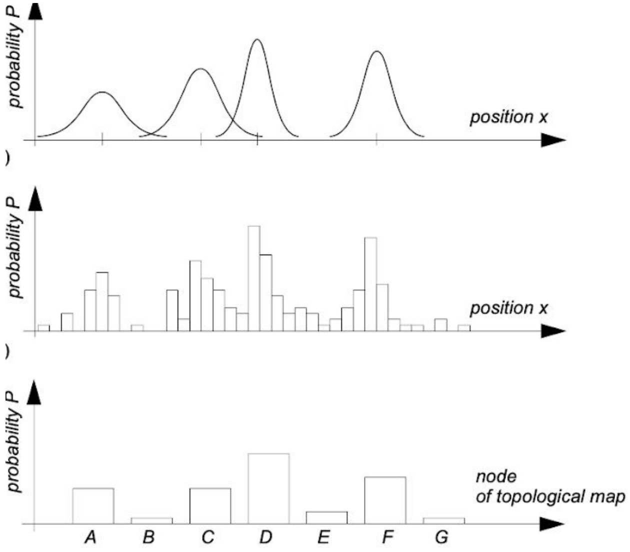
Continuous map, single-hypothesis belief

Belief representation
Multiple-hypothesis belief
Examples



Path of the robot

Belief states at positions 2, 3, and 4



Continuous map, multiple-hypothesis belief

Discretized grid map, p(x) for all possible positions

Discretized topological map, p(x) for all possible positions

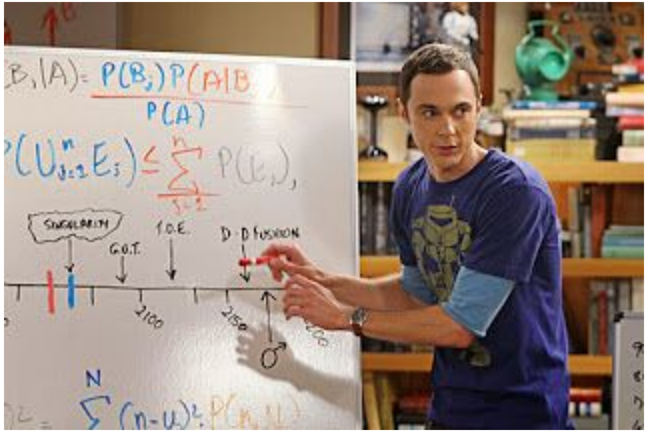
Belief representation

Single-hypothesis belief	*Multiple-hypothesis belief*
No position ambiguity	Explicitly maintain uncertainty. Partial sensor information conceptually incorporated.
Facilitates decision-making at the robot's cognitive level (e.g., path planning)	Reason about the future trajectory of own belief state, e.g choose paths that minimize its future position uncertainty
Updating the robot's belief regarding position is also facilitated	Decision-making hard (consider most probable location)
Forcing a single hypothesis position is challenging	Computationally expensive

Stats

Bayes Rule

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)}$$



- *prior probability distribution p(x)*: the probability distribution of robot location over the whole space **prior** taking into account any **sensor measurement.**
- *posterior probability distribution p(x|y)*: ***after*** reading the **sensor data y**
- *"inverse" conditional probability p(y|x)*: observing those measurements if the robot was at that position

Terminology

$X_T = \{x_0, x_1, x_2, \ldots, x_T\}$	robot **path** is given (location)
$U_T = \{u_0, u_1, u_2, \ldots, u_T\}$	**proprioceptive senso**r readings at time t
$Z_T = \{z_0, z_1, z_2, \ldots, z_T\}$	**observations**, measurement data, or **exteroceptive sensor** readings (sensor reference frame)
$M = \{m_0, m_1, m_2, \ldots, m_{n-1}\}$	map of the environment m_i, e.g., the 2D position of the point or the position and orientation of the line).

Terminology

$$bel(x_t) = p(x_t | z_{1 \to t}, u_{1 \to t})$$

Belief: best guess about the robot state;
probability of the robot being at x_t given *all* its *past observations* $z_{1 \to t}$ and *all* its *past control inputs* $u_{1 \to t}$

$$\overline{bel}(x_t) = p(x_t | z_{1 \to t-1}, u_{1 \to t})$$

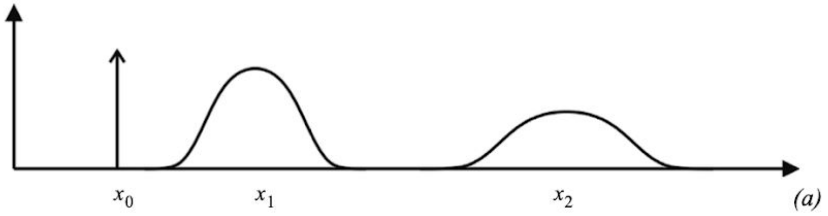
Prediction (or action) update: Belief just before including the new observation z_t;

- robot physically moves (action)
- current robot pose predicted on the basis of the motion control and previous observations.

- **correction (or perception, or measurement) update:** the calculation of bel(x_t) from the prediction update - the robot pose is corrected after the observation.

Ingredients

Initial belief

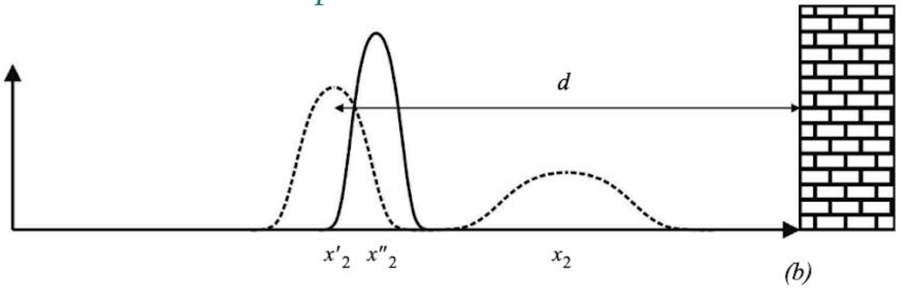


Kown Map: x-axis

Proprioceptive data

Probabilistic motion model used for prediction update

Exteroceptive data



Probabilistic observation model used for perception update

Markov localization

```
for all x_t do
```

$$\overline{bel}(x_t) = \int p(x_t | u_t, x_{t-1}) bel(x_{t-1}) dx_{t-1} \quad \text{(prediction update)}$$

$$bel(x_t) = \eta p(z_t | x_t, M) \overline{bel}(x_t) \quad \text{(measurement update)}$$

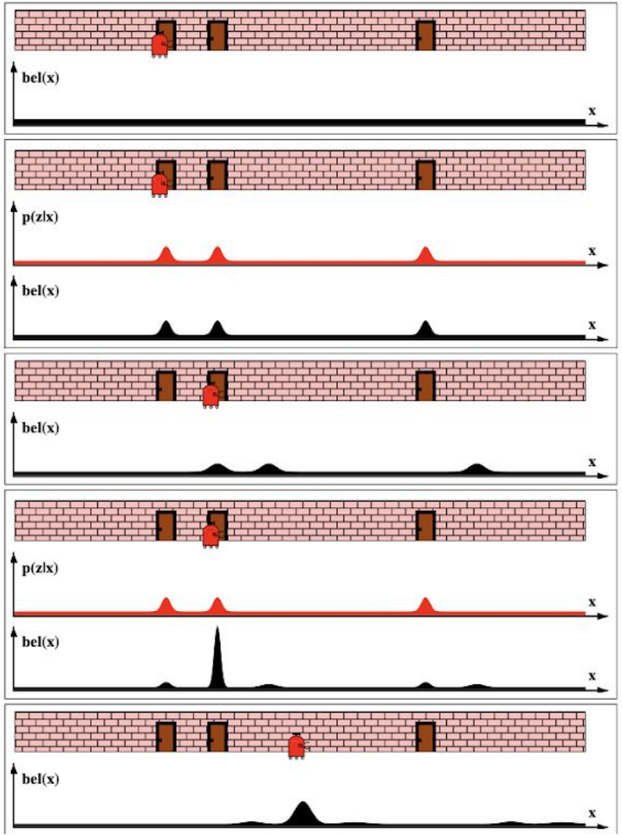
```
endfor
return bel(x_t)
```

Markov assumption:

the ***output*** x_t is a function ***only*** of the robot's **previous state** x_{t-1} and its **most recent *actions*** (odometry) u_t and ***perception*** z_t.

- Of course, not a valid assumption, BUT...
- Simplifies tracking, reasoning, and planning,
- Bayes filters surprisingly robust to violations

Markov localization



Initial belief $bel(x_0)$: uniform over all poses

Observation

Measurement update

Prediction update/ The robot moves to the right; increase in uncertainty

Observation

Measurement update / most of the prob mass focused on correct pos

Prediction update

for all x_t do

$$\overline{bel}(x_t) = \int p(x_t \mid u_t, x_{t-1}) bel(x_{t-1}) dx_{t-1} \quad \text{(prediction update)}$$

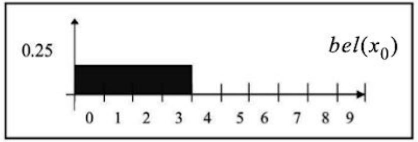
$$bel(x_t) = \eta p(z_t \mid x_t, M) \overline{bel}(x_t) \quad \text{(measurement update)}$$

endfor

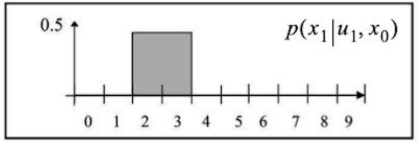
return $bel(x_t)$

Localization

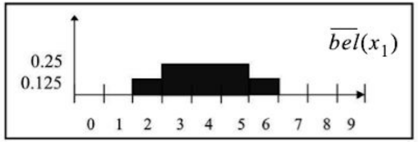
Markov localization



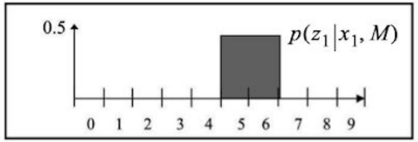
Initial belief $bel(x_0)$: uniform over all poses



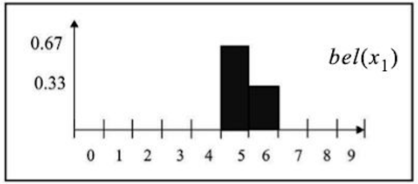
Movement
Probabilistic motion model



Prediction update



Observation
Probabilistic measurement model



Measurement update

for all x_t do

$\overline{bel}(x_t) = \int p(x_t|u_t, x_{t-1})bel(x_{t-1})dx_{t-1}$ (prediction update)

$bel(x_t) = \eta p(z_t|x_t, M)\overline{bel}(x_t)$ (measurement update)

endfor
return $bel(x_t)$

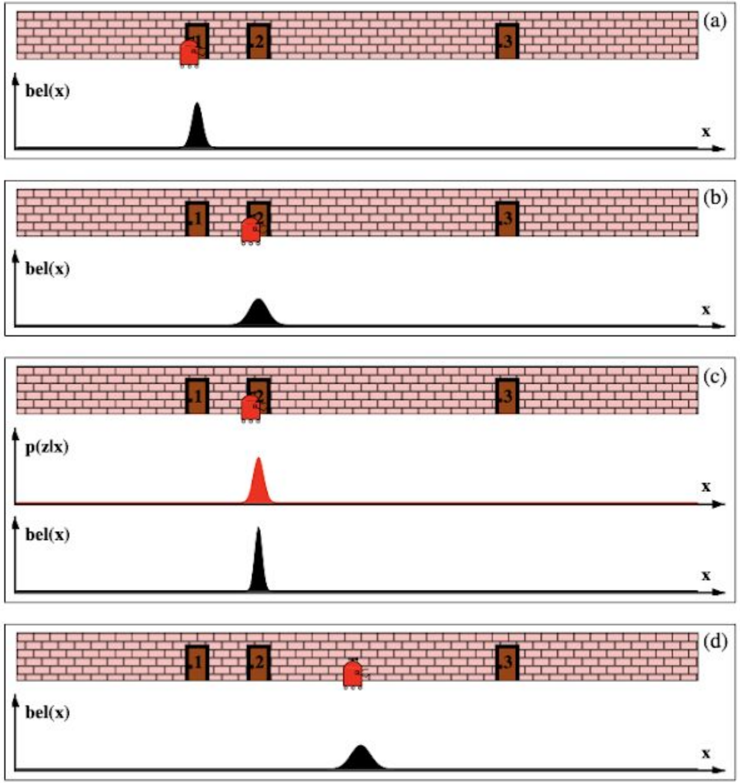
$$p(x_1 = 2) = p(x_0 = 0)p(u_1 = 2) = 0.125\,,$$

$$p(x_1 = 3) = p(x_0 = 0)p(u_1 = 3) + p(x_0 = 1)p(u_1 = 2) = 0.25$$

$$p(x_1 = 4) = p(x_0 = 1)p(u_1 = 3) + p(x_0 = 2)p(u_1 = 2) = 0.25$$

$$p(x_1 = 5) = p(x_0 = 2)p(u_1 = 3) + p(x_0 = 3)p(u_1 = 2) = 0.25$$

$$p(x_1 = 6) = p(x_0 = 3)p(u_1 = 3) = 0.125$$



Localization
Kalman Filter

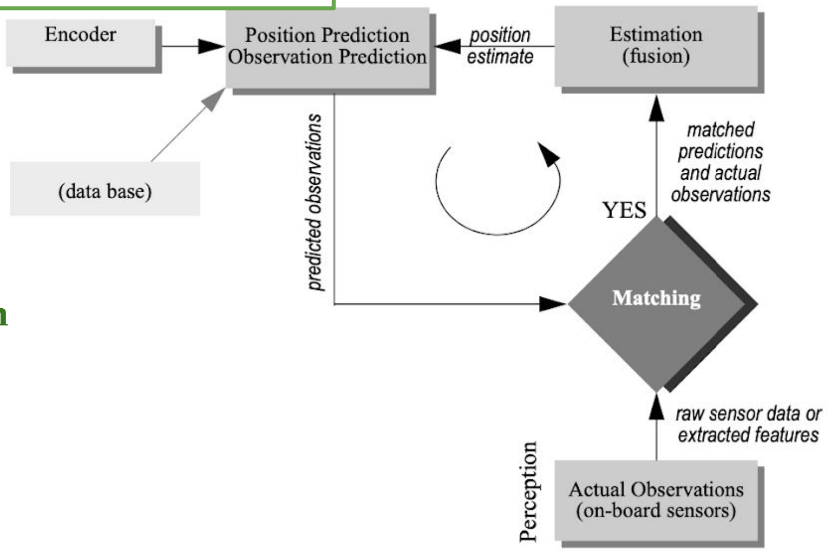
- *Kalman filter localization algorithm*, or KF localization → a special case of Markov localization.
- *Gaussians* to represent:
  - robot belief bel(x_t)
  - motion model
  - the measurement model.
- *Initial location* **must be** *known* with a certain approximation

Localization
Kalman Filter

$$\hat{x}_t = f(x_{t-1}, u_t)$$

$$\hat{P}_t = F_x \cdot P_{t-1} \cdot F_x^T + F_u \cdot Q_t \cdot F_u^T,$$

Probabilistic motion model



Prediction
Update

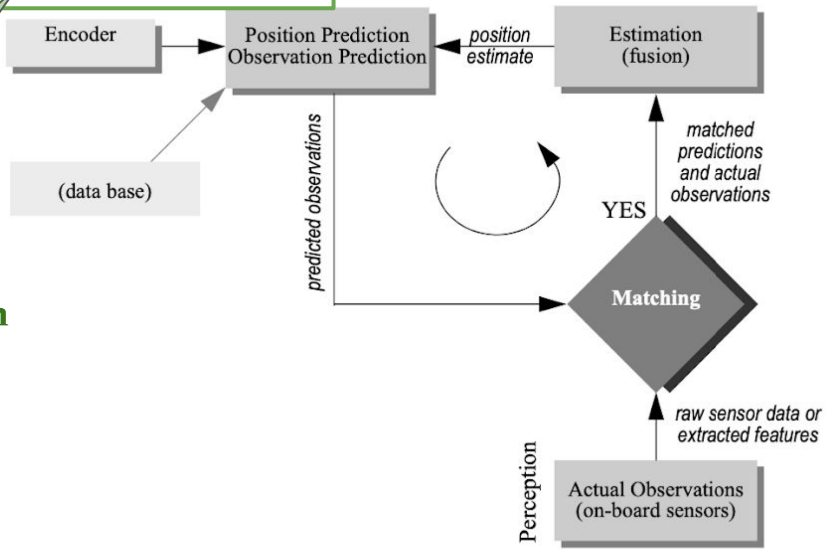
Localization
Kalman Filter

$$\hat{x}_t = f(x_{t-1}, u_t)$$

$$\hat{P}_t = F_x \cdot P_{t-1} \cdot F_x^{\ T} + F_u \cdot Q_t \cdot F_u^{\ T},$$

Motion model error

previous state cov

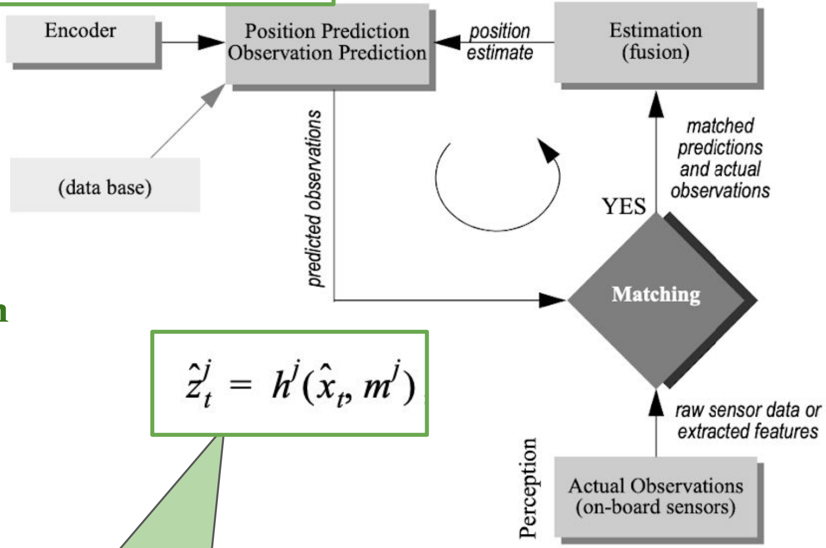


Prediction
Update

Localization
Kalman Filter

$$\hat{x}_t = f(x_{t-1}, u_t)$$

$$\hat{P}_t = F_x \cdot P_{t-1} \cdot F_x^T + F_u \cdot Q_t \cdot F_u^T,$$



Prediction Update

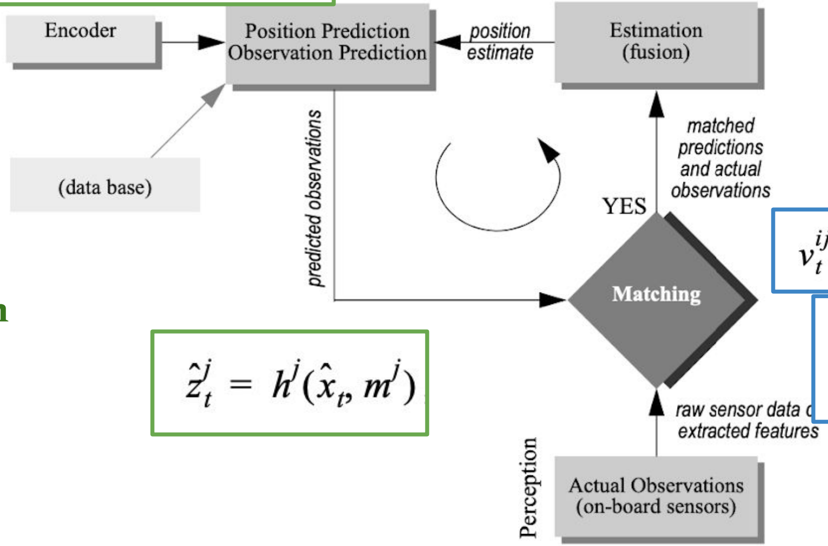
$$\hat{z}_t^j = h^j(\hat{x}_t, m^j)$$

Probabilistic measurement model

Localization
Kalman Filter

$$\hat{x}_t = f(x_{t-1}, u_t)$$

$$\hat{P}_t = F_x \cdot P_{t-1} \cdot F_x^T + F_u \cdot Q_t \cdot F_u^T,$$



Prediction Update

$$\hat{z}_t^j = h^j(\hat{x}_t, m^j)$$

$$v_t^{ijT} \cdot (\Sigma_{IN_t}^{ij})^{-1} \cdot v_t^{ij} \le g^2$$

Validation gate

$$v_t^{ij} = [z_t^i - \hat{z}_t^j] = [z_t^i - h^j(\hat{x}_t, m^j)]$$

$$\Sigma_{IN_t}^{ij} = H^j \cdot \hat{P}_t \cdot H^{jT} + R_t^i$$

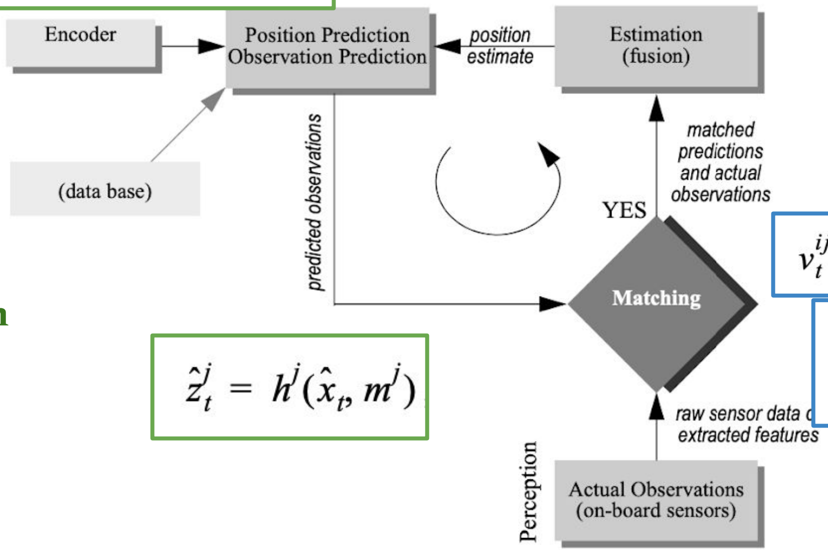
Innovation & Innovation Covariance

Measurement update

Localization
Kalman Filter

$$\hat{x}_t = f(x_{t-1}, u_t)$$

$$\hat{P}_t = F_x \cdot P_{t-1} \cdot F_x^T + F_u \cdot Q_t \cdot F_u^T,$$



$$v_t^{ijT} \cdot (\Sigma_{IN_t}^{ij})^{-1} \cdot v_t^{ij} \le g^2$$

Validation gate

Prediction Update

$$\hat{z}_t^j = h^j(\hat{x}_t, m^j)$$

$$v_t^{ij} = [z_t^i - \hat{z}_t^j] = [z_t^i - h^j(\hat{x}_t, m^j)]$$

$$\Sigma_{IN_t}^{ij} = H^j \cdot \hat{P}_t \cdot H^{jT} + R_t^i$$

Innovation & Innovation Covariance

Sensor error

Measurement update

Localization

Kalman Filter

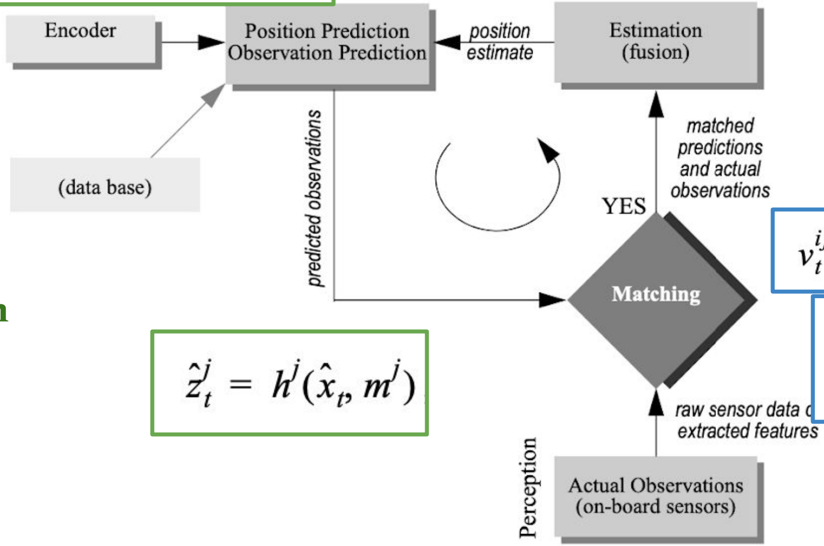
$$K_t = \hat{P}_t \cdot H_t^T \cdot (\Sigma_{IN_t})^{-1}$$

$$x_t = \hat{x}_t + K_t v_t$$

$$P_t = \hat{P}_t - K_t \cdot \Sigma_{IN_t} \cdot K_t^T$$

$$\hat{x}_t = f(x_{t-1}, u_t)$$

$$\hat{P}_t = F_x \cdot P_{t-1} \cdot F_x^T + F_u \cdot Q_t \cdot F_u^T,$$



$$v_t^{ijT} \cdot (\Sigma_{IN_t}^{ij})^{-1} \cdot v_t^{ij} \le g^2$$

Validation gate

$$v_t^{ij} = [z_t^i - \hat{z}_t^j] = [z_t^i - h^j(\hat{x}_t, m^j)]$$

$$\Sigma_{IN_t}^{ij} = H^j \cdot \hat{P}_t \cdot H^{jT} + R_t^i$$

Innovation & Innovation Covariance

Prediction Update

$$\hat{z}_t^j = h^j(\hat{x}_t, m^j)$$

Measurement update

Localization
Kalman Filter

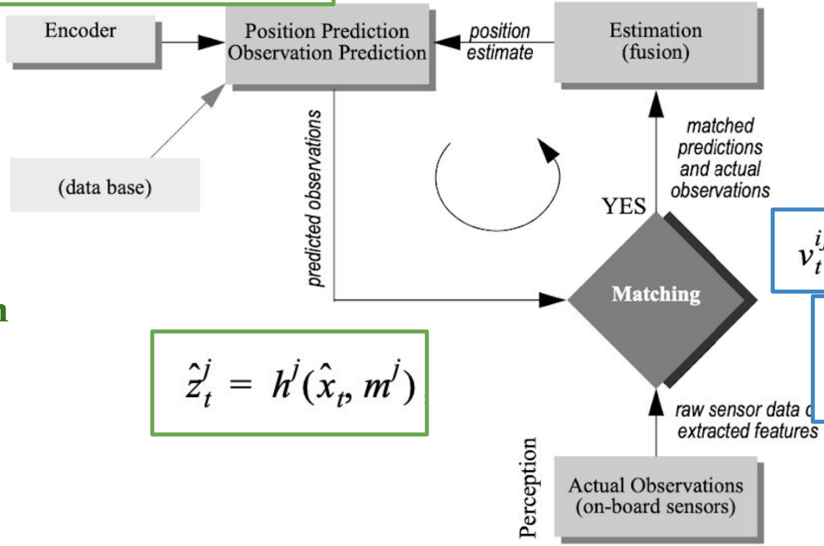
$$\hat{x}_t = f(x_{t-1}, u_t)$$

$$\hat{P}_t = F_x \cdot P_{t-1} \cdot F_x^{\ T} + F_u \cdot Q_t \cdot F_u^{\ T},$$

$$x_t = \hat{x}_t + K_t v_t$$

$$P_t = \hat{P}_t - K_t \cdot \Sigma_{IN_t} \cdot K_t^T$$

$$K_t = \hat{P}_t \cdot H_t^T \cdot (\Sigma_{IN_t})^{-1}$$



$$v_t^{ijT} \cdot (\Sigma_{IN_t}^{ij})^{-1} \cdot v_t^{ij} \le g^2$$

Validation gate

$$v_t^{ij} = [z_t^i - \hat{z}_t^j] = [z_t^i - h^j(\hat{x}_t, m^j)]$$

$$\Sigma_{IN_t}^{ij} = H^j \cdot \hat{P}_t \cdot H^{jT} + R_t^i$$

Innovation & Innovation Covariance

Prediction Update

$$\hat{z}_t^j = h^j(\hat{x}_t, m^j)$$

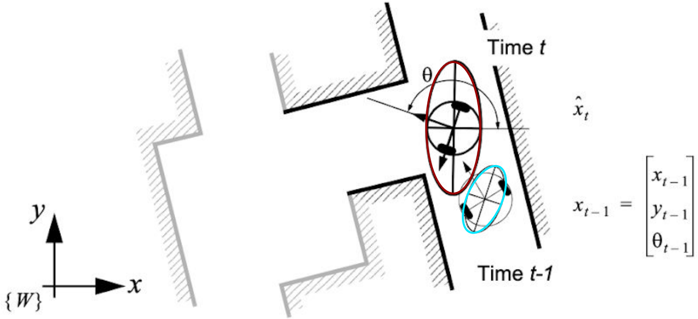
Measurement update

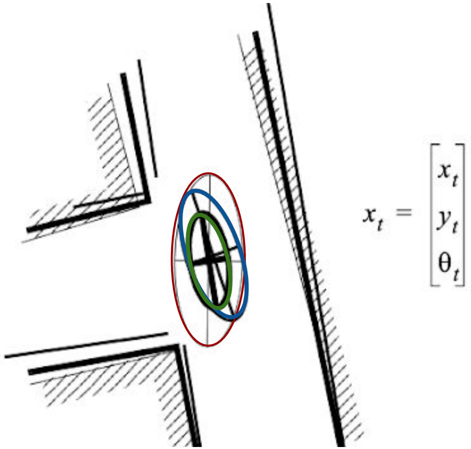
When h is identity (deadreckoning) →

$$x_t = \hat{x}_t + \hat{P}_t(\hat{P}_t + R_t)^{-1}(z_t - \hat{x}_t)$$

$$P_t = \hat{P}_t - \hat{P}_t(\hat{P}_t + R_t)^{-1}\hat{P}_t,$$

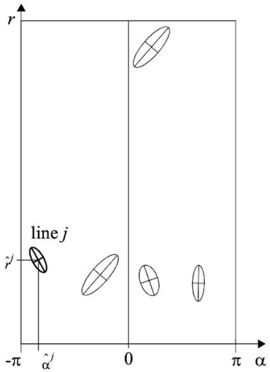
Localization
Kalman Filter



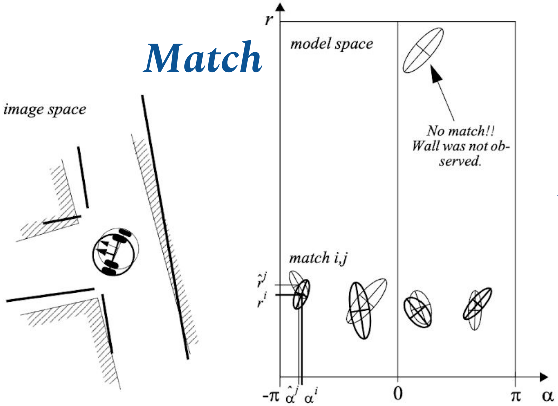


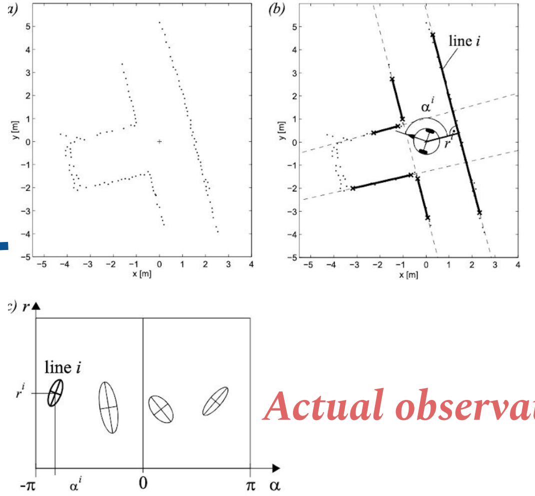
Estimation

Observation prediction



Match

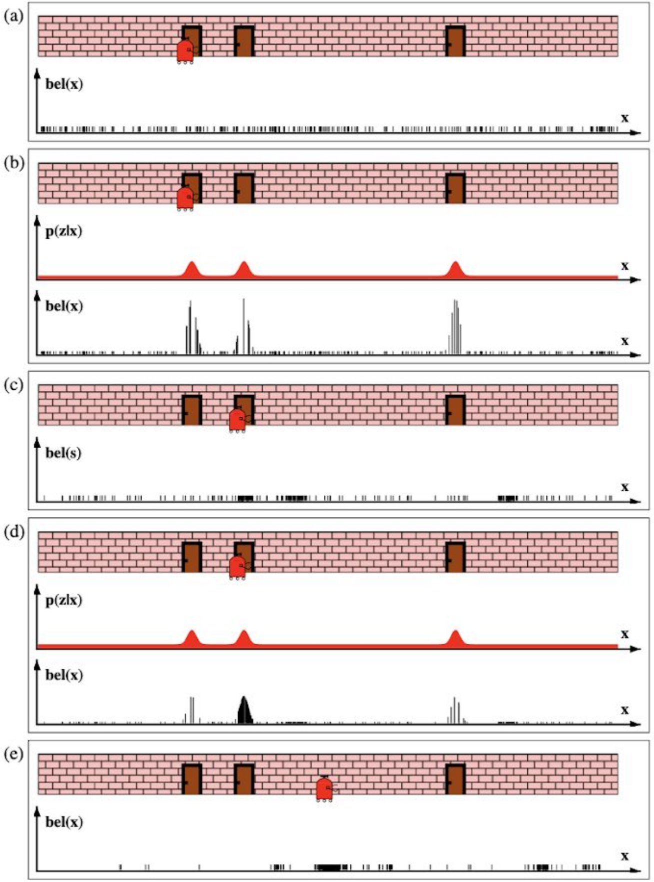




Actual observation

Localization

Monte-Carlo localization



Set of pose particles:
Randomly and uniformly drawn

Observation:
importance factors to each particle -changes height of particles

Robot motion and resampling:
new particle set: uniform importance weights, increased number of particles near the 3 likely places.

Another observation:
most of the cumulative probability mass centered on the second door → most likely location

Localization

Monte-Carlo localization