Mobile robot control 2020: Tutorial #2
Algorithms for robotics

MAY 6^{TH} 2020

Bob Hendrikx, Marzieh Dolatabadi Farahani, Wouter Houtman

Mechanical engineering, Control Systems Technology

EINDHOVEN UNIVERSITY OF TECHNOLOGY

Contents

Robot algorithms and examples in practice:

- *Localization*
- *Feature detection and tracking*
- *Robot motion planning and control*

- **Goal**: provide an overview of algorithms and techniques used for mobile robot control in practice

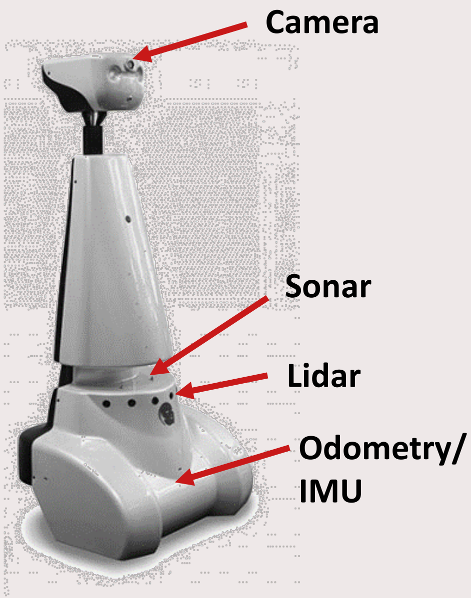
Robot localization

- Robots use ***proprioceptive*** *sensors* for local motion sensing
- Combined with ***exteroceptive*** sensors to *associate* with *external world* in which task is defined

Localization means:

- *Making* ***associations*** *between* ***sensor-data features*** *and* ***objects***
- **Infer** the **location** of *things* based on this **sensor data**

What **algorithms** can we apply to this problem?



TU/e

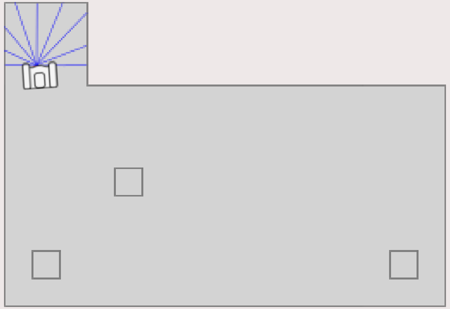
Robot localization

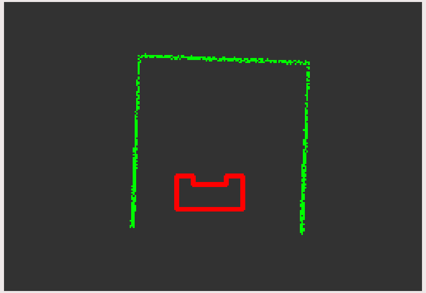
- *Making* ***associations*** *between* ***sensor-data features*** *and* ***objects***
- **Infer** the **location** of *things* based on this **sensor data**

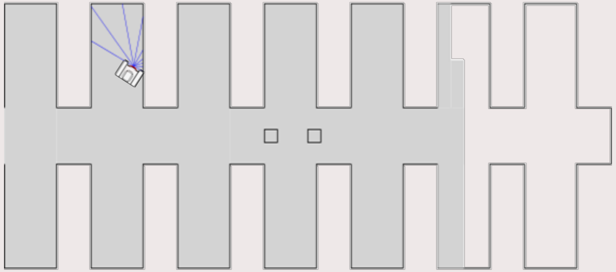
'Classical' localization formulation:
"How to ***infer*** *the* ***robot pose*** *from* ***sensor data****?"*

This is challenging because:
- We often cannot directly *sense* the robot pose
- What we can *sense* is obscured by *noise*
- What we sense does not uniquely determine the robot pose
- Dynamic objects are not on the map

Is every localization problem the same?







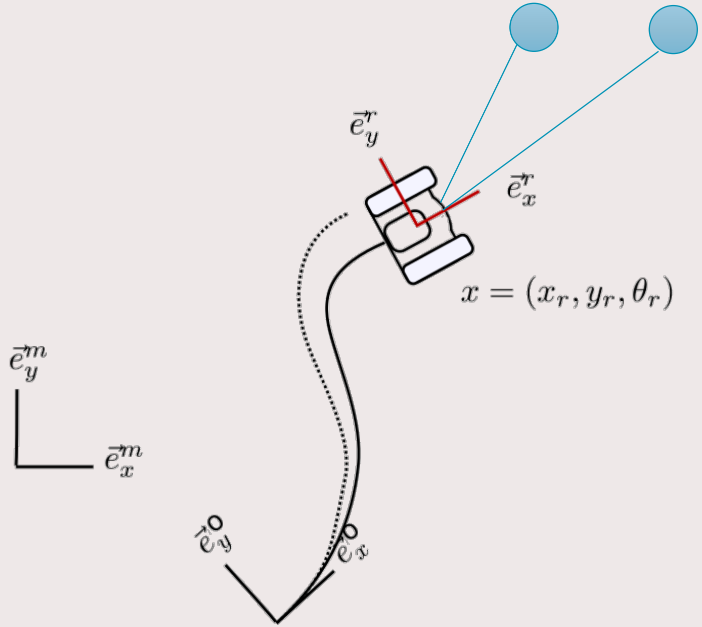
Classical taxonomy of localization problem

- **Tracking** - keeping track of the robot pose **starting** from **known location**
  - Kalman filters / Particle filters
- **Global localization** – Finding the robot pose **without initial knowledge**
  - Particle filters / Multiple hypothesis kalman filters
- **Kidnapped robot problem** – **Changing** the robot pose **without informing** it
  - Heuristic solutions

*All are **inference** and **data association** problems – just different levels of **prior knowledge***

Robot pose

- $x = (x_r, y_r, \theta_r)$ w.r.t. a reference frame
- *Convention: First translate – then rotate*
- *Odometry provides a drifted pose…*
  … w.r.t. wherever the robot was turned on
- *Sensors can help eliminate drift by using a map*



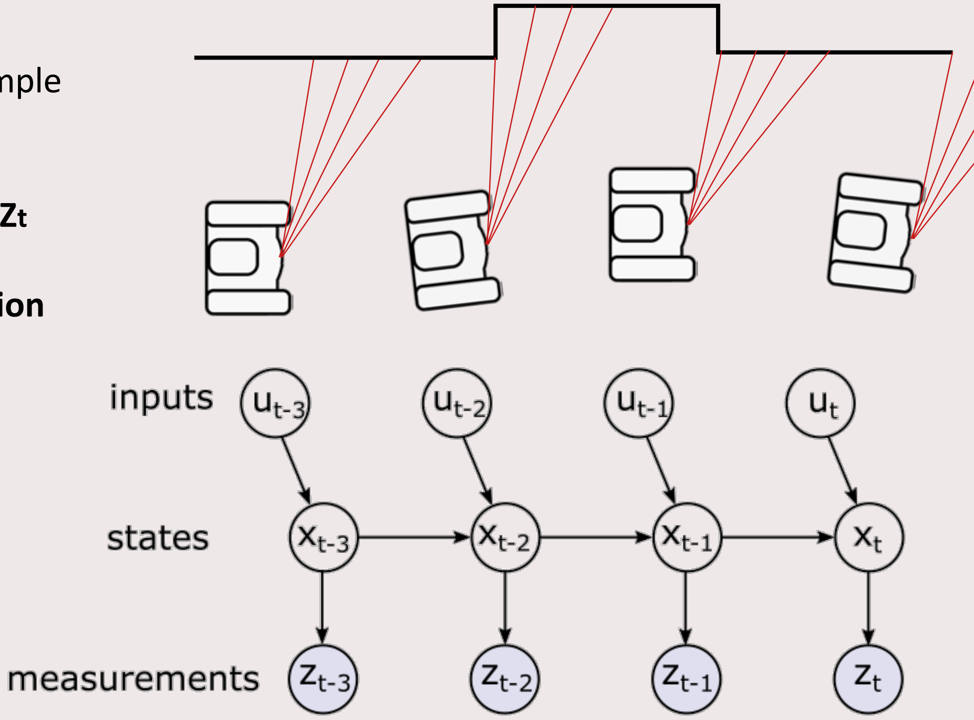
Localization as inference problem

- Estimate a ***state*** using sensor readings, for example **robot poses,** or **objects** *(observer problem!)*
- Often not possible to directly calculate $\mathbf{X_t}$ from $\mathbf{Z_t}$
- How does **noisy sensor data** and **prior information** affect our **state estimate?**

Probabilistic graphical model to capture (in)dependencies between **variables**

Arrows: “provides information about”, causality

Maintain a **probabilistic belief** over state $\mathbf{X_t}$ to systematically and consistently incorporate new information **How?**



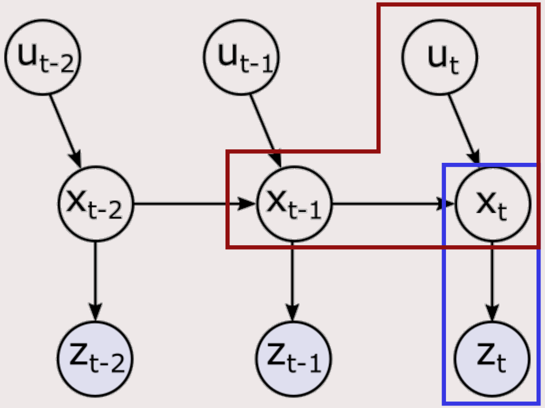
Recursive filtering

Markov assumption → $\mathbf{X_t}$ is *independent* of past variables, given $\mathbf{X_{t-1}}$ (*independent noise*)

Q: Do you think this is a valid assumption?

Maintain a current *belief* over the state $\mathbf{X_t}$,
given $\mathbf{U_t}$, $\mathbf{Z_t}$ and belief $\mathbf{X_{t-1}}$
The belief over $\mathbf{X_t}$ covers all we '*know*'

- Prediction step, **(motion) model**
- update step, **measurement model**



Many algorithms for localization are based on this approach
Q: If state is continuous, what triggers a belief update?

Measurement models

Model the *likelihood* of a measurement **z_t**, given a particular state

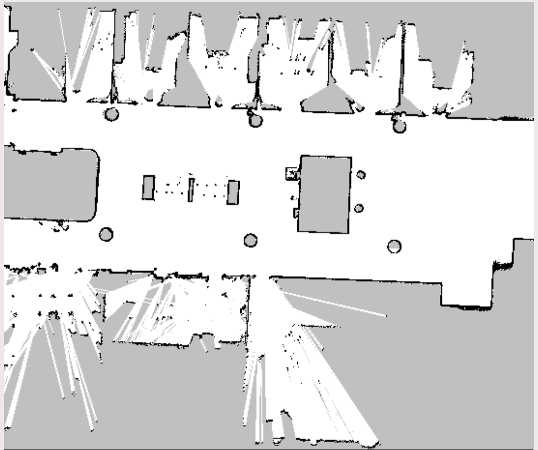
*What do we use for **z_t** ?*

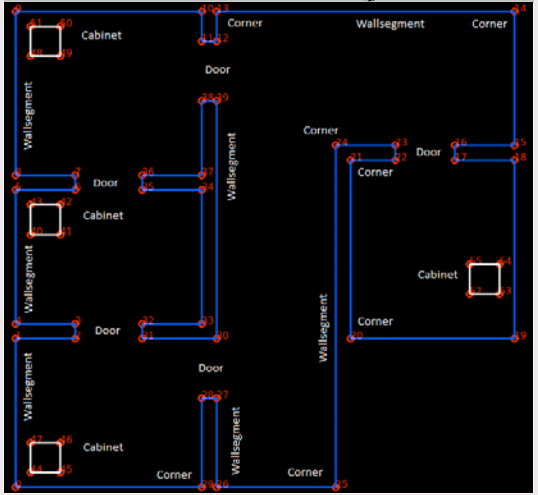
***Beam-based* approach**

- Considers each laser beam individually
- Assumes beams are *statistically independent* (they aren't!)
- Often used together with occupancy grid maps

***Feature-based* approach**

- High-level *features (e.g., lines, corners, circles)*
- Features can be composed into high-level features (e.g., doors)
- Require *feature extraction* from sensor data

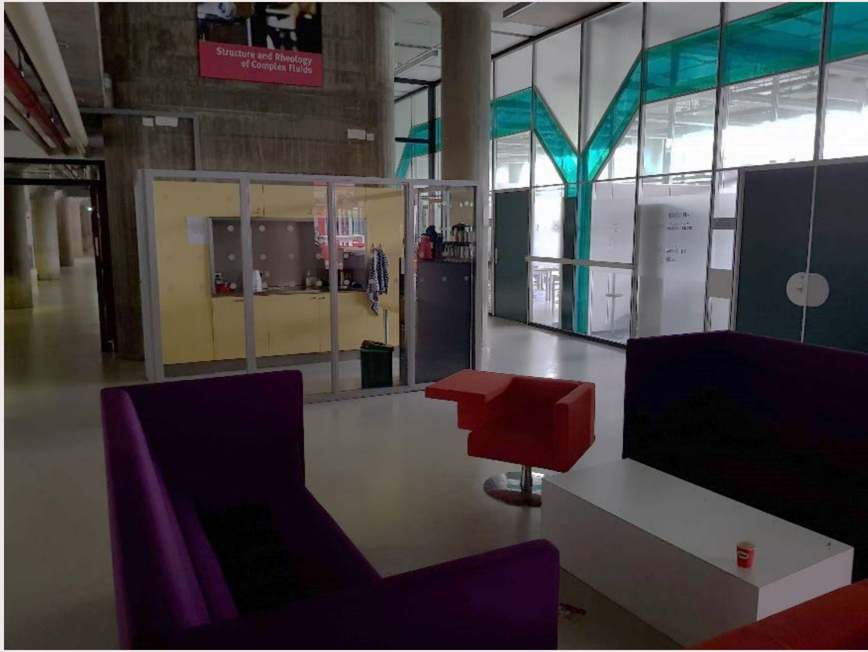




Example location

Driving from A to B in an indoor location

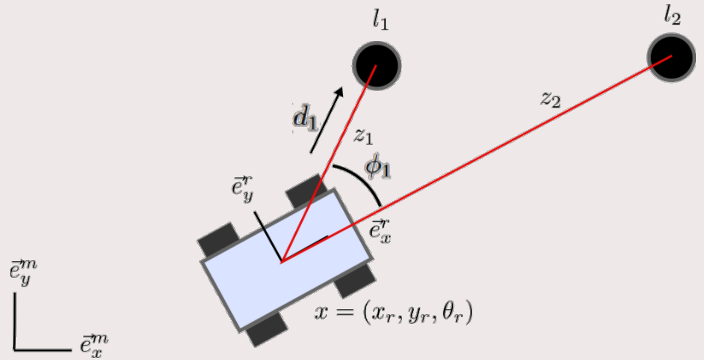
Q: What would you use as a sensor measurement z_t ?



Extended Kalman filters

Motion model

$$x_t = g(x_{t-1}, u_t) + w$$



- w is Gaussian zero-mean error with covariance R
- E.g. odometry uncertainty
- Update mean and covariance of belief:

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

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

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

Gaussians

$p(x) \sim N(\mu, \sigma^2)$:

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}}$$

Univariate

$p(\mathbf{x}) \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$:

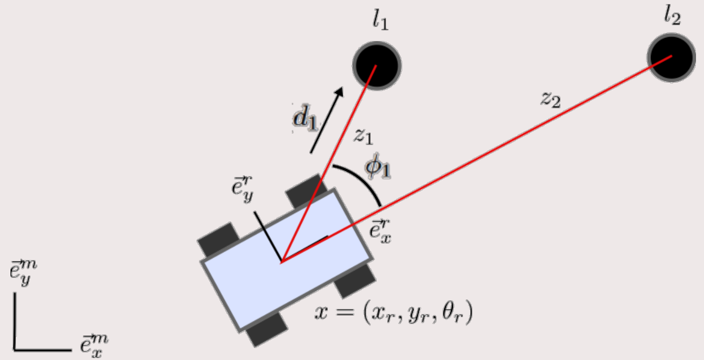
$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}$$

Multivariate

Extended Kalman filters

Measurement model

$$z_t = h(x_t) + v \qquad\qquad z_1 = (d_1, \phi_1)$$



- v is Gaussian zero-mean error with covariance Q
- Measurement uncertainty, linearization error
- Update mean and covariance of belief:

$$\begin{aligned} K_t &= \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1} \\ \mu_t &= \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t)) \\ \Sigma_t &= (I - K_t H_t)\bar{\Sigma}_t \end{aligned} \qquad H_t = \frac{\partial h(\bar{\mu}_t)}{\partial x_t}$$

- Q, R is used to *tune* tradeoff between motion and measurement!

Gaussians

$p(x) \sim N(\mu, \sigma^2)$:

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}}$$

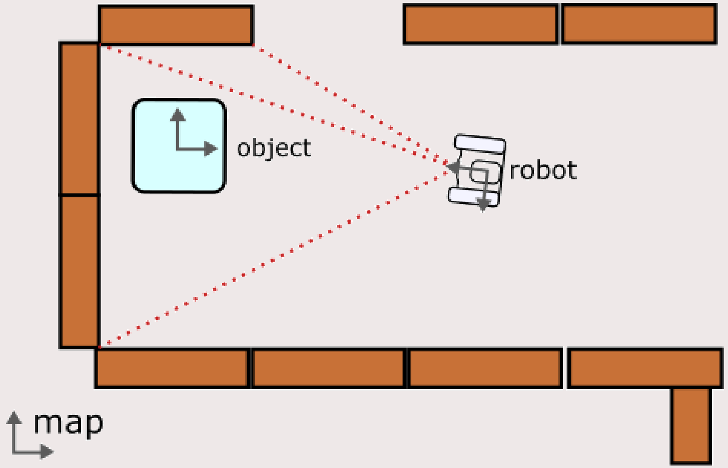
Univariate

$p(\mathbf{x}) \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$:

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^t \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}$$

Multivariate

Localization – *wr.t. map* and *w.r.t. robot*



- **Assuming perfect localization** can lead to **problems**
- Beware of the **cumulative effect** of **uncertainty**
- Sometimes **locating** an object w.r.t. the **robot-fixed frame** is enough!

The data association problem

Problem so far: we assumed **known data associations**

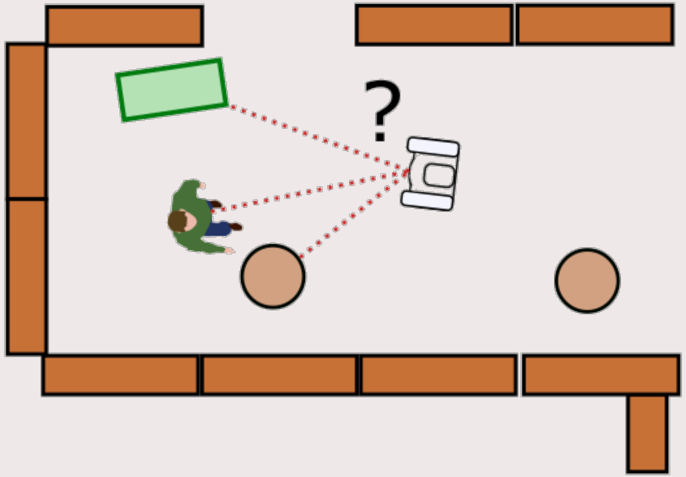
Often we can retrieve the correct data association:

- **nearest neighbor**
- **Uncertainty-based** (choose not to make one)

Making a *wrong association* can be a big problem!

Multiple data association **hypotheses** give rise to **multimodal** probabilities!

How can we deal with this?



Multi-modal beliefs: particle filters

Brute-force implementation of recursive filter

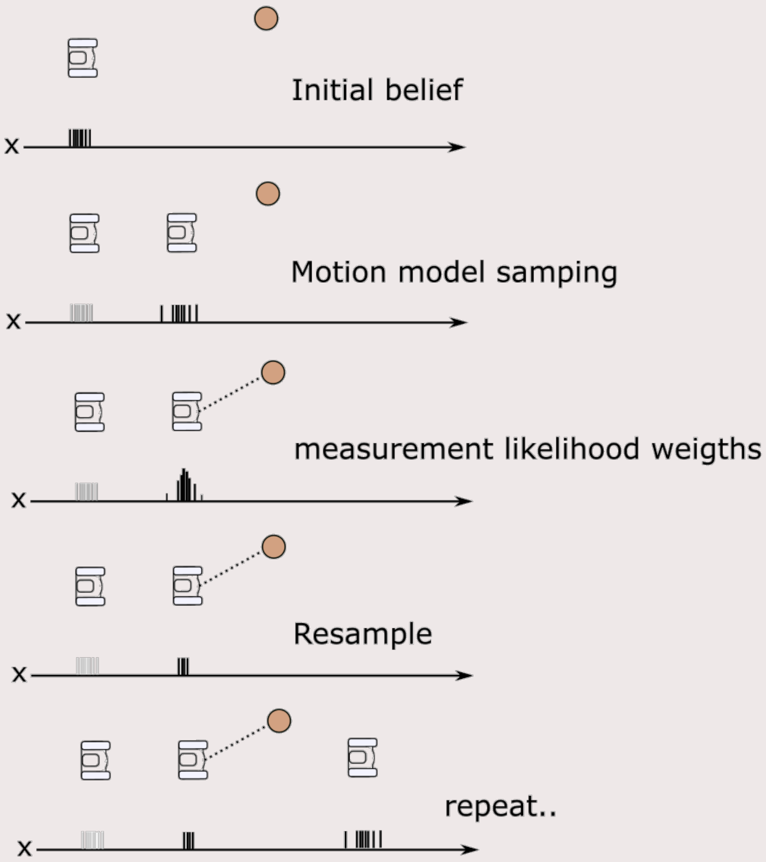
Represents the **belief** as **weighted particles** (often 500+)

Particles are discrete **hypotheses** about the state

Bayesian filter steps

- Particles get propagated according to **motion model**
- Particles get **likelihood weights** based on sensor information
- Requires a **stochastic resampling step** (tuning parameter)
- **Low weight** particles **removed**, **high weight** particles **cloned**

Successful in **low-dimensional** state spaces
Tuning: How many particles? How often resampling?



Take-away message

- Localization requires **data association** and **inference** to calculate the **robot pose** from **sensor data**

- Most approaches use **recursive estimation** (motion prediction-> measurement update)

How do we take robots from *filtering and planning algorithms…*
…to explainable, robust and versatile agents

- Make the **data associations** *as explicit* as possible

- *Does the robot understand that it is lost?*
- *Can the robot explain why it thinks it is lost?*

- We don't expect you to implement everything you've seen here
- Rather, think about how your robot can explain it's assumptions and actions

TU/e

References

Thrun, S., Burgard, W.,, Fox, D. (2005). *Probabilistic robotics*. Cambridge, Mass.: MIT Press.
ISBN: 0262201623 9780262201629