Università degli Studi di Padova

IAS-Lab
Intelligent Autonomous Systems Laboratory

A Very Brief Introduction to the Localization and SLAM Problems
Part 1

Alberto Pretto

Thanks to Wolfram Burgard, Giorgio Grisetti, Davide Scaramuzza and Cyrill Stachniss for some slides!

Outline

- Motivations and problems definition
- Probabilistic tools
- Main ingredients: Motions, observations and maps
- Localization: main paradigms
- SLAM: main paradigms
- Hints on Visual SLAM and current trends

Localization

The problem of estimating the **robot's position given a map** of the environment and a sequence of sensor readings.

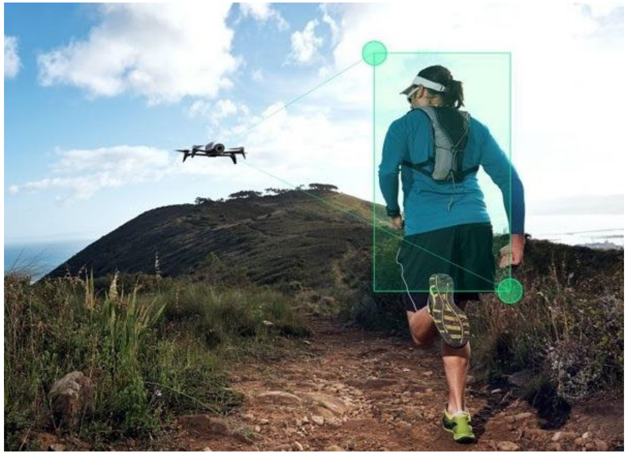
Problem classes:

- Position tracking
- Global localization
- Kidnapped robot problem (recovery)

Robot: a device that moves through the environment, and modify it

State: collection of all aspects of the robot and the environment that may have some impact on the behavior of the robot

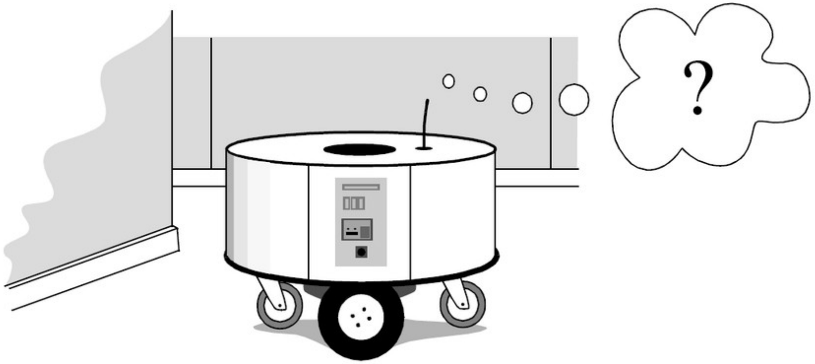
Robots and State



The SLAM Problem

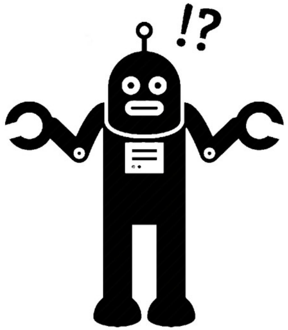
SLAM (acronym for Simultaneous Localization and Mapping) is the problem of computing the **robot's pose and the map of the environment at the same time.**

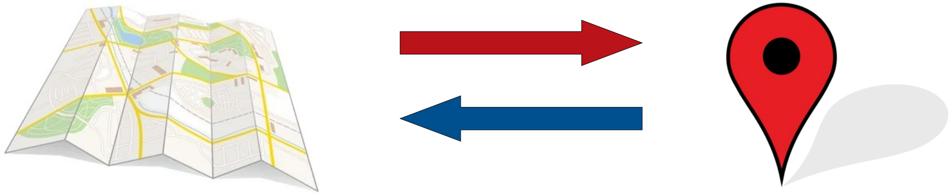
(**Mapping**: building a map given the robot´s location)



The SLAM Problem

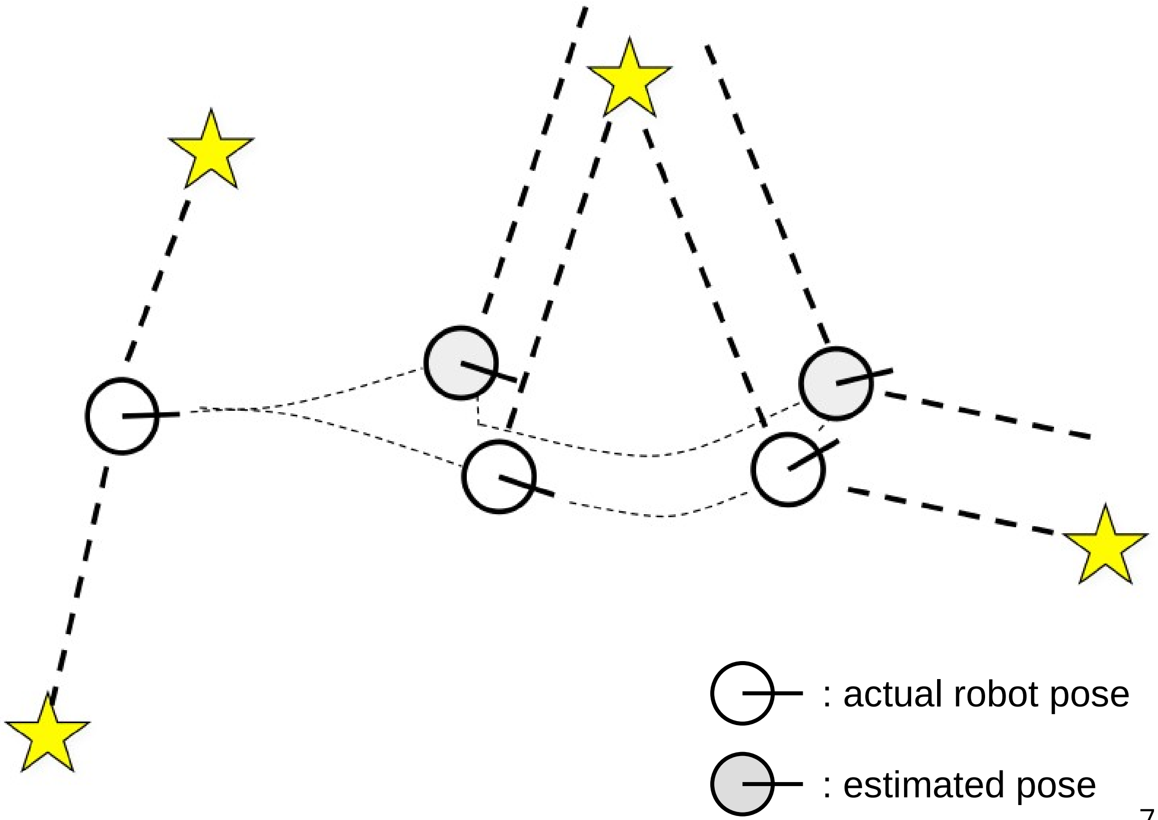
- **Localization**: estimating the robot's position given a map of the environment and a sequence of sensor readings.
- **Mapping**: building a map given the robot's locations
- **SLAM** is a chicken-or-egg problem!





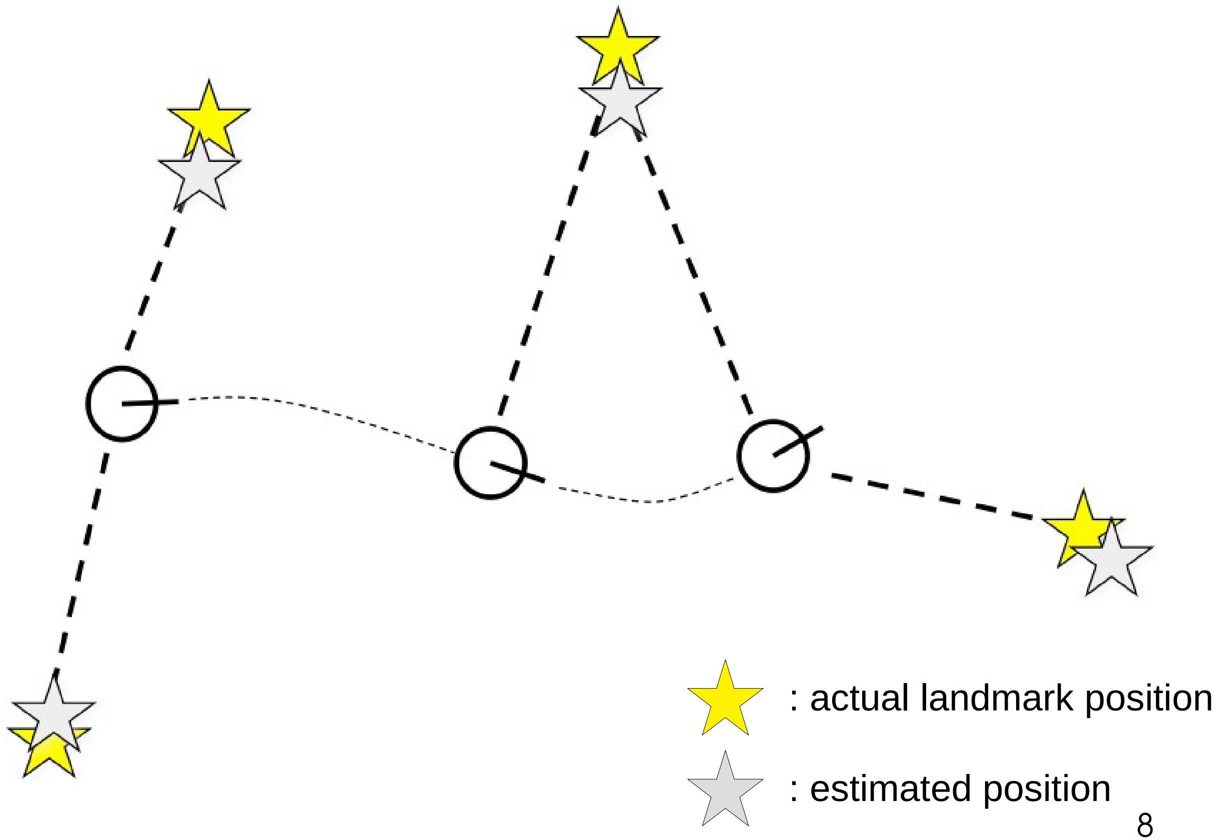
Localization Example

Estimate the robot's poses given landmarks



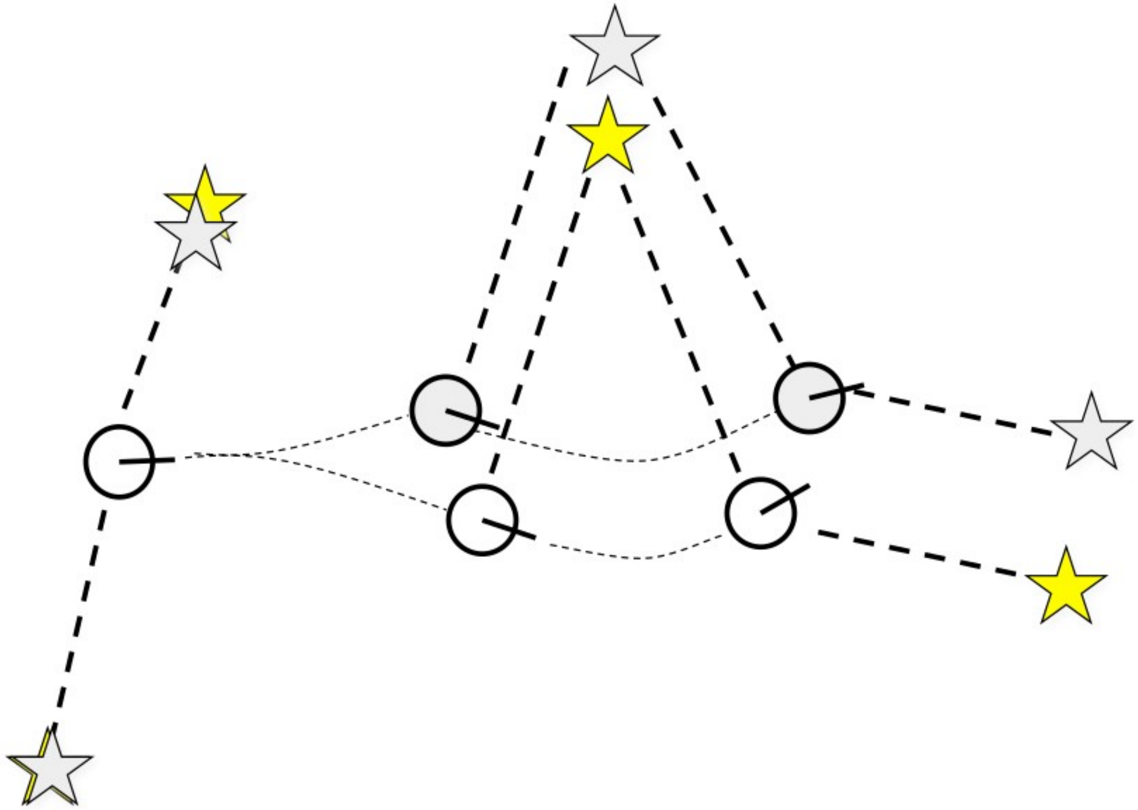
Mapping Example

Estimate the landmarks given the robot's poses

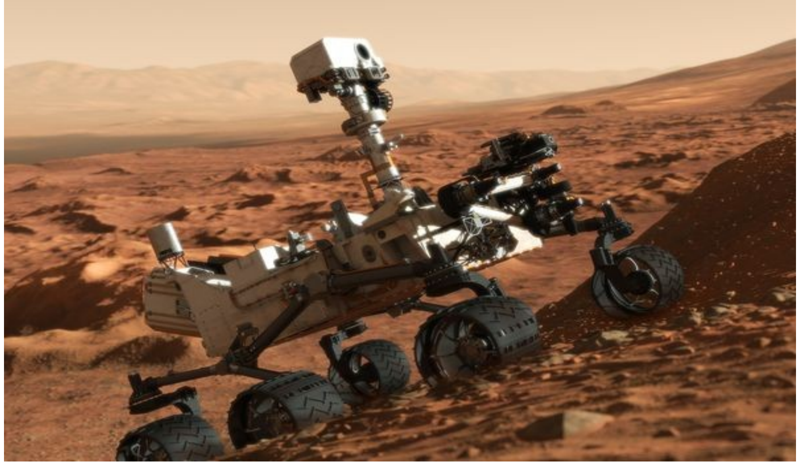


SLAM Example

Estimate the robot's poses **and** the landmarks at the same time



SLAM is Relevant



Definition of the Localization Problem

Given

- The robot controls $u_{1:T} = \{u_1, u_2, u_3 \ldots, u_T\}$
- Observations $z_{1:T} = \{z_1, z_2, z_3 \ldots, z_T\}$

Wanted

- Path (or current position) of the robot $x_{0:T} = \{x_0, x_1, x_2 \ldots, x_T\}$

Definition of the SLAM Problem

Given

- The robot controls $u_{1:T} = \{u_1, u_2, u_3 \ldots, u_T\}$
- Observations $z_{1:T} = \{z_1, z_2, z_3 \ldots, z_T\}$

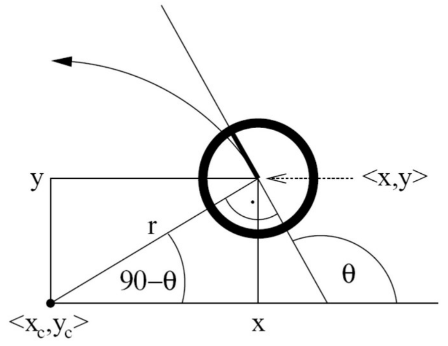
Wanted

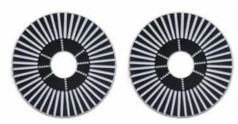
- Map of the environment m
- Path of the robot $x_{0:T} = \{x_0, x_1, x_2 \ldots, x_T\}$

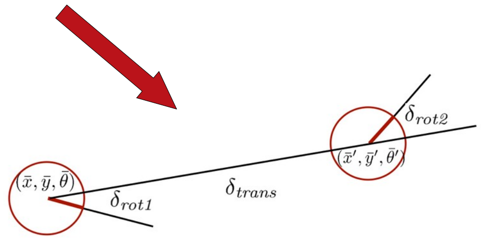
What are Robot Controls?

In localization and SLAM, motion controls are used:

- From controls sent to the **actuators** (e.g., wheel motors) estimate the angular and translational velocity
- Or, when wheel econders are available: use **odometry**, that is actually an output, as an input control





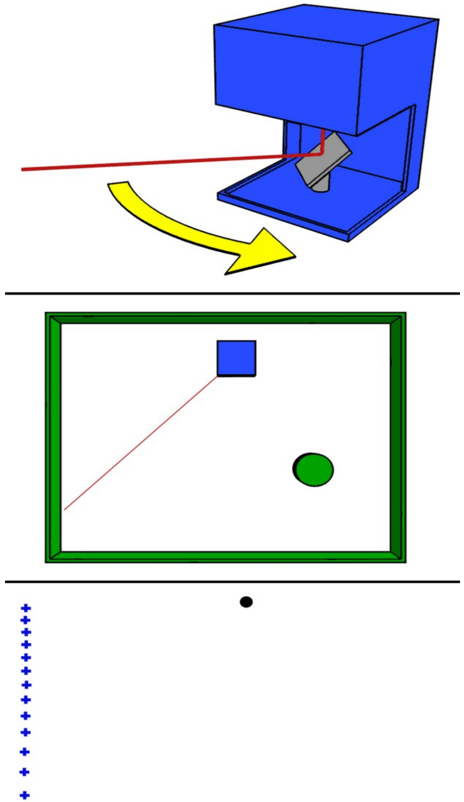


What are Observations?

Landmarks

What are Observations?

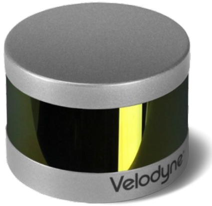
Range scans

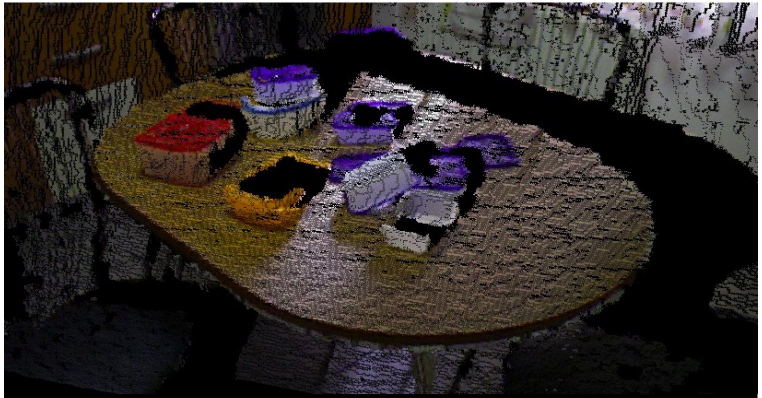


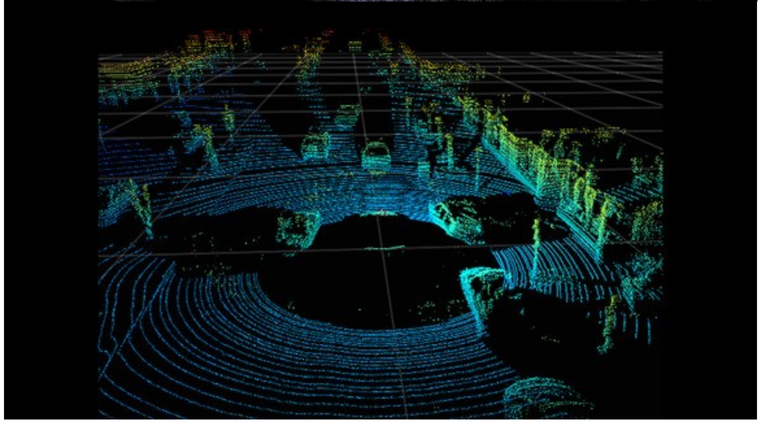
(Source: Wikipedia)

What are Observations?

3D scans/Point clouds

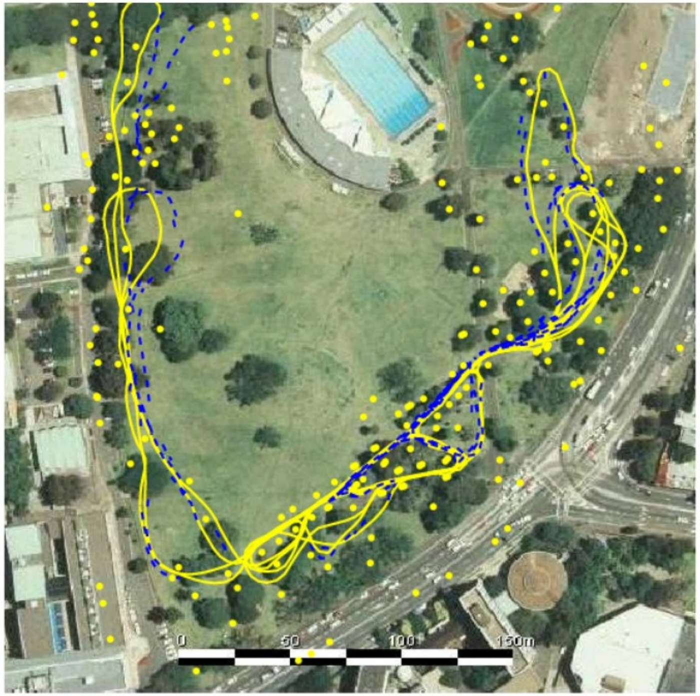




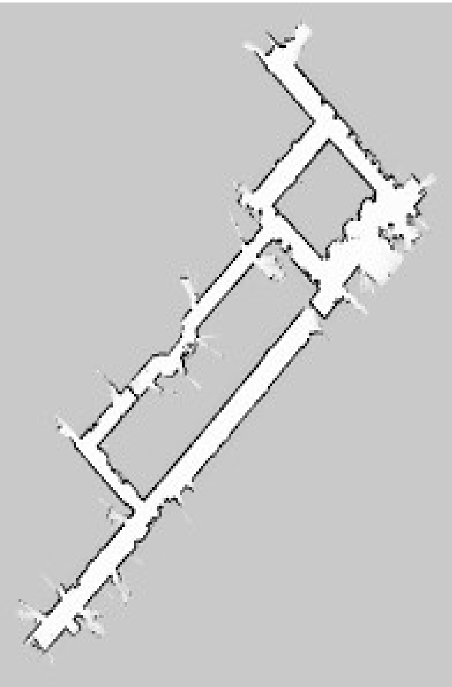


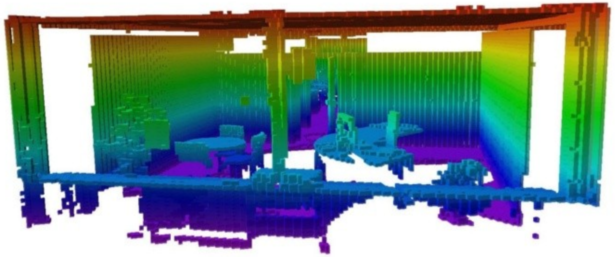
Maps

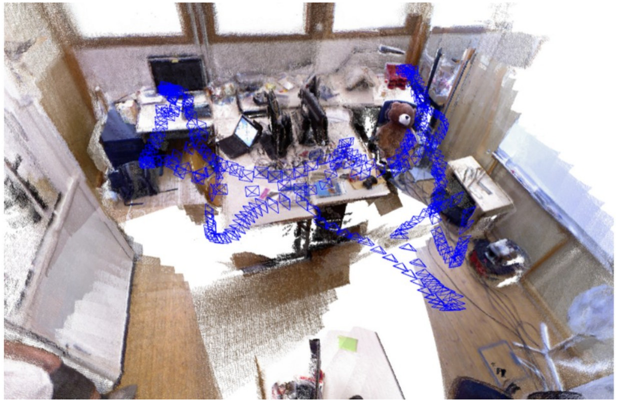
Landmark-based



Volumetric (e.g., grid-based)



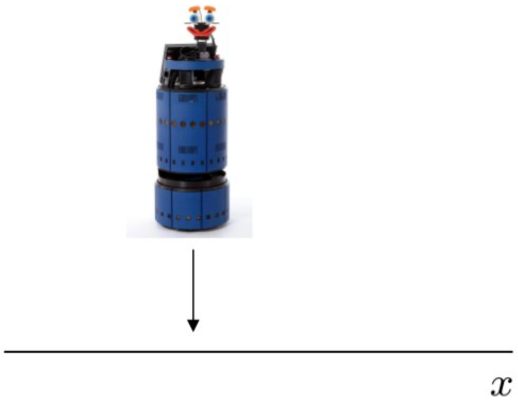




Probabilistic Approach

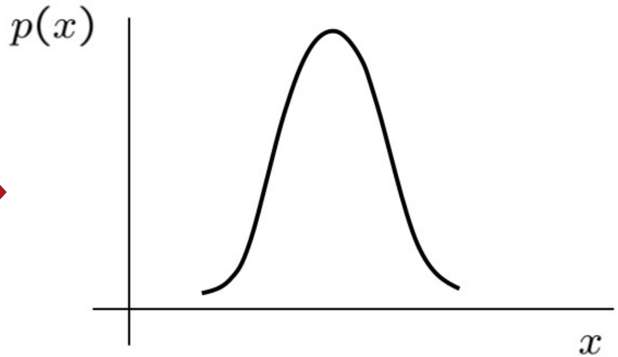
Uncertainty in the robot's motions and observations

Use the probability theory to **explicitly represent the uncertainty**



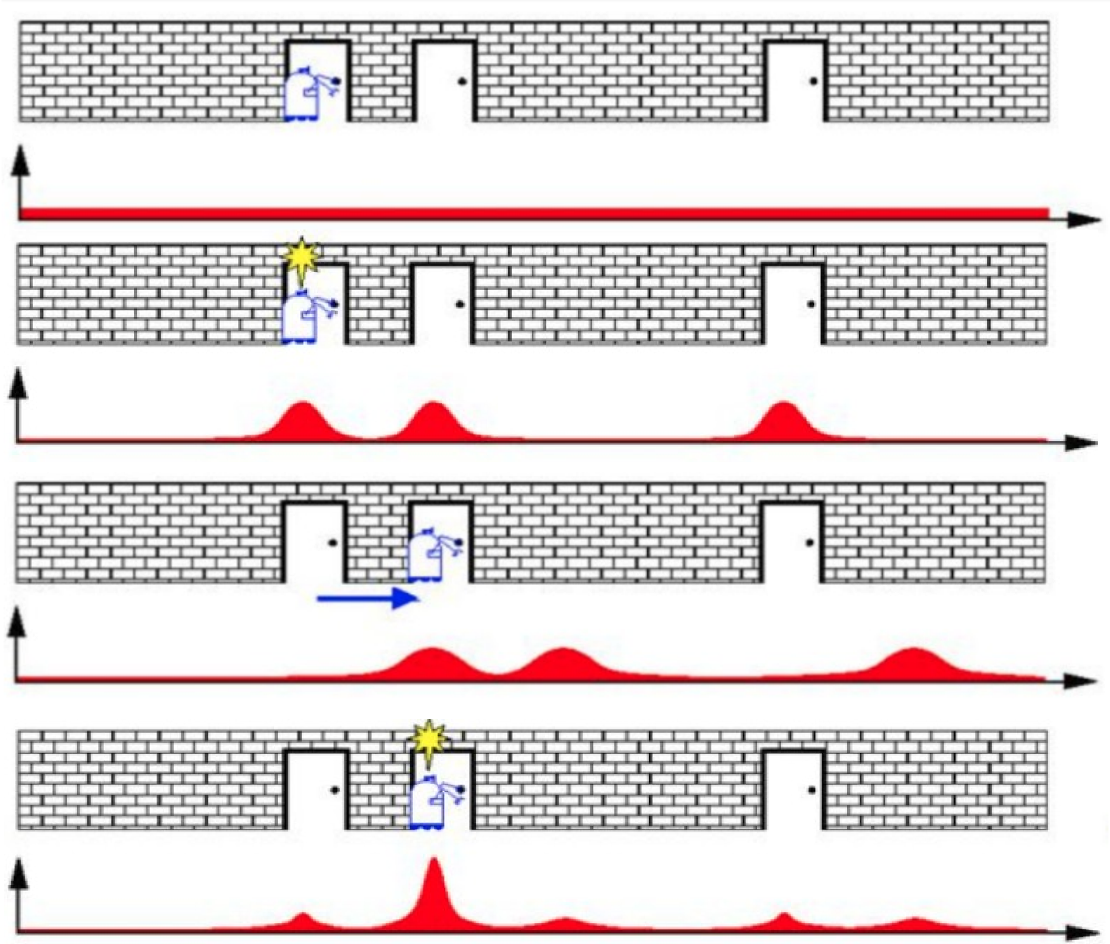
"The robot is exactly here"

Better..



"The robot is somewhere around here"

A simple localization example



Discrete Random Variables

A discrete random variable X can take on a countable number of values, e.g. $\{x_1, x_2, \ldots, x_n\}$.

$P(X=x_i)$, or $P(x_i)$, is the probability that the random variable X takes on value x_i, e.g. {0.1, 0.3, ..., 0.05 }.

P(.) is called probability mass function, with:

$$\sum_x P(x) = 1$$

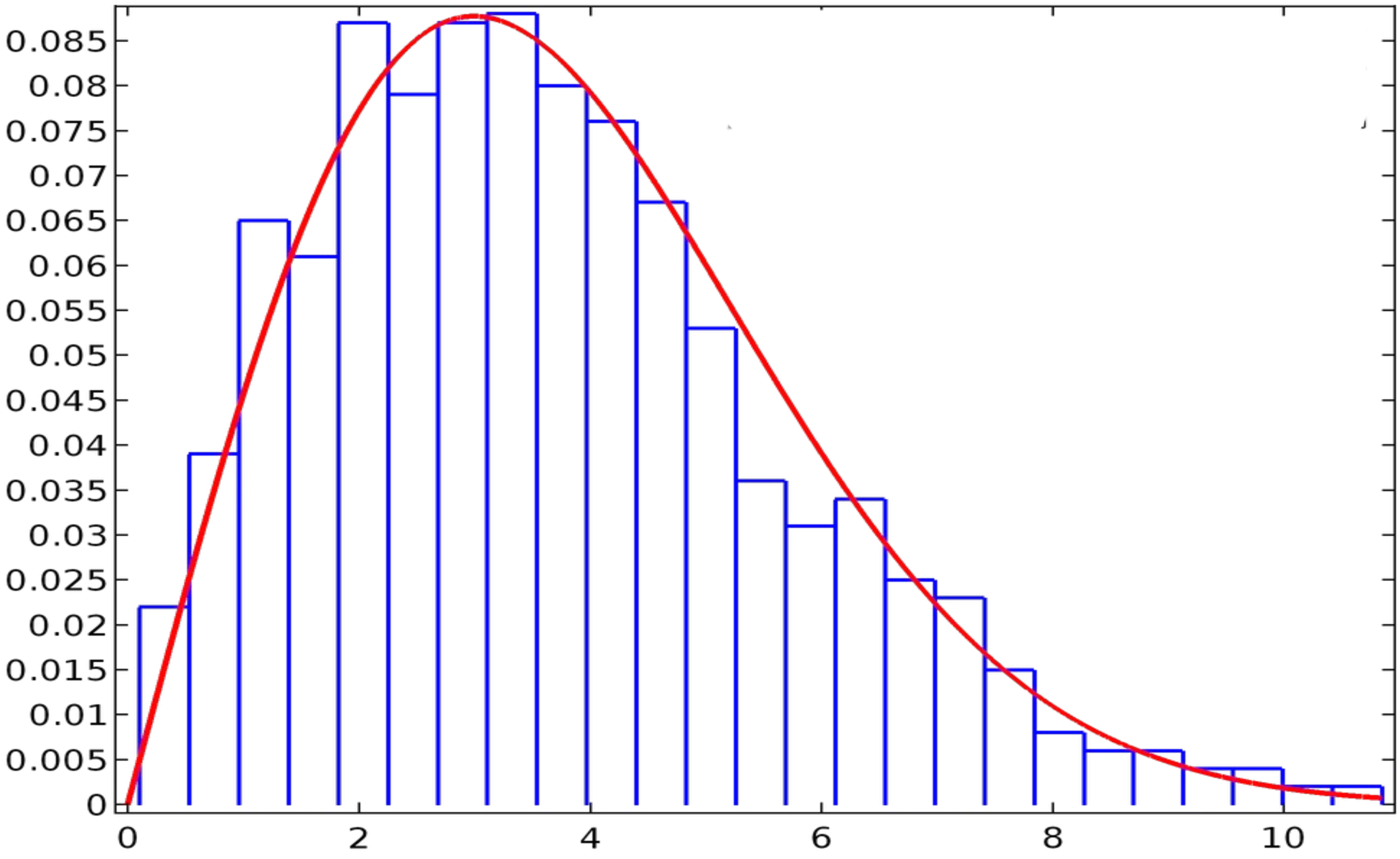
Continuous Random Variables

A continuous random variable *X* can take values in a continuous space.

p(X=x_i), or p(x_i), is a probability density function.

$$\Pr(x \in (a,b)) = \int_a^b p(x)dx \qquad \int p(x)\, dx = 1$$

Discrete vs Continuous RV



Joint and Conditional Probability

- $P(X=x$ **and** $Y=y) = P(x,y)$
- If X and Y are independent then
  - $P(x,y) = P(x)\,P(y)$
- $P(x \mid y)$ is the probability of x given y
  - $P(x \mid y) = P(x,y) / P(y)$
  - $P(x,y) = P(x \mid y)\,P(y)$ (**chain rule**)
- If X and Y are independent then
  - $P(x \mid y) = P(x)$

Marginalization

Discrete case

$$P(x) = \sum_y P(x, y)$$

$$P(x) = \sum_y P(x \mid y) P(y)$$

Continuous case

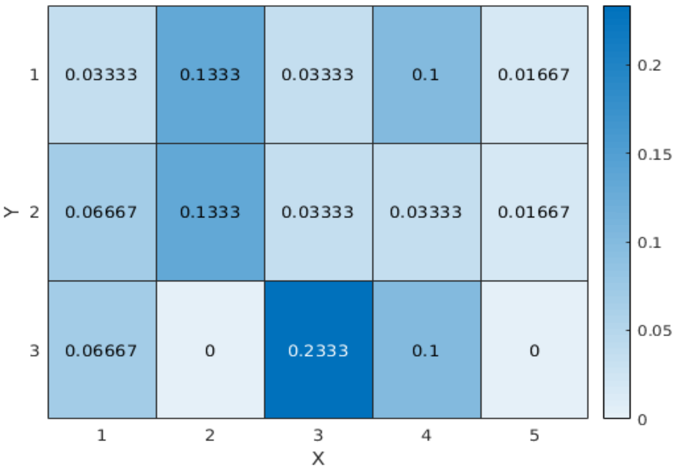
$$p(x) = \int p(x, y)\, dy$$

$$p(x) = \int p(x \mid y)\, p(y)\, dy$$

The second equations represent a variant of the marginalization rule, called **Law of Total Probability**.

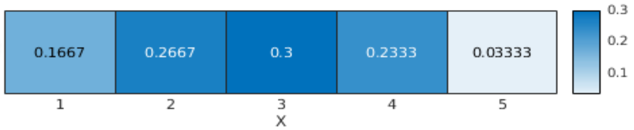
Marginalization

$P(x, y)$

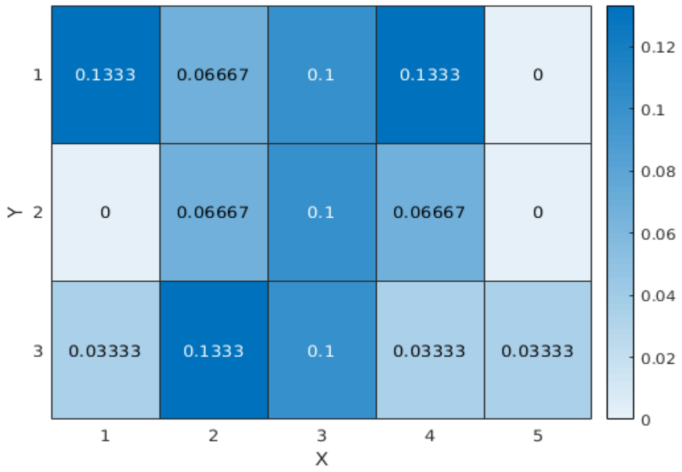


$$P(x) = \sum_{y} P(x, y)$$

$P(x)$

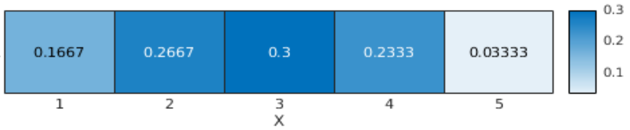


$P(x, y)$



$$P(x) = \sum_{y} P(x, y)$$

$P(x)$



Bayes Formula

Intuition: obtain an unknown target probability density in terms of other, **possibly known**, probability densities

$$P(x, y) = P(x \mid y) P(y) = P(y \mid x) P(x)$$

$$\Rightarrow$$

$$\boxed{P(x \mid y) = \frac{P(y \mid x)\; P(x)}{P(y)} = \frac{\text{likelihood} \cdot \text{prior}}{\text{evidence}}}$$

Normalization

Remove the evidence via normalization

$$P(x \mid y) = \frac{P(y \mid x)\ P(x)}{P(y)} = \eta\ P(y \mid x)\, P(x)$$

$$\eta = P(y)^{-1} = \frac{1}{\sum_x P(y \mid x) P(x)}$$

More conditions? No problem!

$$P(x \mid y \boxed{, z}) = \frac{P(y \mid x \boxed{, z})\, P(x \mid \boxed{z})}{P(y \mid \boxed{z})}$$

Recursive Bayesian Updating

Given a stream of **observations** $z = \{z_1, \ldots, z_t\}$, how can we estimate $P(x \mid z_1, \ldots, z_t)$?

Use the Bayes rule:

$$P(x \mid z_1, \ldots, z_n) = \frac{P(z_n \mid x, z_1, \ldots, z_{n-1})\, P(x \mid z_1, \ldots, z_{n-1})}{P(z_n \mid z_1, \ldots, z_{n-1})}$$

Markov Assumption

z_n is independent of $z_1, \ldots, z_{n-1}$ if we know x

$$P(x \mid z_1, \ldots, z_n) = \frac{P(z_n \mid x)\, P(x \mid z_1, \ldots, z_{n-1})}{P(z_n \mid z_1, \ldots, z_{n-1})}$$

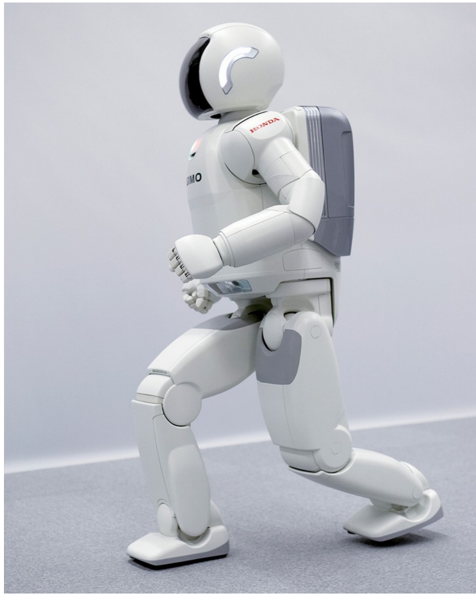
$$= \eta\, P(z_n \mid x)\, P(x \mid z_1, \ldots, z_{n-1})$$

Recursive update

$$= \eta_{1 \ldots n} \prod_{i=1 \ldots n} P(z_i \mid x)\, P(x)$$

Actions

- The robot turns its wheels to move, uses its manipulator to grasp an object, ...
- How can we incorporate such actions $u = \{u_1, \ldots, u_t\}$, i.e. **p(x|u)**?
  - Define a new probability density (also called **state transition**) p(x | u, x') (e.g., x' previous state).



Actions: use marginalization

- Continuous case

$$P(x \mid u) = \int P(x \mid u, x')P(x')dx'$$

- Discrete case

$$P(x \mid u) = \sum P(x \mid u, x')P(x')$$

Reminder: the Localization Problem

Given

- The robot controls $u_{1:T} = \{u_1, u_2, u_3 \dots, u_T\}$
- Observations $z_{1:T} = \{z_1, z_2, z_3 \dots, z_T\}$

Wanted

- Path (or current position) of the robot $x_{0:T} = \{x_0, x_1, x_2 \dots, x_T\}$

Probabilistic Localization Problem

Given a stream of observations *z* and action data u:

$$d_t = \{u_1, z_1 \ldots, u_t, z_t\}$$

Estimate of the robot position *X* as:

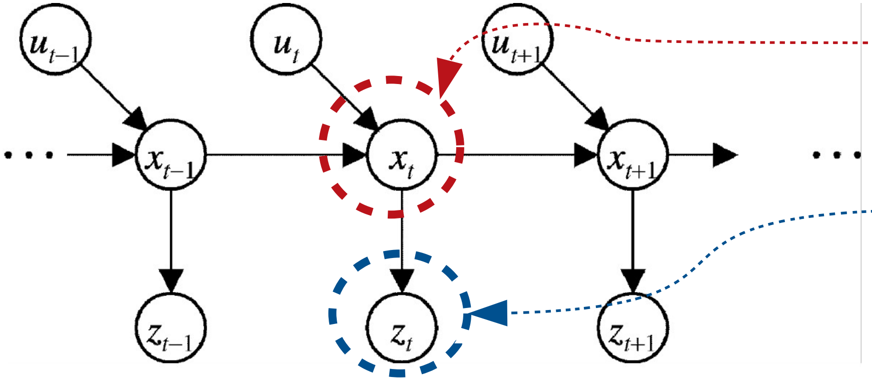
$$P(x_t \mid u_1, z_1 \ldots, u_t, z_t)$$

This *posterior* of the state is also called **Belief**

Markov Assumption (Cont)

A variable x_t depends only on its direct predecessor state x_{t-1} and on the latest action u_t

Bayesian network



$$p(x_t \mid x_{1:t-1}, z_{1:t}, u_{1:t}) = p(x_t \mid x_{t-1}, u_t)$$

$$p(z_t \mid x_{0:t}, z_{1:t}, u_{1:t}) = p(z_t \mid x_t)$$

Another very important assumption: **static world**

Bayes Filters with actions and observations

$$\boxed{Bel(x_t)} = P(x_t \mid u_1, z_1 \ldots, u_t, z_t)$$

Bayes $$= \eta\ P(z_t \mid x_t, u_1, z_1, \ldots, u_t)\ P(x_t \mid u_1, z_1, \ldots, u_t)$$

Markov $$= \eta\ P(z_t \mid x_t)\ P(x_t \mid u_1, z_1, \ldots, u_t)$$

Total prob. $$= \eta\ P(z_t \mid x_t) \int P(x_t \mid u_1, z_1, \ldots, u_t, x_{t-1})\ P(x_{t-1} \mid u_1, z_1, \ldots, u_t)\ dx_{t-1}$$

Markov $$= \eta\ P(z_t \mid x_t) \int P(x_t \mid u_t, x_{t-1})\ P(x_{t-1} \mid u_1, z_1, \ldots, u_t)\ dx_{t-1}$$

Markov $$= \eta P(z_t \mid x_t) \int P(x_t \mid u_t, x_{t-1})\ P(x_{t-1} \mid u_1, z_1, \ldots, z_{t-1})\ dx_{t-1}$$

$$\boxed{= \eta\ P(z_t \mid x_t) \int P(x_t \mid u_t, x_{t-1})\ Bel(x_{t-1})\ dx_{t-1}}$$

Bayes Filter Algorithm

1. Algorithm **Bayes_filter(** ***Bel(x), y*** **)** :
2. $\eta = 0$
3. If *y* is an **observation** *z* then
4. For all *x* do
5. $Bel'(x) = \boldsymbol{P(z \mid x)}\, Bel(x)$
6. $\eta = \eta + Bel'(x)$
7. For all *x* do
8. $Bel'(x) = \eta\, Bel'(x)$
9. Else if *d* is an **action** *u* then
10. For all *x* do
11. $Bel'(x) = \int \boldsymbol{P(x \mid u, x')}\, Bel(x')\, dx'$
12. Return *Bel′(x)*

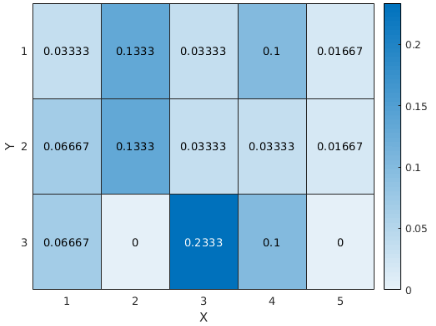
For all x’

Integrate observations

Normalize

Integrate actions

We are considering a discrete case



Action and Sensor Model

In the Bayes filter algorithm, we used two probabilites densities to update the Belief:

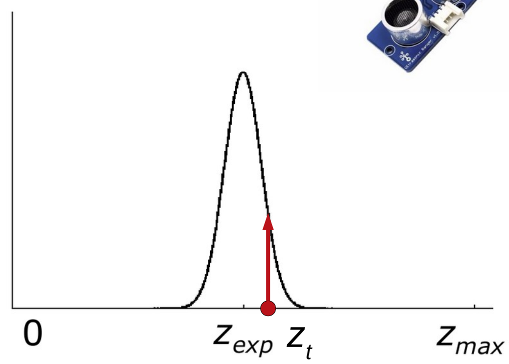
- **Observation**, or **Sensor Model** $P(z_t \mid x_t)$
- **Action Model** $P(x_t \mid u_t, x_{t-1})$
- When the action is the movement of the robot, the action model is called **Motion Model**

Observation Model

Model the uncertainty of the observations, i.e., the probability of a measurement z_t given that the robot is at position x_t.

$$p(z_t \mid x_t)$$

distribution observation given pose



Sensors for Mobile Robots

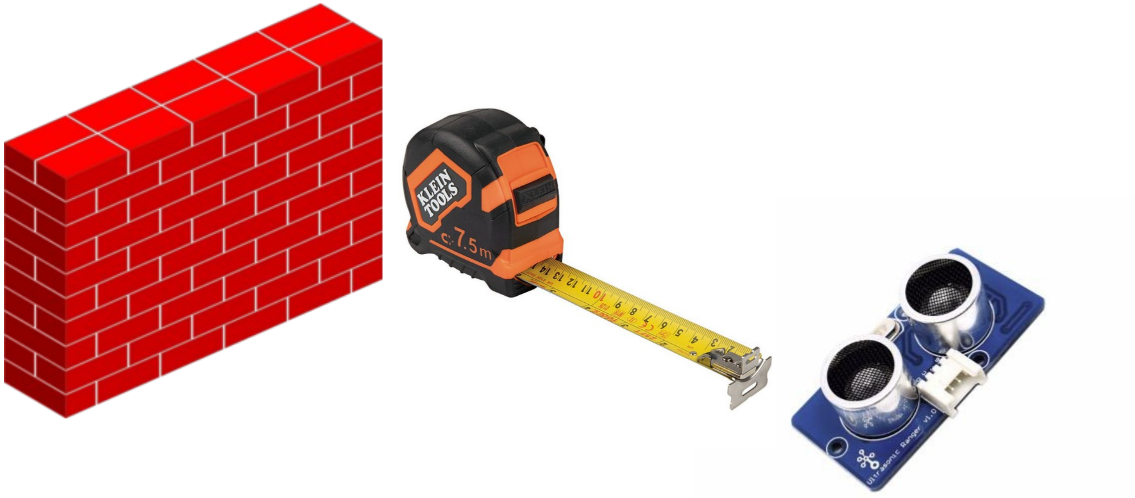
- Contact sensors
  - Bumpers
- Internal sensors
  - Accelerometers (spring-mounted masses)
  - Gyroscopes (spinning mass, laser light)
  - Compasses, inclinometers (earth magnetic field, gravity)
- Proximity (distacen) sensors
  - Sonar (time of flight)
  - Radar (phase and frequency)
  - Laser range-finders (triangulation, tof, phase)
- Visual sensors: Cameras
- Global reference sensors: GPS

And many others...

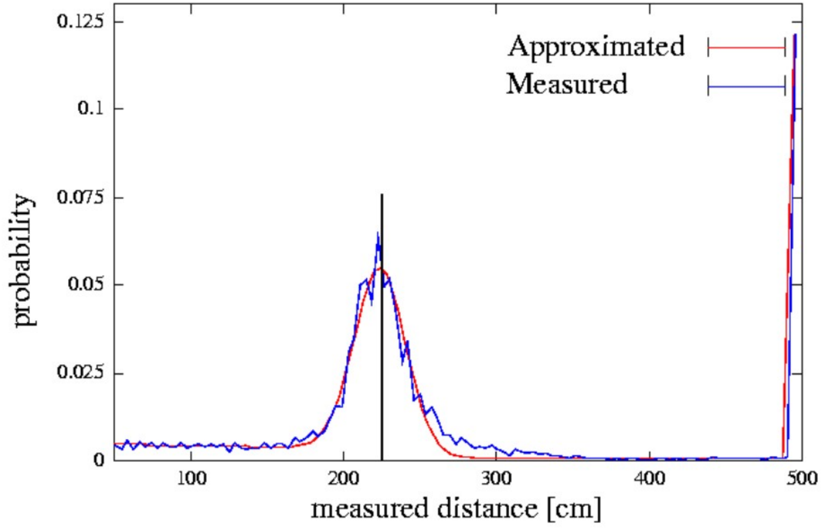
Example: Beam-based Proximity Model

How to estimate the sensor model density?

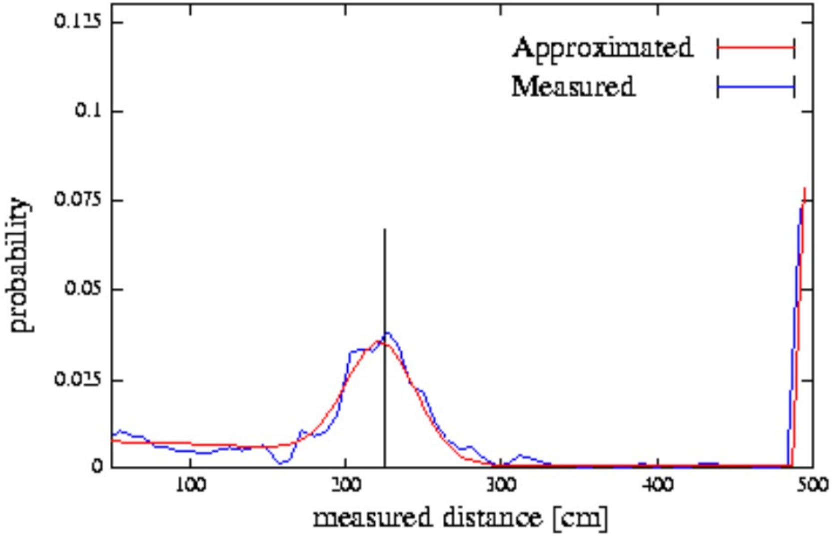
- Put the sensor at several **known** distances from some obstacles
- Collect sensor measurements



Example: Estimate the Model from Real Data

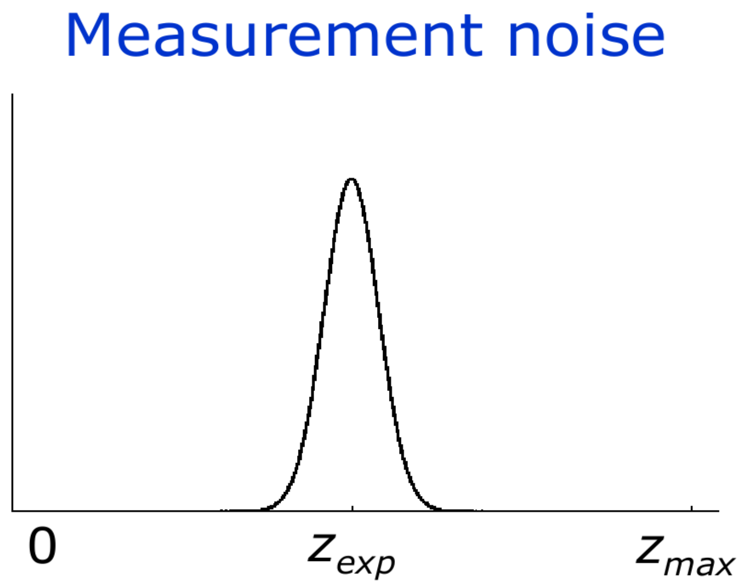


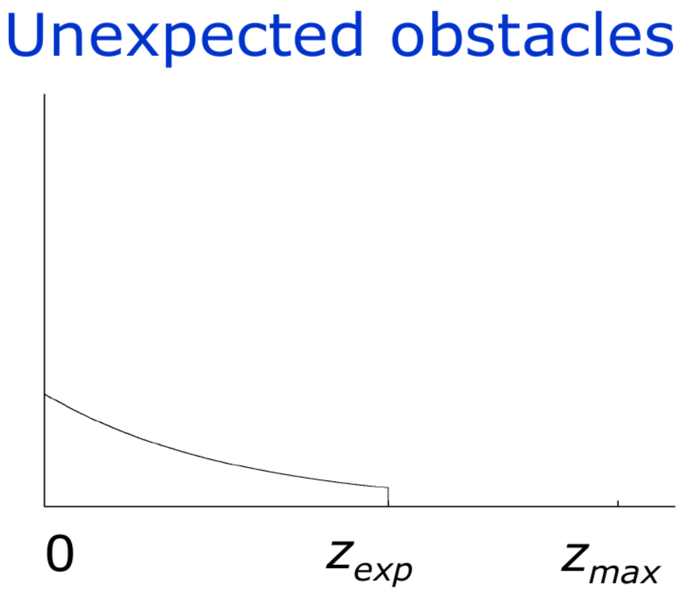
Laser sensor



Sonar sensor

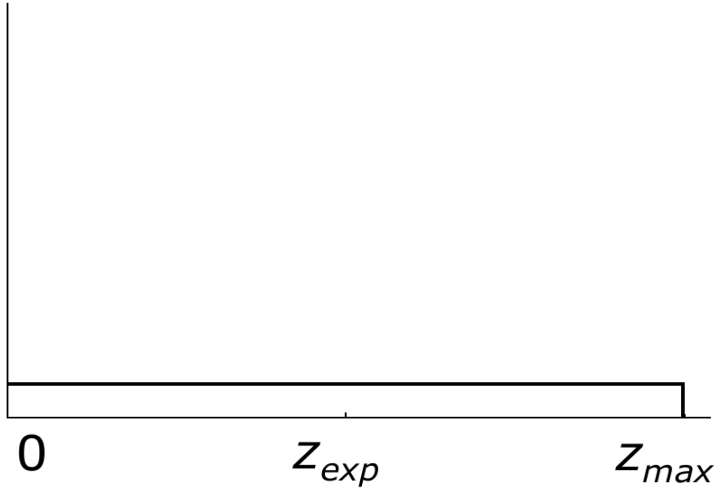
Example: Beam-based Proximity Model



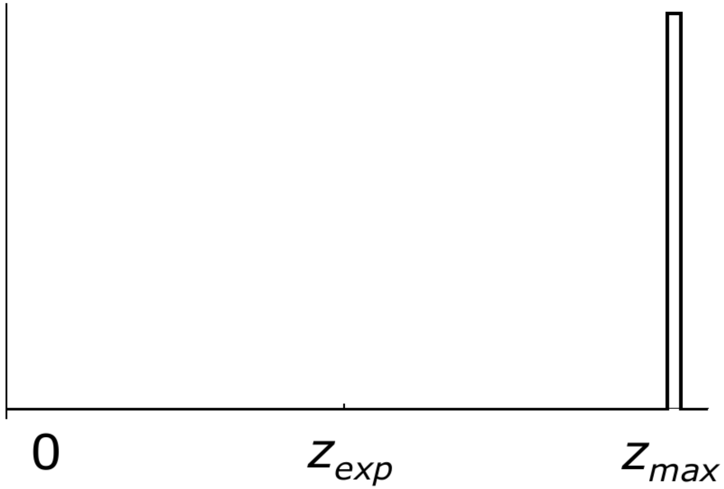


Example: Beam-based Proximity Model

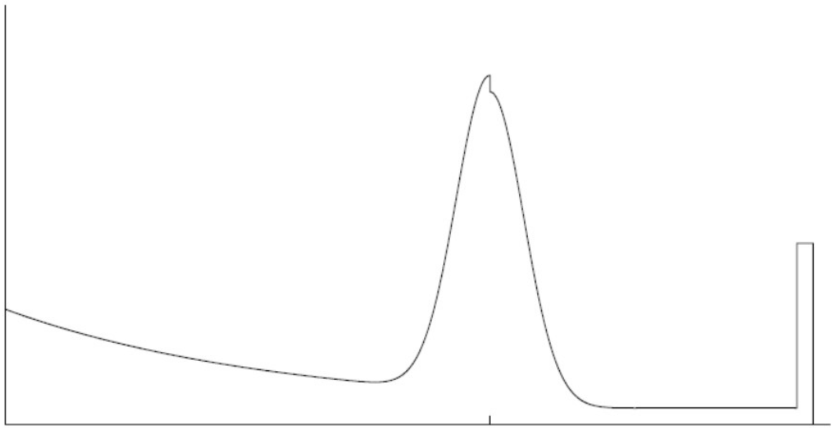
Random measurement



Max range

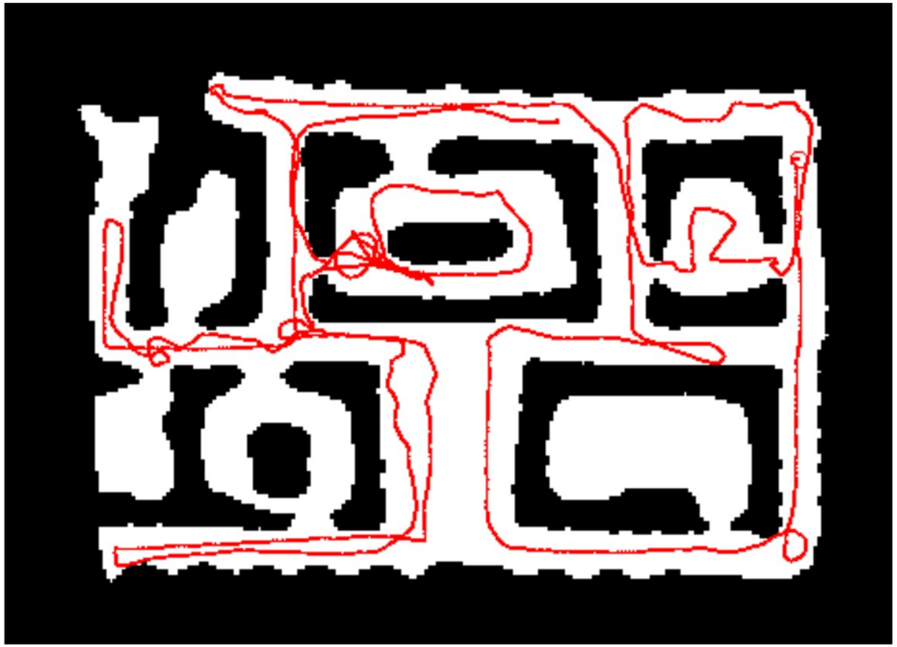


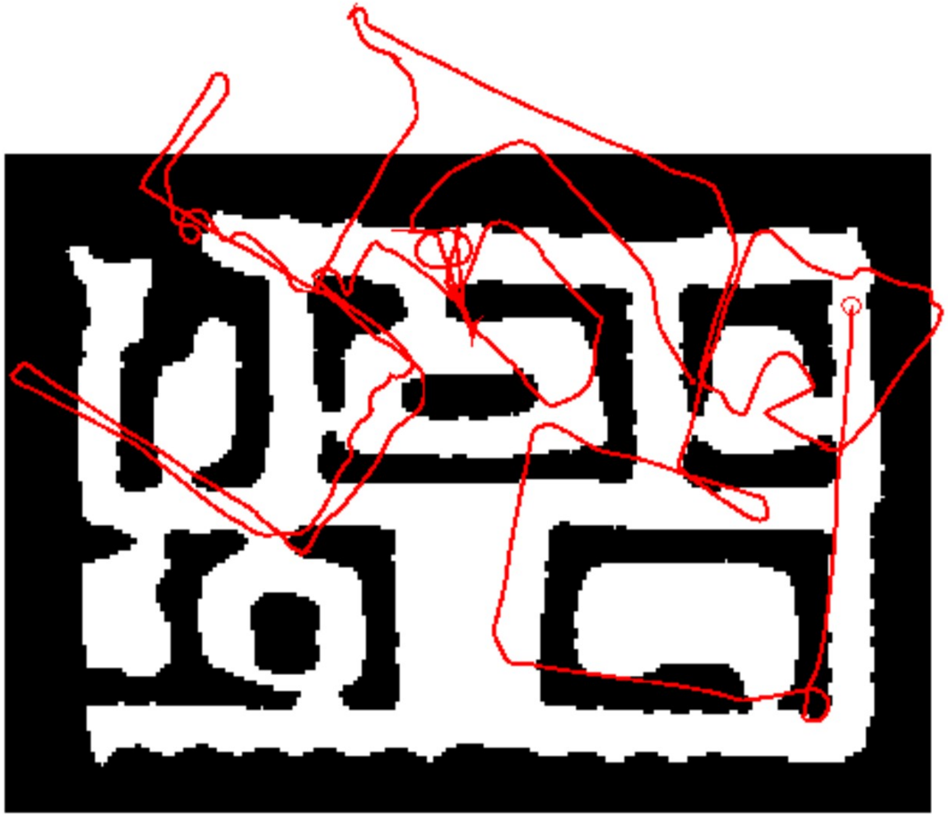
Resulting Mixture Density



$$P(z \mid x,m) = \begin{pmatrix} \alpha_{\text{hit}} \\ \alpha_{\text{unexp}} \\ \alpha_{\max} \\ \alpha_{\text{rand}} \end{pmatrix}^T \cdot \begin{pmatrix} P_{\text{hit}}(z \mid x,m) \\ P_{\text{unexp}}(z \mid x,m) \\ P_{\max}(z \mid x,m) \\ P_{\text{rand}}(z \mid x,m) \end{pmatrix}$$

Robot Motion





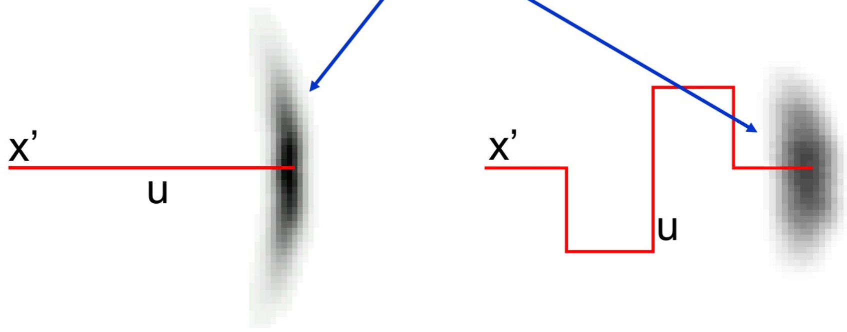
Motion Model

Model the uncertainty of the motion

$$p(x_t \mid x_{t-1}, u_t)$$

distribution new pose given old pose control

$$p(x|u,x')$$



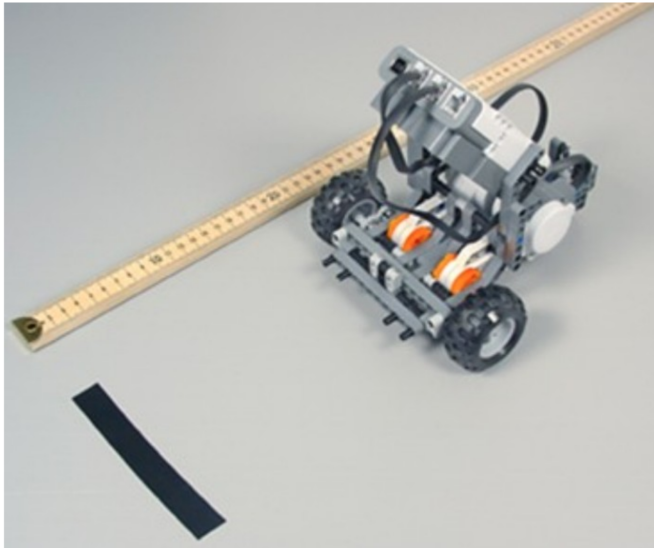
Motion Model

How to estimate the motion model density?

1) Move the robot from position A to position B, and collect the relative motion from the odometry.

2) Measure and collect the actual travelled distance with a meter, and repeat 2) for several trials.

3) Estimate the density parameters given the estimated and travelled distances



Summary

- Localization means estimating the robot´s pose, mapping is the task of modeling the environment
- SLAM does both the previous tasks simultaneously
- Solve such problems in a probabilistc way: not a single solution, but a probability "value" for each possible solution.
- Obtain the unknown target probability densities in terms of other, experimentally estimated, probability densities (e.g., motion and sensor model)