2 – Introduction to Probability and Mapping

Advanced Methods for Mapping and Self-localization in Robotics (MPC-MAP)
Course supervisor: Ing. Lukáš Kopečný, Ph.D.

Ing. Adam Ligocki, Ph.D.

Brno University of Technology
2022

Robotics and AI

Profile

Robotics and AI

Ing. Adam Ligocki, Ph.D.

Position: Research Staff

Research: Data Fusion

Room: SE1.102

Web: https://www.vut.cz/lide/adam-ligocki-154791

Background:
- Artificial Intelligence
- Neural Networks
- Software Development

Normal Distribution

Mean (cz: střední hodnota) symbol: μ	Standard Deviation (cz: Směrodatná odchylka) symbol: σ	Variance (cz: Rozptyl) symbol: $σ^2$

$$\mu = \frac{\sum_{i=1}^{N} x_i}{N} \qquad \sigma = \frac{\sum_{i=1}^{N}(x_i - \mu)^2}{N}$$
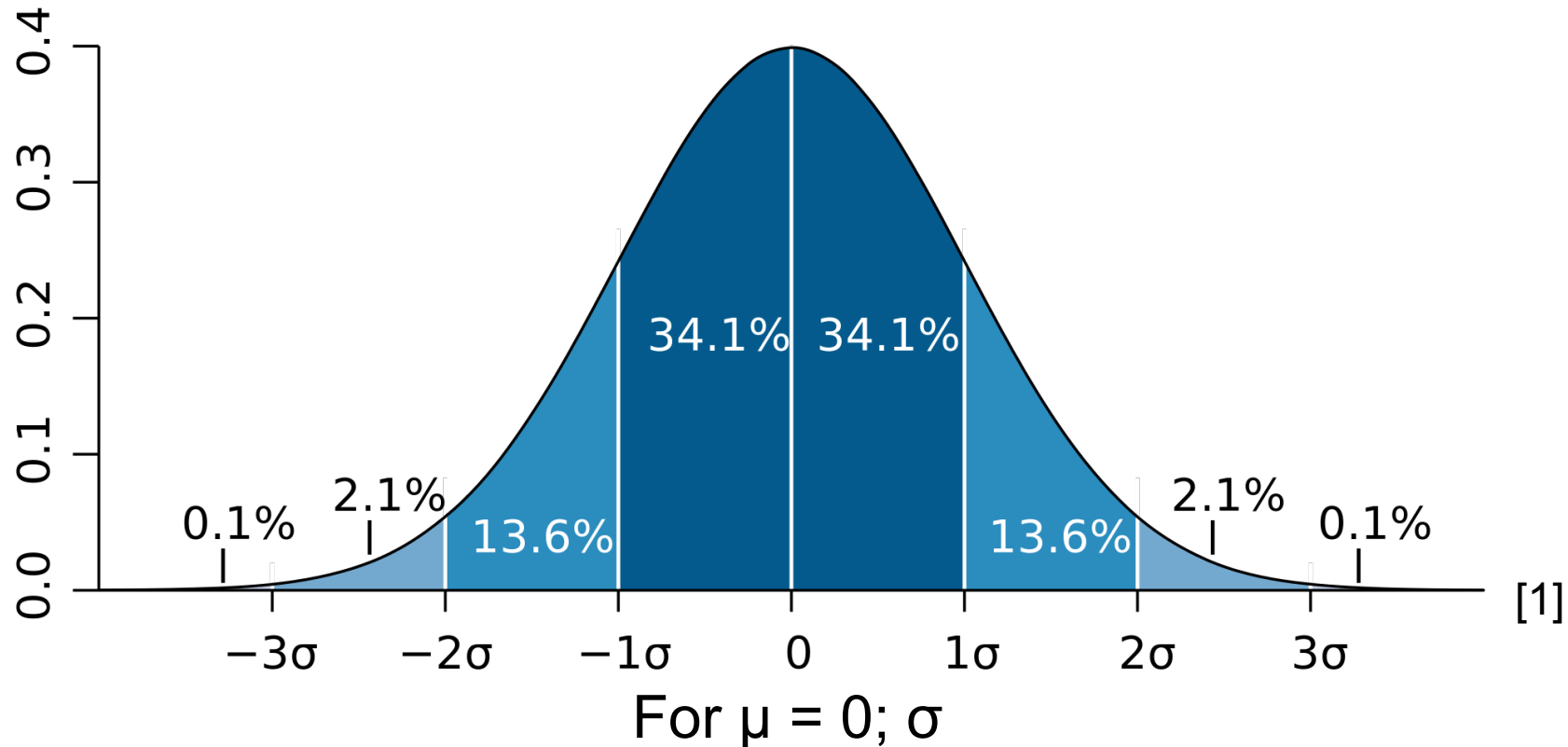
Normal Distribution – Basic Definitions

Mean
(cz: střední hodnota)
symbol: μ

Standard Diviation
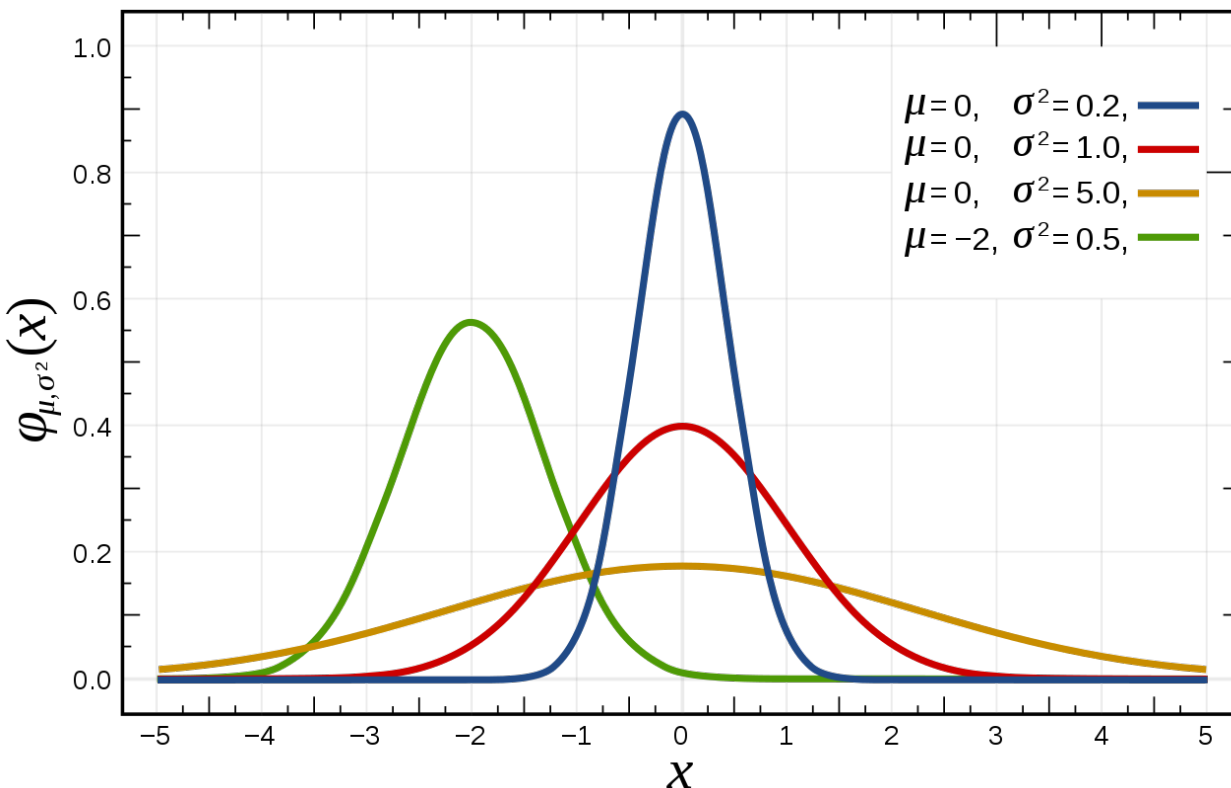(cz: Směrodatná odchylka)
symbol: σ

Variance
(cz: Rozptyl)
symbol: σ^2

34.1% 34.1%

0.1%   2.1%   13.6%   13.6%   2.1%   0.1%

-3σ -2σ -1σ 0 1σ 2σ 3σ
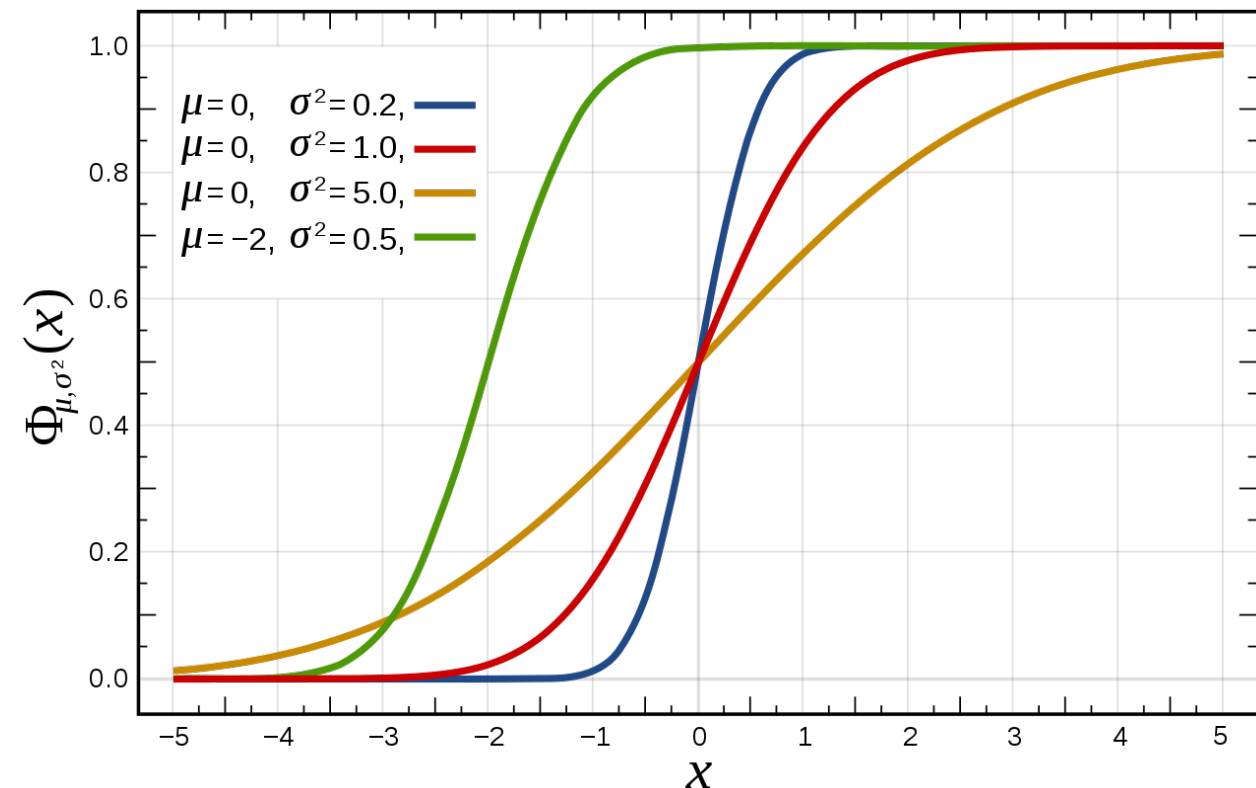
For $\mu = 0$; σ

[1]

Probability Density Function (PDF)

$$f(x, \mu, \sigma^2) = \frac{1}{\sigma * \sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Cumulative Distribution Function (CDF)

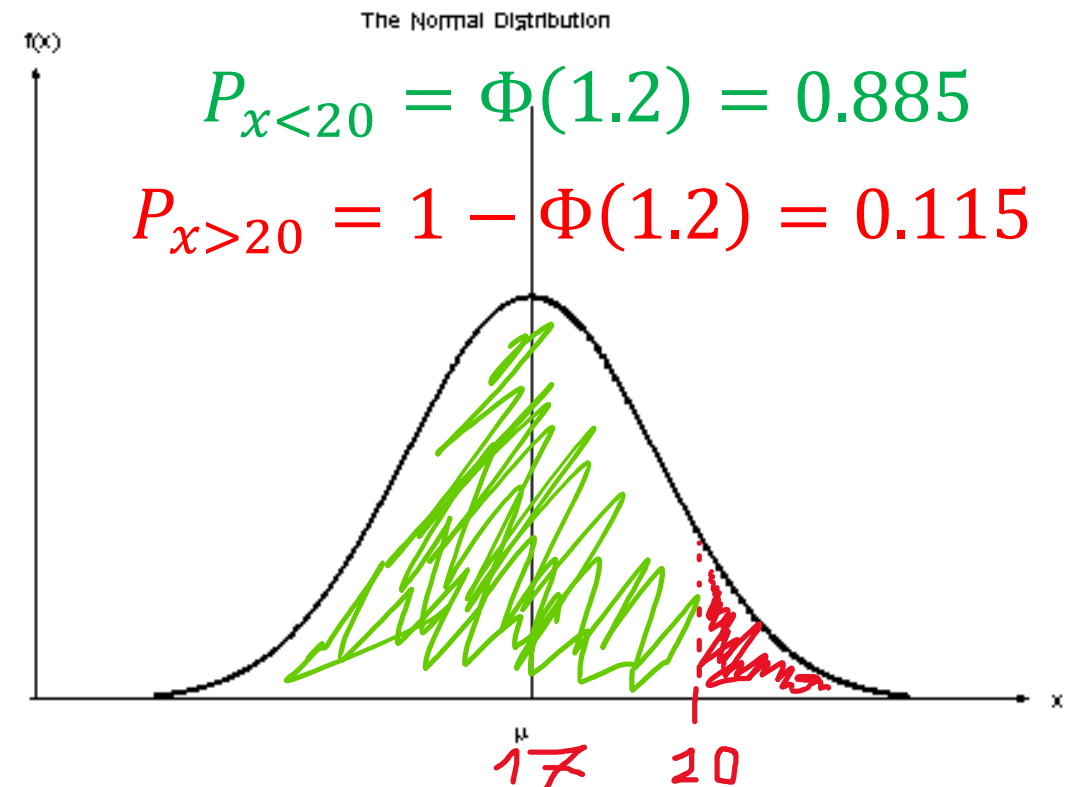$$\Phi(x, \mu, \sigma^2) = \int_{-\infty}^{x} f(t, \mu, \sigma^2) \, dt$$

[1] https://en.wikipedia.org/wiki/Normal_distribution

Normal dist. integration: https://www.youtube.com/watch?v=Bjh5Yvml4RM

Two fast foods (ff1 and ff2) share their extra burned oil resources. The ff1 consumes 10l of oil per day with std. dev. 2l. The ff2 uses 7l of oil per day with std. dev. 1.5l.

What is the probability that they run out of oil if they have 20l on stock for today?

$$\mu_1 = 10$$
$$\sigma_1 = 2$$

$$\mu_c = \mu_1 + \mu_2 = 17$$
$$\sigma_c = \sqrt{\sigma_1{}^2 + \sigma_2{}^2} = 2.5$$

$$\mu_2 = 7$$
$$\sigma_2 = 1.5$$

$$z = \frac{x - \mu_c}{\sigma_c} = 1.2$$

The Normal Distribution

$$P_{x<20} = \Phi(1.2) = 0.885$$
$$P_{x>20} = 1 - \Phi(1.2) = 0.115$$

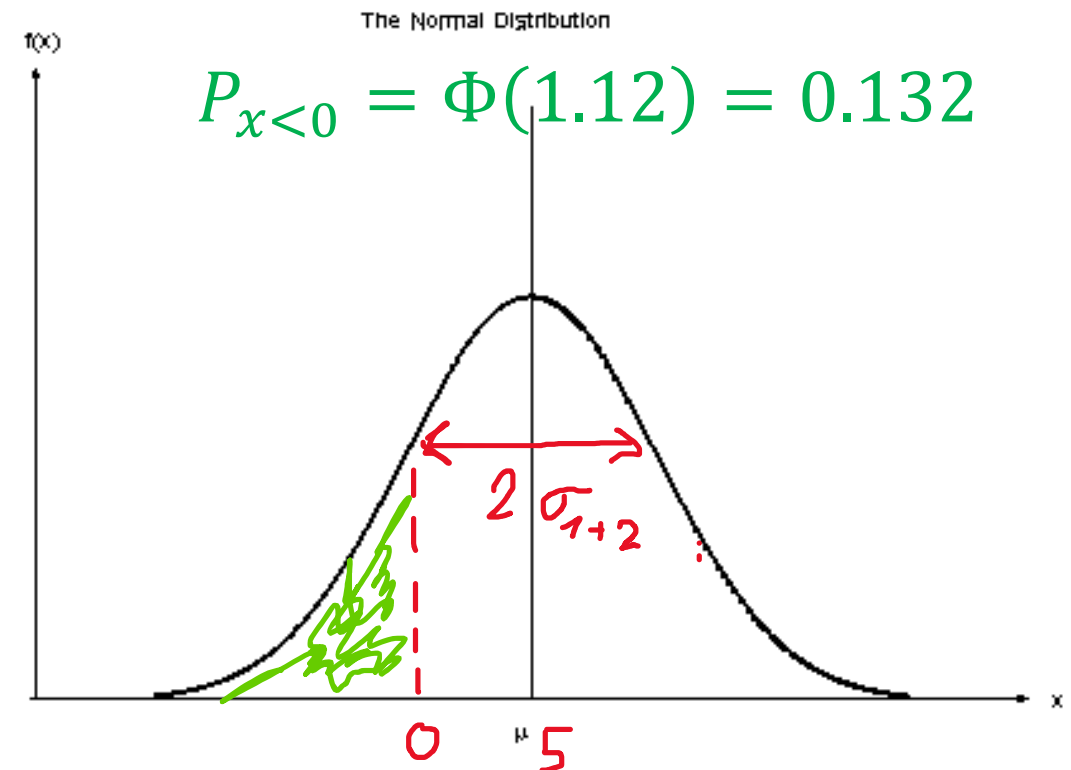The shipping from the USA usually takes 7 days with 2 days std. dev.
The sipping from China usually takes 12 days with 4 days std. dev.
What is the probability that the package from China will be delivered before the one from the USA if both were sent at the same time?

$$\mu_1 = 7 \qquad \mu_c = \mu_2 - \mu_1 = 5$$

$$\sigma_1 = 2 \qquad \sigma_c = \sqrt{\sigma_1{}^2 + \sigma_2{}^2} = 4.47$$

$$\mu_2 = 12$$

$$\sigma_2 = 4 \qquad z = \frac{x - \mu_c}{\sigma_c} = -1.12$$

The Normal Distribution

f(x)

$$P_{x<0} = \Phi(1.12) = 0.132$$

$2\,\sigma_{1+2}$

$0 \qquad \mu \, 5$

x

The sensors are measuring the distance to the obstacle. For both, the noise of the output data has the normal distribution character.

The first sensor measured the distance of 5m with a variance of 0.5m

The second sensor measured the distance of 5.5m with a variance of 0.7m.

Make the best estimation of the real distance to the obstacle.

$$\mu_1 = 5$$
$$\sigma_1^2 = 0.5$$

$$\mu_c = \frac{\mu_1 \sigma_2^2 + \mu_2 \sigma_1^2}{\sigma_1^2 + \sigma_2^2} = 5.208$$

$$\mu_2 = 5.5$$
$$\sigma_2^2 = 0.7$$

$$\sigma_c = \sqrt{\frac{\sigma_1^2 * \sigma_2^2}{\sigma_1^2 + \sigma_2^2}} = 0.540$$

Very nice example: https://datascopeanalytics.com/blog/sensor-fusion-tutorial/

The sensors are measuring the distance to the obstacle. For both, the noise of the output data has the normal distribution character.

The first sensor measured the distance of 5m with a variance of 0.5m

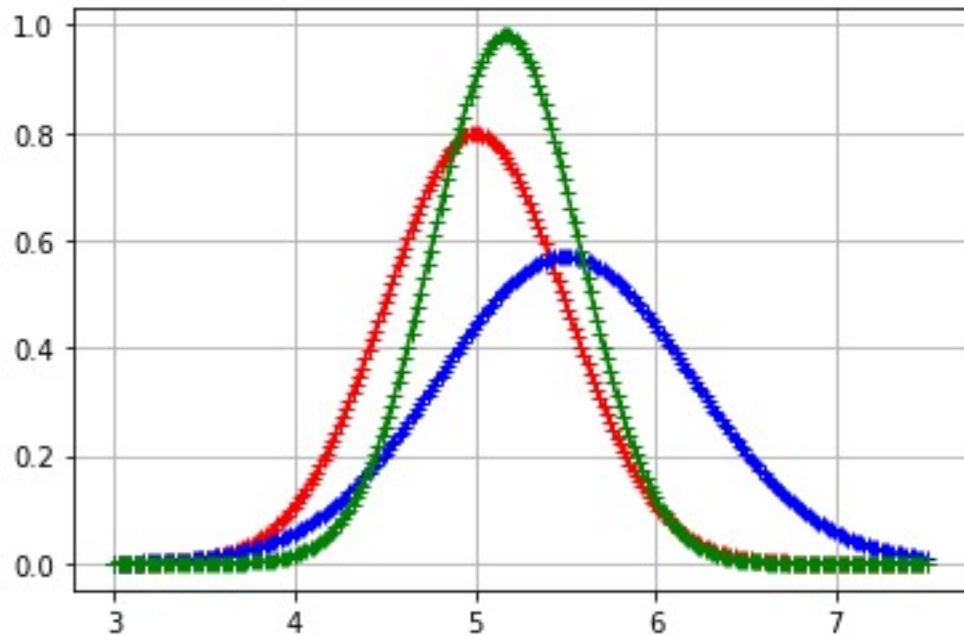The second sensor measured the distance of 5.5m with a variance of 0.7m.

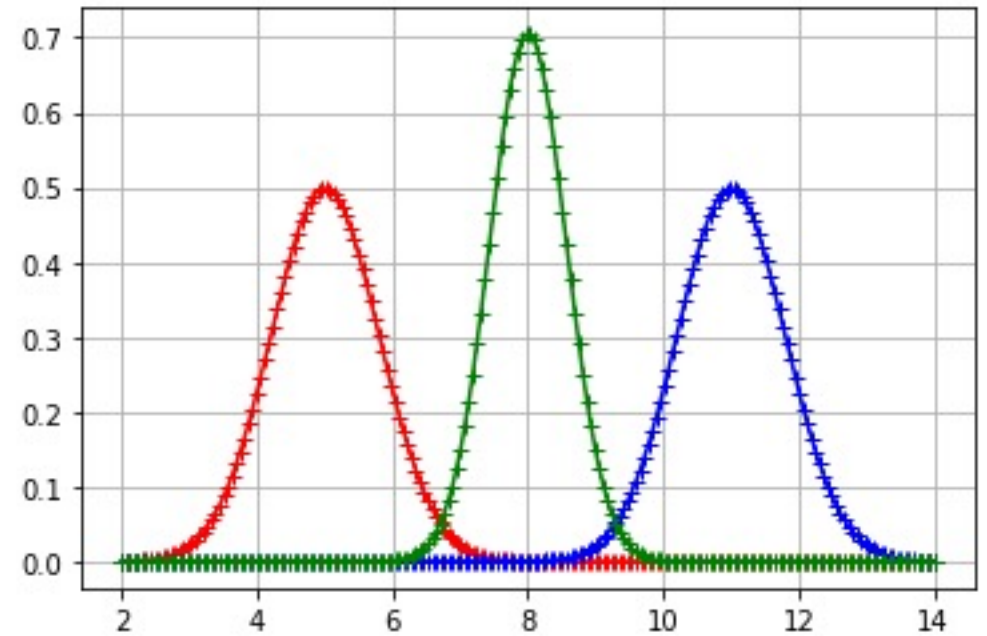Make the best estimation of the real distance to the obstacle.

Very nice example: https://datascopeanalytics.com/blog/sensor-fusion-tutorial/

$$\mu_1 = 5.0 \quad \mu_2 = 5.5 \quad \mu_c = 5.17$$
$$\sigma_1 = 0.5 \quad \sigma_2 = 0.7 \quad \sigma_c = 0.41$$

$$\mu_1 = 5 \quad \mu_2 = 11 \quad \mu_c = 8.0$$
$$\sigma_1 = 0.8 \quad \sigma_2 = 0.8 \quad \sigma_c = 0.56$$

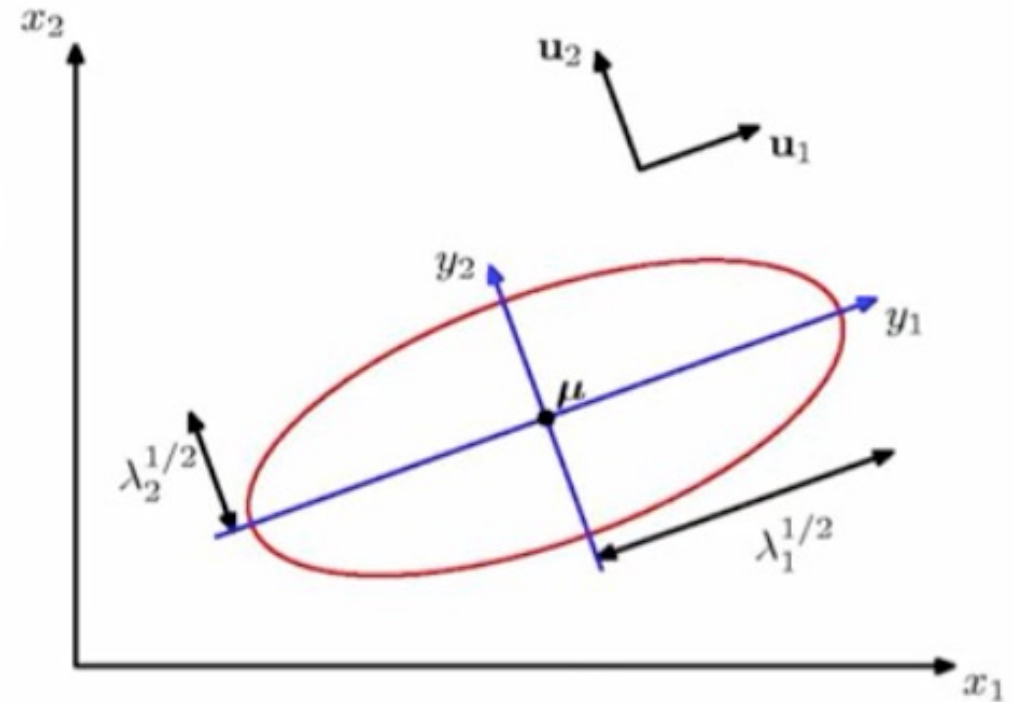Very nice example: https://datascopeanalytics.com/blog/sensor-fusion-tutorial/

2D Normal Distribution

$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} 1 & 3/5 \\ 3/5 & 2 \end{bmatrix}$$
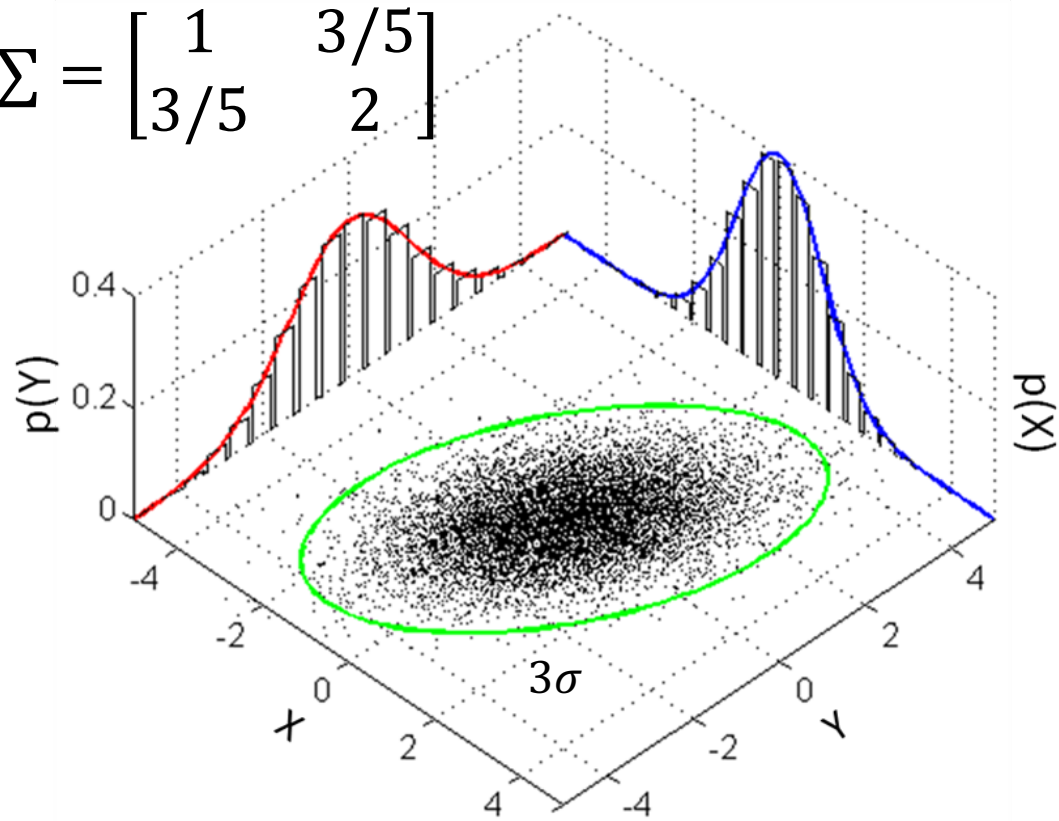
$$\Sigma = U\Lambda U^T = \begin{bmatrix} u_1 & u_2 \\ u_1 & u_2 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} u_1 & u_1 \\ u_2 & u_2 \end{bmatrix}$$
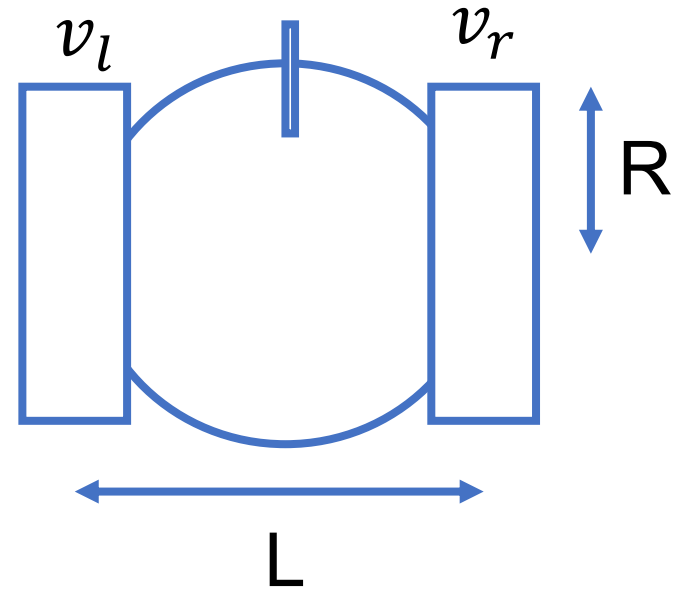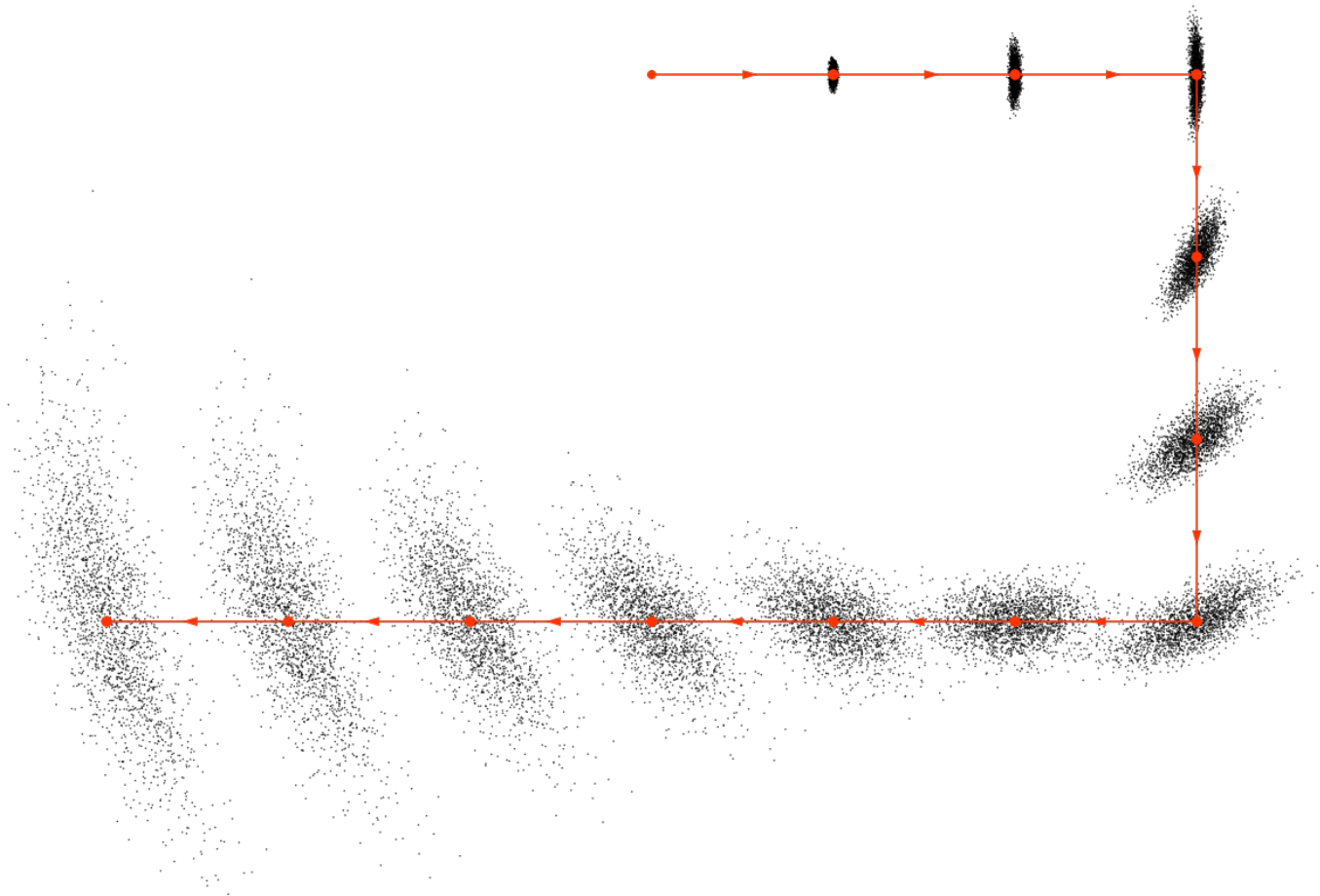
[1] https://en.wikipedia.org/wiki/Multivariate_normal_distribution
[2] https://www.youtube.com/watch?v=eho8xH3E6mE

Growing unsertancy of the position of the moving robot

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \dfrac{R}{2}(v_l + v_r) \\[2mm] \dfrac{R}{L}(v_r - v_l) \end{bmatrix}$$

[1] https://en.wikipedia.org/wiki/Monte_Carlo_localization#/media/File:Particle2dmotion.svg

Bayes Theorem

Thomas Bayes (c. 1701 – 7 April 1761)
was an English statistician, philosopher
and Presbyterian minister who is known
for formulating a specific case of the theorem
that bears his name: Bayes' theorem.

[1] https://cs.wikipedia.org/wiki/Thomas_Bayes

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)} = \frac{P(E|H)P(H)}{P(E|H)P(H) + P(E|\neg H)P(\neg H)}$$

H ... Hypothesis

E ... Evidence

$P(H)$ ... Probability of hipothesis is true, when there is no evidence

$P(E|H)$ ... Probability of seeing evidence if the hypothesis is true

$P(E)$ ... Probability of seeing evidence

$P(H|E)$ ... Probability of hypothesis when seeing evidence

[1] https://www.youtube.com/watch?v=HZGCoVF3YvM

$$P(\text{☀️} | \text{🌱}) = \frac{P(\text{🌱}|\text{☀️})P(\text{☀️})}{P(\text{🌱})} = \frac{P(\text{🌱}|\text{☀️})P(\text{☀️})}{P(\text{☀️})P(\text{🌱}|\text{☀️}) + P(\text{🌧️})P(\text{🌱}|\text{🌧️})}$$
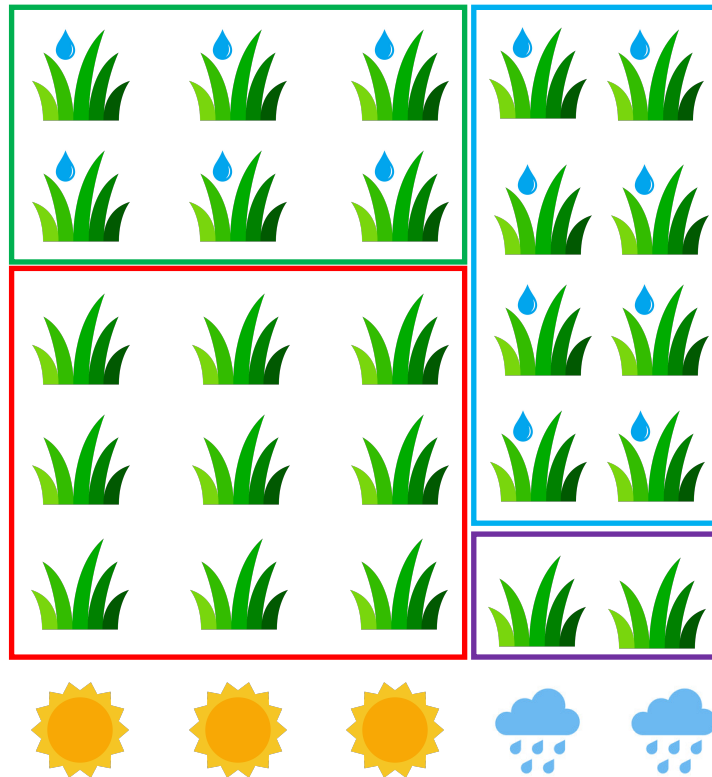
H ... Hypothesis: ☀️

E ... Evidence: 🌱

$\neg H$ ... Hypothesis: 🌧️

$\neg E$ ... Evidence: 🌱

[1] https://www.youtube.com/watch?v=HZGCoVF3YvM

$$P(☀️ \mid 🌱) = \frac{P(🌱 \mid ☀️)P(☀️)}{P(🌱)} = \frac{P(🌱 \mid ☀️)P(☀️)}{P(☀️)P(🌱 \mid ☀️) + P(🌧️)P(🌱 \mid 🌧️)}$$

$$P(☀️ \mid 🌱) = ?$$

[1] https://www.youtube.com/watch?v=HZGCoVF3YvM

$$P(\text{☀}|\text{🌱}) = \frac{P(\text{🌱}|\text{☀})P(\text{☀})}{P(\text{🌱})} = \frac{P(\text{🌱}|\text{☀})P(\text{☀})}{P(\text{☀})P(\text{🌱}|\text{☀}) + P(\text{🌧})P(\text{🌱}|\text{🌧})}$$

$$\frac{6}{25} = 0.24$$

$$\frac{9}{25} = 0.36$$

$$\frac{8}{25} = 0.32$$

*

* 2/25 = 0.08

$$P(\text{☀}) = 0.6 \quad P(\text{🌧}) = 0.4$$

$$P(\text{🌱}|\text{☀}) = \frac{\Box}{\Box + \Box} = 0.6$$

$$P(\text{🌱}|\text{🌧}) = \frac{\Box}{\Box + \Box} = 0.2$$

$$P(\text{☀}|\text{🌱}) = \frac{\Box}{\Box + \Box} = 0.818$$

[1] https://www.youtube.com/watch?v=HZGCoVF3YvM

$$P(\text{☀️}|\text{🌱}) = \frac{P(\text{🌱}|\text{☀️})P(\text{☀️})}{P(\text{🌱})} = \frac{P(\text{🌱}|\text{☀️})P(\text{☀️})}{P(\text{☀️})P(\text{🌱}|\text{☀️}) + P(\text{🌧️})P(\text{🌱}|\text{🌧️})}$$
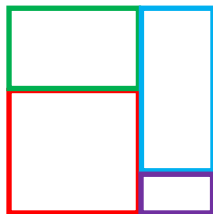
$P(\text{☀️}) = 0.6$

$P(\text{🌧️}) = 0.4$
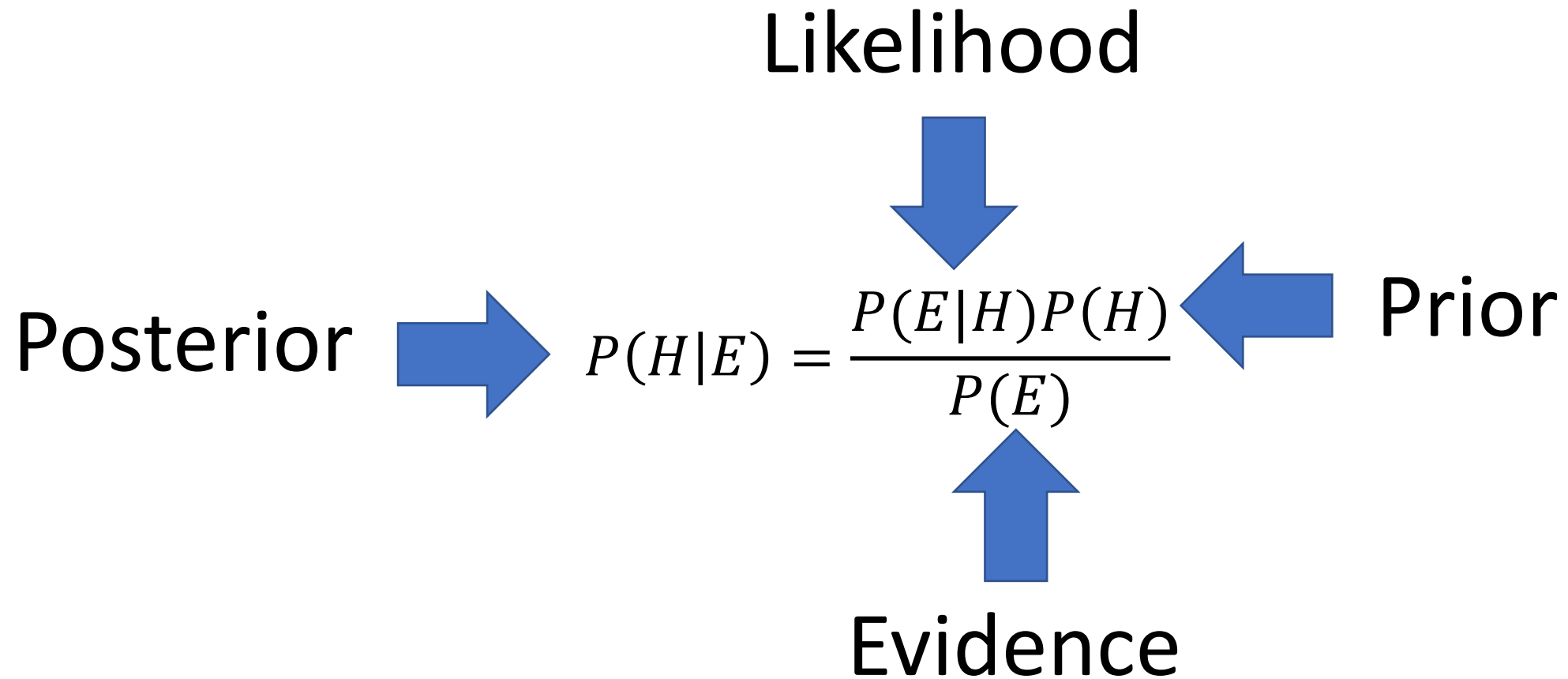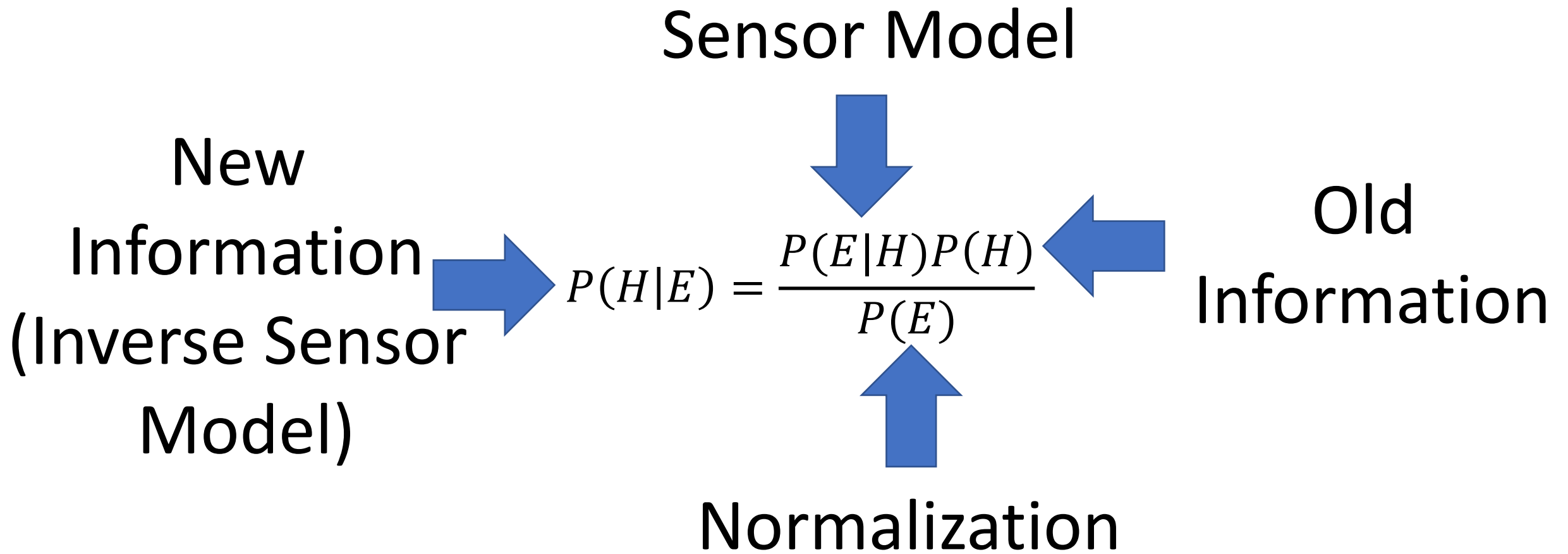
$P(\text{🌱}|\text{☀️}) = 0.6$

$P(\text{🌱}|\text{🌧️}) = 0.2$

$$P(\text{☀️}|\text{🌱}) = \frac{0.6 * 0.6}{P(\text{🌱})} = \frac{0.6 \ * \ 0.6}{0.6 * 0.6 + 0.4 * 0.2} = 0.818$$

$$P(\text{☀️}|\text{🌱}) = \frac{\square}{\square + \square} = 0.818$$

[1] https://www.youtube.com/watch?v=HZGCoVF3YvM

Likelihood

Posterior $\longrightarrow$

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)}$$

Prior

Evidence

[1] https://www.youtube.com/watch?v=HZGCoVF3YvM

Sensor Model

New Information (Inverse Sensor Model)

Old Information

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)}$$

Normalization

[1] https://www.youtube.com/watch?v=HZGCoVF3YvM

Bayes Filter is **generic mathematical framework** to estimate inner states of the system (world).

Prediction:

Motion Model

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

Correction:

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

Observation Model

BRNO
UNIVERSITY
OF TECHNOLOGY

FACULTY OF ELECTRICAL
ENGINEERING
AND COMMUNICATION

Sensors

Typically:

Lasers, LiDARs

Ultrasound sensors

IR range sensor

...

Provides information about the distance and the azimuth of the object w.r.t. robot's coordinates system.

2D LiDAR scan

Ultrasound sensor characteristics

$$\begin{bmatrix} r \\ \phi \end{bmatrix} = \begin{bmatrix} \sqrt{(x_{obj} - x_{robot})^2 + (y_{obj} - y_{robot})^2} \\ atan2(y_{obj} - y_{robot}, x_{obj} - x_{robot}) \end{bmatrix} + Q$$

Camera Sensor

Sensing usually "2D" data represented as an image

Types of camera sensors:

Common RGB/grayscale camera

Multispectral Camera

RGBD camera (kinect)

IR camera
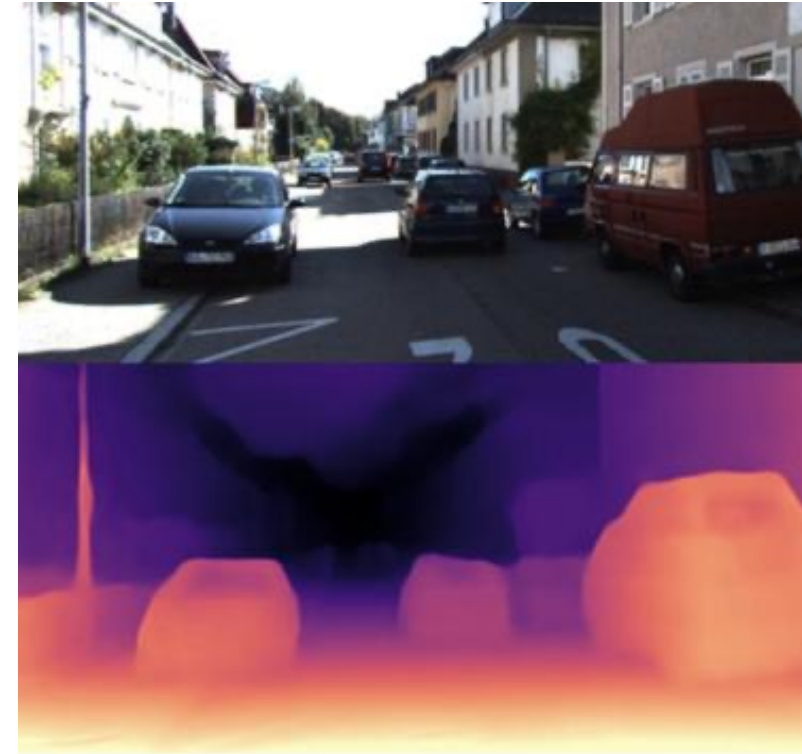
RGB-D Image: RGB+Depth

IR Image

RGB Image

Mapping World (3D) to Camera (2D) - projection

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} u/w \\ v/w \\ w \end{bmatrix} = C_{int} * H_{world\_to\_cam} * x_{world} =$$

$$= \begin{bmatrix} f_x & \gamma & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Mapping Camera (2D) to World (3D)

- During projection we lost depth information
- Each pixel can be represented as ray in 3D
- Expand $\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$ for with scale factor: $\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} * s = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$

https://www.researchgate.net/profile/Luis-Ortiz-25/publication/326518096
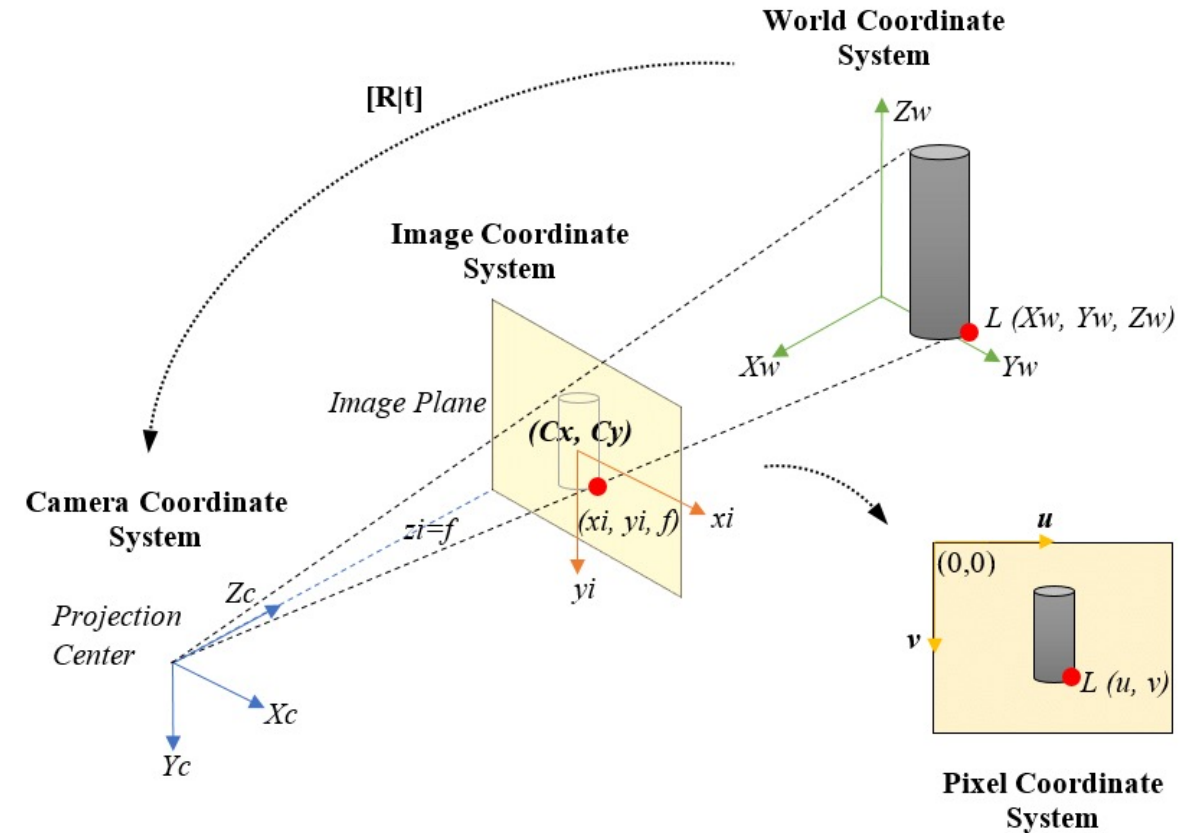
Line Follower
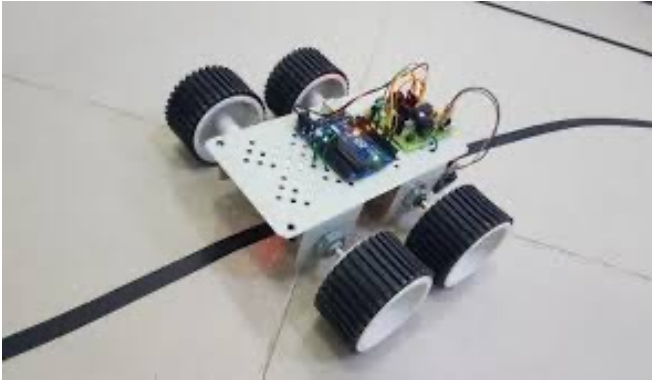
electronicshub.com

GNSS

http://what-when-how.com/

Motion Capture

https://deva90sapmc8w.cloudfront.net/roboticsPrime17w_8-450h.png
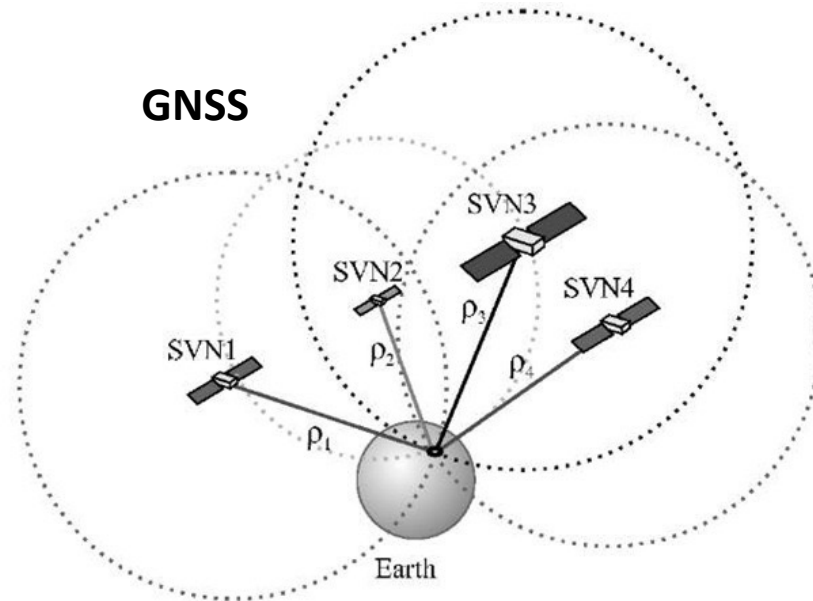
AR/QR/April Tags
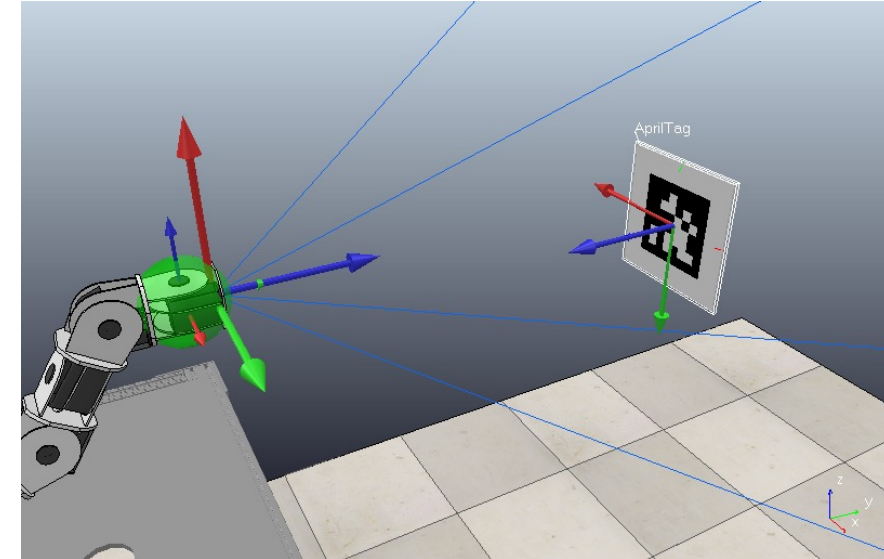
https://haowensh.com/assets/img/2020/kdc/frame_confirmation.png

## Beacon loclalization (UWB, BT, ...)

https://www.researchgate.net/publication/336832935
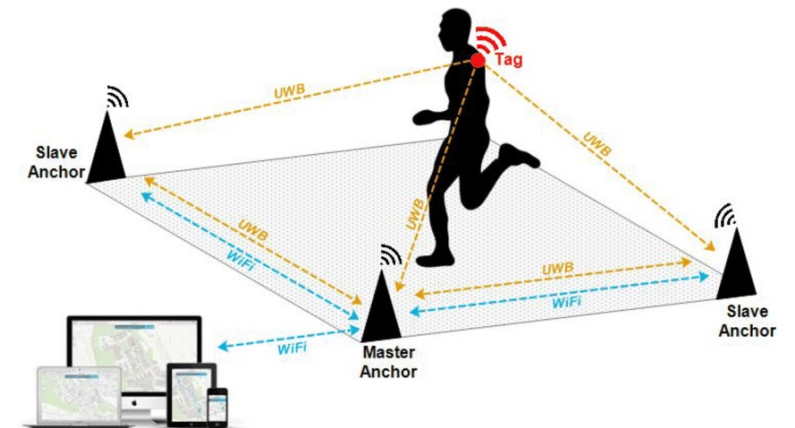
Map Types

Map is represented as a set of "Landmarks" (graph nodes)

Robot measures distance to the landmarks (distances are graph edges)

Robot estimates position by building a graph with minimal error w.r.t. measurements.

If the robot is also capable of expanding the map by adding new landmarks, we talk about SLAM (Simultaneous Localization and Mapping)

[1] https://jay.tech.blog/2017/03/18/graph-slam/
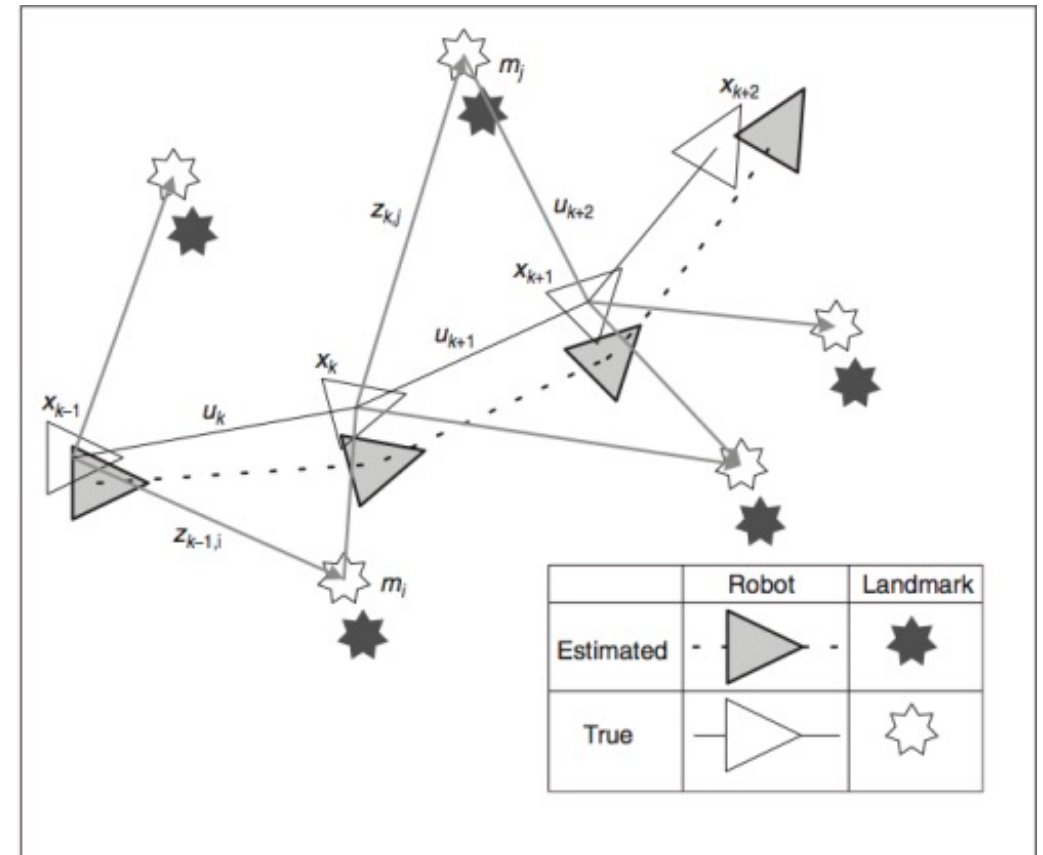
The world is represented as a 2D/3Dcell grid.

Every cell contains a number that holds information about the probability, the cell is occupied by an obstacle.

$p(M^O)$ ... probability, cell is Occupied

$p(M^E)$ ... probability, cell is Empty

$p(M^O) = 0 \sim p(M^E) = 1$ ... cell is Empty

$p(M^O) = 0.5 \sim p(M^E) = 0.5$ ... no information about cell

$p(M^O) = 1 \sim p(M^E) = 0$ ... cell is Occupied
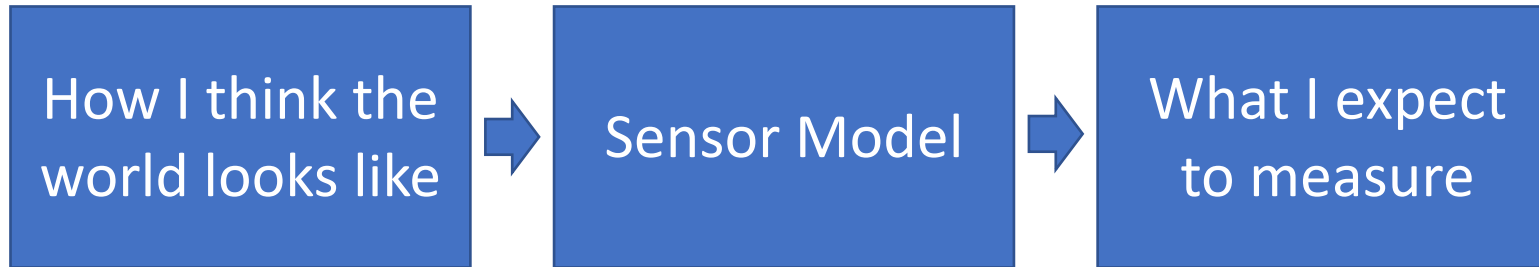
$$0 \leq p(M^O) \leq 1$$
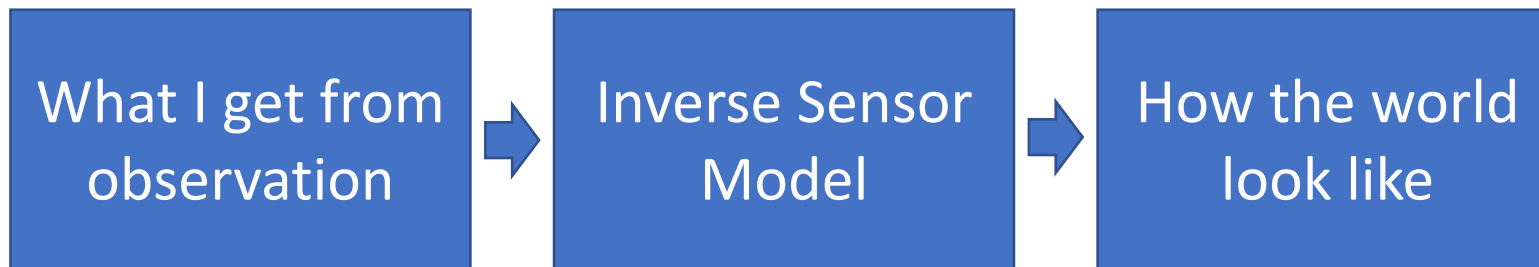
$$0 \leq p(M^E) \leq 1$$

$$p(M^O) + p(M^E) = 1$$

[1] Tomáš Lázna, Mapy a navigace v mobilní robotice, 2019
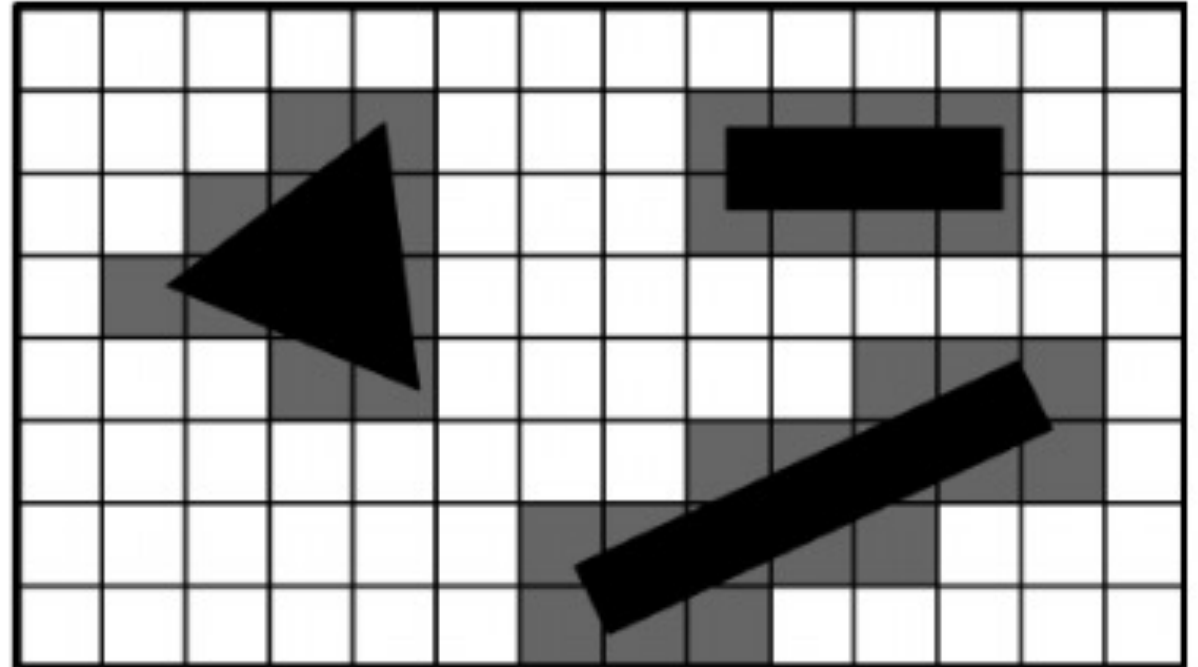
Sensor Modeling

Sensor Model

How I think the world looks like	→	Sensor Model	→	What I expect to measure

Inverse Sensor Model

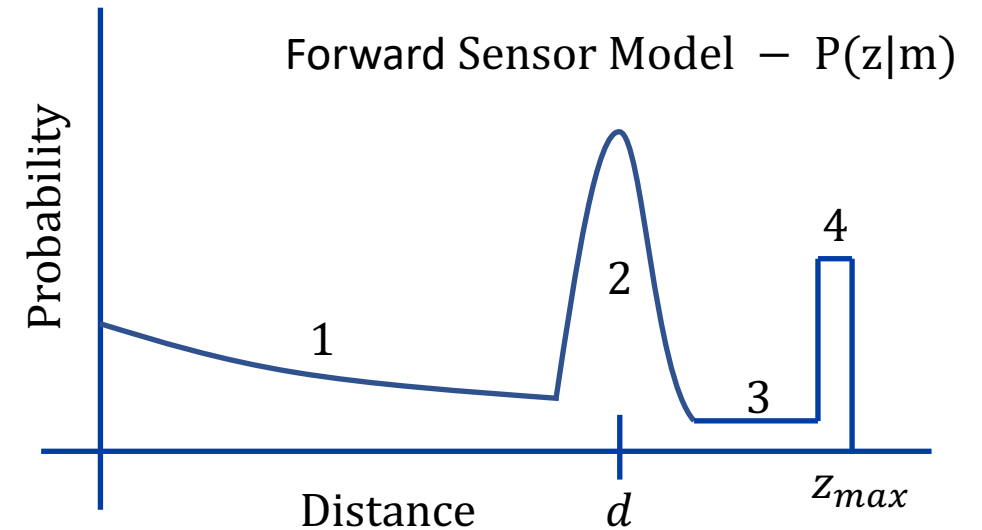| What I get from observation	→	Inverse Sensor Model	→	How the world look like

Common Laser Sensor Model:

Probability of getting measurement
of an object at distance d.
z_{max} is sensor's max range.

1 ... Approximation of dynamic obstacles (exp decay)
2 ... Distribution around obstacle (normal dist.)
3 ... Unknown area (uniform dist.)
4 ... Max sensor's range

Common Laser Inverse Sensor Model:

Probability of object's presence
at the distance of z_t.

1 ... Free space (low prob.)
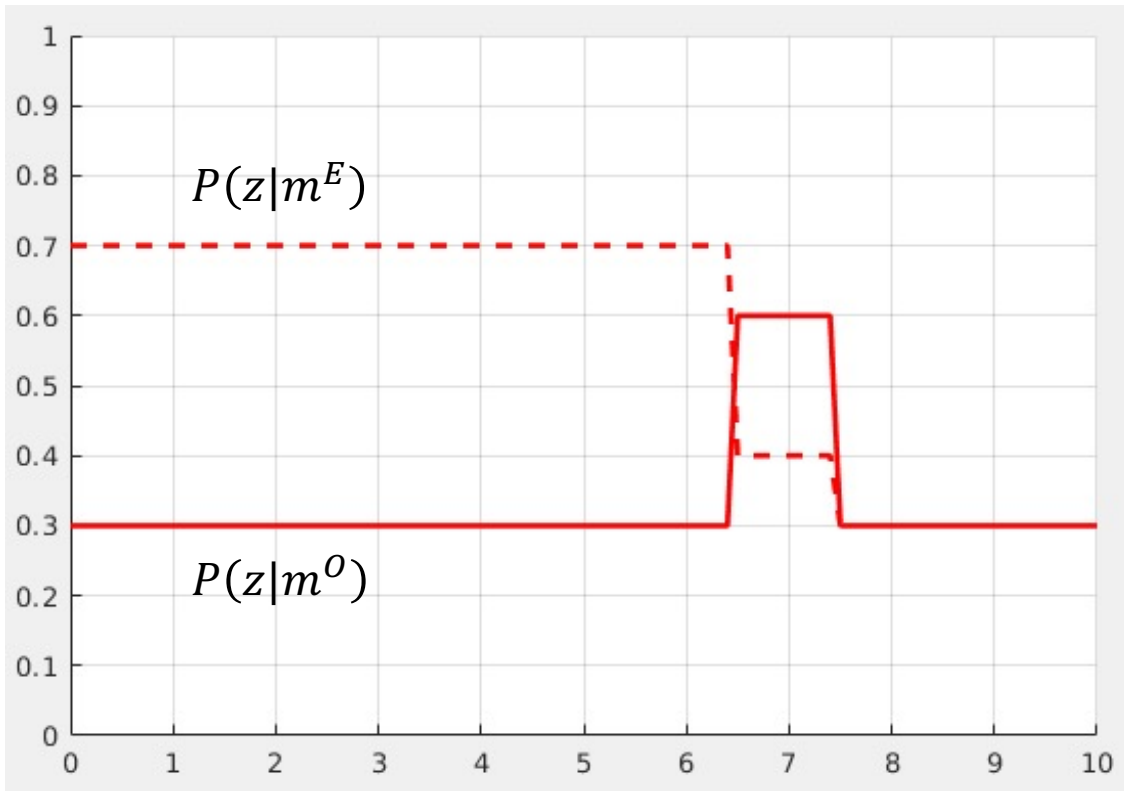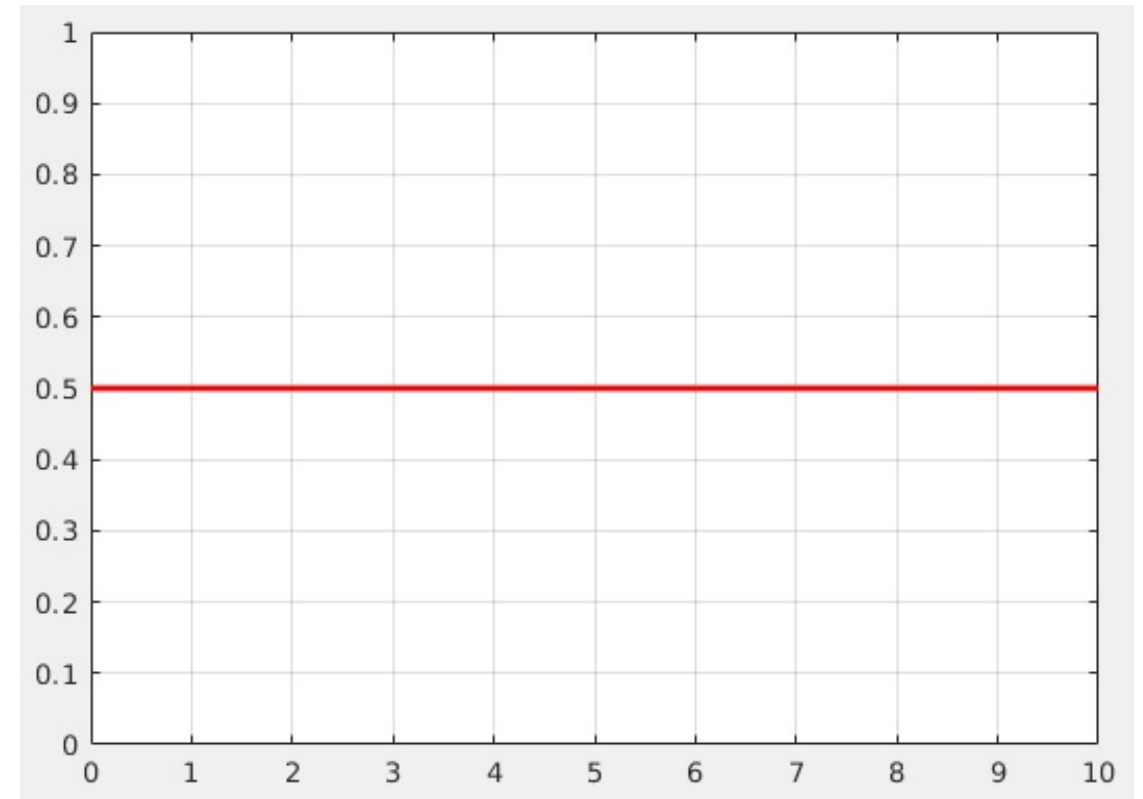2 ... Occupied space (high prob.)
3 ... Unknown area (p=0.5)

Forward Sensor Model — P(z|m)

Probability

Distance d z_{max}

Inverse Sensor Model — P(m|z)

Probability

z
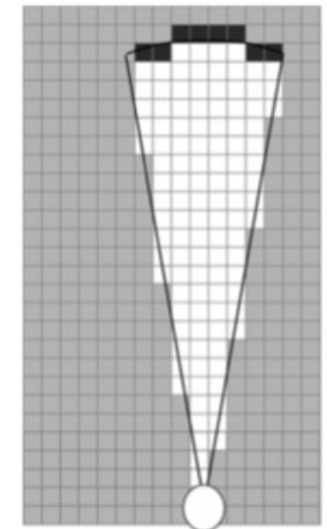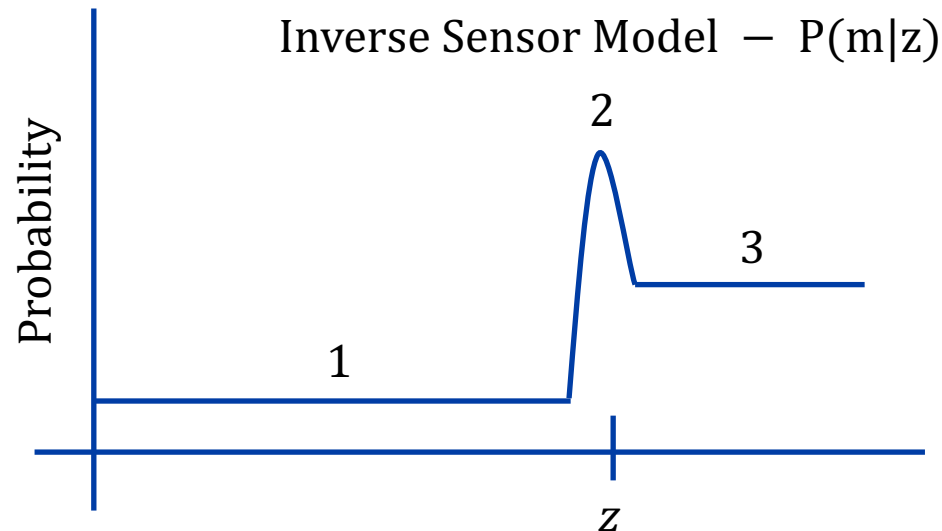
ISM 2D

$Measurement\ z = 7m$

Forward Sensor Model $-$ P(z|m)
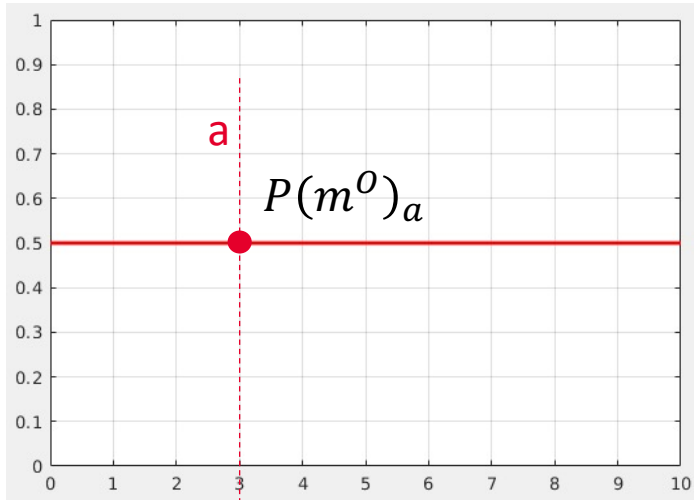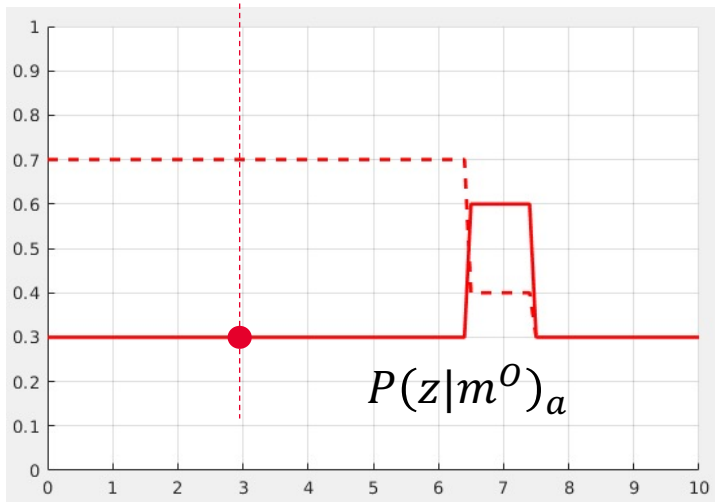
Map $-$ P(m^O)

Measurement $z = 7m$

Map

Forward Sensor Model

0) Init

$$P(m^O)_a = 0.5 \quad \rightarrow \quad P(m^E)_a = 1 - P(m^O)_a = 0.5$$
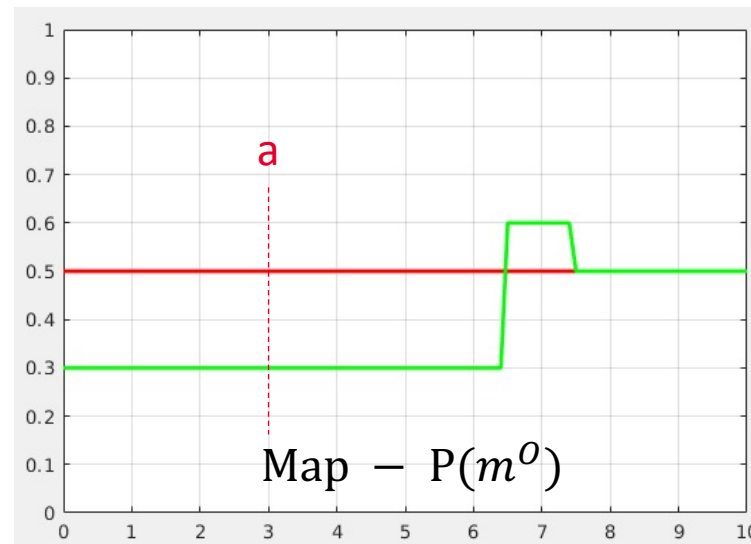
1) First measurement

$$P(m^O)_a = 0.5 \quad \rightarrow \quad P(m^E)_a = 1 - P(m^O)_a = 0.5$$
$$P(z|m^O)_a = 0.3 \quad \rightarrow \quad P(z|m^E)_a = 0.7$$
$$P(z)_a = P(z|m^O)_a * P(m^O)_a + P(z|m^E)_a * P(m^E)_a =$$
$$= 0.3 * 0.5 * 0.7 * 0.5 = 0.5;$$

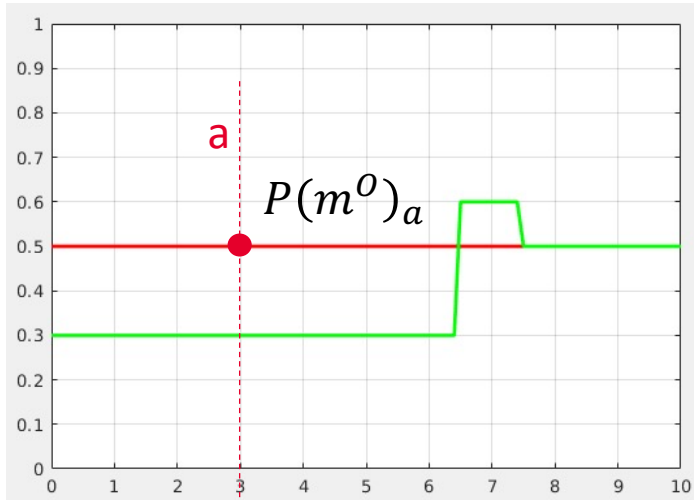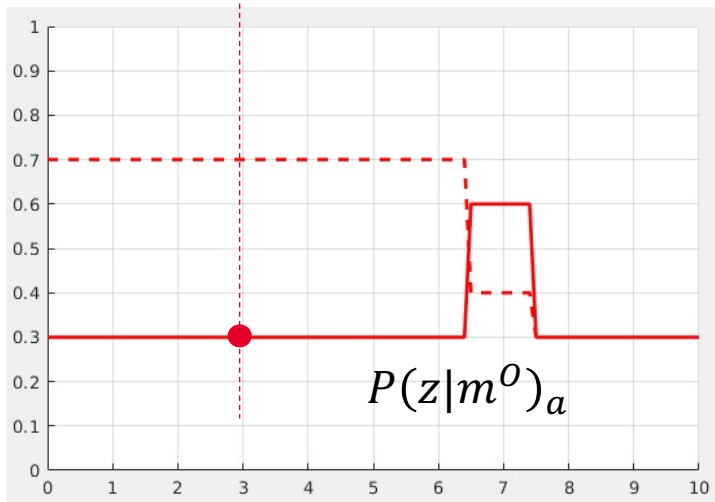$$P(m^O|z)_a = \frac{P(z|m^O)_a * P(m^O)_a}{P(z)_a} = \frac{0.3 * 0.5}{0.5} = 0.3$$

$\text{Map} - \text{P}(m^O)$

Measurement $z = 7m$

Map

Forward Sensor Model

2) Second measurement

$$P(m^O)_a = 0.3 \quad \rightarrow \quad P(m^E)_a = 1 - P(m^O)_a = 0.7$$
$$P(z|m^O)_a = 0.3 \quad \rightarrow \quad P(z|m^E)_a = 0.7$$
$$P(z)_a = P(z|m^O)_a * P(m^O)_a + P(z|m^E)_a * P(m^E)_a =$$
$$= 0.3 * 0.3 * 0.7 * 0.7 = 0.58;$$

$$P(m^O|z)_a = \frac{P(z|m^O)_a * P(m^O)_a}{P(z)_a} = \frac{0.3 * 0.3}{0.58} = 0.155$$
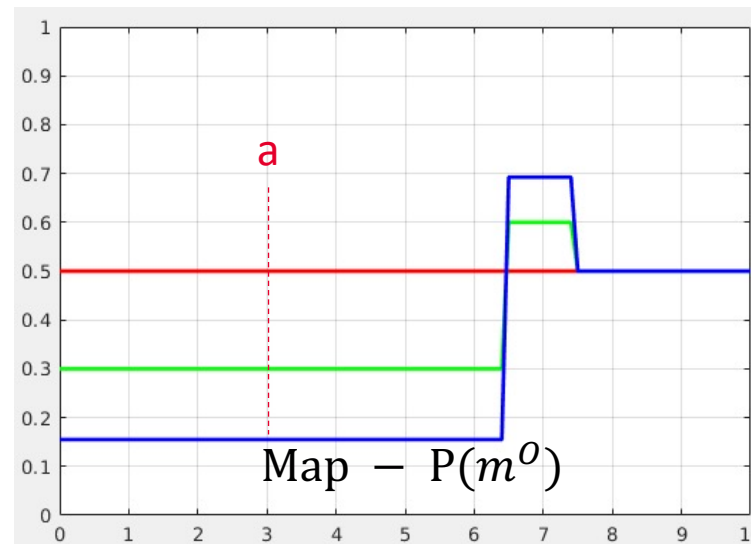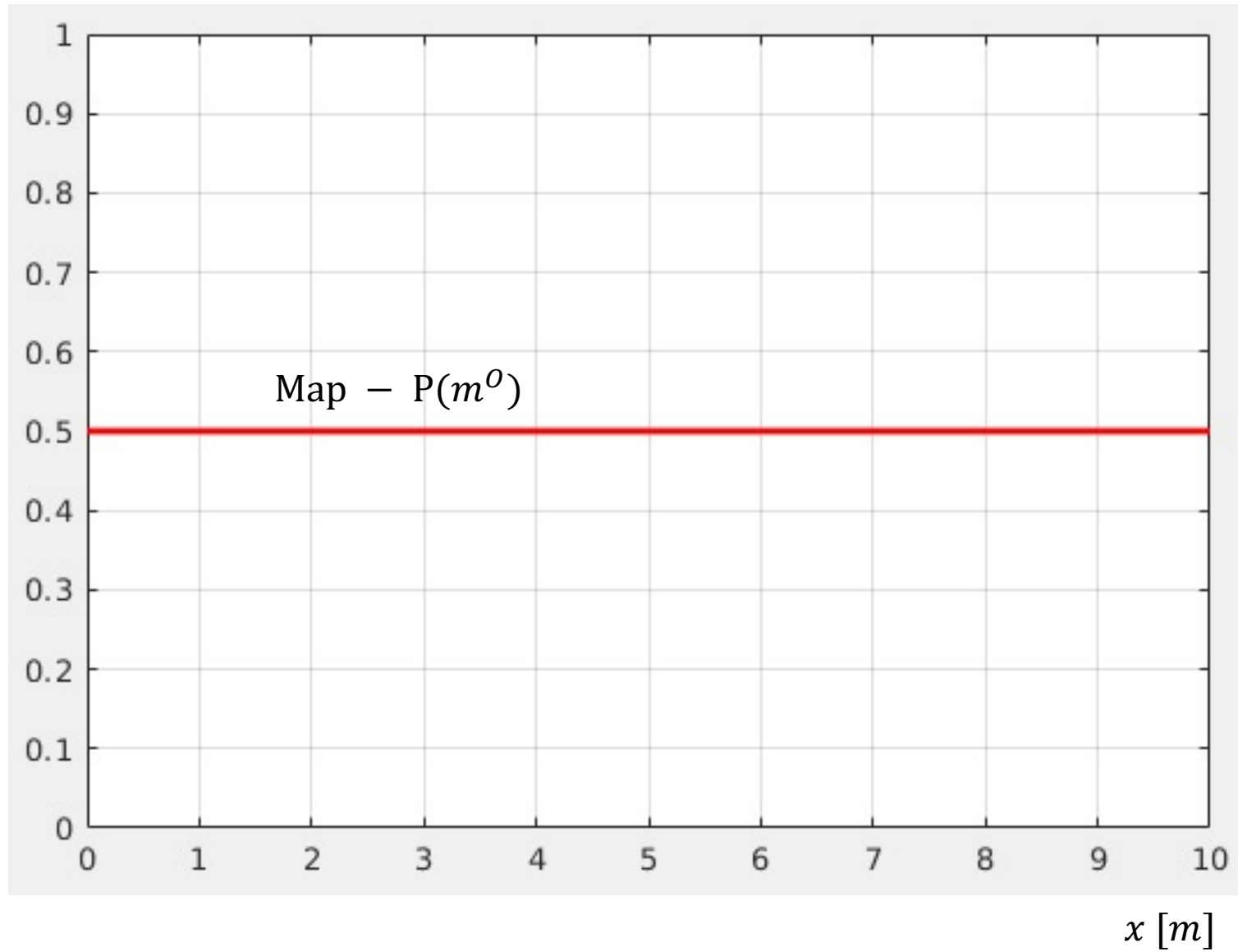
Map $-$ P(m^O)
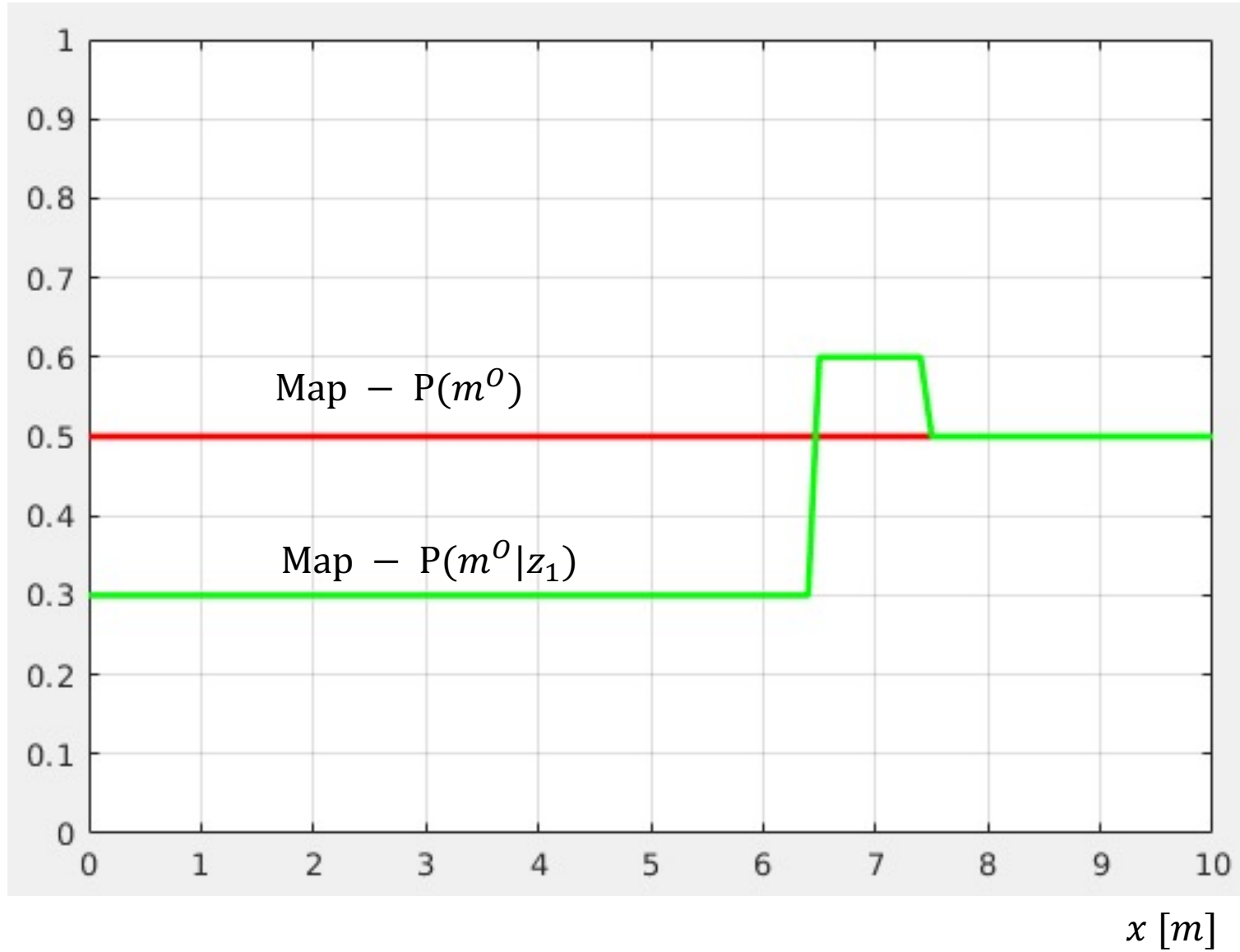
Map – $P(m^O)$
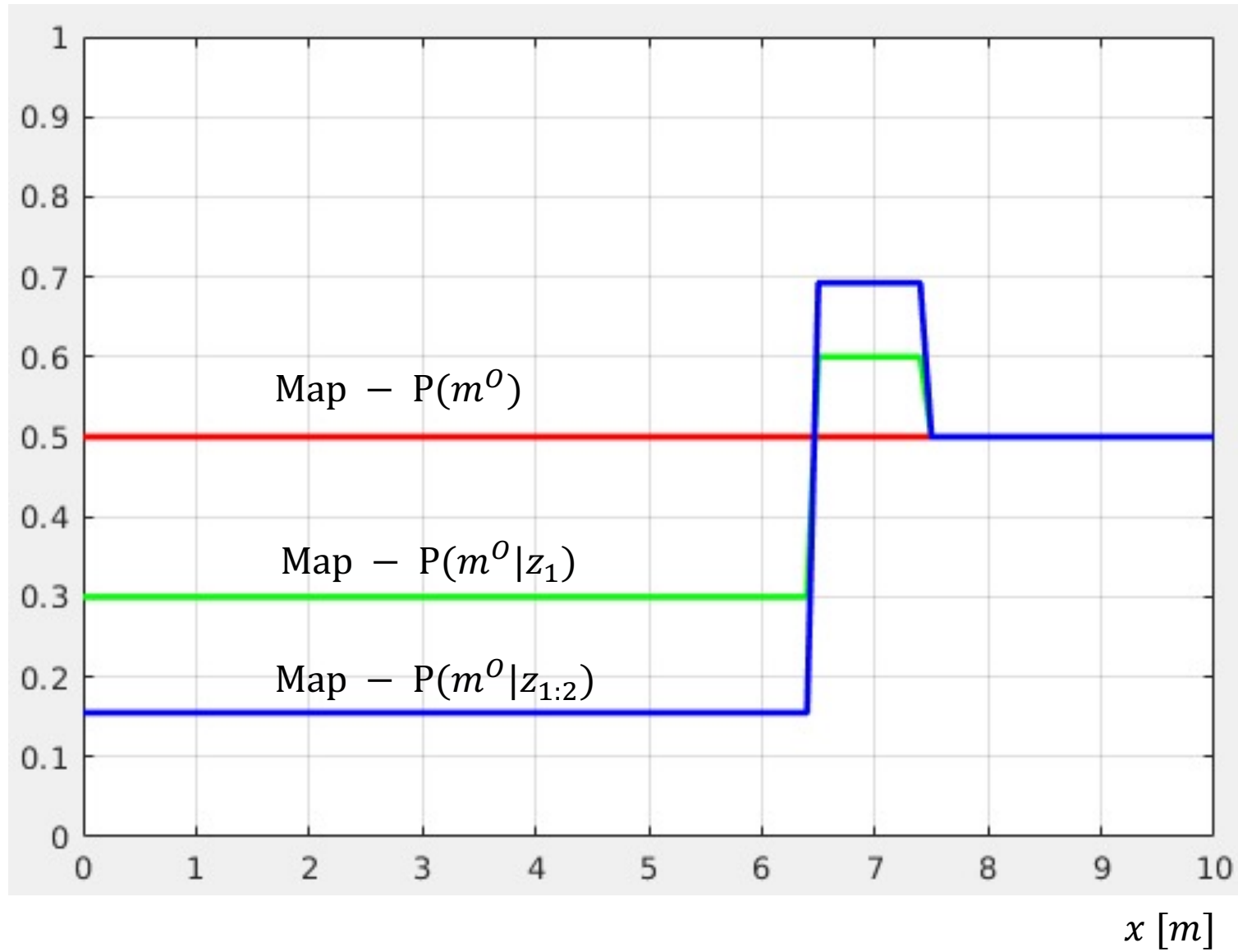
$x\,[m]$

Map − $\mathrm{P}(m^O)$

Map − $\mathrm{P}(m^O|z_1)$

$x\,[m]$

Map – $P(m^O)$

Map – $P(m^O|z_1)$

Map – $P(m^O|z_{1:2})$

$x\ [m]$

Map $-$ P$(m^O|z_{1:4})$

Map $-$ P($m^O|z_{1:5}$)
Map $-$ P($m^O|z_{1:4}$)
Map $-$ P(m^O)
Map $-$ P($m^O|z_1$)
Map $-$ P($m^O|z_{1:2}$)
Map $-$ P($m^O|z_{1:3}$)
Map $-$ P($m^O|z_{1:5}$)
x [m]
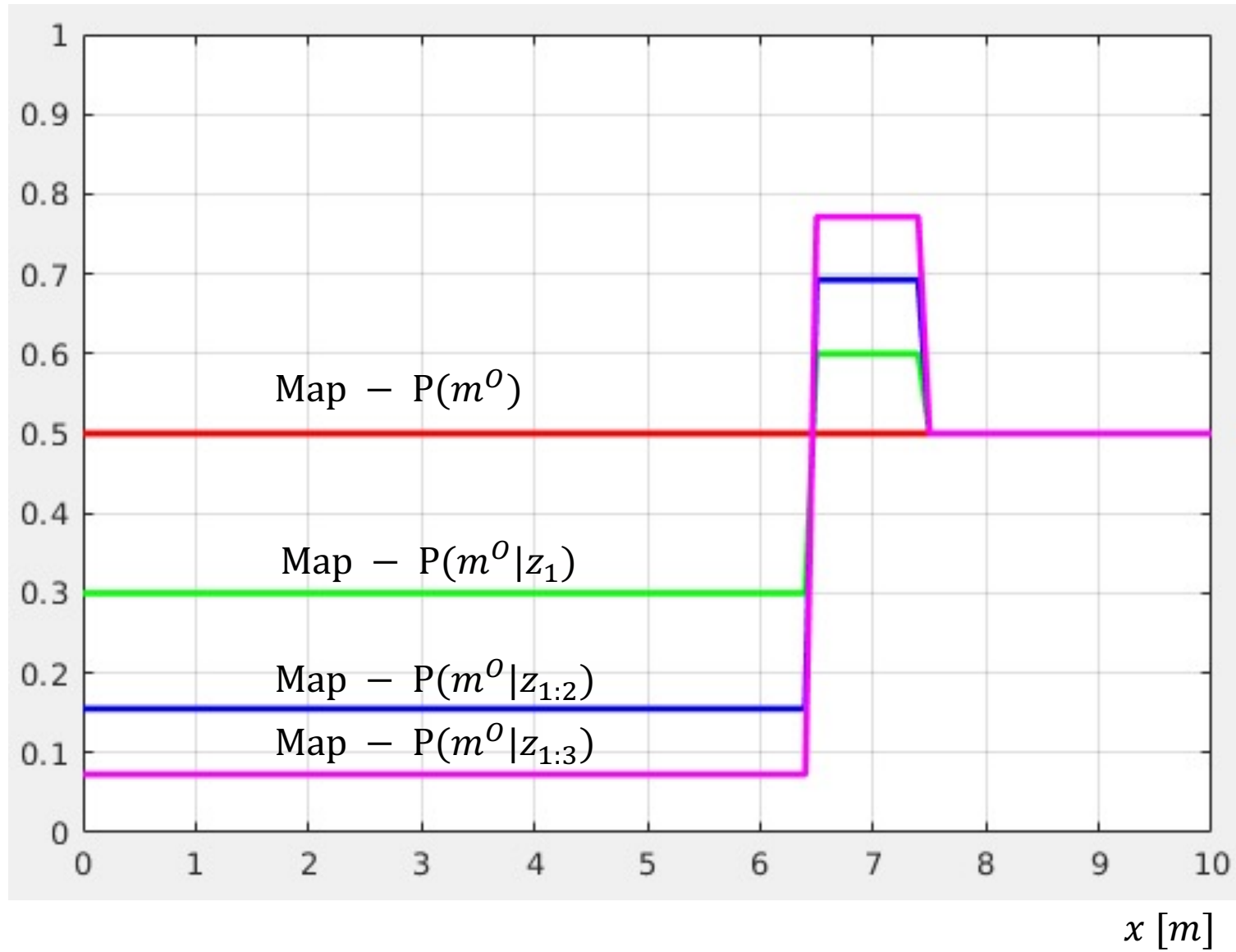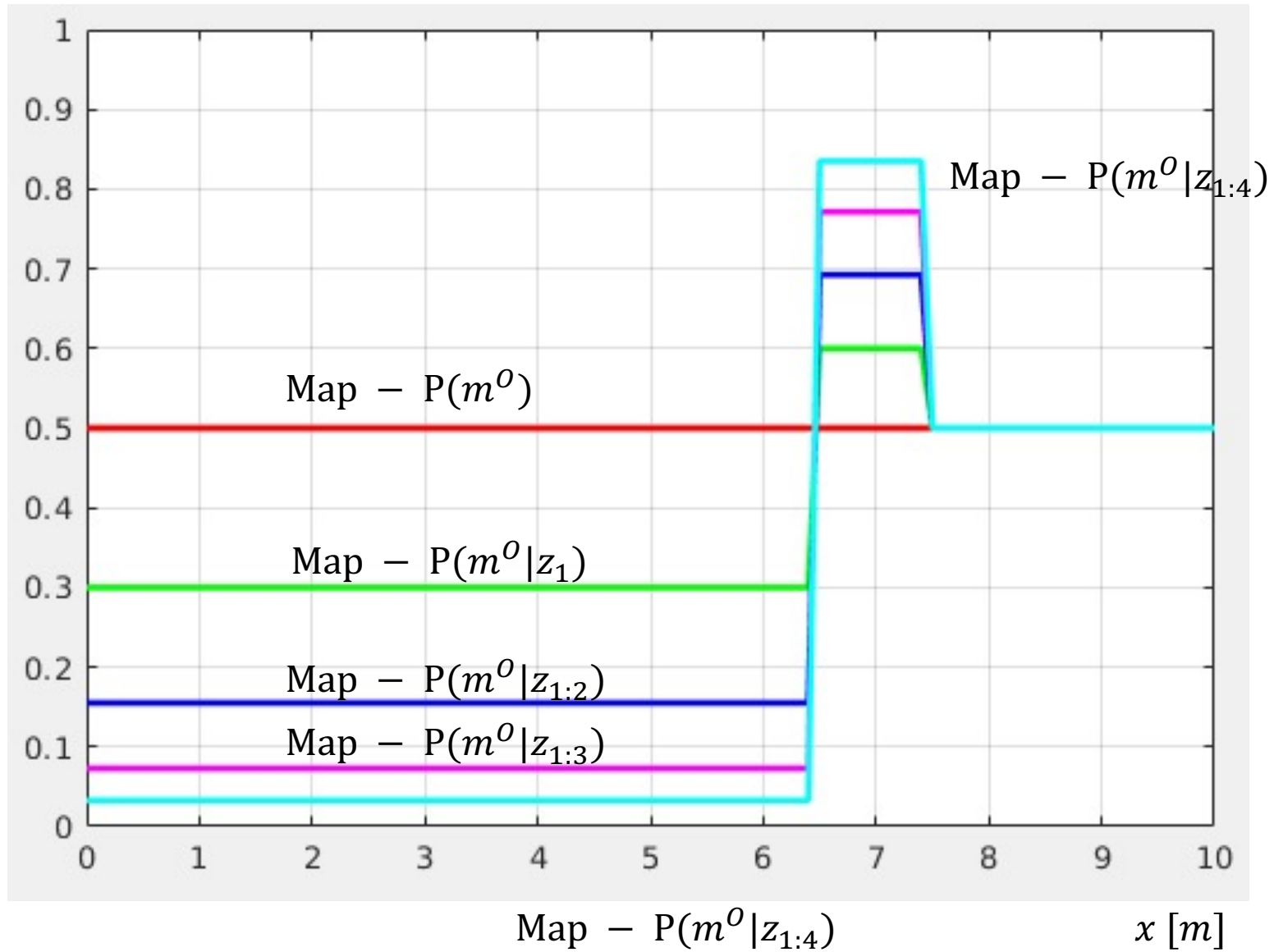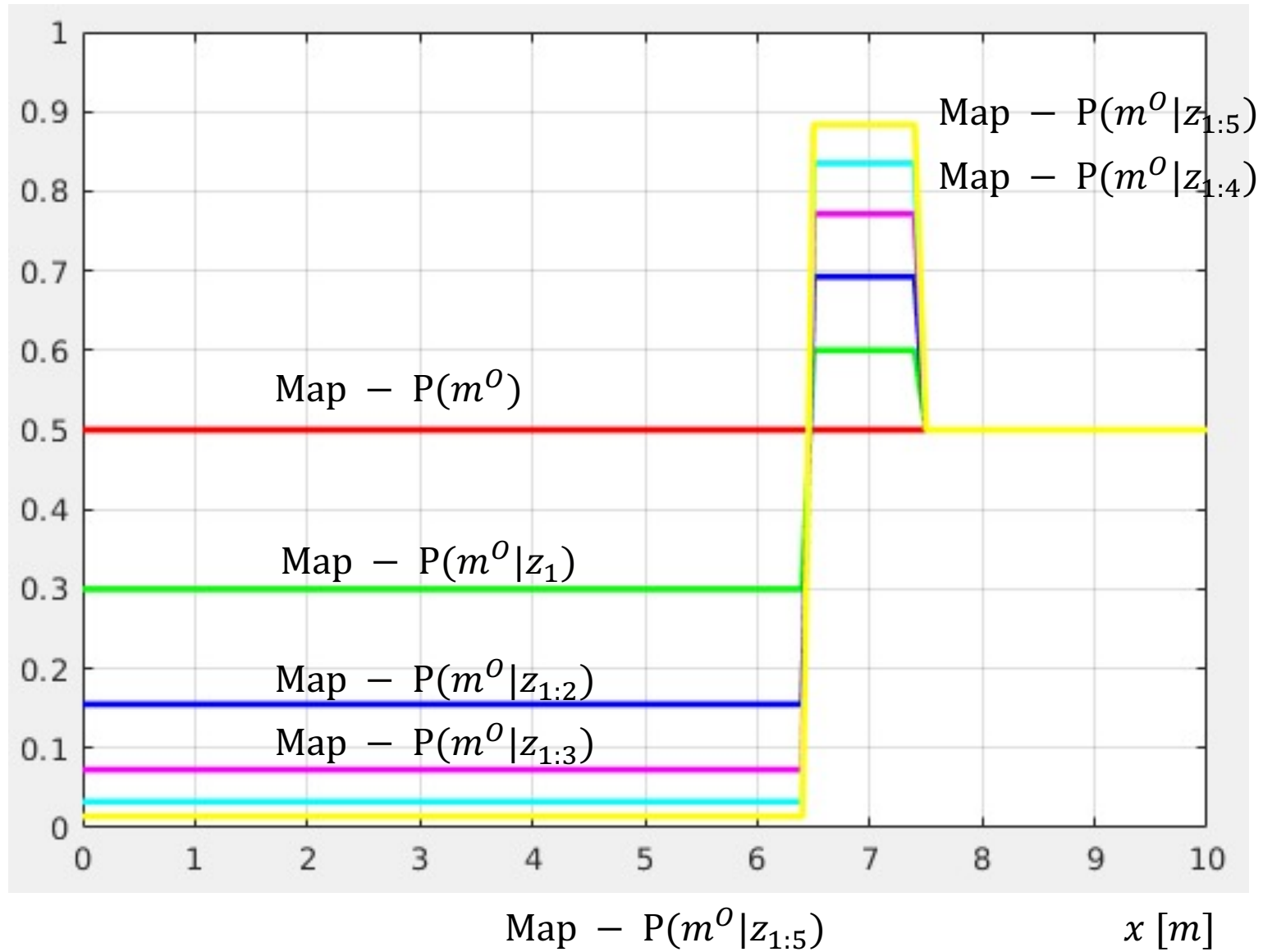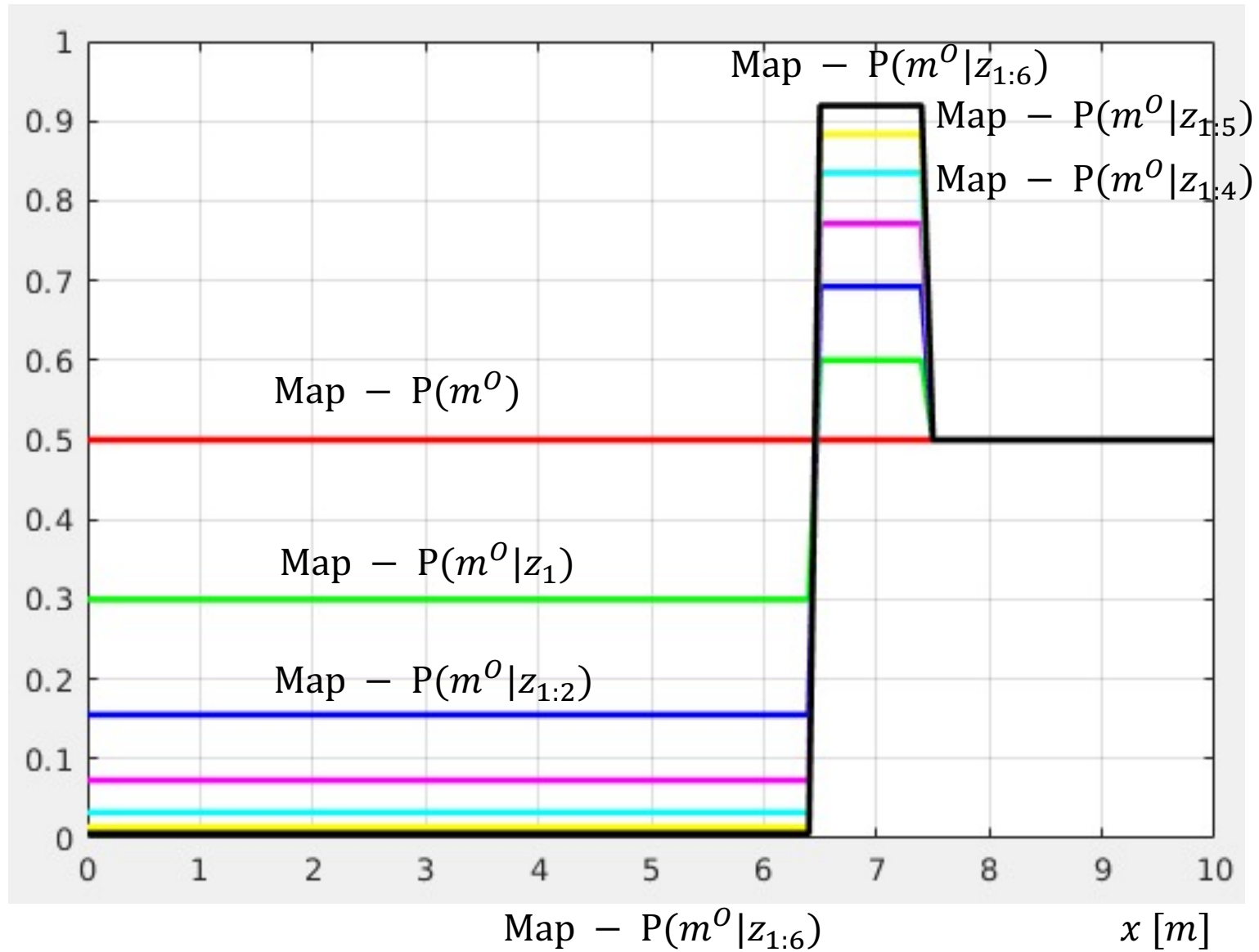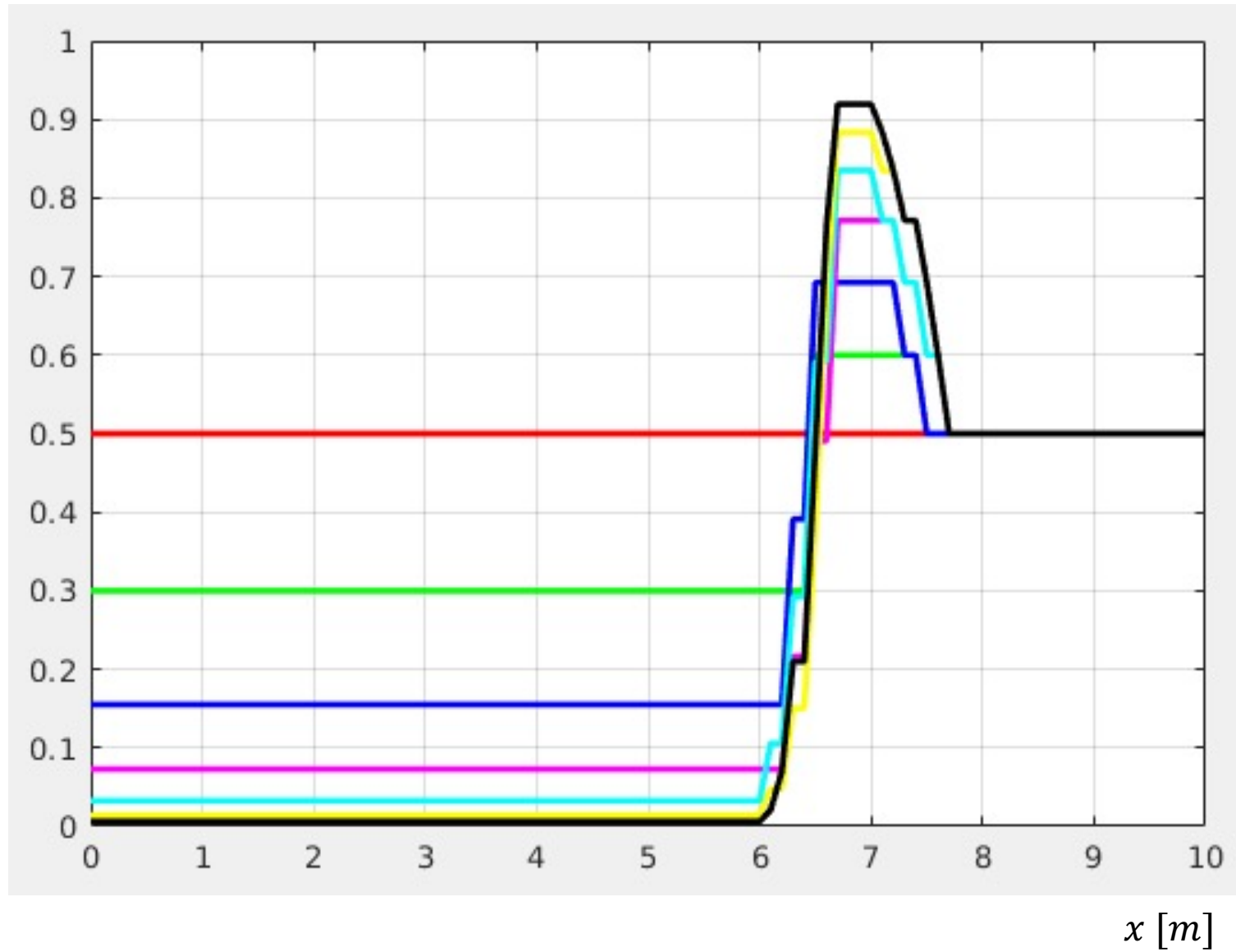
Sensor Modeling in ND Space

(a) (b) (c)

(d) (e) conflict (f)

Inverse Sensor Model

500 particles with 100 data points

Ground Truth

Noisy map

Noisy Map

https://github.com/khazit/CrazySLAM
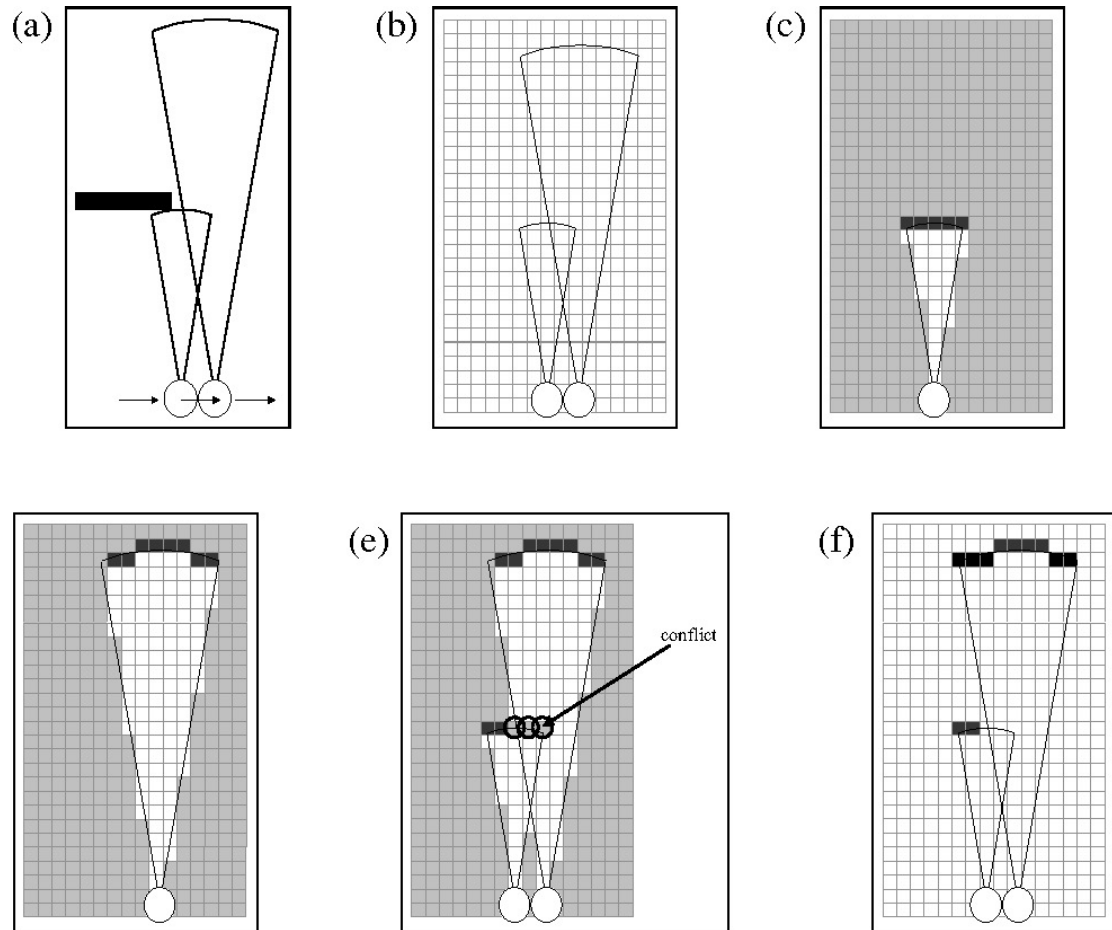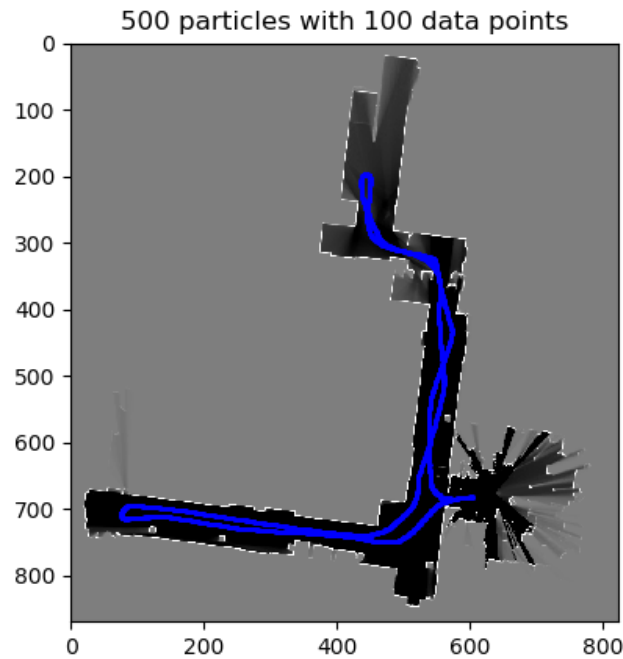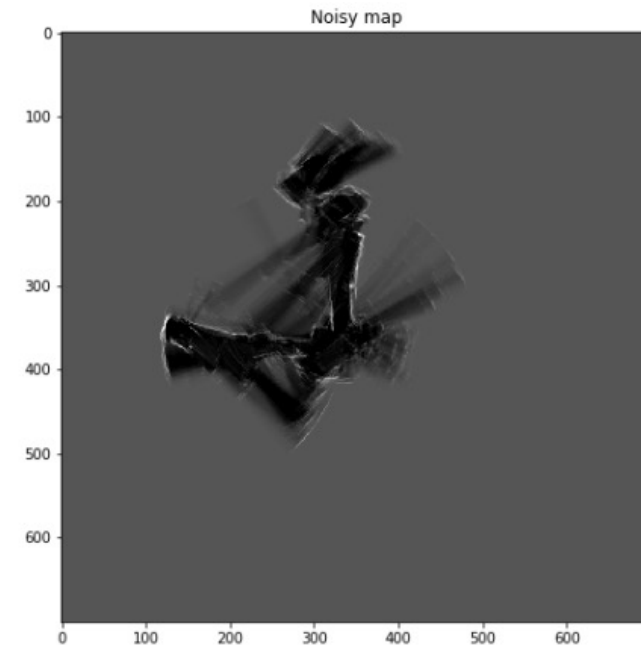
Summary

Normal Distribution

    Combining independent variables
    Combining multiple observations

Bayes rule

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)}$$

Bayes Filter

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

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

Sensor Model

    Map -> expected sensor output

Inverse Sensor Model

    Sensor output -> map

Maps

    Occupancy grid
    Topological / Graph

Assignment

Assignment No. 1

Using MATLAB script, simulate the simple 1D map building process using the simulated measurement and self-proposed Inverse Sensor Model (ISM).

Create a single A4 report that will describe your approach to the exercise (3-6 sentences for each task and picture, if makes sense).

Submission

Send the report and all related MATLAB scripts at adam.ligocki@vutbr.cz.

MATLAB script must be executable without errors and has to generate all graphical outputs that are in the report.

Deadline: 20th Feb 2022, 23:59.

MPC-MAP Assignment No. 1

Using MATLAB script, simulate the simple 1D map building process using the simulated measurement and self-proposed Sensor Model.

Create a single A4 report that will describe your approach to the exercise (3-6 sentences for each task and picture, if makes sense).

Task 1

Define parameters of your simulation
- Distance of the obstacle x
- Standard deviation of your sensor σ
- Discrete map parameters (cell size c and covered area d)

Note: Choose these values with the idea of visualizing your simulation. The visual outputs simulated for parameters of x=100m and σ=0.001m c=0.001m, d=200m will be useless.

Task 2

In the script file, implement the following functions

The "measure" function will return a single measurement sample with a normal distribution of (μ=obstacle_distance, σ=sensor_std_dev).

The "get_forward_sensor_model" function will return an gaussian-based obstacle probability distribution function for the entire map space with respect to the currently measured distance. Complementary implement also the "get_neg_forward_sensor_model".

Task 3

Create a simulation of the mapping process. Generate 5-10 measurement samples and for each of them generate $P(m^o|z)$ distribution using the sensor model and update the probability distribution of the obstacles in the map $P(m^o)$.

Document the simulation of the map building process with several images, especially the final state of the map model.

Submission

Send the report and all related MATLAB scripts at adam.ligocki@vutbr.cz.
MATLAB script must be executable without errors and has to generate all graphical outputs that are in the report.
Deadline: 20th Feb 2022, 23:59.

Assignment

MPC-MAP Assignment No. 1 - Report

Author: Adam Ligocki
Date: 7st Feb 2022

Task 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque faucibus, mi eu pellentesque venenatis, ligula leo tincidunt mauris, in tempus lectus erat eget purus. Fusce quis urna dolor. Phasellus tristique felis justo, vel consectetur magna luctus a. Nulla pharetra magna non pellentesque vestibulum.

Task 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque faucibus, mi eu pellentesque venenatis, ligula leo tincidunt mauris, in tempus lectus erat eget purus. Fusce quis urna dolor. Phasellus tristique felis justo, vel consectetur magna luctus a. Nulla pharetra magna non pellentesque vestibulum.

Figure 1 - Lorem Ipsum

Task 3

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque faucibus, mi eu pellentesque venenatis, ligula leo tincidunt mauris, in tempus lectus erat eget purus. Fusce quis urna dolor. Phasellus tristique felis justo, vel consectetur magna luctus a. Nulla pharetra magna non pellentesque vestibulum.
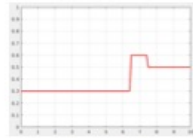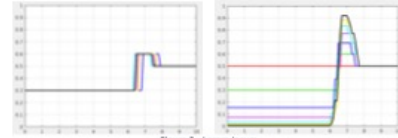
Figure 2 - Lorem Ipsum

Report (placeholder)

Adam Ligocki

adam.ligocki@vutbr.cz

Brno University of Technology
Faculty of Electrical Engineering and Communication
Department of Control and Instrumentation

Robotics and AI
Research Group