2 – Introduction to Probability and Mapping

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

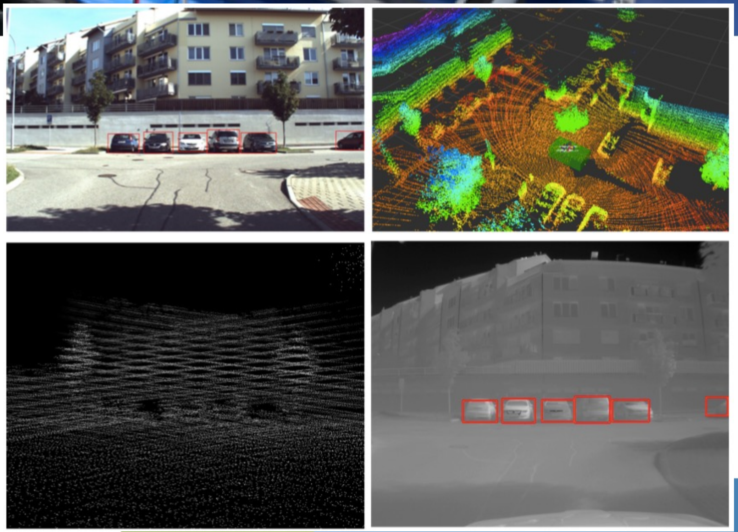
Ing. Adam Ligocki, Ph.D.

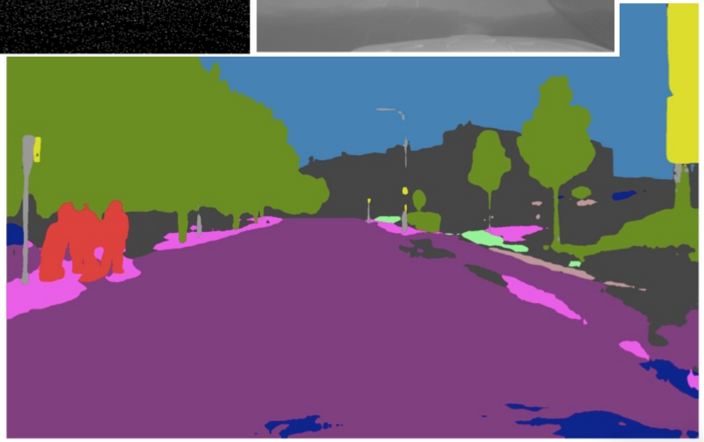
Brno University of Technology
2022

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

Normal Distribution – Basic Definitions

Mean
(cz: střední hodnota)
symbol: μ

Standard Deviation
(cz: Směrodatná odchylka)
symbol: σ

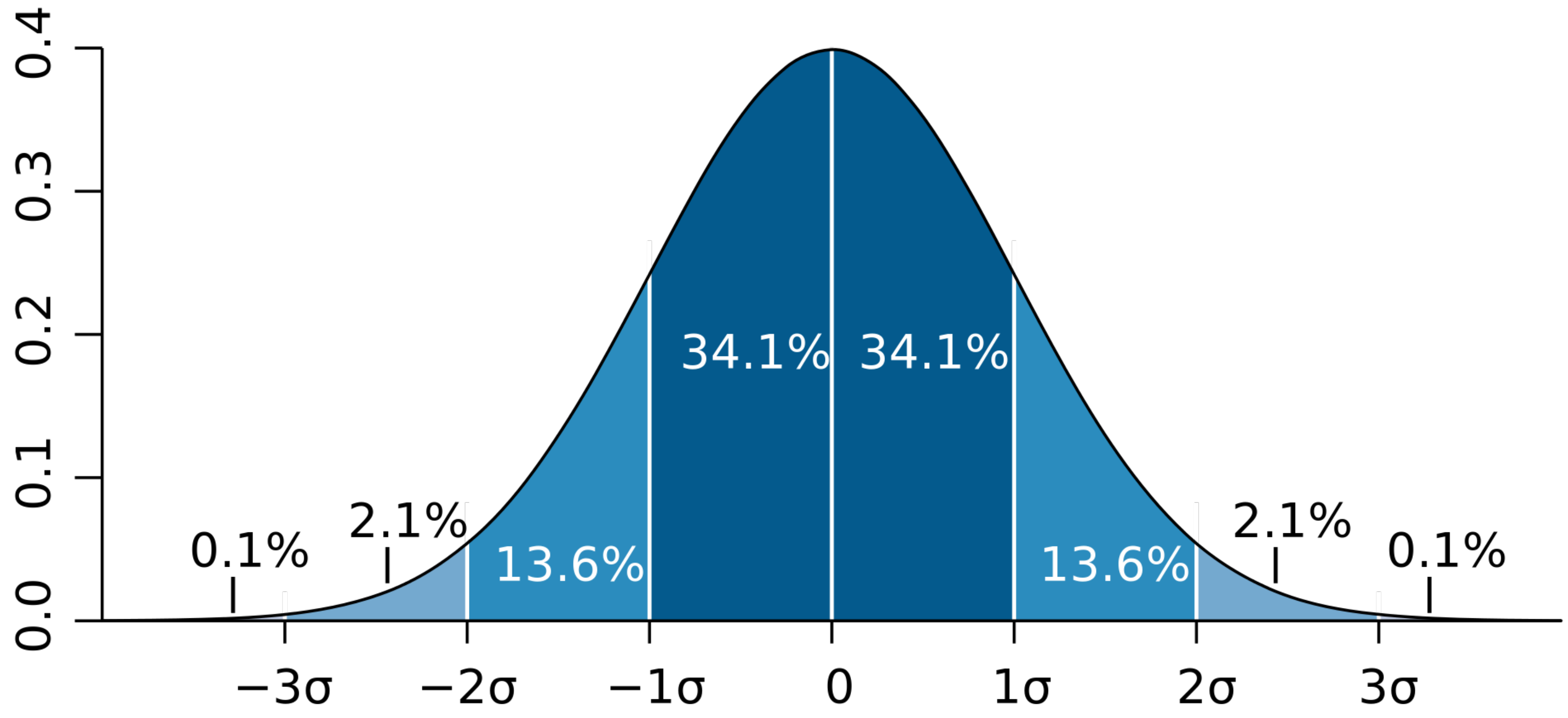
Variance
(cz: Rozptyl)
symbol: σ^2

$$\mu = \frac{\sum_{i=1}^{N} x_i}{N} \qquad \sigma = \frac{\sum_{i=1}^{N} (x_i - \mu)^2}{N}$$

Mean
(cz: střední hodnota)
symbol: μ

Standard Diviation
(cz: Směrodatná odchylka)
symbol: σ

Variance
(cz: Rozptyl)
symbol: σ^2



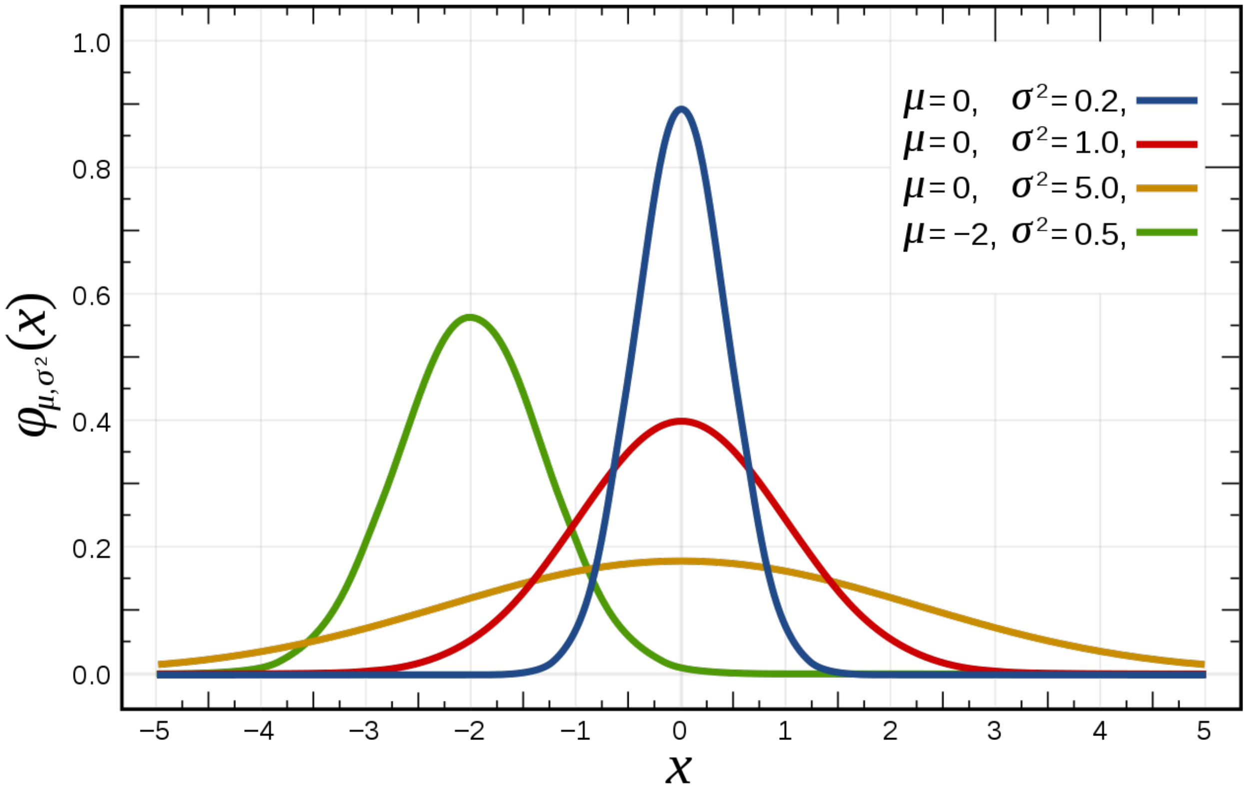
[1]

For μ = 0; σ

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

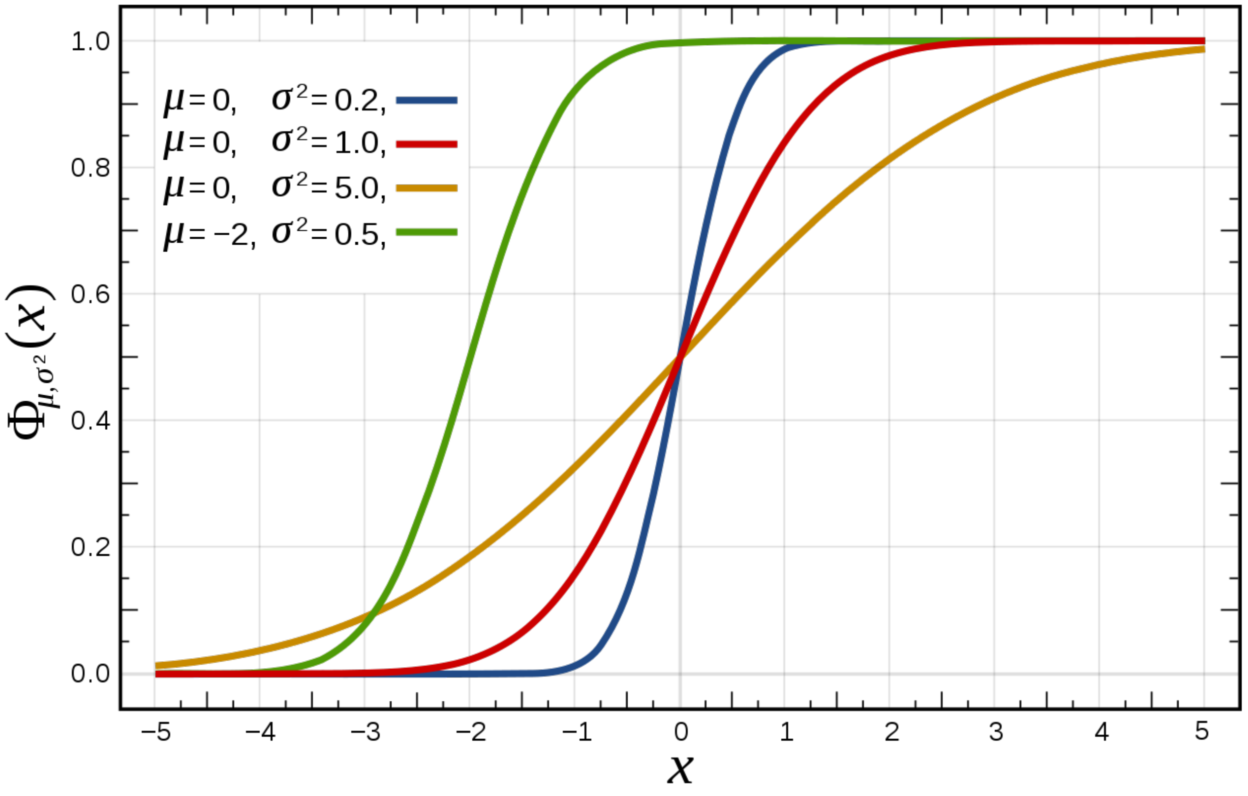
Probability Density Function (PDF)

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



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

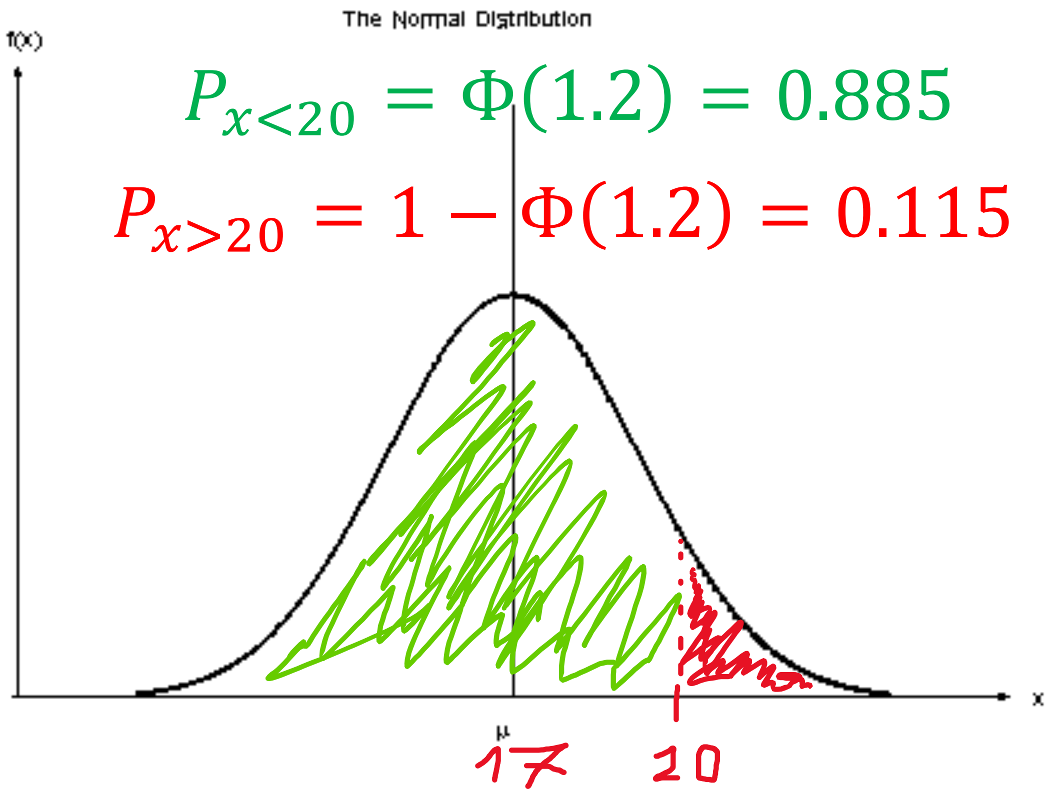
Two fast foods (ff1 and ff2) share their extra burned oil resources. The ff1 consumes 10l of oil per day with std. dev. 2l. The ff2 uses 7l of oil per day with std. dev. 1.5l.

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

$$\mu_1 = 10 \qquad \mu_c = \mu_1 + \mu_2 = 17$$
$$\sigma_1 = 2 \qquad \sigma_c = \sqrt{\sigma_1{}^2 + \sigma_2{}^2} = 2.5$$

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

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



Normal Distribution – Combining Normal Distributions (Independent Variables)

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?

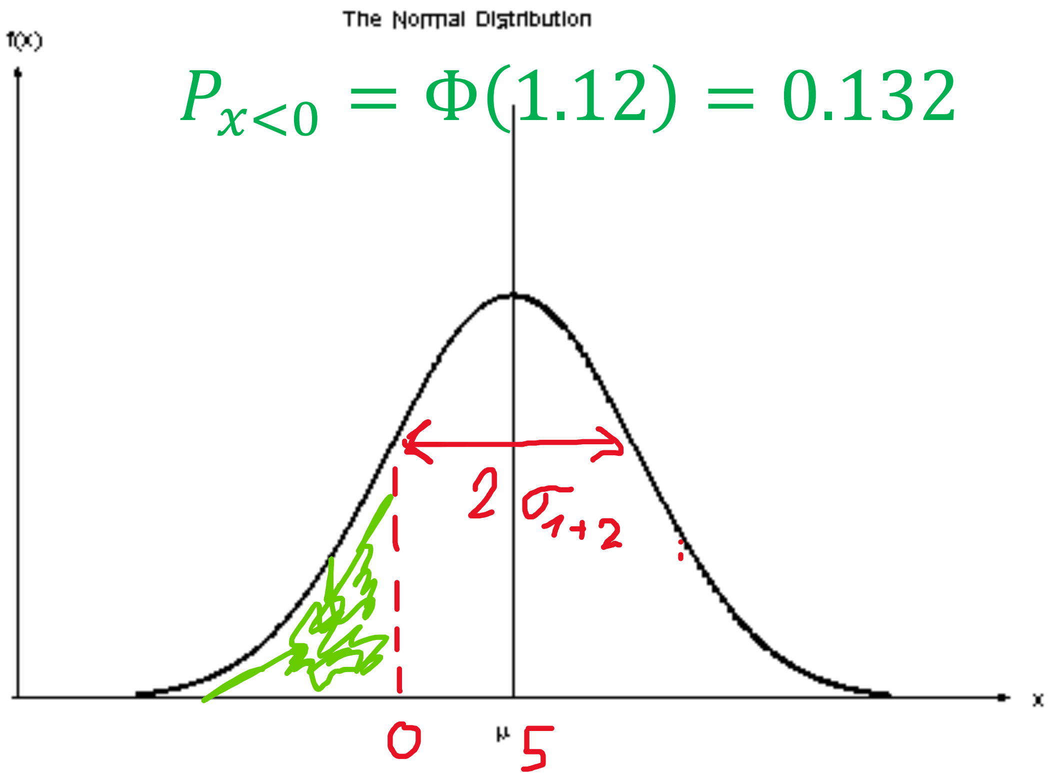
$$\mu_1 = 7$$
$$\sigma_1 = 2$$

$$\mu_c = \mu_2 - \mu_1 = 5$$
$$\sigma_c = \sqrt{\sigma_1{}^2 + \sigma_2{}^2} = 4.47$$

$$\mu_2 = 12$$
$$\sigma_2 = 4$$

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

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



Normal Distribution – Combining Normal Distributions (Fusing Observations)

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/

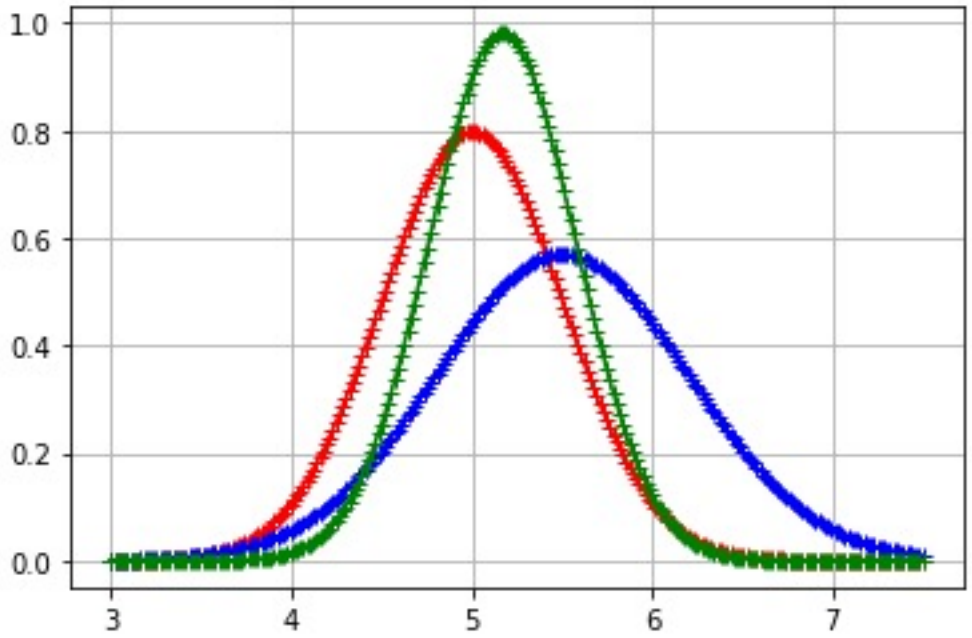
Normal Distribution – Combining Normal Distributions (Fusing Observations)

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

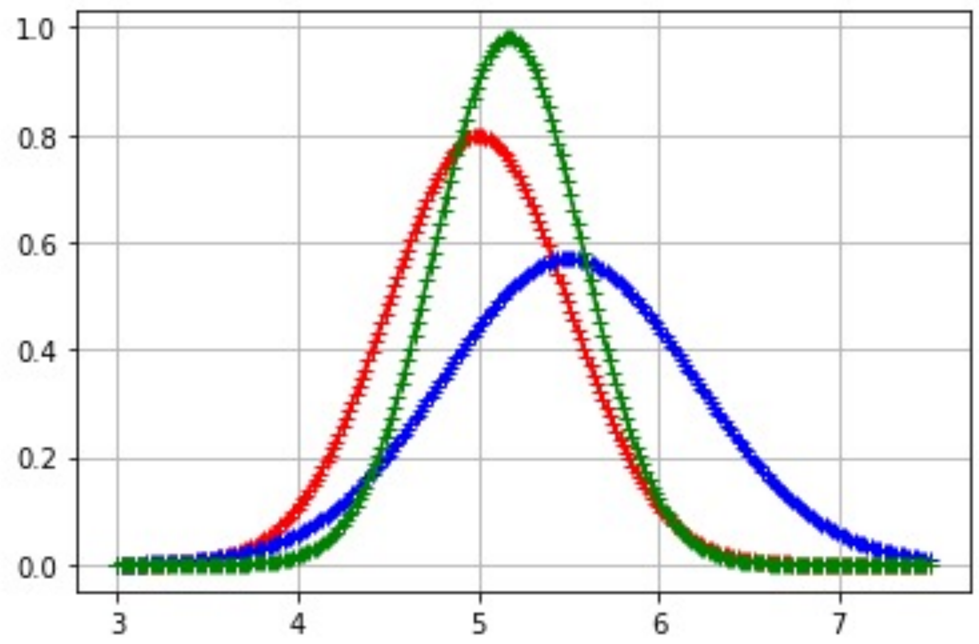
The second sensor measured the distance of 5.5m with a variance of 0.7m.

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



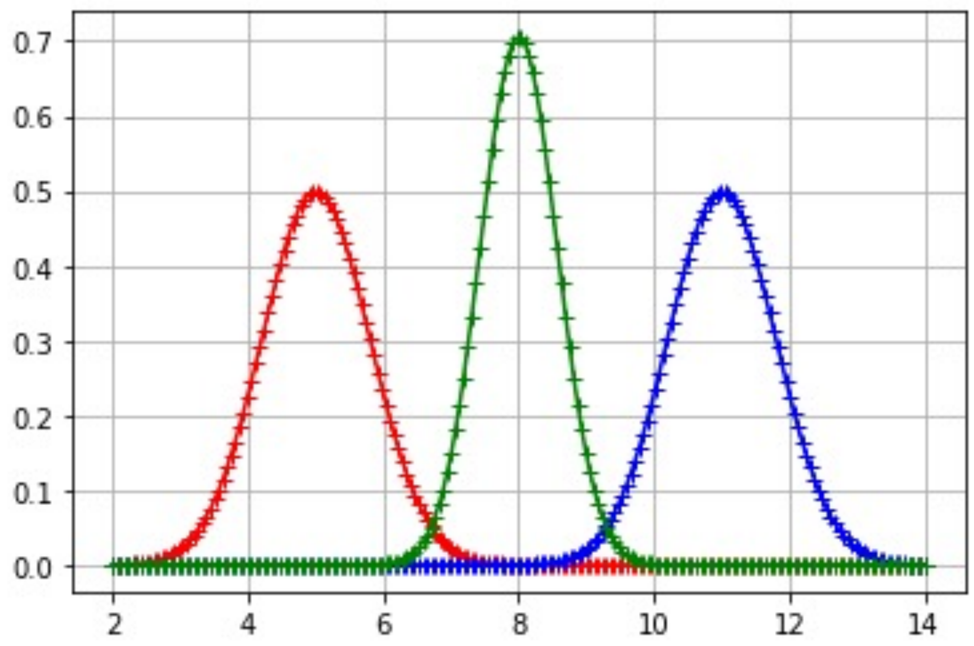
Very nice example: https://datascopeanalytics.com/blog/sensor-fusion-tutorial/

Normal Distribution – Combining Normal Distributions (Fusing Observations)



$\mu_1 = 5.0$ $\mu_2 = 5.5$ $\mu_c = 5.17$

$\sigma_1 = 0.5$ $\sigma_2 = 0.7$ $\sigma_c = 0.41$



$\mu_1 = 5$ $\mu_2 = 11$ $\mu_c = 8.0$

$\sigma_1 = 0.8$ $\sigma_2 = 0.8$ $\sigma_c = 0.56$

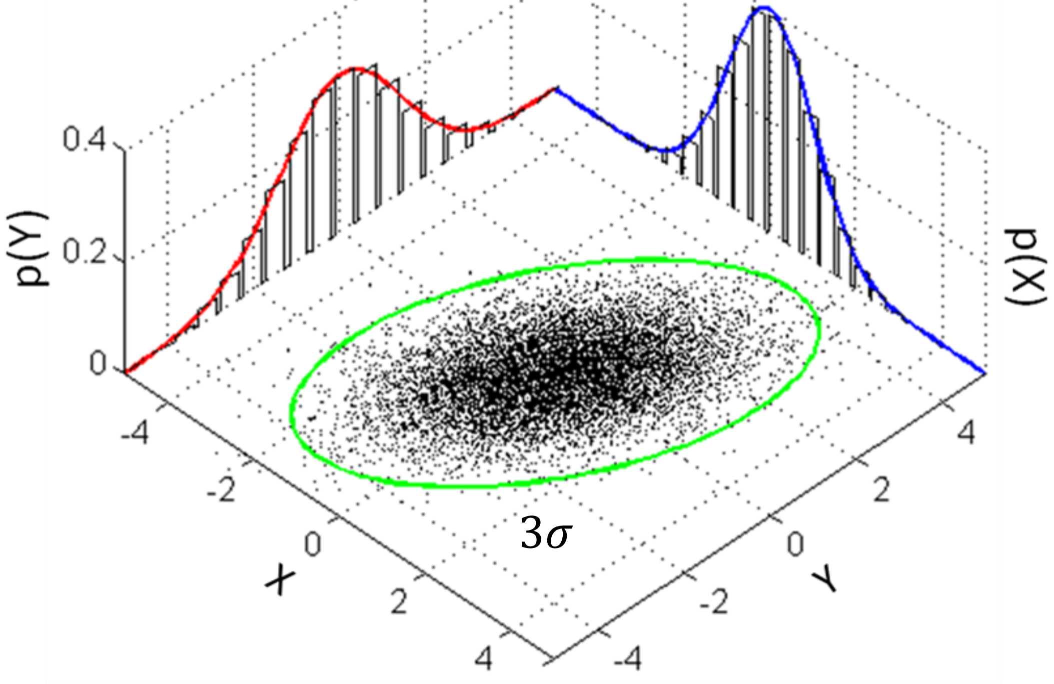
Very nice example: https://datascopeanalytics.com/blog/sensor-fusion-tutorial/

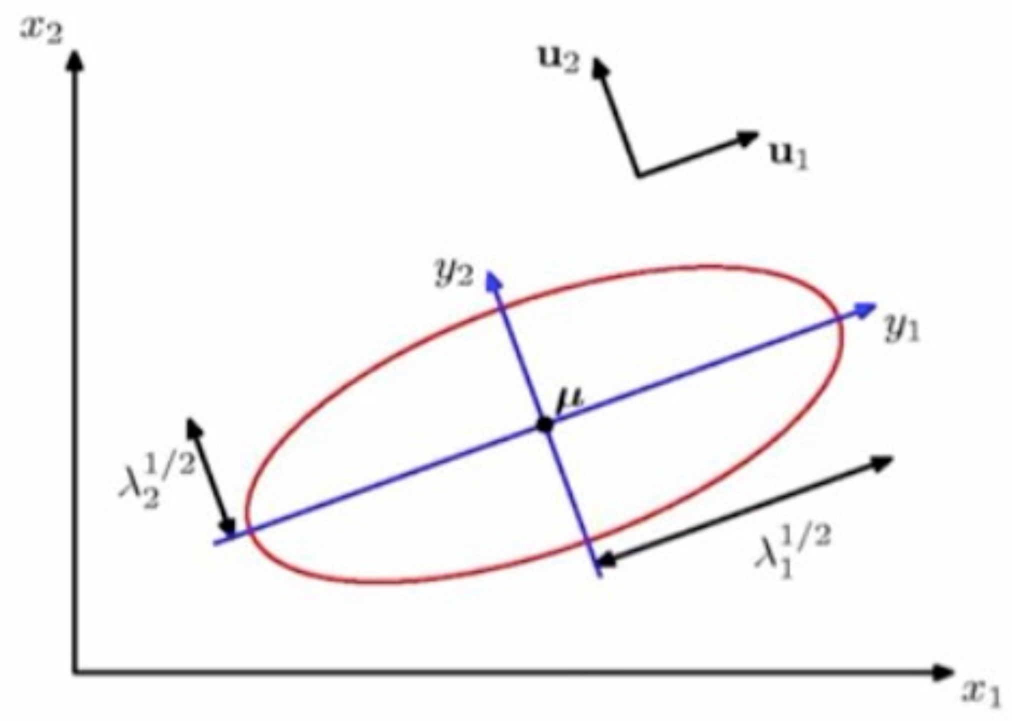
2D Normal Distribution

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

$$\Sigma = \begin{bmatrix} 1 & 3/5 \\ 3/5 & 2 \end{bmatrix}$$

$$\Sigma = U\Lambda U^T = \begin{bmatrix} u_1 & u_2 \\ u_1 & u_2 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} u_1 & u_1 \\ u_2 & u_2 \end{bmatrix}$$

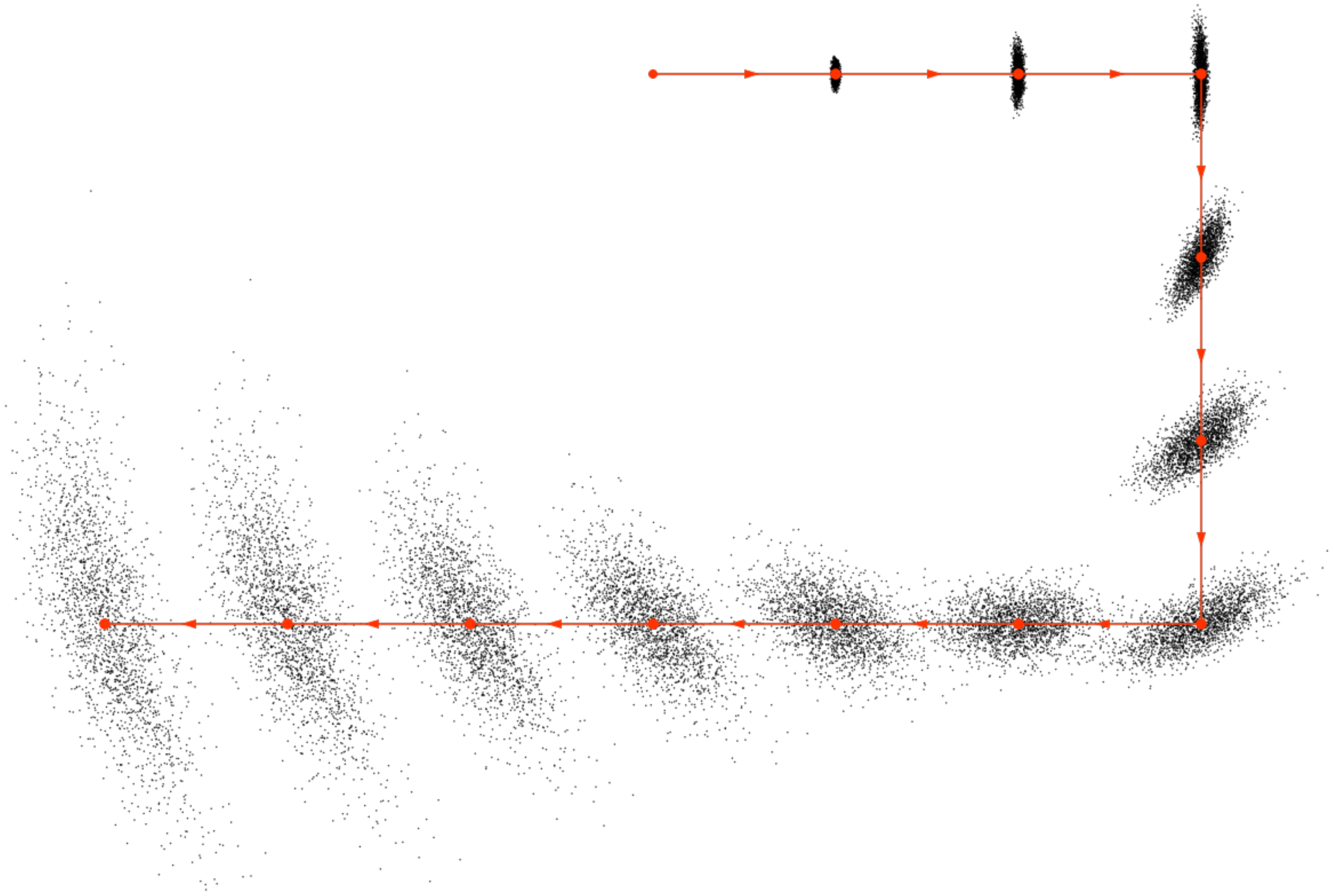


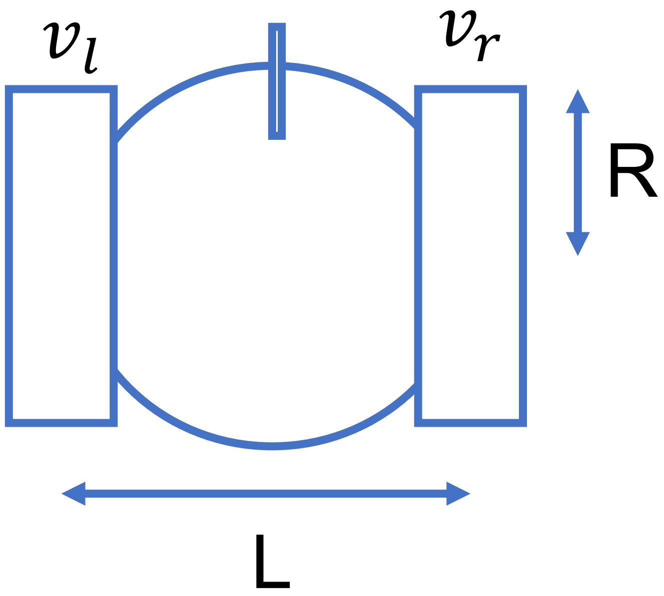


[1] https://en.wikipedia.org/wiki/Multivariate_normal_distribution
[2] https://www.youtube.com/watch?v=eho8xH3E6mE

2D Normal Distribution – Example of Usage

Growing unsertancy of the position of the moving robot





$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{R}{2}(v_l + v_r) \\ \frac{R}{L}(v_r - v_l) \end{bmatrix}$$

[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

Bayes Theorem

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

H ... Hypothesis

E ... Evidence

$P(H)\ldots$ 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

Bayes Theorem

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

H ... Hypothesis: ☀️

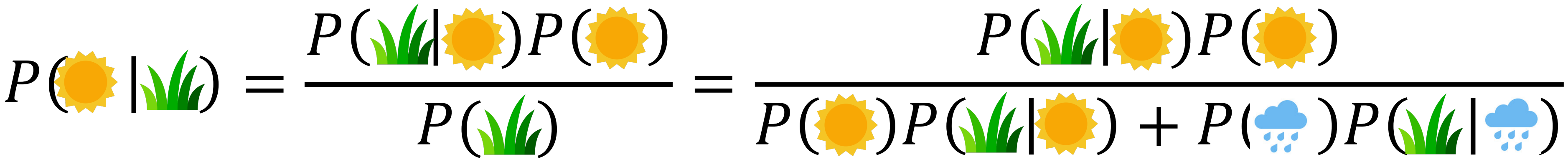
E ... Evidence: 🌿

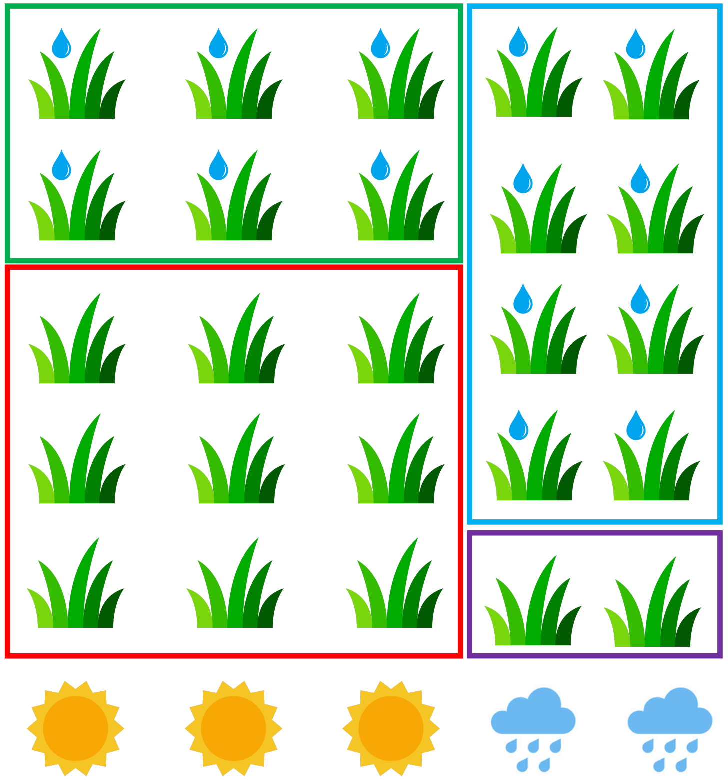
$\neg H$ ... Hypothesis: 🌧️

$\neg E$ ... Evidence: 💧🌿

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

Bayes Theorem





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

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

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

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

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

*

* 2/25 =0.08

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

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

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

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

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

Bayes Theorem

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

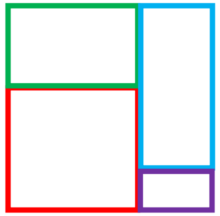
$$P(\text{☀}) = 0.6$$

$$P(\text{🌧}) = 0.4$$

$$P(\text{🌱}|\text{☀}) = 0.6$$

$$P(\text{🌱}|\text{🌧}) = 0.2$$

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



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

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

Bayes Theorem

Likelihood

Posterior

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

Prior

Evidence

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

Sensor Model

New Information (Inverse Sensor Model)

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

Old Information

Normalization

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

Bayes Filter

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

Sensors

Sensors: Range-bearing sensor

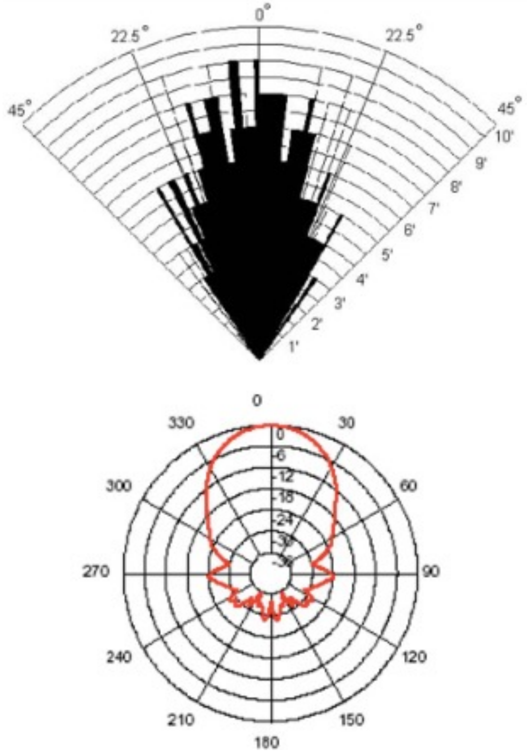
Typically:

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.

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


2D LiDAR scan



Ultrasound sensor characteristics

Sensors: Camera

Camera Sensor

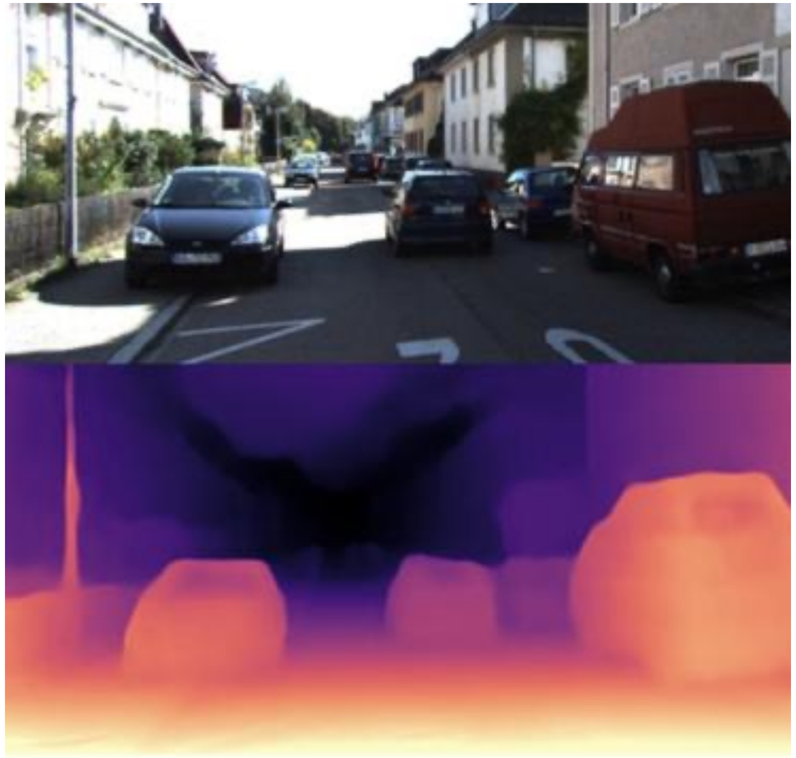
Sensing usually “2D” data represented as an image

Types of camera sensors:

- Common RGB/grayscale camera
- Multispectral Camera
- RGBD camera (kinect)
- IR camera

IR Image

RGB Image

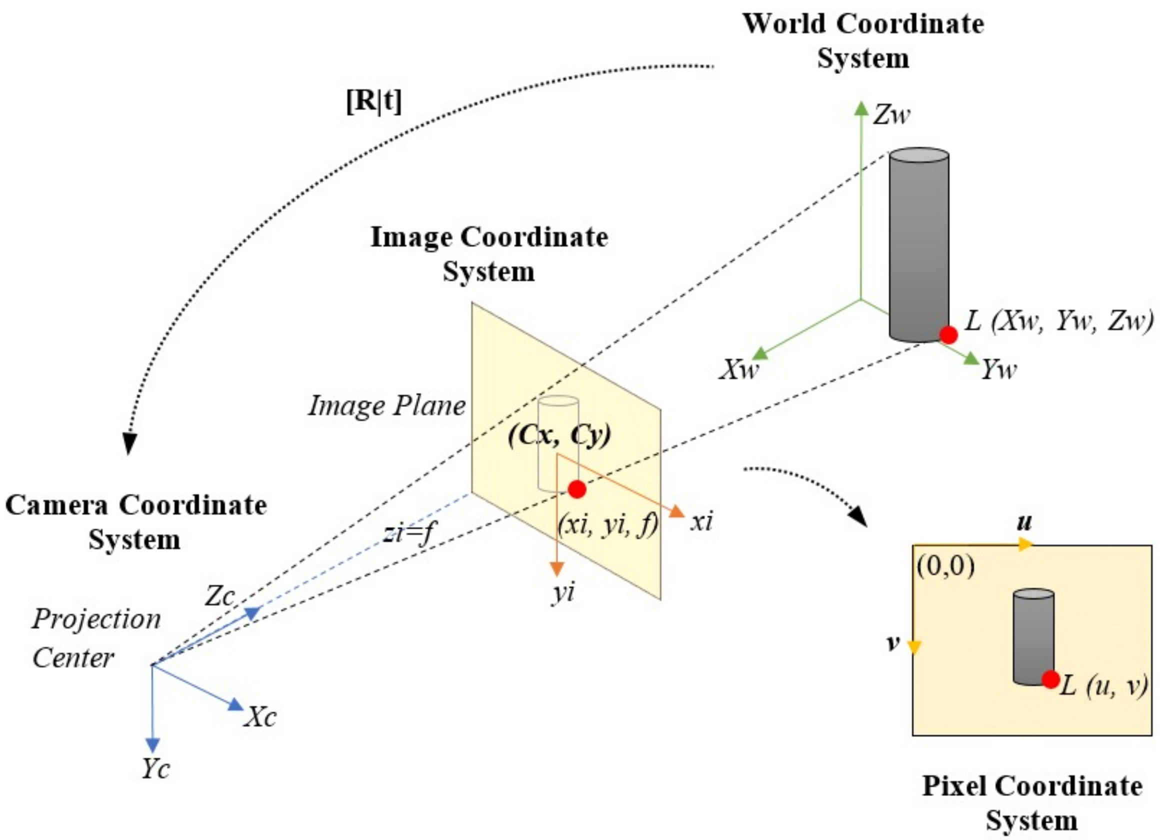


RGB-D Image: RGB+Depth

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

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

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



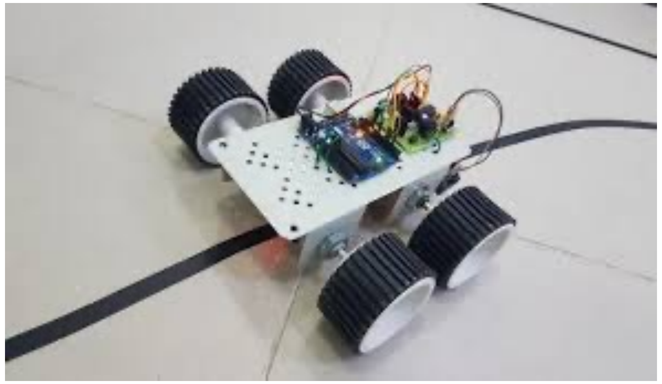
https://www.researchgate.net/profile/Luis-Ortiz-25/publication/326518096

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: $\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} * s = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$

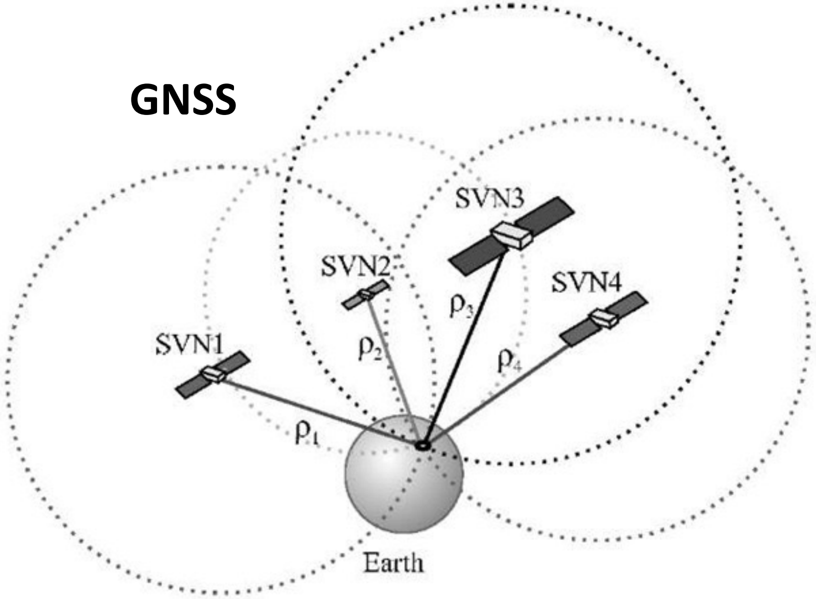
Sensors: External Pose Measurement (GNSS, motion capture, ...)

Line Follower



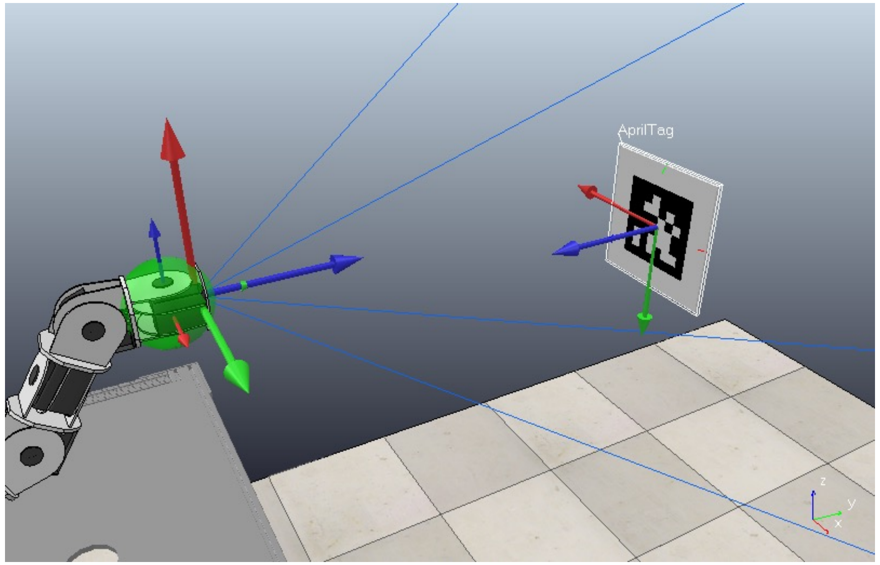
electronicshub.com

GNSS



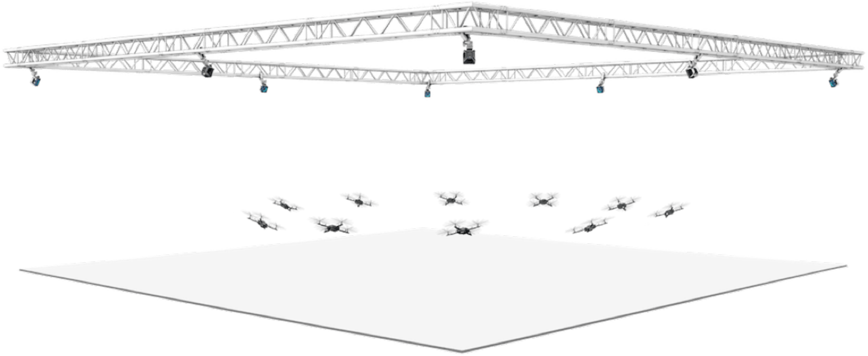
http://what-when-how.com/

AR/QR/April Tags

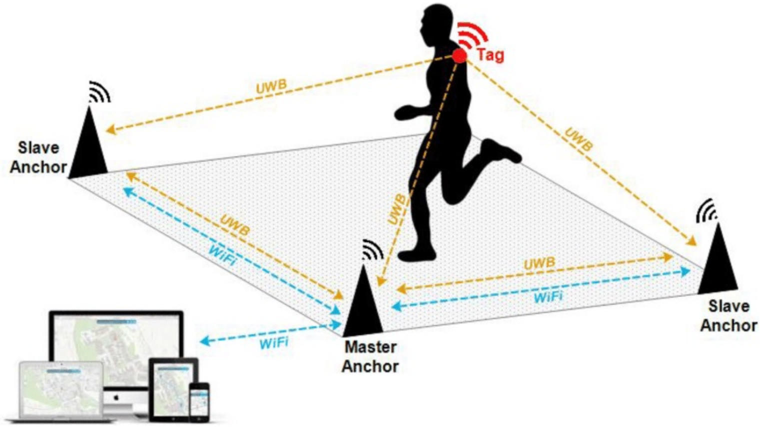


https://haowensh.com/assets/img/2020/kdc/frame_confirmation.png

Motion Capture



https://deva90sapmc8w.cloudfront.net/roboticsPrime17w_8-450h.png



Beacon loclalization (UWB, BT, ...)

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

Map Types

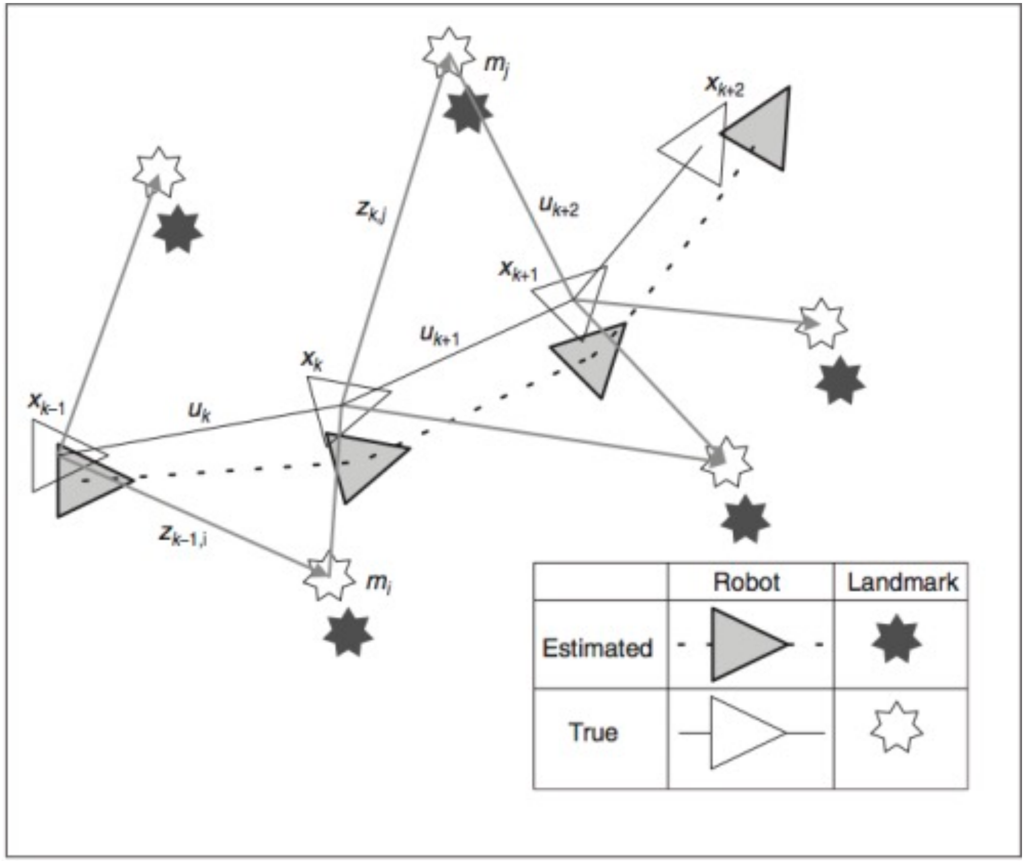
Map Types – Topological Map (Graph)

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)

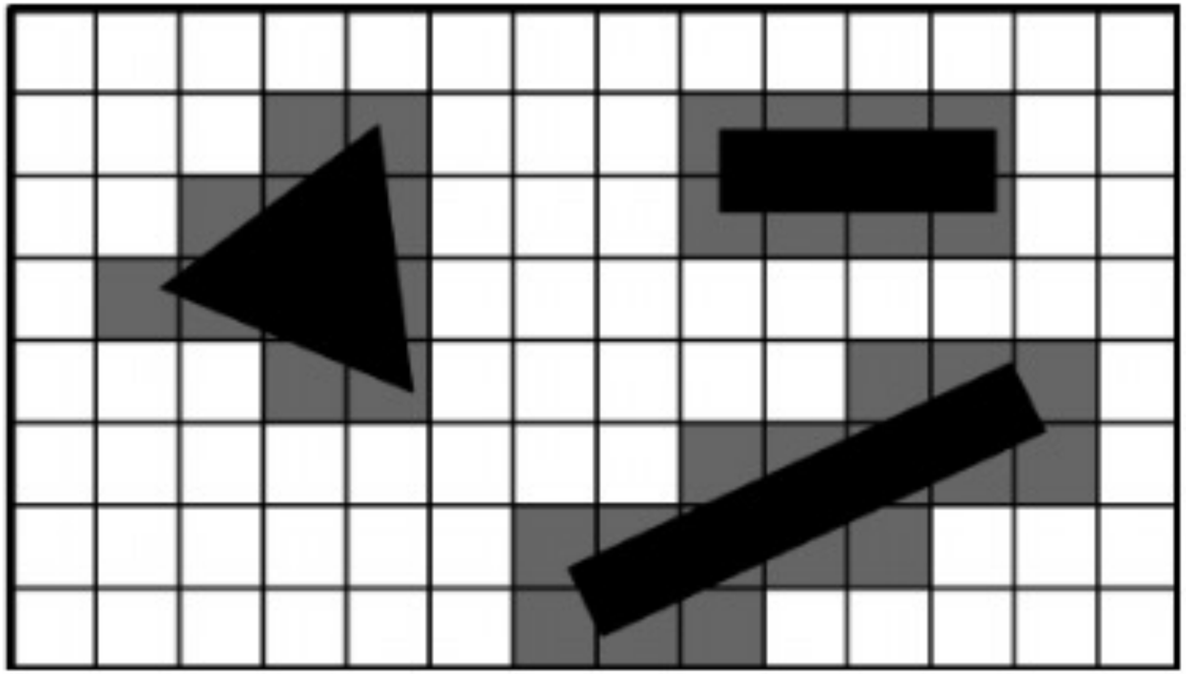


[1] https://jay.tech.blog/2017/03/18/graph-slam/

Map Types – Occupancy Grid

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.



$\mathrm{p}(M^O)$ ... probability, cell is Occupied

$\mathrm{p}(M^E)$ ... probability, cell is Empty

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

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

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

$$0 \leq \mathrm{p}(M^O) \leq 1$$

$$0 \leq \mathrm{p}(M^E) \leq 1$$

$$\mathrm{p}(M^O) + \mathrm{p}(M^E) = 1$$

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

Sensor Modeling

Sensor Modeling

Sensor Model

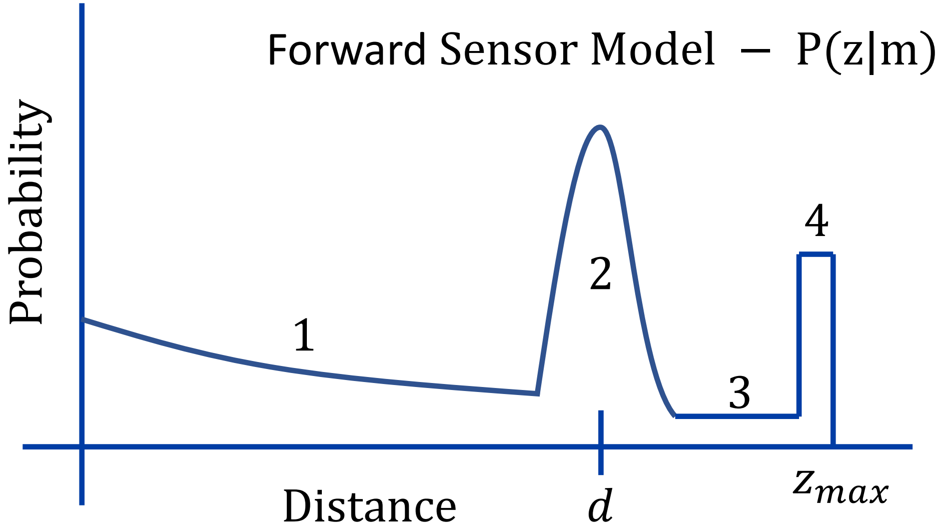
How I think the world looks like → Sensor Model → What I expect to measure

Inverse Sensor Model

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

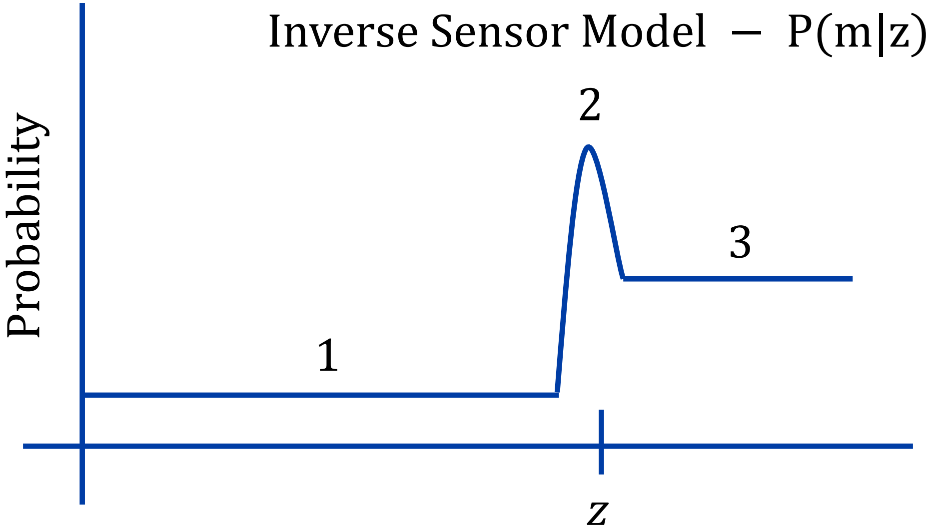
Common Laser Sensor Model:
Probability of getting measurement of an object at distance *d*.
z_{max} is sensor's max range.

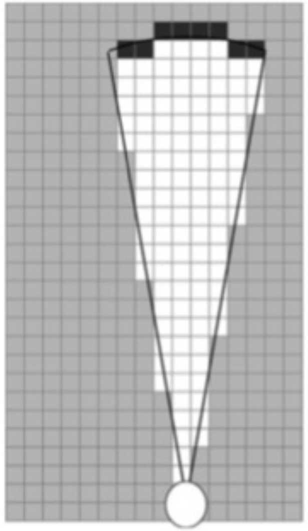
1 ... Approximation of dynamic obstacles (exp decay)
2 ... Distribution around obstacle (normal dist.)
3 ... Unknown area (uniform dist.)
4 ... Max sensor's range



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)





ISM 2D

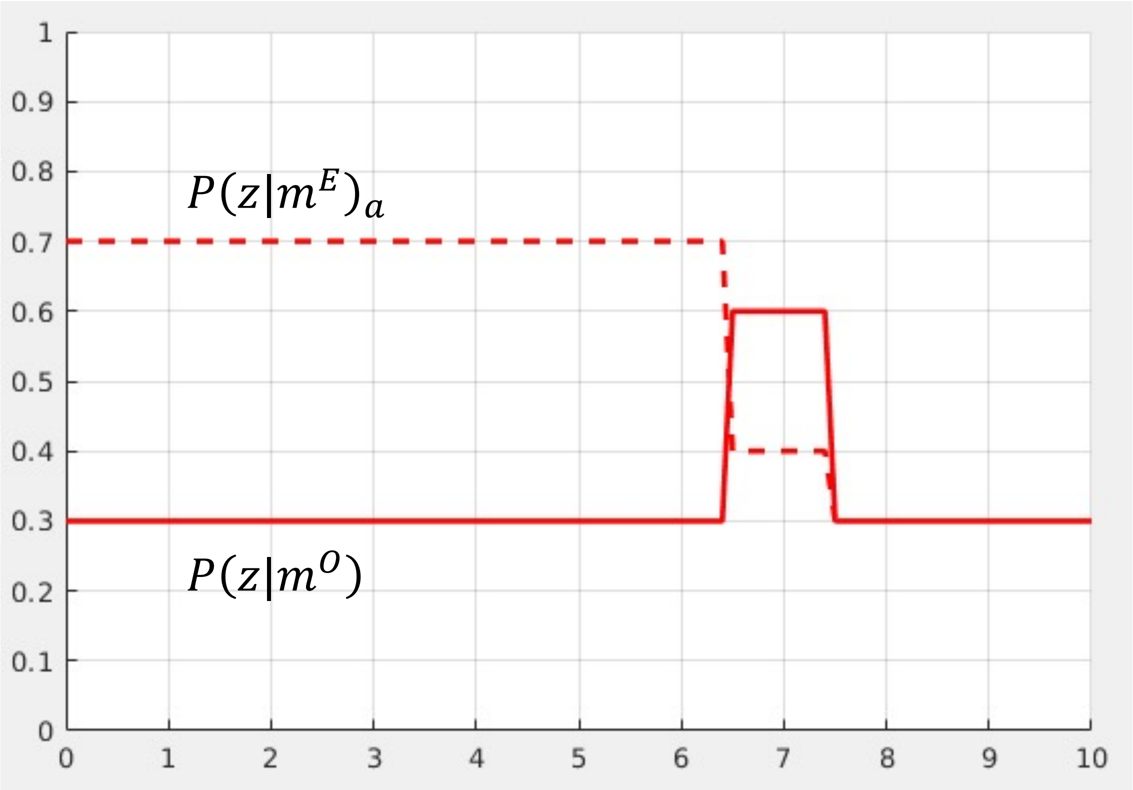
Sensor Modeling – Use Case

Measurement $z = 7m$

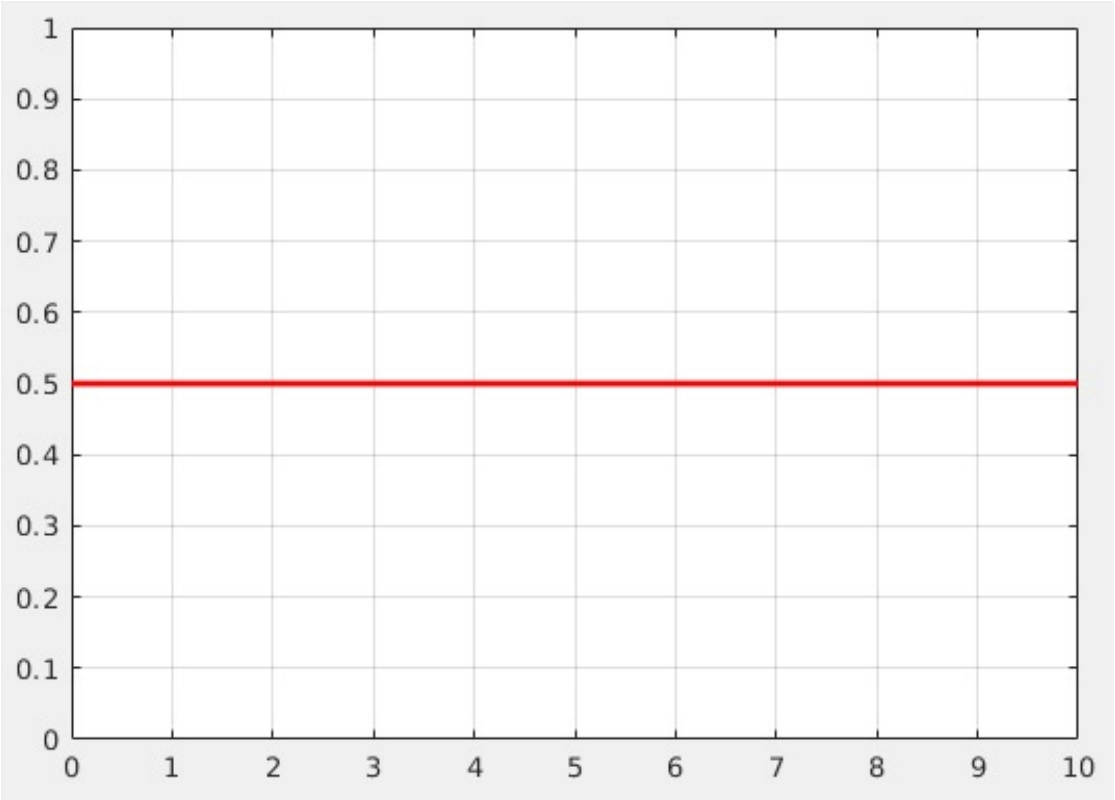
$P(z|m^O)_a$: probabiliy to measure that cell a is occupied

$P(z|m^E)_a$: probabiliy to measure that cell a is empty

$P(z|m^O)_a \neq 1 - P(z|m^E)_a$



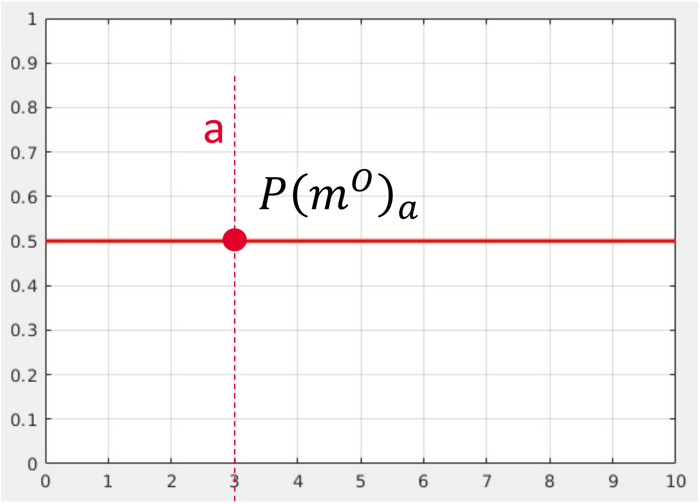
Forward Sensor Models $-$ $\mathrm{P}(z|m^O)$



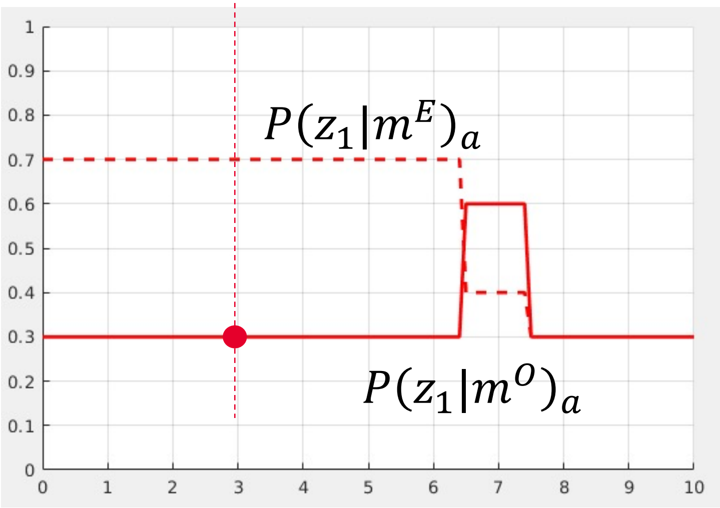
Initial map $-$ $\mathrm{P}(m^O)$

Sensor Modeling – Use Case

$Measurement\ z = 7m$



Map



Forward Sensor Model

0) Init

$P(m^O)_a = 0.5$ -> $P(m^E)_a = 1 - P(m^O)_a = 0.5$

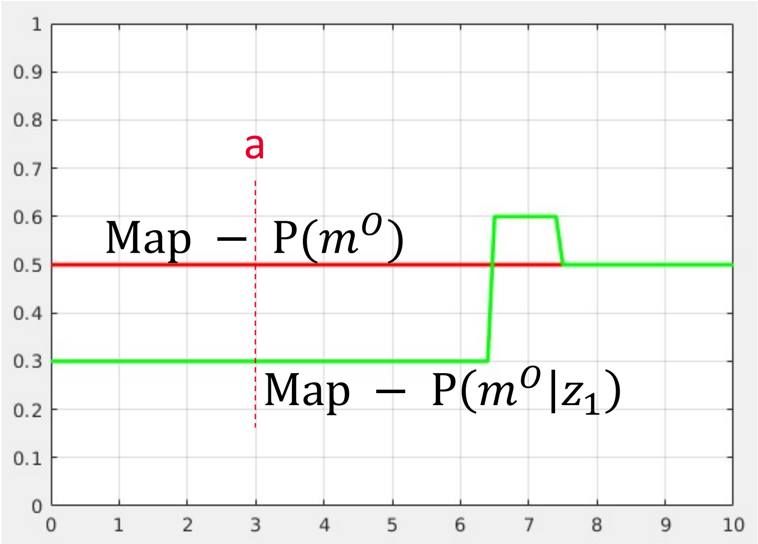
1) First measurement

$P(m^O)_a = 0.5$ -> $P(m^E)_a = 1 - P(m^O)_a = 0.5$

$P(z_1|m^O)_a = 0.3$ $P(z_1|m^E)_a = 0.7$

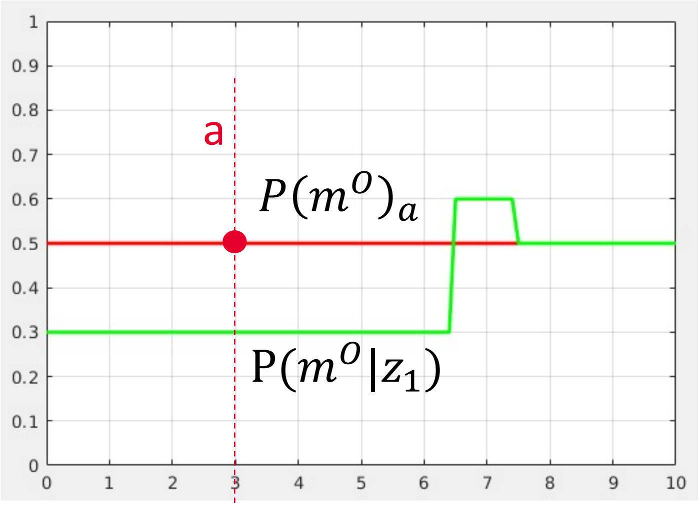
$$P(z)_a = P(z_1|m^O)_a * P(m^O)_a + P(z_1|m^E)_a * P(m^E)_a =$$
$$= 0.3 * 0.5 * 0.7 * 0.5 = 0.5;$$

$$P(m^O|z_1)_a = \frac{P(z_1|m^O)_a * P(m^O)_a}{P(z)_a} = \frac{0.3*0.5}{0.5} = 0.3$$

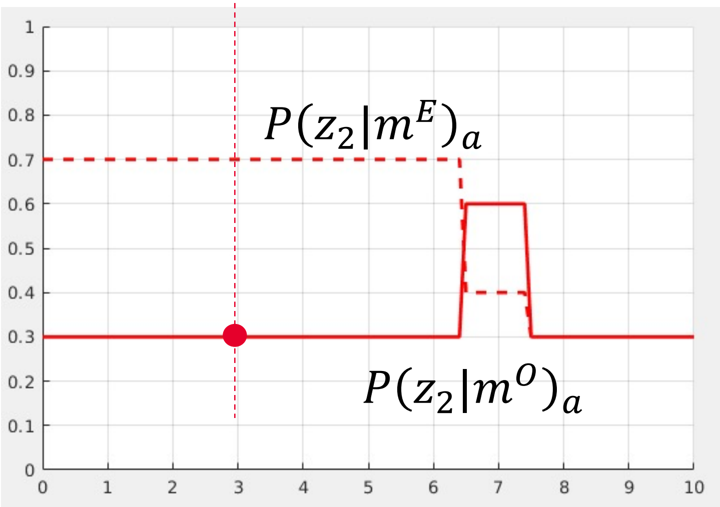


Sensor Modeling – Use Case

$Measurement\ z = 7m$



Map



Forward Sensor Model

2) Second measurement

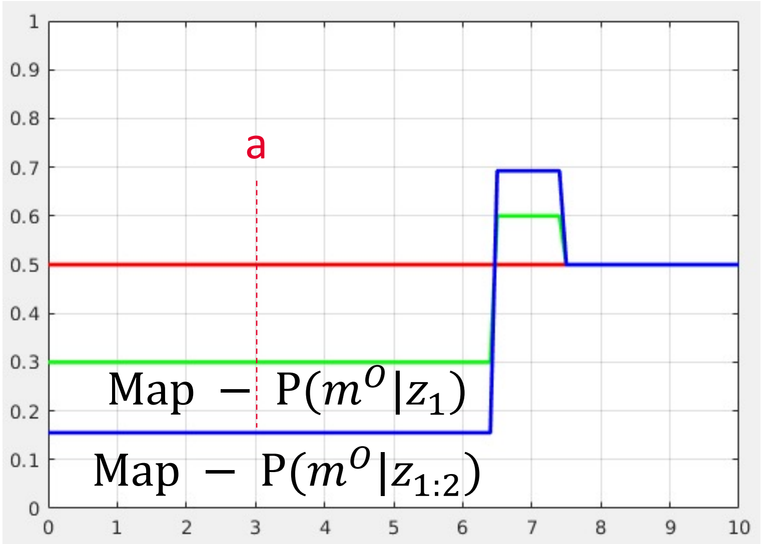
$$P(m^O)_a \sim p(m^O|z_1) = 0.3 \quad \text{->} \quad P(m^E)_a \sim p(m^E|z_1) = 1 - p(m^O|z_1) = 0.3$$

$$P(z_2|m^O)_a = 0.3 \qquad P(z_2|m^E)_a = 0.7$$

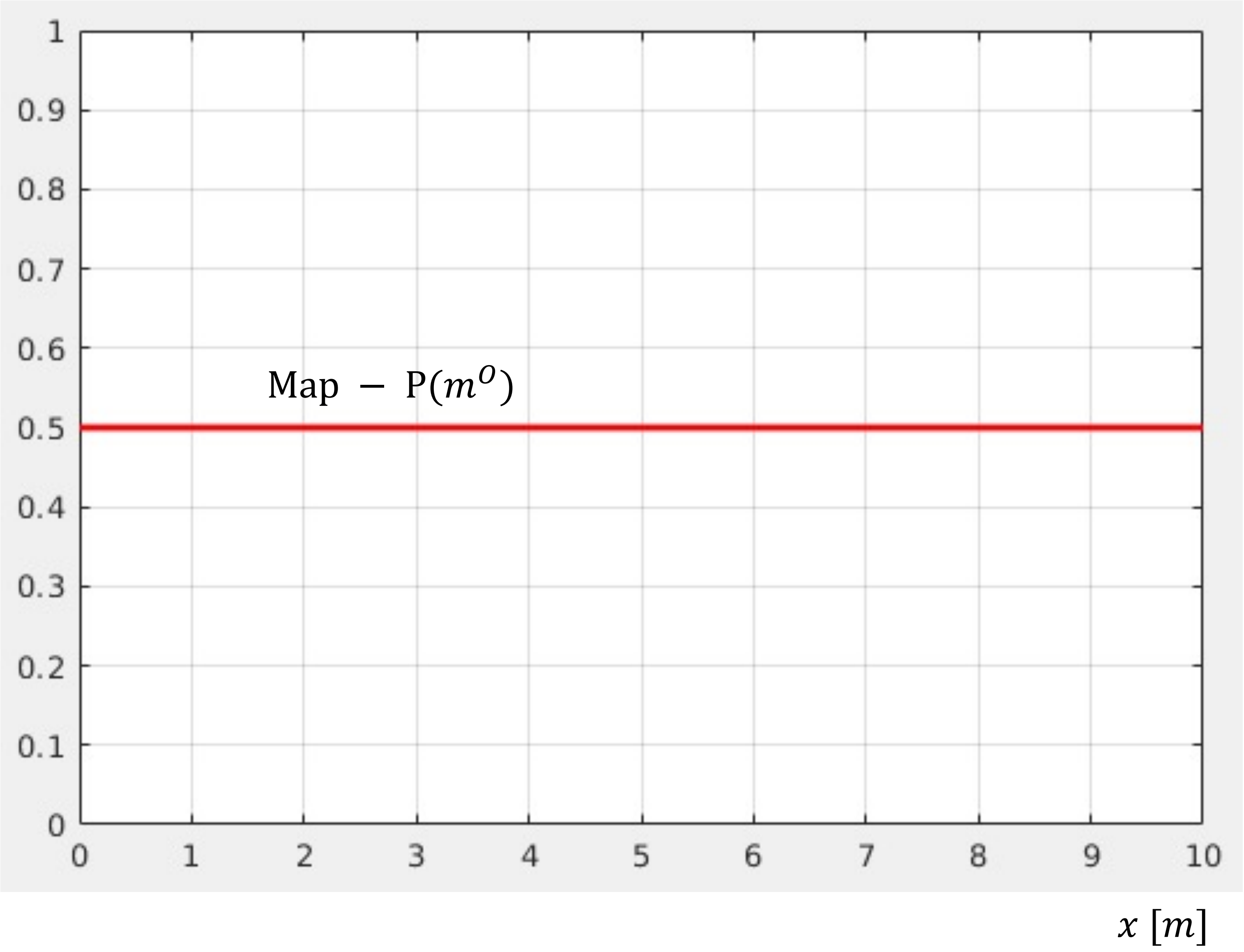
$$P(z)_a = P(z_2|m^O)_a{}^* \, P(m^O)_a + P(z_2|m^E)_a{}^* \, P(m^E)_a =$$

$$= 0.3 * 0.3 * 0.7 * 0.7 = 0.58;$$

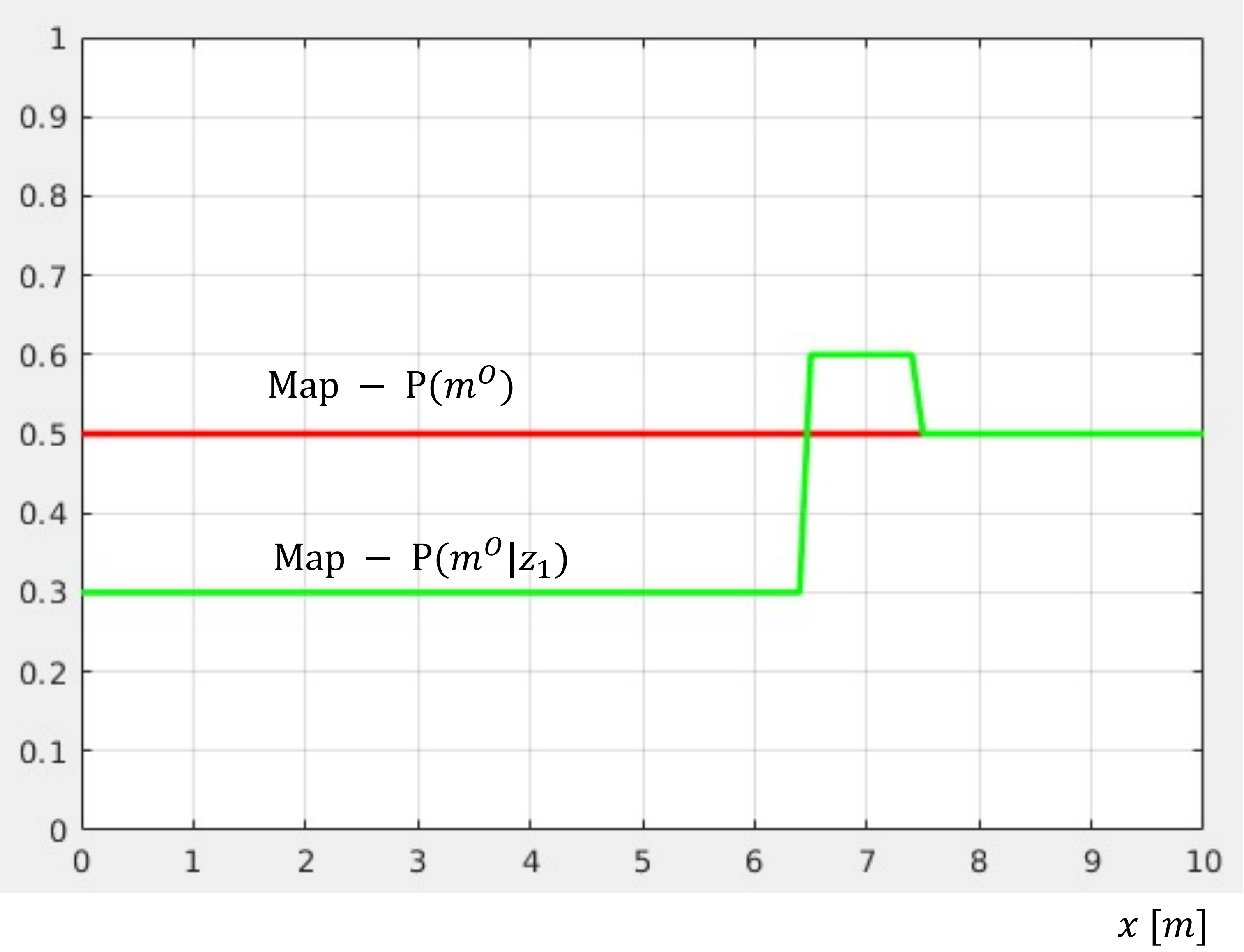
$$P(m^O|z_{1:2})_a = \frac{P(z_2|m^O)_a * P(m^O)_a}{P(z)_a} = \frac{0.3*0.3}{0.58} = 0.155$$



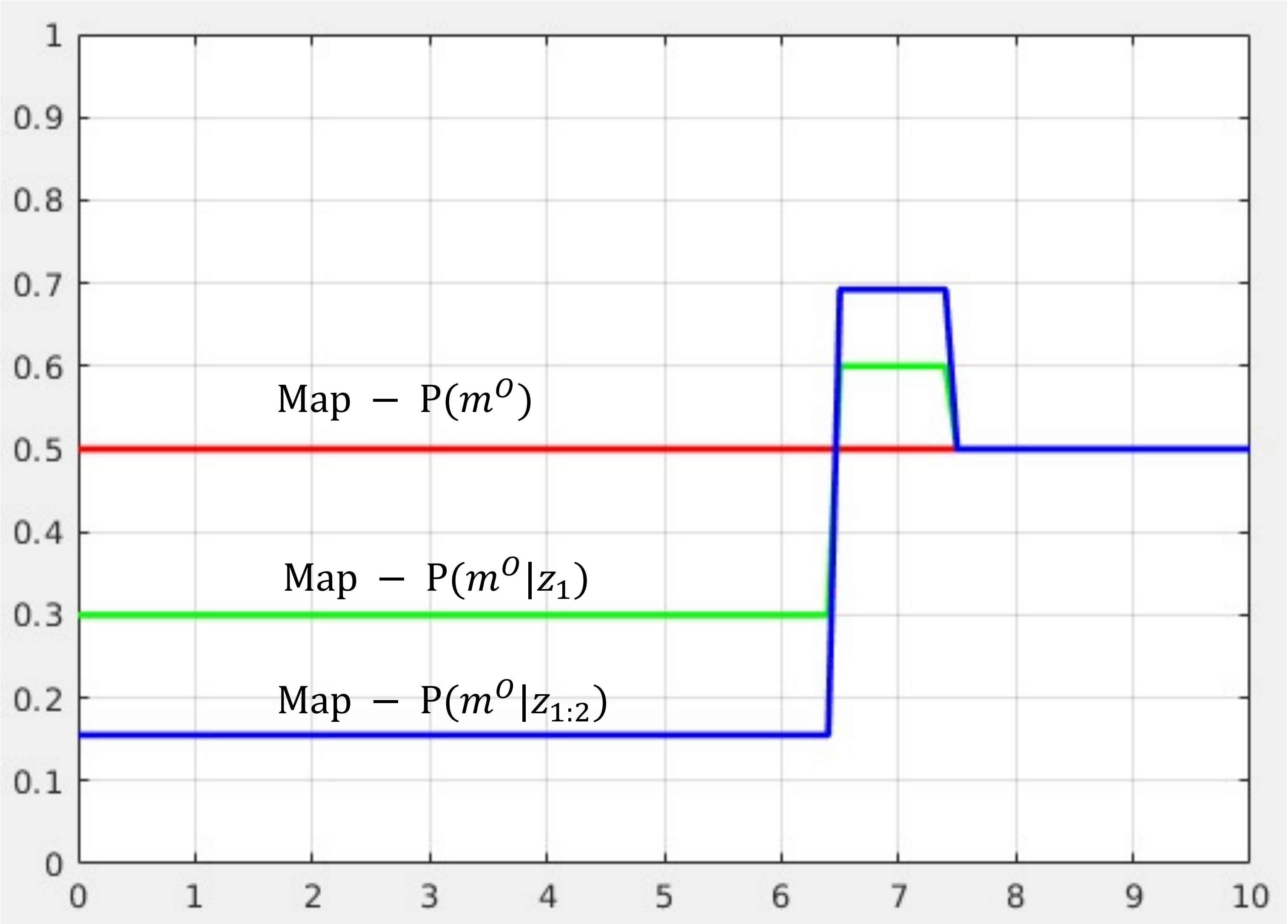
Sensor Modeling – init



Sensor Modeling – z_1

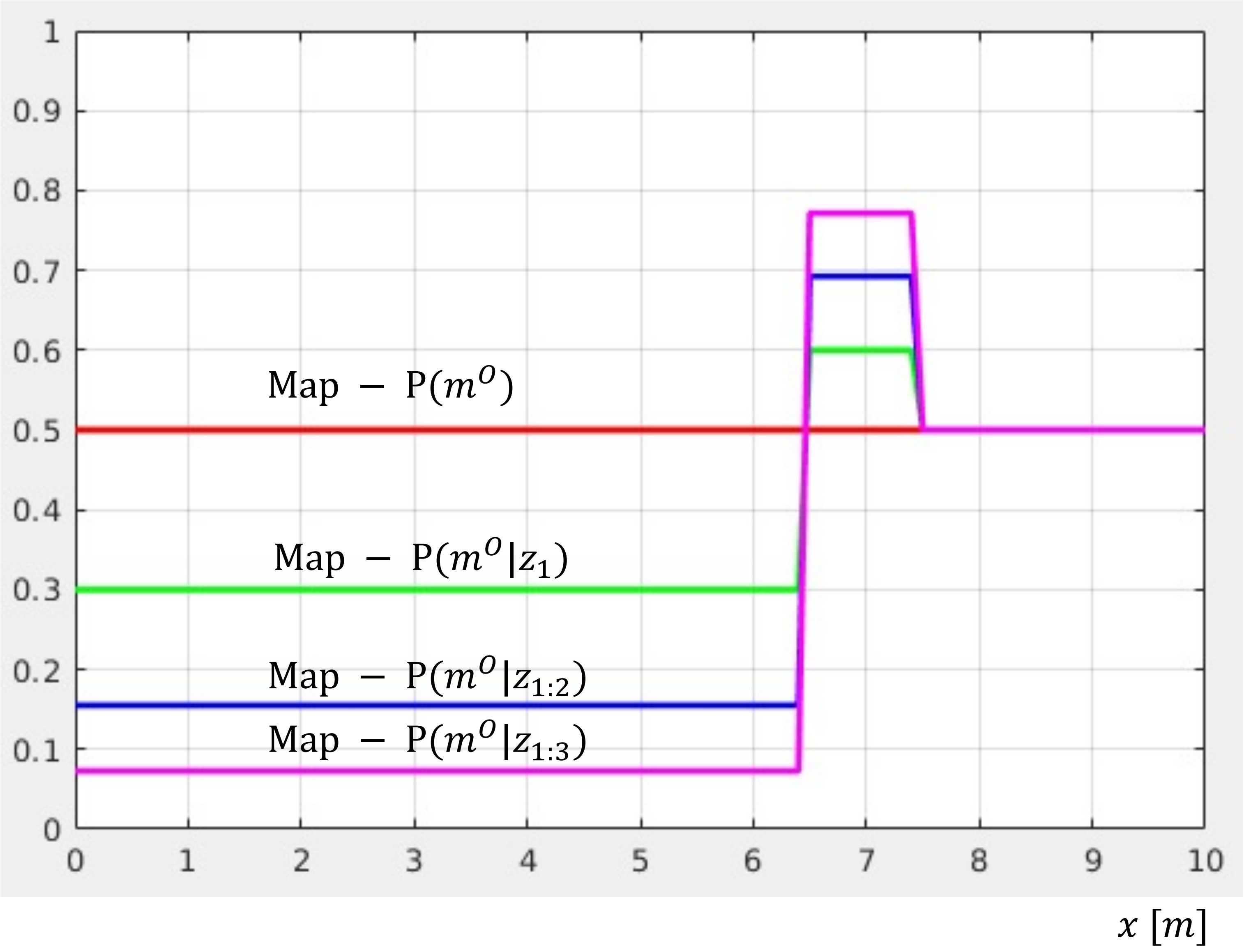


Sensor Modeling – z_2

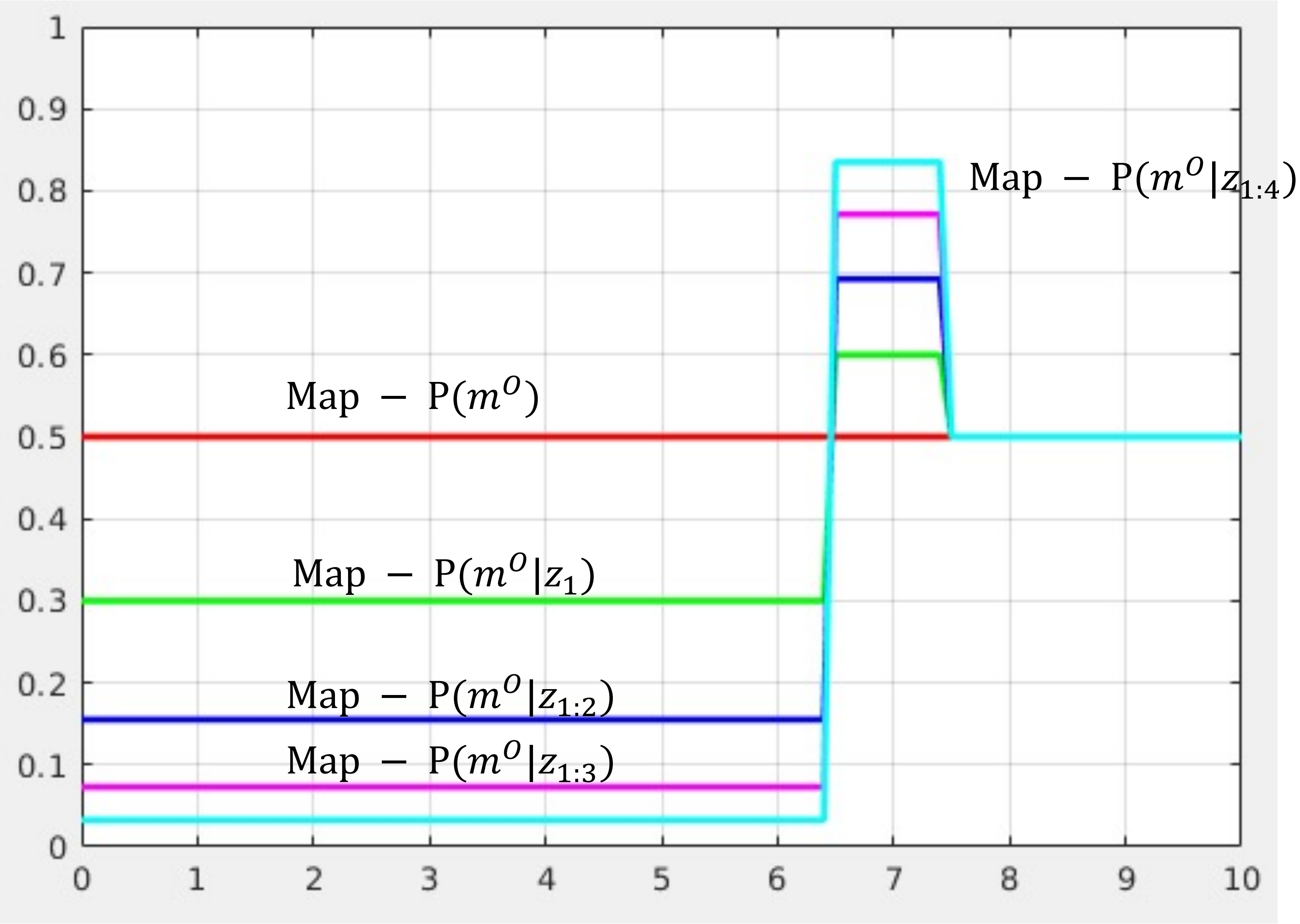


$x\ [m]$

Sensor Modeling – z_3

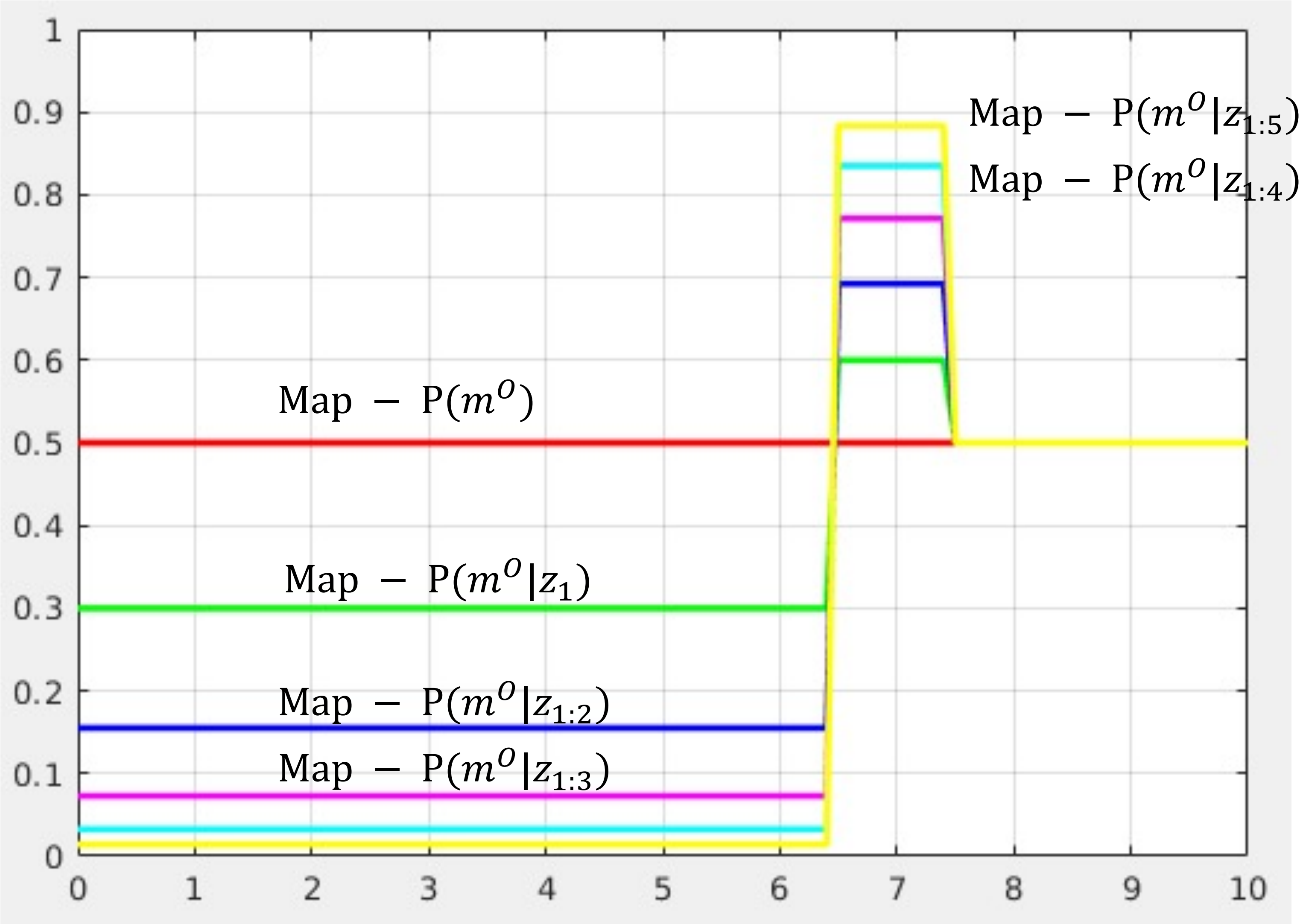


Sensor Modeling – z_4



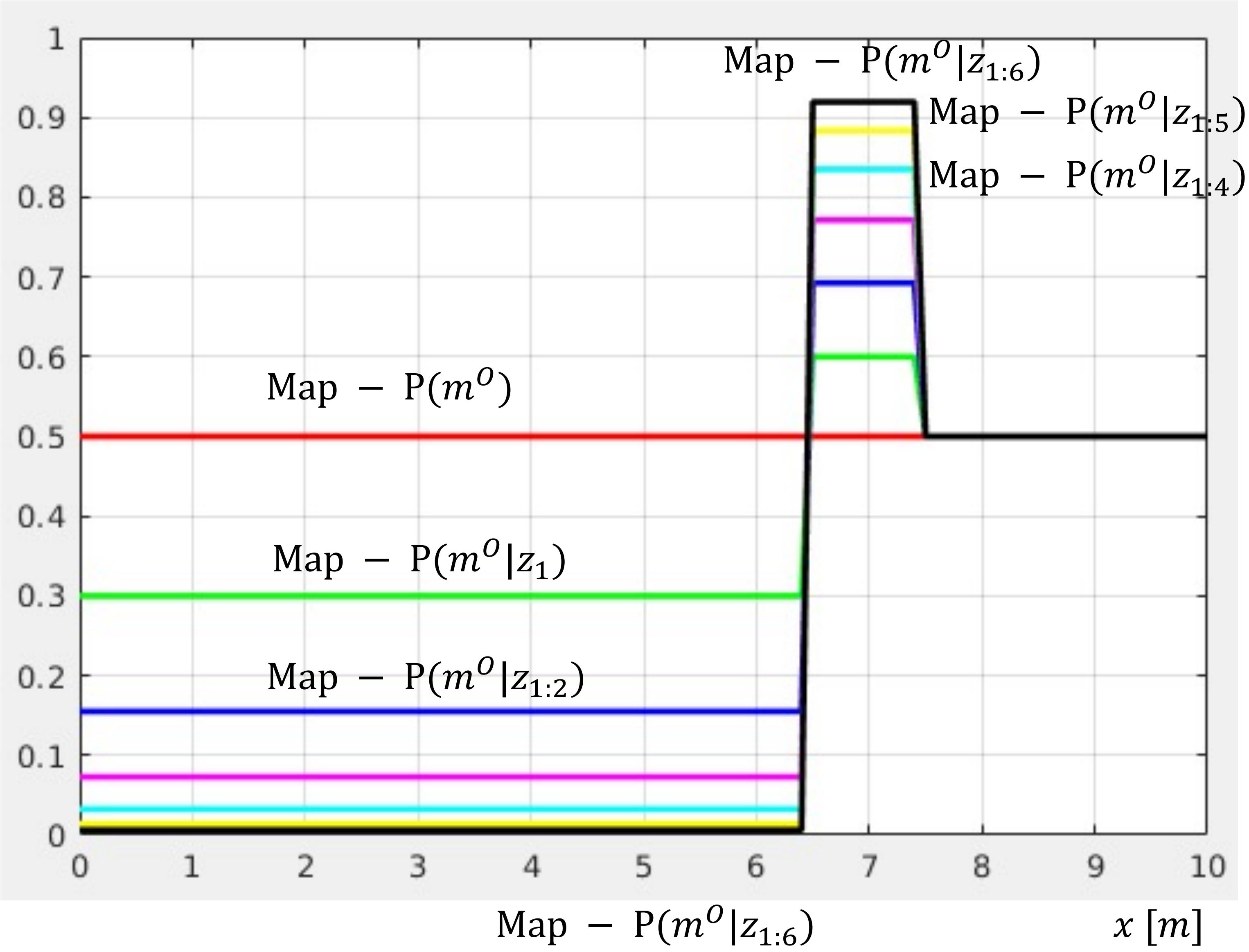
Map – $P(m^O|z_{1:4})$ $x\ [m]$

Sensor Modeling – z_5

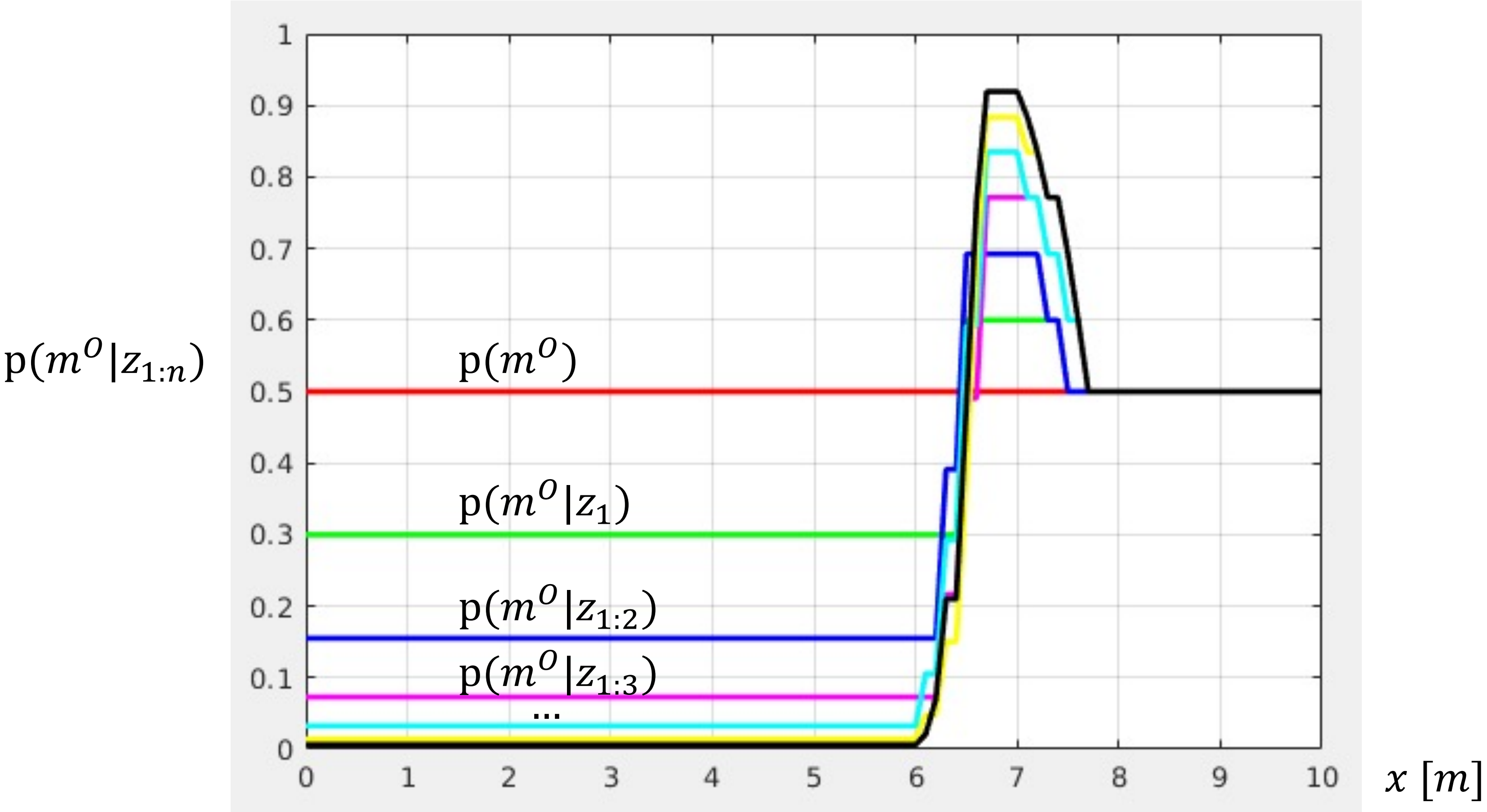


Map − $P(m^O|z_{1:5})$ x [m]

Sensor Modeling – z_6



Sensor Modeling – Non-Zero Measurement Sigma



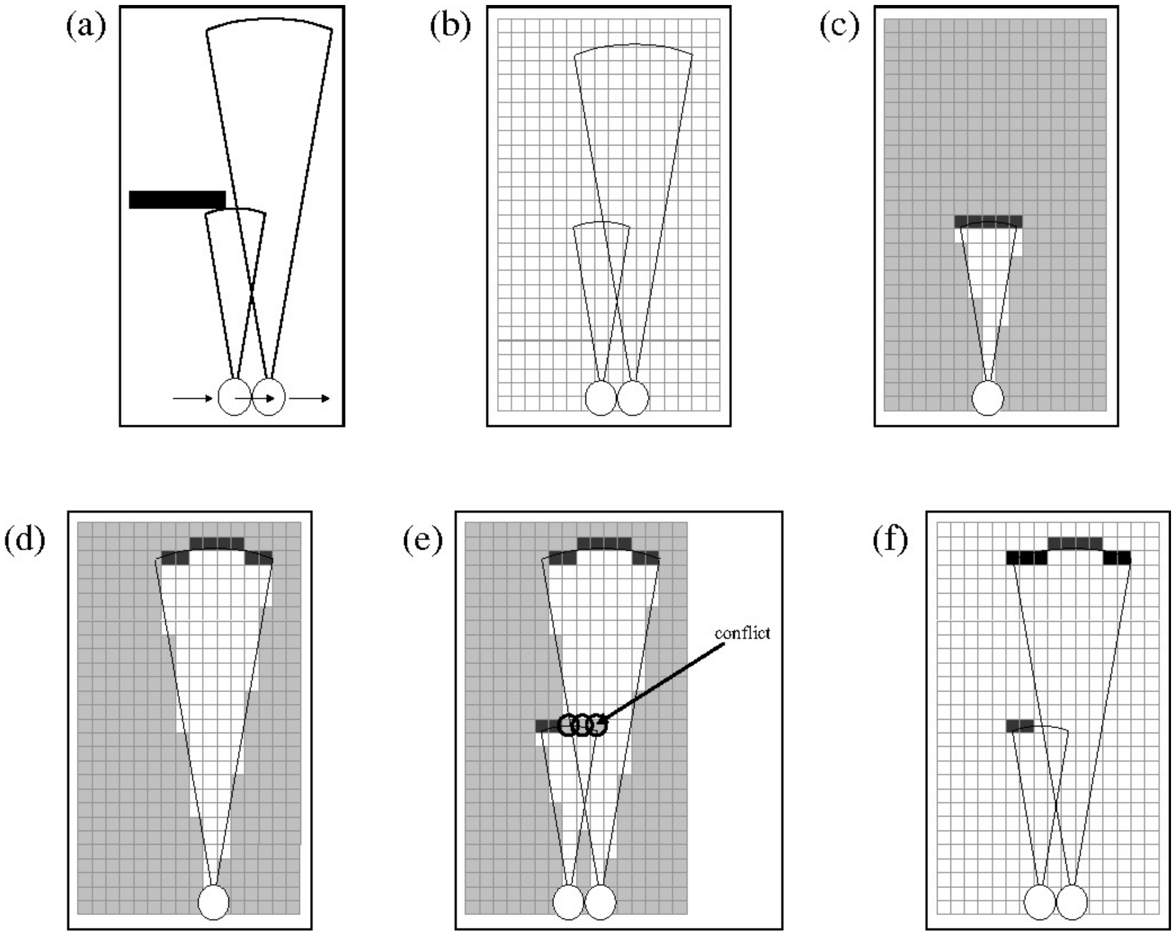
$p(z_n|m^O)$ – forward sensor model

$p(z_n|m^E)$ – negative forward sensor model

$p(m^O|z_n)$ – inverse sensor model – map based on single measurement

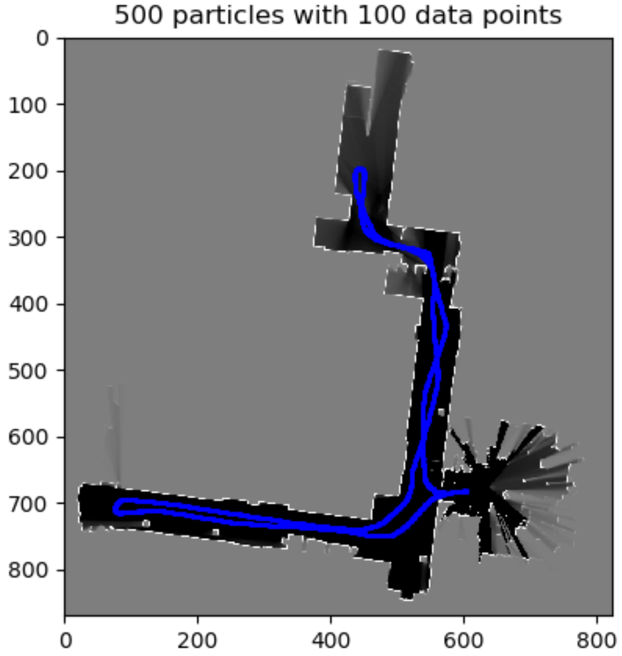
$p(m^O|z_{1:n})$ – final map – product of all inverse models

Sensor Modeling in ND Space

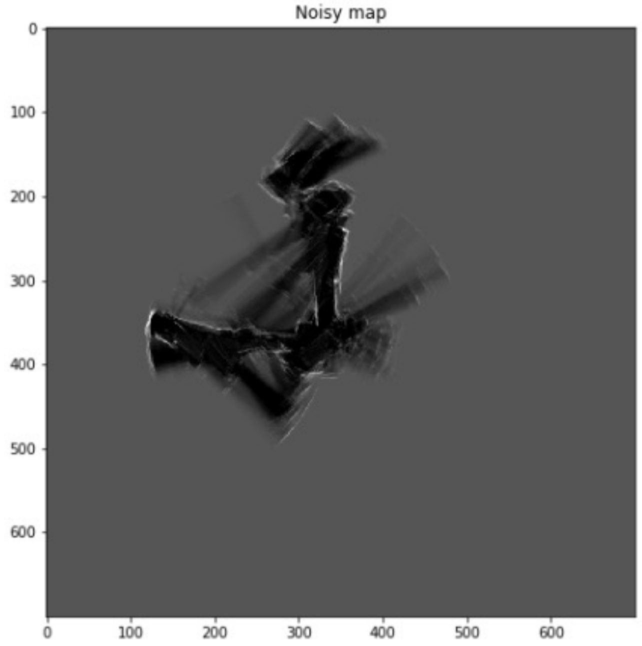


Inverse Sensor Model

https://d3i71xaburhd42.cloudfront.net/bee34b7ac00
2c6b395f384d91968643e6cb99ef3/3-Figure2-1.png



Ground Truth



Noisy Map

https://github.com/khazit/CrazySLAM

Summary

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

Assignment No. 1

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

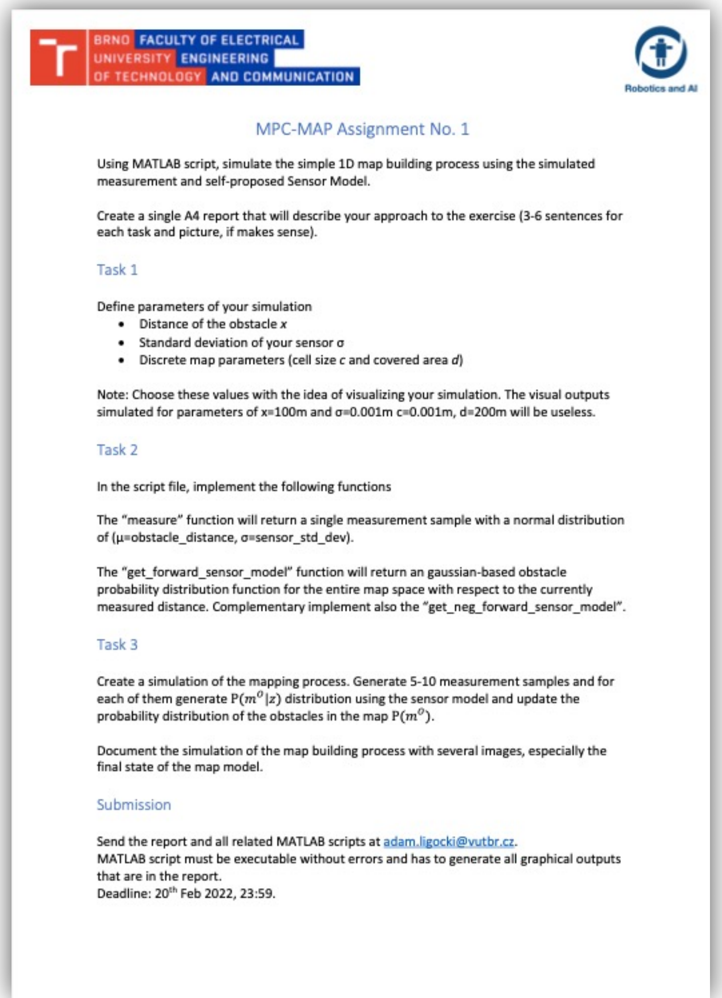
Create a single A4 report that will describe your approach to the exercise (3-6 sentences for each task and picture, if makes sense).

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: 20^{th} Feb 2022, 23:59.



BRNO UNIVERSITY OF TECHNOLOGY | FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION | Robotics and AI

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

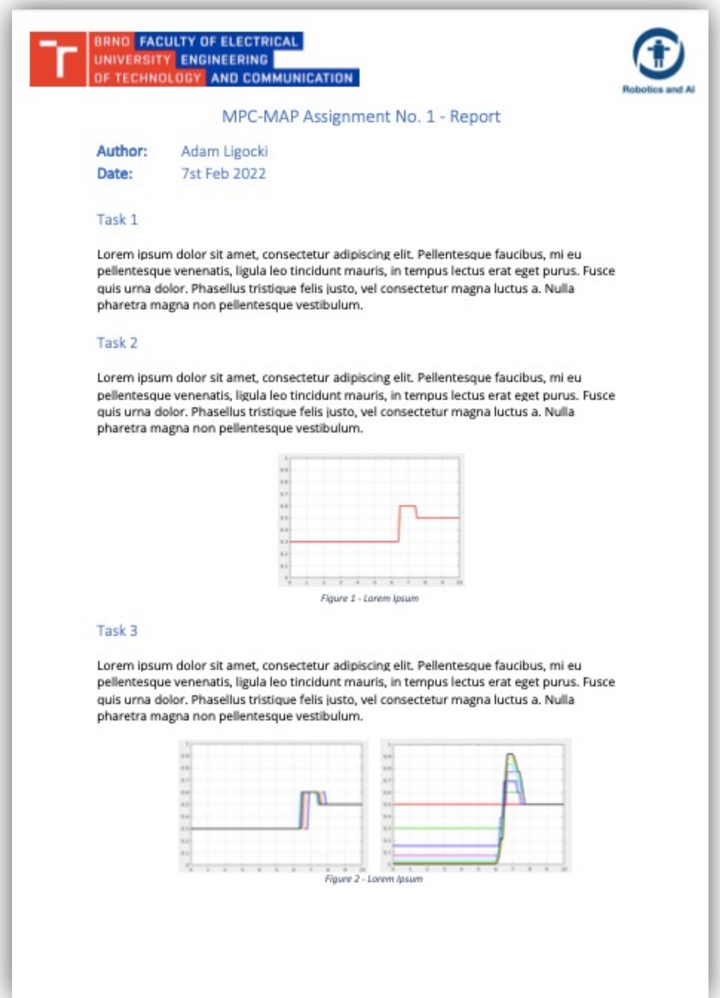
Create a simulation of the mapping process. Generate 5-10 measurement samples and for each of them generate $P(m^o|z)$ distribution using the sensor model and update the probability distribution of the obstacles in the map $P(m^o)$.

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: 20^{th} Feb 2022, 23:59.

Assignment



BRNO UNIVERSITY OF TECHNOLOGY | FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION | Robotics and AI

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