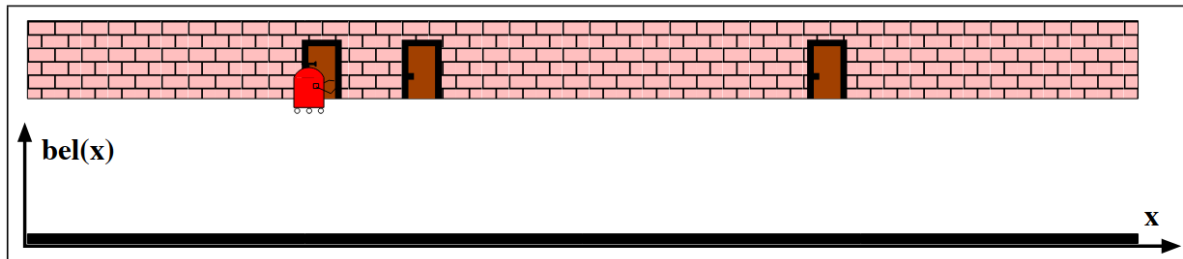


Basics and the prediction step

1D Bayesian Filtering – Robot Localization on a Grid (Grid Localization)

- **Problem:** The robot is stuck in a hallway. It doesn't know its exact location. It has a map, imperfect propulsion, and a noisy sensor.
- **Goal:** To estimate the most probable position of the robot based on a sequence of movements and measurements.
- **Method:** Discrete Bayesian Filter (Histogram Filter).



Source: www.probabilistic-robotics.org



Fundusze Europejskie
dla Rozwoju Społecznego



Rzeczpospolita
Polska

Dofinansowane przez
Unię Europejską



World Representation (Discrete Map)

- **1D world:** Corridor divided into N equal cells (e.g. 10 squares).
- **State** (x): Index of the cell where the robot is located ($x \in \{0, 1, \dots, N-1\}$).
- **Belief:** The probability distribution $P(x_t)$ over all possible states.
 - In Python: a NumPy array of size N .
 - Key condition: The sum of all array elements must be 1.0
- **Example:** `belief = [0.1, 0.1, ..., 0.1]` means total uncertainty (uniform distribution)



Fundusze Europejskie
dla Rozwoju Społecznego



Rzeczpospolita
Polska

Dofinansowane przez
Unię Europejską



Intuition – How does it work?

Bayesian filtering is a cyclical process consisting of two steps:

- **Prediction (Movement):** The robot moves. Each movement increases uncertainty (blurs the distribution) because the wheels can slip.
- **Correction (Measurement):** The robot "sees" the surroundings (e.g., doors). The measurement provides information that allows "tightening" the probability distribution in places that match the map.



Fundusze Europejskie
dla Rozwoju Społecznego



Rzeczpospolita
Polska

Dofinansowane przez
Unię Europejską



Gaussian Noise Modeling

The mathematical foundation is recursive updating:

$$bel(x_t) = \eta \cdot P(z_t | x_t) \cdot \overline{bel}(x_t)$$

Where:

- $\overline{bel}(x_t)$ (**Prior**): Belief after the movement but before the measurement.
- $P(z_t | x_t)$ (**Likelihood**): How closely the z_t measurement matches the x_t position (sensor model).
- η (**Eta**): Normalization constant – we simply divide the result by the sum to get back to the probability.

Tasks for today

Implementation instructions

- **Initialization:** Creating a probability table.
- **Movement Model (Convolution):** Moving the probability “cloud” with error taken into account (e.g. 80% chance of hitting the target, 10% chance of missing, 10% chance of passing).
- **Sensor Model:** Multiplying by the probability derived from the map.