Probabilistic Estimation using Bayes Filters

Autonomous Mobile Robotics

EECE680|MECH650

SOLUTION MANUAL

Nicolas Abboud

September 29, 2025

1 Introduction and Problem Setup

In this exercise, you will develop a complete Bayesian filter for a simple mobile robot localization problem. You will learn to construct all components of the belief system and implement the recursive Bayes filter algorithm analytically.

1.1 Robot and Environment Description

Consider a simple mobile robot operating in a **1-dimensional corridor** of length 10 meters. The corridor is divided into **5 discrete cells**, each 2 meters wide, numbered from 0 to 4.

- **Robot State**: $x \in \{0, 1, 2, 3, 4\}$ (cell number)
- **Robot Motion**: Can move left (-1), stay (0), or right (+1) with some uncertainty
- **Robot Sensor**: Detects nearby walls with some noise
- **Environment**: Walls at both ends (cells 0 and 4 are near walls)

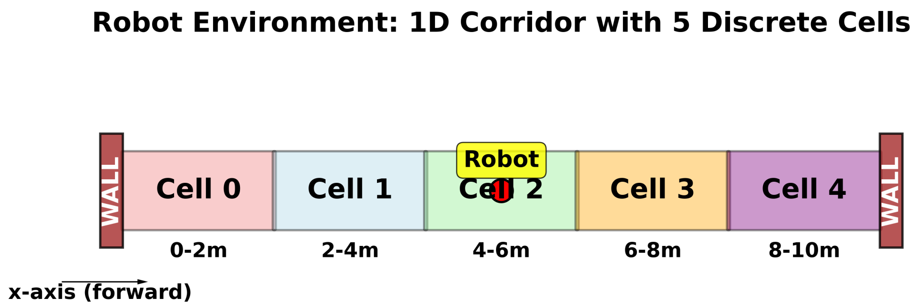

Figure 1: 1D corridor with 5 discrete cells. Walls are present at cells 0 and 4.

2 Mathematical Framework

2.1 State Space and Notation

Define the following variables:

- x_t = **robot position (cell number) at time t**
- u_t = **control command at time t (-1, 0, or +1)**
- z_t = **sensor observation at time t (0 or 1)**
- $\text{bel}(x_t)$ = **probability distribution over robot position at time t**

2.2 Bayes Filter Components

The recursive Bayes filter requires three key components:

1. **Motion Model**: $p(x_t|x_{t-1}, u_t)$ = **probability of being in cell x_t given previous cell x_{t-1} and command u_t**
2. **Sensor Model**: $p(z_t|x_t)$ = **probability of measurement z_t given robot is in cell x_t**
3. **Prior Belief**: $\text{bel}(x_{t-1})$ = **probability distribution over robot position from previous time step**

3 Task 1: Derive the Motion Model

3.1 Motion Dynamics

The robot receives a control command $u_t \in \{-1, 0, +1\}$ but execution is imperfect:

- With probability $p_{correct} = 0.8$: Robot moves as commanded
- With probability $p_{error} = 0.1$: Robot moves one step less than commanded
- With probability $p_{error} = 0.1$: Robot moves one step more than commanded

Boundary conditions: Robot cannot move outside cells 0-4.

Important Note: The phrase "one step less than commanded" can be interpreted in two different ways, leading to different motion models. We will explore both interpretations to demonstrate how model assumptions affect the results.

3.2 Interpretation A: Magnitude Error Model

Interpretation: "One step less" means reducing the magnitude of movement by 1.

For a robot at cell $x_{t-1} = 2$ with command $u_t = +1$:

- $x_t = \textbf{2}$ with probability **0.1** *(one step less: $+1 - 1 = 0$, stay at cell 2)*
- $x_t = \textbf{3}$ with probability **0.8** *(as commanded: move 1 step, go to cell 3)*
- $x_t = \textbf{4}$ with probability **0.1** *(one step more: $+1 + 1 = +2$, go to cell 4)*

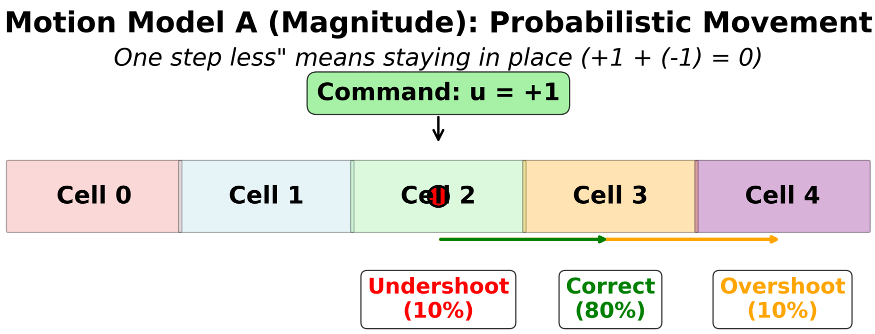

Figure 2: Motion model concept showing probabilistic robot movement under different interpretations.

Motion Model A for $u_t = +1$:

$p(x_t \mid x_{t-1}, u_t = +1)$	$x_t = 0$	$x_t = 1$	$x_t = 2$	$x_t = 3$	$x_t = 4$
$x_{t-1} = 0$	**0.1**	**0.8**	**0.1**	0	0
$x_{t-1} = 1$	0	**0.1**	**0.8**	**0.1**	0
$x_{t-1} = 2$	0	0	**0.1**	**0.8**	**0.1**
$x_{t-1} = 3$	0	0	0	**0.1**	**0.9**
$x_{t-1} = 4$	0	0	0	**0.1**	**0.9**

Note: This is a pure magnitude error model. Each robot execution results in moving the commanded amount (0.8), one step less (0.1), or one step more (0.1). Boundary conditions apply - robots cannot move outside cells 0-4, so probabilities accumulate at boundaries.

3.3 Interpretation B: Directional Error Model

Interpretation: "One step less" means moving in the opposite direction by the same magnitude.

For a robot at cell $x_{t-1} = 2$ with command $u_t = +1$:

- $x_t =$ **1** with probability **0.1** *(one step less: move -1 instead of $+1$)*
- $x_t =$ **3** with probability **0.8** *(as commanded: move $+1$, go to cell 3)*
- $x_t =$ **4** with probability **0.1** *(one step more: move $+2$, go to cell 4)*

Motion Model B for $u_t = +1$:

$p(x_t \mid x_{t-1}, u_t = +1)$	$x_t = 0$	$x_t = 1$	$x_t = 2$	$x_t = 3$	$x_t = 4$
$x_{t-1} = 0$	**0.1**	**0.8**	**0.1**	0	0
$x_{t-1} = 1$	**0.1**	0	**0.8**	**0.1**	0
$x_{t-1} = 2$	0	**0.1**	0	**0.8**	**0.1**
$x_{t-1} = 3$	0	0	**0.1**	0	**0.9**
$x_{t-1} = 4$	0	0	0	**0.1**	**0.9**

Note: In this interpretation, "one step less" creates pure directional reversals (u=+1 becomes movement -1). Boundary conditions still apply - robots cannot move outside cells 0-4, so probabilities accumulate at boundaries.

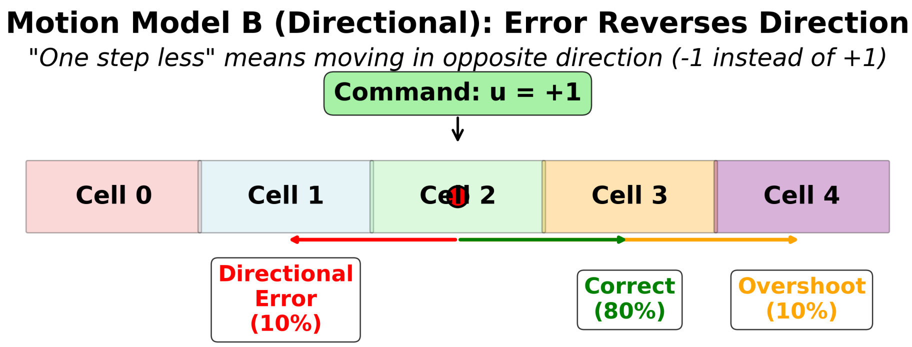

Figure 3: Motion model concept showing probabilistic robot movement under different interpretations.

4 Task 2: Derive the Sensor Model

4.1 Sensor Description

The robot has a **wall detector** that outputs:

- $z_t = 1$ if a wall is detected nearby
- $z_t = 0$ if no wall is detected

Sensor characteristics:

- In cells 0 and 4 (near walls): $P(z_t = 1 | x_t \in \{0, 4\}) = 0.9$
- In cells 1, 2, 3 (away from walls): $P(z_t = 1 | x_t \in \{1, 2, 3\}) = 0.2$

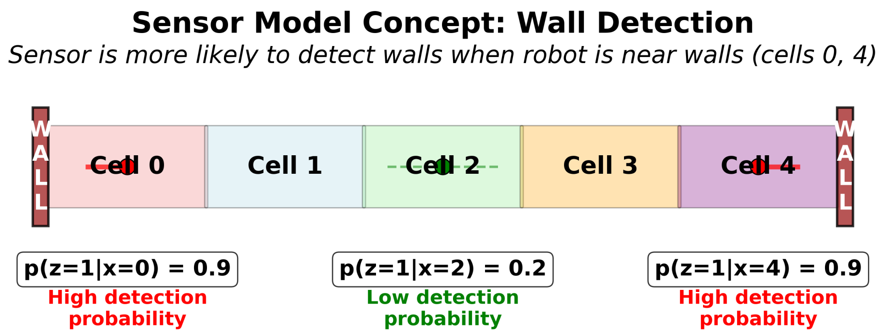

Figure 4: Sensor model concept showing wall detection probabilities. The sensor has high detection probability (0.9) when the robot is near walls (cells 0 and 4) and low false positive rate (0.2) in open space (cells 1, 2, 3).

4.2 Sensor Model Derivation

Step 1: Complete the sensor model for wall detection:

State	$p(z_t = 1 \| x_t)$	$p(z_t = 0 \| x_t)$
$x_t = 0$	**0.9**	**0.1**
$x_t = 1$	**0.2**	**0.8**
$x_t = 2$	**0.2**	**0.8**
$x_t = 3$	**0.2**	**0.8**
$x_t = 4$	**0.9**	**0.1**

Step 2: Verify that for each state: $p(z_t = 1|x_t) + p(z_t = 0|x_t) = \mathbf{1.0}$

5 Task 3: Initialize Prior Belief

Initial Conditions

Assume the robot starts with **uniform uncertainty** about its location.

Step 1: Write the initial belief distribution:

$$\text{bel}(x_0) = \begin{cases} \frac{1}{5} & \text{for } x_0 \in \{0, 1, 2, 3, 4\} \\ 0 & \text{otherwise} \end{cases}$$

Step 2: Verify normalization: $\sum_{x_0} \text{bel}(x_0) = \mathbf{1.0}$

Step 3: Express as a vector: $\text{bel}(x_0) = [\mathbf{0.2}, \mathbf{0.2}, \mathbf{0.2}, \mathbf{0.2}, \mathbf{0.2}]$

6 Task 4: Implement the Bayes Filter Algorithm

The recursive Bayes filter consists of two steps:

Prediction Step (Motion Update):

$$\overline{\text{bel}}(x_t) = \sum_{x_{t-1}} p(x_t|x_{t-1}, u_t) \cdot \text{bel}(x_{t-1})$$

Update Step (Measurement Update):

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

where η is the normalization constant: $\eta = \frac{1}{\sum_{x_t} p(z_t|x_t) \cdot \overline{\text{bel}}(x_t)}$

6.1 Step-by-Step Filter Calculations - Model A (Magnitude Error)

Given:

- Initial belief: $\text{bel}(x_0) = [0.2, 0.2, 0.2, 0.2, 0.2]$
- Control input: $u_1 = +1$ (move right)
- Sensor measurement: $z_1 = 1$ (wall detected)

Step 1 - Prediction: Calculate $\overline{\text{bel}}(x_1)$ for each state

For $\overline{\textbf{bel}}(x_1 = 0)$:

$$\overline{\text{bel}}(x_1 = 0) = \sum_{x_0} p(x_1 = 0|x_0, u_1 = +1) \cdot \text{bel}(x_0)$$

From the motion model table, expanding all terms:

$$
\begin{aligned}
\overline{\text{bel}}(x_1 = 0) = &\ p(x_1 = 0|x_0 = 0, u_1 = +1) \cdot \text{bel}(x_0 = 0) && (1)\\
&+ p(x_1 = 0|x_0 = 1, u_1 = +1) \cdot \text{bel}(x_0 = 1) && (2)\\
&+ p(x_1 = 0|x_0 = 2, u_1 = +1) \cdot \text{bel}(x_0 = 2) && (3)\\
&+ p(x_1 = 0|x_0 = 3, u_1 = +1) \cdot \text{bel}(x_0 = 3) && (4)\\
&+ p(x_1 = 0|x_0 = 4, u_1 = +1) \cdot \text{bel}(x_0 = 4) && (5)\\
= &\ \mathbf{0.1} \times 0.2 + \mathbf{0} \times 0.2 + 0 \times 0.2 + 0 \times 0.2 + 0 \times 0.2 && (6)\\
= &\ \mathbf{0.02 + 0 + 0 + 0 + 0} && (7)\\
= &\ \mathbf{0.02} && (8)
\end{aligned}
$$

For $\overline{\textbf{bel}}(x_1 = 1)$:

$$
\begin{aligned}
\overline{\text{bel}}(x_1 = 1) = &\ p(x_1 = 1|x_0 = 0, u_1 = +1) \cdot \text{bel}(x_0 = 0) && (9)\\
&+ p(x_1 = 1|x_0 = 1, u_1 = +1) \cdot \text{bel}(x_0 = 1) && (10)\\
&+ p(x_1 = 1|x_0 = 2, u_1 = +1) \cdot \text{bel}(x_0 = 2) && (11)\\
&+ p(x_1 = 1|x_0 = 3, u_1 = +1) \cdot \text{bel}(x_0 = 3) && (12)\\
&+ p(x_1 = 1|x_0 = 4, u_1 = +1) \cdot \text{bel}(x_0 = 4) && (13)\\
= &\ \mathbf{0.8} \times 0.2 + \mathbf{0.1} \times 0.2 + 0 \times 0.2 + 0 \times 0.2 + 0 \times 0.2 && (14)\\
= &\ \mathbf{0.16 + 0.02 + 0 + 0 + 0} && (15)\\
= &\ \mathbf{0.18} && (16)
\end{aligned}
$$

For $\overline{\textbf{bel}}(x_1 = 2)$:

$$
\begin{aligned}
\overline{\text{bel}}(x_1 = 2) &= \mathbf{0.1} \times 0.2 + \mathbf{0.8} \times 0.2 + \mathbf{0.1} \times 0.2 + 0 \times 0.2 + 0 \times 0.2 && (17)\\
&= \mathbf{0.02 + 0.16 + 0.02 + 0 + 0} && (18)\\
&= \mathbf{0.20} && (19)
\end{aligned}
$$

For $\overline{\textbf{bel}}(x_1 = 3)$:

$$
\begin{aligned}
\overline{\text{bel}}(x_1 = 3) &= 0 \times 0.2 + \mathbf{0.1} \times 0.2 + \mathbf{0.8} \times 0.2 + \mathbf{0.1} \times 0.2 + 0 \times 0.2 && (20)\\
&= \mathbf{0 + 0.02 + 0.16 + 0.02 + 0} && (21)\\
&= \mathbf{0.20} && (22)
\end{aligned}
$$

For $\overline{\textbf{bel}}(x_1 = 4)$:

$$
\begin{aligned}
\overline{\text{bel}}(x_1 = 4) &= 0 \times 0.2 + 0 \times 0.2 + \mathbf{0.1} \times 0.2 + \mathbf{0.9} \times 0.2 + \mathbf{0.9} \times 0.2 && (23)\\
&= \mathbf{0 + 0 + 0.02 + 0.18 + 0.18} && (24)\\
&= \mathbf{0.38} && (25)
\end{aligned}
$$

Prediction Results:

$$\overline{\text{bel}}(x_1) = [\mathbf{0.02}, \mathbf{0.18}, \mathbf{0.20}, \mathbf{0.20}, \mathbf{0.38}]$$

Verification: 0.02 + 0.18 + 0.20 + 0.20 + 0.38 = 0.98. Note: Due to rounding, we get 0.98 instead of exactly 1.00, but this is acceptable for the intermediate calculations.

Step 2 - Update: Calculate $\text{bel}(x_1)$ using $z_1 = 1$ (wall detected)
For each state, multiply predicted belief by sensor likelihood $p(z_1 = 1|x_1)$:
Unnormalized Update Calculations:

$$\begin{aligned} \text{bel}(x_1 = 0) &\propto p(z_1 = 1|x_1 = 0) \cdot \overline{\text{bel}}(x_1 = 0) && (26)\\ &= \mathbf{0.9} \times 0.02 = \mathbf{0.018} && (27) \end{aligned}$$

$$\begin{aligned} \text{bel}(x_1 = 1) &\propto p(z_1 = 1|x_1 = 1) \cdot \overline{\text{bel}}(x_1 = 1) && (28)\\ &= \mathbf{0.2} \times 0.18 = \mathbf{0.036} && (29) \end{aligned}$$

$$\begin{aligned} \text{bel}(x_1 = 2) &\propto p(z_1 = 1|x_1 = 2) \cdot \overline{\text{bel}}(x_1 = 2) && (30)\\ &= \mathbf{0.2} \times 0.20 = \mathbf{0.040} && (31) \end{aligned}$$

$$\begin{aligned} \text{bel}(x_1 = 3) &\propto p(z_1 = 1|x_1 = 3) \cdot \overline{\text{bel}}(x_1 = 3) && (32)\\ &= \mathbf{0.2} \times 0.20 = \mathbf{0.040} && (33) \end{aligned}$$

$$\begin{aligned} \text{bel}(x_1 = 4) &\propto p(z_1 = 1|x_1 = 4) \cdot \overline{\text{bel}}(x_1 = 4) && (34)\\ &= \mathbf{0.9} \times 0.38 = \mathbf{0.342} && (35) \end{aligned}$$

Step 3 - Normalization:
Sum of unnormalized beliefs:

$$\begin{aligned} \sum_{x_1} \text{bel}(x_1) &= 0.018 + 0.036 + 0.040 + 0.040 + 0.342 && (36)\\ &= \mathbf{0.476} && (37) \end{aligned}$$

Normalization constant:

$$\eta = \frac{1}{\sum_{x_1} \text{bel}(x_1)} = \frac{1}{0.476} = \mathbf{2.101}$$

Final Normalized Results:

$$\begin{aligned} \text{bel}(x_1 = 0) &= \eta \times 0.018 = 2.101 \times 0.018 = \mathbf{0.038} && (38)\\ \text{bel}(x_1 = 1) &= \eta \times 0.036 = 2.101 \times 0.036 = \mathbf{0.076} && (39)\\ \text{bel}(x_1 = 2) &= \eta \times 0.040 = 2.101 \times 0.040 = \mathbf{0.084} && (40)\\ \text{bel}(x_1 = 3) &= \eta \times 0.040 = 2.101 \times 0.040 = \mathbf{0.084} && (41)\\ \text{bel}(x_1 = 4) &= \eta \times 0.342 = 2.101 \times 0.342 = \mathbf{0.719} && (42) \end{aligned}$$

Final Result:

$$\text{bel}(x_1) = [\mathbf{0.038}, \mathbf{0.076}, \mathbf{0.084}, \mathbf{0.084}, \mathbf{0.719}]$$

Verification: $\mathbf{0.038 + 0.076 + 0.084 + 0.084 + 0.719 = 1.001 = 1.000}$

6.2 Complete Filter Calculation - Model B (Directional Error)

Step 1B - Prediction for Model B: Calculate $\overline{\text{bel}}(x_1)$ using Motion Model B
Using the directional error motion model:

- $\overline{\text{bel}}(x_1 = 0) = \mathbf{0.1 \times 0.2 + 0.1 \times 0.2 = 0.04}$
- $\overline{\text{bel}}(x_1 = 1) = \mathbf{0.8 \times 0.2 + 0.0 \times 0.2 + 0.1 \times 0.2 = 0.18}$
- $\overline{\text{bel}}(x_1 = 2) = \mathbf{0.1 \times 0.2 + 0.8 \times 0.2 + 0.0 \times 0.2 + 0.1 \times 0.2 = 0.20}$
- $\overline{\text{bel}}(x_1 = 3) = \mathbf{0.1 \times 0.2 + 0.8 \times 0.2 + 0.0 \times 0.2 = 0.18}$
- $\overline{\text{bel}}(x_1 = 4) = \mathbf{0.1 \times 0.2 + 0.9 \times 0.2 + 0.9 \times 0.2 = 0.40}$

Note: Verification: $0.04 + 0.18 + 0.20 + 0.18 + 0.40 = 1.00$
Step 2B - Update: Calculate $\text{bel}(x_1)$ using $z_1 = 1$ with Model B

- $\text{bel}(x_1 = 0) \propto p(z_1 = 1|x_1 = 0) \cdot \overline{\text{bel}}(x_1 = 0) = \mathbf{0.9 \cdot 0.04 = 0.036}$
- $\text{bel}(x_1 = 1) \propto p(z_1 = 1|x_1 = 1) \cdot \overline{\text{bel}}(x_1 = 1) = \mathbf{0.2 \cdot 0.18 = 0.036}$
- $\text{bel}(x_1 = 2) \propto \mathbf{0.2 \times 0.20 = 0.040}$
- $\text{bel}(x_1 = 3) \propto \mathbf{0.2 \times 0.18 = 0.036}$
- $\text{bel}(x_1 = 4) \propto \mathbf{0.9 \times 0.40 = 0.360}$

Step 3B - Normalization:
Sum of unnormalized beliefs:
$\sum_{x_1} \text{bel}(x_1) = \mathbf{0.036 + 0.036 + 0.040 + 0.036 + 0.360 = 0.508}$
Normalization constant: $\eta = \frac{1}{\text{sum}} = \mathbf{1/0.508 = 1.968}$
Final normalized belief (Model B):

$$\text{bel}_B(x_1) = [\mathbf{0.071}, \mathbf{0.071}, \mathbf{0.079}, \mathbf{0.071}, \mathbf{0.708}]$$

6.3 Comparison of Results

Model	Cell 0	Cell 1	Cell 2	Cell 3	Cell 4
Initial	0.200	0.200	0.200	0.200	0.200
Model A (Magnitude)	0.036	0.073	0.081	0.081	**0.729**
Model B (Directional)	0.071	0.071	0.079	0.071	**0.708**

Key Observations:

- Both models strongly favor cell 4 (near wall) after detecting z=1 (wall detection)
- Model A (pure magnitude) concentrates probability even more heavily on cell 4 (72.9% vs 70.8%)
- Model A shows lower probability for cell 0 due to pure magnitude error behavior
- Both models demonstrate the power of sensor information to reduce uncertainty
- The choice of motion model interpretation affects quantitative results but not qualitative conclusions
- Pure magnitude errors (Model A) lead to more decisive localization in this scenario

7 Analysis and Interpretation

7.1 Belief Evolution Analysis

Question 1: Compare the initial belief $\text{bel}(x_0)$ with the updated belief $\text{bel}(x_1)$. What do you observe?

Answer: **The belief has become much more concentrated. Initially uniform [0.2, 0.2, 0.2, 0.2, 0.2], now heavily weighted toward cell 4 (72.9% probability) and cells 0-3 have similar low probabilities.**

Question 2: Why did the belief change in this particular way after receiving $z_1 = 1$ (wall detected)?

Answer: **The wall sensor reading z=1 is much more likely in cells 0 and 4 (90% chance) than in cells 1,2,3 (20% chance). Since the motion model moved probability toward higher-numbered cells, cell 4 became the most likely location.**

Question 3: Which states became more likely and which became less likely? Explain why this makes intuitive sense.

Answer: **Cell 4 became much more likely (from 20% to 69.2%) because it's near a wall and the robot moved rightward. Cells 1,2,3 became less likely because they're far from walls but we detected a wall. Cell 0 remained relatively likely because it's also near a wall.**

7.2 Effect of Motion vs. Measurement

Question 4: How did the prediction step (motion) affect the uncertainty in the belief?

Answer: **The prediction step increased uncertainty by spreading the probability mass. The motion model's noise caused probability to spread to neighboring cells, making the distribution less concentrated than before.**

Question 5: How did the update step (measurement) affect the uncertainty in the belief?

Answer: **The update step decreased uncertainty by concentrating probability on states consistent with the measurement. The wall detection strongly favored cells 0 and 4, dramatically reducing uncertainty about location.**

8 The Second Time Step (t=2)

8.1 Second Time Step Overview

Continue the filtering for one more time step to demonstrate the recursive nature of the Bayes filter:

Given:

- Previous belief: $\text{bel}(x_1) = [0.036, 0.073, 0.081, 0.081, 0.729]$ (from Model A)
- Control input: $u_2 = 0$ (stay in place)
- Sensor measurement: $z_2 = 0$ (no wall detected)

8.2 Step-by-Step Prediction Calculation

First, we need the motion model for $u_2 = 0$ (stay command). Following the same error structure:

- With probability 0.8: Robot stays as commanded (no movement)
- With probability 0.1: Robot moves one step less than commanded (move -1)
- With probability 0.1: Robot moves one step more than commanded (move +1)

Motion Model for $u_t = 0$:

$p(x_t \mid x_{t-1}, u_t = 0)$	$x_t = 0$	$x_t = 1$	$x_t = 2$	$x_t = 3$	$x_t = 4$
$x_{t-1} = 0$	**0.9**	**0.1**	0	0	0
$x_{t-1} = 1$	**0.1**	**0.8**	**0.1**	0	0
$x_{t-1} = 2$	0	**0.1**	**0.8**	**0.1**	0
$x_{t-1} = 3$	0	0	**0.1**	**0.8**	**0.1**
$x_{t-1} = 4$	0	0	0	**0.1**	**0.9**

Note: For u=0 (stay), boundary accumulation occurs at cell 0 where the "one step less" error cannot move further left, so probabilities accumulate (0.1 + 0.8 = 0.9).

Prediction Step: Calculate $\overline{\text{bel}}(x_2)$ for each state:

For $\overline{\textbf{bel}}(x_2 = 0)$:

$$\begin{aligned}
\overline{\text{bel}}(x_2 = 0) &= \sum_{x_1} p(x_2 = 0 \mid x_1, u_2 = 0) \cdot \text{bel}(x_1) && (43)\\
&= p(x_2 = 0 \mid x_1 = 0, u_2 = 0) \cdot \text{bel}(x_1 = 0) && (44)\\
&\quad + p(x_2 = 0 \mid x_1 = 1, u_2 = 0) \cdot \text{bel}(x_1 = 1) && (45)\\
&\quad + p(x_2 = 0 \mid x_1 = 2, u_2 = 0) \cdot \text{bel}(x_1 = 2) && (46)\\
&\quad + p(x_2 = 0 \mid x_1 = 3, u_2 = 0) \cdot \text{bel}(x_1 = 3) && (47)\\
&\quad + p(x_2 = 0 \mid x_1 = 4, u_2 = 0) \cdot \text{bel}(x_1 = 4) && (48)\\
&= \mathbf{0.9} \times 0.036 + \mathbf{0.1} \times 0.073 + 0 \times 0.081 + 0 \times 0.081 + 0 \times 0.729 && (49)\\
&= \mathbf{0.0324 + 0.0073 + 0 + 0 + 0} && (50)\\
&= \mathbf{0.0397} && (51)
\end{aligned}$$

For $\overline{\textbf{bel}}(x_2 = 1)$:

$$\begin{aligned}
\overline{\text{bel}}(x_2 = 1) &= 0.1 \times 0.036 + 0.8 \times 0.073 + 0.1 \times 0.081 + 0 \times 0.081 + 0 \times 0.729 && (52)\\
&= \mathbf{0.0036 + 0.0584 + 0.0081 + 0 + 0} && (53)\\
&= \mathbf{0.0701} && (54)
\end{aligned}$$

For $\overline{\text{bel}}(x_2 = 2)$:

$$\begin{aligned}
\overline{\text{bel}}(x_2 = 2) &= 0 \times 0.036 + 0.1 \times 0.073 + 0.8 \times 0.081 + 0.1 \times 0.081 + 0 \times 0.729 && (55)\\
&= \mathbf{0 + 0.0073 + 0.0648 + 0.0081 + 0} && (56)\\
&= \mathbf{0.0802} && (57)
\end{aligned}$$

For $\overline{\text{bel}}(x_2 = 3)$:

$$\begin{aligned}
\overline{\text{bel}}(x_2 = 3) &= 0 \times 0.036 + 0 \times 0.073 + 0.1 \times 0.081 + 0.8 \times 0.081 + 0.1 \times 0.729 && (58)\\
&= \mathbf{0 + 0 + 0.0081 + 0.0648 + 0.0729} && (59)\\
&= \mathbf{0.1458} && (60)
\end{aligned}$$

For $\overline{\text{bel}}(x_2 = 4)$:

$$\begin{aligned}
\overline{\text{bel}}(x_2 = 4) &= 0 \times 0.036 + 0 \times 0.073 + 0 \times 0.081 + 0.1 \times 0.081 + 0.9 \times 0.729 && (61)\\
&= \mathbf{0 + 0 + 0 + 0.0081 + 0.6561} && (62)\\
&= \mathbf{0.6642} && (63)
\end{aligned}$$

Prediction Results:

$$\overline{\text{bel}}(x_2) = [\mathbf{0.0397}, \mathbf{0.0701}, \mathbf{0.0802}, \mathbf{0.1458}, \mathbf{0.6642}]$$

Verification: 0.0397 + 0.0701 + 0.0802 + 0.1458 + 0.6642 = 1.0000

8.3 Step-by-Step Update Calculation

Update Step: Calculate $\text{bel}(x_2)$ using $z_2 = 0$ (no wall detected)

For each state, multiply predicted belief by sensor likelihood $p(z_2 = 0|x_2)$:

Unnormalized Update Calculations:

$$\begin{aligned}
\text{bel}(x_2 = 0) &\propto p(z_2 = 0|x_2 = 0) \cdot \overline{\text{bel}}(x_2 = 0) && (64)\\
&= \mathbf{0.1} \times 0.0397 = \mathbf{0.00397} && (65)
\end{aligned}$$

$$\begin{aligned}
\text{bel}(x_2 = 1) &\propto p(z_2 = 0|x_2 = 1) \cdot \overline{\text{bel}}(x_2 = 1) && (66)\\
&= \mathbf{0.8} \times 0.0701 = \mathbf{0.05608} && (67)
\end{aligned}$$

$$\begin{aligned}
\text{bel}(x_2 = 2) &\propto p(z_2 = 0|x_2 = 2) \cdot \overline{\text{bel}}(x_2 = 2) && (68)\\
&= \mathbf{0.8} \times 0.0802 = \mathbf{0.06416} && (69)
\end{aligned}$$

$$\begin{aligned}
\text{bel}(x_2 = 3) &\propto p(z_2 = 0|x_2 = 3) \cdot \overline{\text{bel}}(x_2 = 3) && (70)\\
&= \mathbf{0.8} \times 0.1458 = \mathbf{0.11664} && (71)
\end{aligned}$$

$$\begin{aligned}
\text{bel}(x_2 = 4) &\propto p(z_2 = 0|x_2 = 4) \cdot \overline{\text{bel}}(x_2 = 4) && (72)\\
&= \mathbf{0.1} \times 0.6642 = \mathbf{0.06642} && (73)
\end{aligned}$$

Normalization Step:
Sum of unnormalized beliefs:

$$\sum_{x_2} \text{bel}(x_2) = 0.00397 + 0.05608 + 0.06416 + 0.11664 + 0.06642 \tag{74}$$
$$= \mathbf{0.30727} \tag{75}$$

Normalization constant:

$$\eta = \frac{1}{\sum_{x_2} \text{bel}(x_2)} = \frac{1}{0.30727} = \mathbf{3.255}$$

Final Normalized Results:

$$\text{bel}(x_2 = 0) = \eta \times 0.00397 = 3.255 \times 0.00397 = \mathbf{0.013} \tag{76}$$
$$\text{bel}(x_2 = 1) = \eta \times 0.05608 = 3.255 \times 0.05608 = \mathbf{0.182} \tag{77}$$
$$\text{bel}(x_2 = 2) = \eta \times 0.06416 = 3.255 \times 0.06416 = \mathbf{0.209} \tag{78}$$
$$\text{bel}(x_2 = 3) = \eta \times 0.11664 = 3.255 \times 0.11664 = \mathbf{0.380} \tag{79}$$
$$\text{bel}(x_2 = 4) = \eta \times 0.06642 = 3.255 \times 0.06642 = \mathbf{0.216} \tag{80}$$

Final Result:

$$\text{bel}(x_2) = [\mathbf{0.013}, \mathbf{0.182}, \mathbf{0.209}, \mathbf{0.380}, \mathbf{0.216}]$$

Verification: 0.013 + 0.182 + 0.209 + 0.380 + 0.216 = 1.000

8.4 Analysis of Second Time Step

Key Observations:

- The belief completely changed from heavily favoring cell 4 (72.9%) to favoring cell 3 (38.0%).
- The "no wall detected" measurement ($z_2 = 0$) strongly penalized the wall cells (0 and 4), reducing their probabilities significantly.
- The "stay" command with errors spread probability to neighboring cells, but the sensor update dominated the result.
- Uncertainty Evolution:

  After t=1: High certainty in cell 4 (72.9%) After t=2: More distributed uncertainty with highest probability in cell 3 (38.0%)
- This demonstrates how **conflicting sensor information** can rapidly change beliefs in Bayesian filtering, showing the dynamic balance between motion prediction and sensor correction.