Exercise 8 – Markov Localization and Kalman Filter

Overview

This exercise session will illustrate two common approaches to robot localization, namely the Markov and (Extended) Kalman Filter algorithms. Both exercises are based on exams from previous years. They closely match the expected level in terms of difficulty, but have been extended slightly to provide more comprehensive coverage of the theory.

Q1 Markov Localization

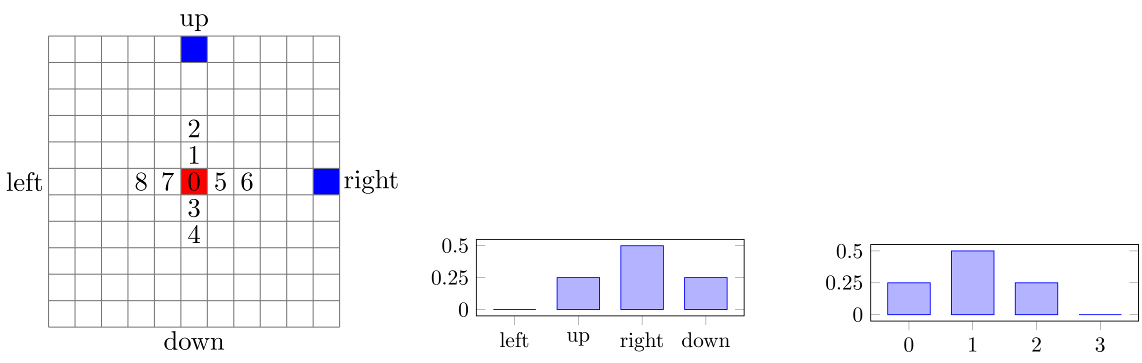
This first problem considers a robot moving in a plane, which we represent as a discretized 2D grid. The robot state x consists of its position and heading $x = \left(\mathrm{pos}_x, \mathrm{pos}_y, \theta\right)$, where its heading is discretized as $\theta \in \{\mathrm{left}, \mathrm{right}, \mathrm{up}, \mathrm{down}\}$. Since we are considering a probabilistic approach, we will introduce a belief state $\mathrm{bel}(\mathcal{X}_t)$ that represents our belief of the robot state (as a distribution) at time t. Since we have a discrete spate space, we will use the Markov localization algorithms (Bayes filter), summarized by the following equations

$$\overline{\mathrm{bel}}(\mathcal{X}_t) = \sum_{\mathcal{X}_{t-1}} p(\mathcal{X}_t | u_t, \mathcal{X}_{t-1}) \mathrm{bel}(\mathcal{X}_{t-1}) \tag{1}$$

$$\mathrm{bel}(\mathcal{X}_t) = \eta p(z_t | \mathcal{X}_t, M) \overline{\mathrm{bel}}(\mathcal{X}_t) \tag{2}$$

We know the initial position of the robot with zero uncertainty, and mark it with a red square in figure 1a. Obstacles are marked with blue squares. Some of the cells in the figure contain numbers, which will be used to refer to them in the following questions. The robot's initial heading is uncertain and its distribution is shown in figure 1b.

The robot starts by moving in the direction it is facing. The motion is uncertain in the sense that the distance it moves in this direction is a random variable whose distribution is given by figure 1c.



(a) Discretized world (b) Initial heading distribution (c) Forward motion distribution

Q1.1

Calculate the probability distribution of the robot being at positions 0 through 8 after one motion step and draw it onto figure 2. In addition to filling the bars, don't forget to also annotate the y-axis.

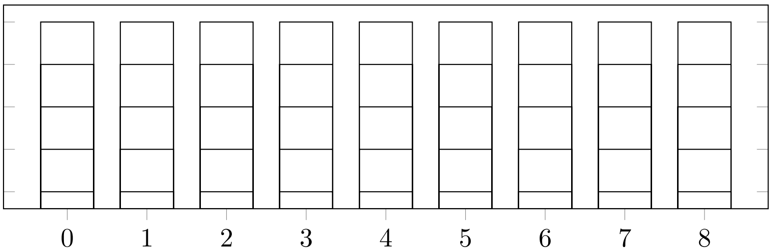


Figure 2: Fill in the probabilities of the robot being in cells 0 through 8 after one motion step.

Answer: We know from the problem description that the initial position of the robot is cell_0 with zero uncertainty and that the robot's initial heading is given by 1b. The initial state's belief distribution $\text{bel}(\mathcal{X}_0)$ is therefore

$$
\begin{aligned}
\text{bel}(x = (0, 0, \text{left})) &= 0 \\
\text{bel}(x = (0, 0, \text{up})) &= 0.25 \\
\text{bel}(x = (0, 0, \text{right})) &= 0.5 \\
\text{bel}(x = (0, 0, \text{down})) &= 0.25
\end{aligned}
$$

From the forward motion model 1c, we see that the robot might move forward by 0, 1 or 2 cells in each step. For the first motion update, equation 1 is equal to $\overline{\text{bel}}(\mathcal{X}_1) = \sum_{\mathcal{X}_0} p(\mathcal{X}_1|u_1, \mathcal{X}_0)\text{bel}(\mathcal{X}_0)$.

Enumerating all combinations of (non-zero) initial states and possible motions, we get

		initial state			
		x=(0,0,left)	x=(0,0,up)	x=(0,0,right)	x=(0,0,down)
steps	0	$0.25 \cdot 0 = 0$	$0.25 \cdot 0.25 = 1/16$	$0.25 \cdot 0.5 = 1/8$	$0.25 \cdot 0.25 = 1/16$
	1	$0.5 \cdot 0 = 0$	$0.5 \cdot 0.25 = 1/8$	$0.5 \cdot 0.5 = 1/4$	$0.5 \cdot 0.25 = 1/8$
	2	$0.25 \cdot 0 = 0$	$0.25 \cdot 0.25 = 1/16$	$0.25 \cdot 0.5 = 1/8$	$0.25 \cdot 0.25 = 1/16$

Notice that the above initial state and motion combinations bring the robot to the following cells

		initial state			
		x=(0,0,left)	x=(0,0,up)	x=(0,0,right)	x=(0,0,down)
steps	0	cell_0	cell_0	cell_0	cell_0
	1	cell_7	cell_1	cell_5	cell_3
	2	cell_8	cell_2	cell_6	cell_4

Combining the probabilities for each cell yields

$$
\begin{aligned}
\text{cell}_0 &= 0 + \frac{1}{16} + \frac{1}{8} + \frac{1}{16} = 1/4 \\
\text{cell}_1 &= 1/8 \\
\text{cell}_2 &= 1/16 \\
\text{cell}_3 &= 1/8 \\
\text{cell}_4 &= 1/16 \\
\text{cell}_5 &= 1/4 \\
\text{cell}_6 &= 1/8 \\
\text{cell}_7 &= 0 \\
\text{cell}_8 &= 0
\end{aligned}
$$

Drawing these probabilities on top of figure 2 results in figure 3.

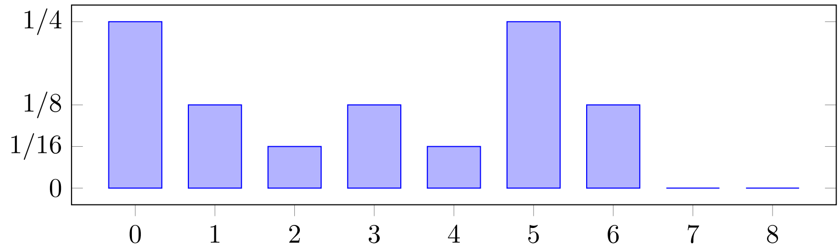


Figure 3: Probabilities of the robot being in cells 0 through 8 after one motion step.

Q1.2

The robot is equipped with a sensor which detects the presence of an obstacle (at any range) in the direction it is facing with probability 1 (no sensor noise). Following its movement, the robot receives a sensor measurement indicating the presence of an obstacle. Fill out the probabilities of the robot being at positions 0 through 8 after fusing in the measurement in graph 4.

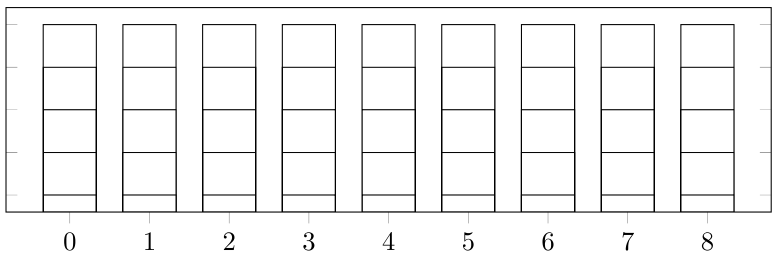


Figure 4: Fill in the probabilities of the robot being in cells 0 through 8 after one motion step and measurement update.

Answer: The robot moved forward without rotating. Its heading θ therefore did not change between t_0 and t_1, i.e. $\theta_{t_0} = \theta_{t_1}$. This means that for cell 0, its heading uncertainty distribution still corresponds to the initial distribution given in figure 1b. Furthermore, it could only reach cells 1 and 2 if it's heading θ_{t_0} = up, cells 5 and 6 if θ_{t_0} = right and so forth.

As described in the assignment, the robot observes obstacles with probability 1 if they are present in the direction it is facing. This also means that the probability of the robot not observing an obstacle despite one being in the direction it is facing is 0. Intuitively, we would therefore expect that after applying the measurement update the beliefs for all cells are zero except for cells 1, 2, 5 and 6, and for cell 0 if the robot is facing up or to the right.

For the first measurement update, equation 2 is equal to $\text{bel}(\mathcal{X}_1) = \eta p(z_1|\mathcal{X}_1, M)\overline{\text{bel}}(\mathcal{X}_1)$.

Enumerating all combinations yields

$$
\begin{aligned}
&\text{cell}_0 : p\left(1 | x = (0,0,\text{left}), M\right) \cdot \overline{\text{bel}}\left(x = (0,0,\text{left})\right) && = 0 \cdot 1/4 \cdot 0 && = 0 \\
&\text{cell}_0 : p\left(1 | x = (0,0,\text{up}), M\right) \cdot \overline{\text{bel}}\left(x = (0,0,\text{up})\right) && = 1 \cdot 1/4 \cdot 0.25 && = 1/16 \\
&\text{cell}_0 : p\left(1 | x = (0,0,\text{right}), M\right) \cdot \overline{\text{bel}}\left(x = (0,0,\text{right})\right) && = 1 \cdot 1/4 \cdot 0.5 && = 1/8 \\
&\text{cell}_0 : p\left(1 | x = (0,0,\text{down}), M\right) \cdot \overline{\text{bel}}\left(x = (0,0,\text{down})\right) && = 0 \cdot 1/4 \cdot 0.25 && = 0 \\
&\text{cell}_1 : p\left(1 | x = (0,0,\text{up}), M\right) \cdot \overline{\text{bel}}\left(x = (0,0,\text{up})\right) && = 1 \cdot 1/8 && = 1/8 \\
&\text{cell}_2 : p\left(1 | x = (0,0,\text{up}), M\right) \cdot \overline{\text{bel}}\left(x = (0,0,\text{up})\right) && = 1 \cdot 1/16 && = 1/16 \\
&\text{cell}_3 : p\left(1 | x = (0,0,\text{down}), M\right) \cdot \overline{\text{bel}}\left(x = (0,0,\text{down})\right) && = 0 \cdot 1/8 && = 0 \\
&\text{cell}_4 : p\left(1 | x = (0,0,\text{down}), M\right) \cdot \overline{\text{bel}}\left(x = (0,0,\text{down})\right) && = 0 \cdot 1/16 && = 0 \\
&\text{cell}_5 : p\left(1 | x = (0,0,\text{right}), M\right) \cdot \overline{\text{bel}}\left(x = (0,0,\text{right})\right) && = 1 \cdot 1/4 && = 1/4 \\
&\text{cell}_6 : p\left(1 | x = (0,0,\text{right}), M\right) \cdot \overline{\text{bel}}\left(x = (0,0,\text{right})\right) && = 1 \cdot 1/8 && = 1/8 \\
&\text{cell}_7 : p\left(1 | x = (0,0,\text{left}), M\right) \cdot \overline{\text{bel}}\left(x = (0,0,\text{left})\right) && = 0 \cdot 0 && = 0 \\
&\text{cell}_8 : p\left(1 | x = (0,0,\text{left}), M\right) \cdot \overline{\text{bel}}\left(x = (0,0,\text{left})\right) && = 0 \cdot 0 && = 0
\end{aligned}
$$

After combining the beliefs for each cell and normalizing, we get

$$
\begin{aligned}
\text{bel}(\text{cell}_0) &= \frac{16}{12} \cdot \left(\frac{1}{16} + \frac{1}{8}\right) = 1/4 \\
\text{bel}(\text{cell}_1) &= \frac{16}{12} \cdot \frac{1}{8} = 1/6 \\
\text{bel}(\text{cell}_2) &= \frac{16}{12} \cdot \frac{1}{16} = 1/12 \\
\text{bel}(\text{cell}_3) &= 0 \\
\text{bel}(\text{cell}_4) &= 0 \\
\text{bel}(\text{cell}_5) &= \frac{16}{12} \cdot \frac{1}{4} = 1/3 \\
\text{bel}(\text{cell}_6) &= \frac{16}{12} \cdot \frac{1}{8} = 1/6 \\
\text{bel}(\text{cell}_7) &= 0 \\
\text{bel}(\text{cell}_8) &= 0
\end{aligned}
$$

Representing these numbers as a bar plot results in figure 5.

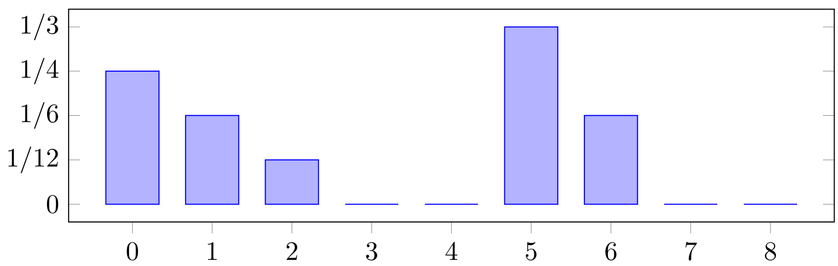


Figure 5: Probabilities of the robot being in cells 0 through 8 after one motion step and measurement update.

Q2 Kalman Filter

A spaceship is approaching the moon and preparing for a landing. To ensure a safe and precise landing, it needs to continuously monitor its current height above ground as well as its velocity so that it can predict its landing space and time. To perform this task, a distance ranging sensor is mounted outside the spacecraft, angled at

45 deg downwards toward the ground. Because the state space is now continuous, we will estimate the state of the vehicle using a Kalman filter. Such a filter can generally be structured into a prediction step and an update step. In the following, we will look at the two steps separately.

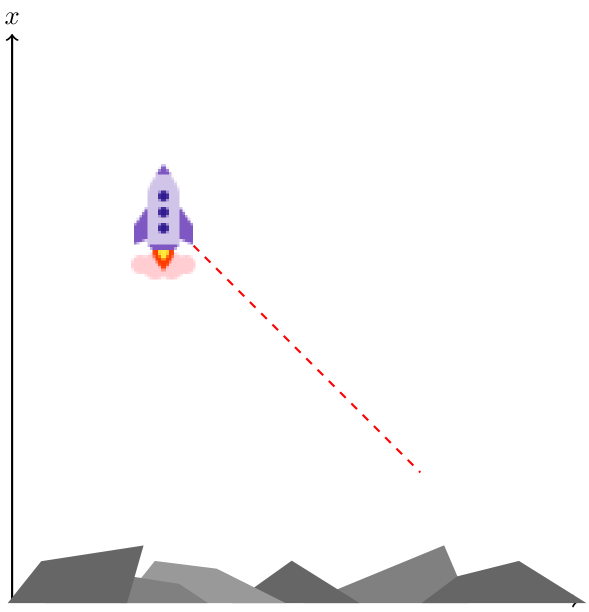


Figure 6: Spaceship approaching the surface of the moon, equipped with a distance ranging sensor (red dashed line).

Q2.1 State Prediction

For now, it can be assumed that the spacecraft moves only in a vertical line towards the surface of the moon. It can therefore be assumed that the problem is one dimensional. There are two forces acting on the spacecraft: the gravity of the moon ($g = 1.62\,\mathrm{m/s^2}$) as well as the force of the rocket engines ($a = 1.52\,\mathrm{m/s^2}$) used to decelerate the spacecraft. We will assume mass is constant, and to make the maths easier you can consider the control as $u = a - g$.

Write down the discrete state equations of the spaceship. Formulate them as a function of the spacecraft's height $x(t)$, velocity $\dot{x}(t)$, and control input u. Assume the time between two prediction steps is given by ΔT.

Answer: The state equations can be derived from the system kinematics:

$$\ddot{x}(t) = a - g \tag{3}$$

$$\dot{x}(t + \Delta T) = \dot{x}(t) + \Delta T(a - g) \tag{4}$$

$$x(t + \Delta T) = x(t) + \Delta T\dot{x}(t) + 0.5\Delta T^2(a - g) \tag{5}$$

We are interested in the *height* and *velocity* of the spacecraft. We therefore define our state as $s(k) = \begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix}$ and our control input as $u = (a - g)$. The state equations can then be formulated as:

$$s(k+1) = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix} s(k) + \begin{bmatrix} 0.5\Delta T^2 \\ \Delta T \end{bmatrix} u \tag{6}$$

$$= \mathbf{F}s(k) + \mathbf{B}u \tag{7}$$

Q2.2 State Update

As neither gravity is constant[1], nor the rocket engines perform perfectly, it can be assumed that the rocket's motion is noisy. We will model this noise as additive Gaussian noise, where at each time step an additional

[1] Depending on the location, the Moon's gravity can vary by up to $0.0253\,\mathrm{m/s^2}$.

random velocity, sampled i.i.d from $v_r \sim \mathcal{N}\left(0, \sigma_v^2\right)$ will be added to the velocity (and $v_r \Delta T$ to the position).

If we would only rely on our motion model, the believed state of the spacecraft would drift away over time from the actual true state, due to motion noise. This is why we employ various sensors to update the belief of our state and remain closer to the true state. In this case, the spacecraft uses a Laser-Rangefinder angled at 45 deg towards the ground, which provides a distance measurement. As the lunar surface is uneven and rocky and the laser device is not perfect, there is a certain measurement noise associated with each measurement. This noise can equally be modelled as additive i.i.d Gaussian noise $\xi \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$, which is applied to the sensor measurement. Furthermore, assume that the sensor provides a new measurement every ΔT.

Write down the measurement as a function of the state and determine the measurement Jacobian with respect to the state.

Answer: The observation model can be formulated as follow: As the laser is at an angle towards the ground, the distance d measured does not directly correspond to the spacecraft's height x. Instead $d = x * \frac{1}{\sin(45°)}$, so that our measurement model can be formulated as:

$$\begin{aligned} z(k) &= g(s) + \xi && (8) \\ &= \begin{bmatrix} \frac{1}{\sin(45°)} & 0 \end{bmatrix} s(k) + \xi && (9) \\ &= \mathbf{H}s(k) + \xi && (10) \end{aligned}$$

As we have two state variables and a 1-dimensional measurement function our measurement Jacobian will be a matrix with the dimensions $J(s) \in \mathbb{R}^{1\times 2}$. The Jacobian is given by:

$$\begin{aligned} J(s) &= \begin{bmatrix} \frac{d}{dx} g(s) & \frac{d}{d\dot{x}} g(s) \end{bmatrix} && (11) \\ &= \begin{bmatrix} \frac{1}{\sin(45°)} & 0 \end{bmatrix} && (12) \end{aligned}$$

Q2.3 Kalman Filter Implementation in Python

A Kalman Filter can be easily implemented to fuse a variety of sensors and motion models. Have a look at the attached python script and **fill in the missing parts of the Kalman Filter that you have derived in the previous questions**. You will have to:

1. Define the state-transition model $\mathbf{F}$, the control-input model $\mathbf{B}$ and observation model $\mathbf{H}$
2. Implement the motion model function
3. Implement the prediction step using the motion model function inside the `kalman_filter` function
4. Compute the Kalman gain K, the posterior for the filter state and the covariance matrix (P) inside the `kalman_filter` function

Now, adjust the following parameters and observe the behavior of the filter.

- Motion noise (σ_v)
- Measurement noise (R)
- Measurement frequency/size of timesteps (ΔT)
- Engine thrust
- Error in the initial state estimate

Q2.4 Comprehension Questions

Q2.4.1 What happens to the Kalman Gain K if the sensor noise R is increased?

Answer: As the sensor noise increases, the Innovation covariance S increases. The Kalman Gain K depends on the inverse of the Innovation covariance, it therefore decreases. This has the effect, that measurements are weighted less and filter overall relies more on the predictions of the motion model.

Q2.4.2 Assume now that the Spaceship can in addition move laterally above the surface (i.e. in the yz-plane). If we add these dimensions to our EKF, what will happen to its position belief in each individual axis?

Answer: Even though our distance sensor is angled at 45 deg, with our known map of the flat lunar surface, we can not obtain any information about the lateral movement of the spaceship from the available measurements. This means, that neither of the lateral axes (y and z) are observable. The Kalman filter will only perform *prediction* steps for these dimensions, and the state belief will diverge quickly. The vertical x axis remains unchanged, and is still observable via the distance sensor.