Probabilistic Robotics, BAIPR6, Fall 2010

Exam, Thursday December 23th, 9:00 - 12:00, room A1.06

Arnoud Visser

December 22, 2010

Question 1

(a) *EKF versus Graph SLAM*

Describe the essential difference between the two algorithms described in Chapter 10 and 11 of the book. In addition, describe the consequence of this difference.

(b) *Graph versus SEIF SLAM*

Describe the essential difference between the two algorithms described in Chapter 11 and 12 of the book. In addition, describe the consequence of this difference.

(c) *EKF versus Fast SLAM*

Describe the essential difference between the two algorithms described in Chapter 10 and 13 of the book. In addition, describe the consequence of this difference.

Note

For each of the previous questions, give enough detail to demonstrate that you are comfortable with this subject. On the other hand, keep your answer concise, because your description can be easily become fishy from a few incorrect associations.

Question 2

Suppose an indoor robot uses sonar sensors with a 15 degree opening cone, mounted on a fixed height so that they point out horizontally and parallel to the ground. This is a common configuration for an indoor robot. Discuss what happens when the robot faces an obstacle whose height is just below the height of the sensor (for example, 15 cm below). Specially, answer the following questions:

- **(a)** Under what conditions will the robot detect the obstacle? Under what conditions will it fail to detect it? Be concise.
- **(b)** What implications does this all have for the binary Bayes filter and the underlying Markov assumption? How can you make the occupancy grid algorithm fail?
- **(c)** Based on your answer to the previous question, can you provide an improved occupancy grid mapping algorithm that will detect the obstacle more reliably than the plain occupancy grid mapping algorithm?

Question 3

Suppose that a robot operates in the environment shown in Figure 1 using the same motion and sensing model as below. Design an algorithm that is as simple as possible and successfully localizes the robot, regardless of its initial state. Assume as initial condition that there is an uniform distribution over all 7 grid-positions and all 4 orientations (28 states: $1^+ \cup 2^+ \cup 3^+ \cup 4^+ \cup 5^+ \cup 6^+ \cup 7^+$).

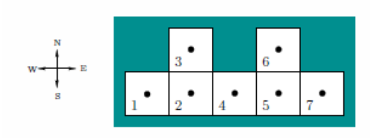

Figure 1: An environment for grid-based localization (Courtesy LaValle [1])

Thus, a state, $x \in X$, is written as $x = (p, d)$, in which p is a position on the grid and d is one of the four directions. The directions are labeled N, E, W, and S, for 'north', 'east', 'west', and 'south', respectively. States can be combined into sets, for example $3^{\dashv}$ indicates the set of states where position $p = 3$ and the orientation $d = \{N, W, S\}$.

The robot is given four actions, $U = \{F, B, R, L\}$, which represent 'forward', 'backward', 'right' and 'left', repsectively. These motions occur with respect of the current orientation of the robot, which may be unknown. In Figure 2.a this motion model is illustrated. For the F action, the robot moves forward one grid cell and maintains its orientation. For the B action, the robot changes its orientation by 180 degrees and then moves forward. For the R action, the robot turns right by 90 degrees and than moves forward. The L action behaves similarly. If it is not possible to move because of an obstacle, it is assumed that the robot changes its orientation (in the case of B, R, or L), but does not change its position. This is depicted in Figure 2.b.

(a) The four possible actions; take one step, if possible, and reorient the robot

(b) If a direction is blocked because of an obstacle, then the orientation changes, but the position remains fixed

Figure 2: Motion model of the robot (Courtesy LaValle [1])

The robot has one simple sensor that can only detect whether it was able to move in the direction that was attempted. The sensor space = $Y = \{0, 1\}$. This yields $y = h(x_{k-1}, x_k) = 1$ if x_{k-1} and x_k are different positions, and $h(x_{k-1}, x_k) = 0$ otherwise. Thus the sensor indicates whether the robot has moved after the application of an action.

The algorithm doesn't have to be complete, it is enough to extend the decision tree until a quarter of the state space can be identified.

Success!

References

[1] S. M. LaValle, *Planning Algorithms*, Cambridge University Press, 2006.