

Autonomous Robotic Systems

Bayes Filter

Christoph Emunds (i6146758)

Dominik Nerger (i6146759)

March 18, 2017

Contents

1	Introduction	1
2	The world model	1
3	The Bayes Filter	2
3.1	Update step	2
3.2	Measurement step	3
4	Results	3
5	Conclusion	3

1 Introduction

This report is part of the third assignment for the course Autonomous Robotic Systems. In this report, we present our implementation of a Bayes Filter for a world model containing a corridor with several doors. Furthermore, we discuss the results that we obtained.

2 The world model

The robot can move between 10 grid cells in the world model. Because the robot is able to traverse through them in both directions, 20 states are available.

Therefore, the world model needs to store information about 20 states. It is necessary that through the sensing measurements it can be deduced which states match the measured state because this needs to be used in the calculations for the sensing update.

The possible values for each state are represented in the enum *WORLD_MEASURE*. This enum contains the following four values: *L,LR,LRF* and *R*

The characters in each value represent whether a wall has been measured on the left (L), right (R) or front (F) side, with a combination indicating that multiple walls have been found.

Therefore, the world model is represented through a vector with size 20 and information about which measurements should be true in this state contained in the respective enum value.

3 The Bayes Filter

The Bayes Filter is initialized with a uniformly distributed belief state, i.e. 0.05 for each state because there are 20 states.

3.1 Update step

The update step is executed whenever the robot moves forward or makes a 180 degree turn. To calculate the probability for being in a state x after a motion command, the probabilities of all possible starting states is weighted according to the given transition probabilities and summed up. For the *move forward* command, this equates to:

$$p(x_i) = 0.8 \times p(x_{i-1}) + 0.1 \times p(x_{i-2}) + 0.1 \times p(x_i)$$

The world model is not cyclic, therefore two additional calculations are necessary for certain states. States for which $i\%10 == 1$ holds true, the equation is:

$$p(x_i) = 0.8 \times p(x_{i-1}) + 0.1 \times p(x_i)$$

Furthermore, the equation for states for which $i\%10 == 0$ holds true is:

$$p(x_i) = 0.1 \times p(x_i)$$

This is necessary because, after a movement update, the states 0 and 10 can only be reached by being in the respective state already and the move failing. The same holds true for states 1 and 11 , which can be reached by correctly moving from states 0 or 10 or failing to move when they are in their state already.

For the *turn* command, the equation results in:

$$p(x_i) = 0.9 \times p(x_{N-1-i}) + 0.1 \times p(x_i)$$

3.2 Measurement step

The probability for being in a state that is conform with the measurement is multiplied by a factor of 0.6, the probability for every other state is multiplied by 0.2.

4 Results

Most states in the corridor have a wall on both sides, which makes them indistinguishable. When issuing the *measurement* command in a state that has a door, the probabilities of the three states containing a door raises. Multiple continuous measurements cause the robot's confidence for being in one of these three states to grow.

5 Conclusion