Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Lecture: Introduction to Mobile Robotics
Summer term 2014
Prof. Dr. Wolfram Burgard
Juniorprof. Dr. Maren Bennewitz
Dr. Gian Diego Tipaldi
Dr. Luciano Spinello

# Sheet 8

Topic: Velocity Motion Model, Discrete Filter

Submission deadline: June 17, 2013

Submit to: `mobilerobotics@informatik.uni-freiburg.de`

## Exercise 1: Velocity-Based Motion Model

*Remark: This exercise is to be solved without Octave.*

Consider a robot which moves on a circular trajectory with noise-free constant velocities $(v, w)$ (this situation is shown on page 30 of the *Probabilistic Motion Models* slides). The current pose of the robot is $(x, y, \theta)$.

(a) Derive the following expression for the center of the circle, $(x_c, y_c)$:

$$\begin{pmatrix} x_c \\ y_c \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} -\frac{v}{w} sin\theta \\ \frac{v}{w} cos\theta \end{pmatrix}$$

(b) Now consider the situation where we are given a start pose $(x, y, \theta)$ and an end pose $(x', y', \theta')$, connected by a circular movement. Prove that the center of the circle can be expressed as

$$\begin{pmatrix} x_c \\ y_c \end{pmatrix} = \frac{1}{2} \begin{pmatrix} x + x' \\ y + y' \end{pmatrix} + \mu \begin{pmatrix} y - y' \\ x' - x \end{pmatrix} \tag{1}$$

with some $\mu \in \mathbb{R}$

*Hint*: The center of the circle lies on a ray that lies on the half-way point between $(x, y)$ and $(x', y')$ and is orthogonal to the line between these coordinates. Use the parametric equation for a line to represent this ray.

(c) Show that the value of $\mu$ is given by

$$\mu = \frac{1}{2} \frac{(x - x')cos\theta + (y - y')sin\theta}{(y - y')cos\theta - (x - x')sin\theta}.$$

*Hint*: $\mu$ can be calculated by using the fact that the line described by equation (1) and the line from $(x_c, y_c)$ to $(x, y)$ intersect at $(x_c, y_c)$.

**Exercise 2: Discrete Filter**

In this exercise you will be implementing a discrete Bayes filter accounting for the motion of a robot on a 1-D constrained world.

Assume that the robot lives in a world with 20 cells and is positioned on the 10th cell. The world is bounded, so the robot cannot move to outside of the specified area. Assume further that at each time step the robot can execute either a *move forward* or a *move backward* command. Unfortunately, the motion of the robot is subject to error, so if the robot executes an action it will sometimes fail. When the robot moves forward we know that the following might happen:

1. With a 25% chance the robot will not move
2. With a 50% chance the robot will move to the next cell
3. With a 25% chance the robot will move two cells forward
4. There is a 0% chance of the robot either moving in the wrong direction or more than two cells forwards

Assume the same model also when moving backward, just in the opposite direction.

Since the robot is living on a bounded world it is constrained by its limits, this changes the motion probabilities on the boundary cells, namely:

1. If the robot is located at the last cell and tries to move forward, it will stay at the same cell with a chance of 100%
2. If the robot is located at the second to last cell and tries to move forward, it will stay at the same cell with a chance of 25%, while it will move to the next cell with a chance of 75%

Again, assume the same model when moving backward, just in the opposite direction.

Implement in `octave` a discrete Bayes filter and estimate the final belief on the position of the robot after having executed 9 consecutive *move forward* commands and 3 consecutive *move backward* commands. Plot the resulting belief on the position of the robot.

*Hints*: Start from an initial belief of:

```
bel = [ zeros(10,1); 1; zeros(9,1) ];
```

You can check if your implementation has bugs by noting that the belief needs to sum to one (within a very small error, due to the limited precision of the computer). If it doesn't there is a mistake.

Careful about the bounds in the world, those need to be handled ad-hoc.