Albert-Ludwigs-Universität Freiburg
Lecture: Introduction to Mobile Robotics
Summer term 2016

`mobilerobotics@informatik.uni-freiburg.de`

Institut für Informatik

Prof. Dr. Wolfram Burgard
Dr. Michael Ruhnke
Dr. Bastian Steder

Sheet 11

Topic: FastSLAM

Due date: 08.07.2016

Exercise: FastSLAM Implementation

Implement the landmark-based FastSLAM algorithm as presented in the lecture. Assume known feature correspondences.

To support this task, we provide a detailed listing of the algorithm as a PDF file and a small *Python* framework (see course website). The framework contains the following folders:

data This folder contains files representing the world definition and sensor readings used by the filter.

code This folder contains the FastSLAM framework with stubs for you to complete.

To run the FastSLAM , just run in the terminal `python fastslam_framework.py sensor.dat world.dat N` , where N is the number of the particles. *Note: You first have to complete the function correction_step in order to get the framework working correctly.* We are not using any external library for visualizing the particle cloud. The stub for plotting the particles is included in the framework. Once you have completed the framework, you should be able to see a simple matplotlib figure popped up for visualization. Some implementation tips:

- Turn off the visualization to speed up the computation by commenting the lines in the function `get_position`.

- To read in the sensor and world data, we have used dictionaries. Dictionaries provide an easier way to access data structs based on single or multiple keys. The functions `read_sensor_data` and `read_world_data` read in the data from the files and build a dictionary for each of them with timestamps as the primary keys. To access the sensor data from the data_dict, you can use
```
data_dict[timestamp,'sensor']['id']
data_dict[timestamp,'sensor']['range']
data_dict[timestamp,'sensor']['bearing']
```

and for odometry you can access the dictionary as `data_dict[timestamp,’odom’][’r1’]`
`data_dict[timestamp,’odom’][’t’]`
`data_dict[timestamp,’odom’][’r2’]`

To access the positions of the landmarks from world_dict, you can simply use

`world_dict[id]`