Lab3 - Particle Filter

1 Introduction

This lab is designed to implement a Particle Filter or Montecarlo Localization (MCL) algorithm to localize a two dimensional robot (turtlebot) in a given map. The bagfile provided is the same as in the previous lab.

1.1 Particle Filter

The MCL localization is an implementation of the Markovian localization problem where the involved pdfs are represented through samples (particles) and the Bayes filter is implemented through the Particle Filter. Markov localization addresses the problem of state estimation from sensor data. Markov localization is a probabilistic algorithm: Instead of maintaining a single hypothesis as to where in the world a robot might be, Markov localization maintains a probability distribution over the space of all such hypotheses. The probabilistic representation allows it to weigh these different hypotheses in a mathematically sound way.

2 Pre-lab

Review the particle filter explained on lectures and write the prediction step in Python code, given the particles position in (x, y) as a numpy array named `p_xy` of shape $2 \times N$ and their orientations being an array named `p_ang` of dimension N.

3 Lab work

The folder named `particle_filter` contains the code for running this lab and a bagfile obtained from a turtlebot with `nav_msgs/Odometry` and `sensor_msgs/LaserScan` messages. You need to copy your `splitandmerge.py` solution into the `src/` folder. Then you can test the code by coping the folder in your workspace and running:

```
roslaunch particle_filter particle_filter.launch
```

As you can see the dummy Particle Filter algorithm implemented, only initializes the particles along the extension of the map, nothing else. The three parts of the filter that must be implemented are:

Prediction Compute the new position (x, y, θ) of the particles given an odometry measurement $(\Delta x, \Delta y, \Delta\theta)$ in the vehicle frame. Do not forget to add the odometry Gaussian noise in the increments.

Weighting Compute the weight of each particle by comparing the lines obtained from the measurements with the lines of the given map. To be compared, the lines have to be transformed from the form [x1 y1 x2 y2] to the representation [range angle] defined by the shortest distance from the origin to the line (perpendicular distance) being the range, and the angle of this range seen from the origin (Fig. 1).

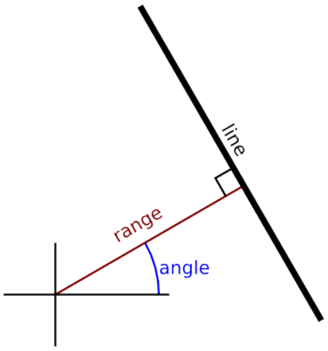

Figure 1: Representation of a line in (range, angle).

The comparison must be computed with a Gaussian as

$$w = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right] \tag{1}$$

being x the measured value (range or angle), μ the expected value (extracted from given map lines) and σ the uncertainty of the measurement. Since range and angle uncertainties are independent they can be multiplied to obtain the final weight.

Lines measured can be compared to all lines in the map and take the best results or associating them by searching the most likely one and then computing the weight. Remember that map lines are represented on the world frame and the measured lines in the robot frame (each particle own frame).

Resampling Systematic resampling, as explained in lectures, will be used in this lab.

4 Optional

Improve the localization by taking into account if the segments extracted from the measurements coincide with the segments in the given map. Not only compare the representation in range and angle (Fig. 2). This will discern more properly which measurements can be associated with which map features.

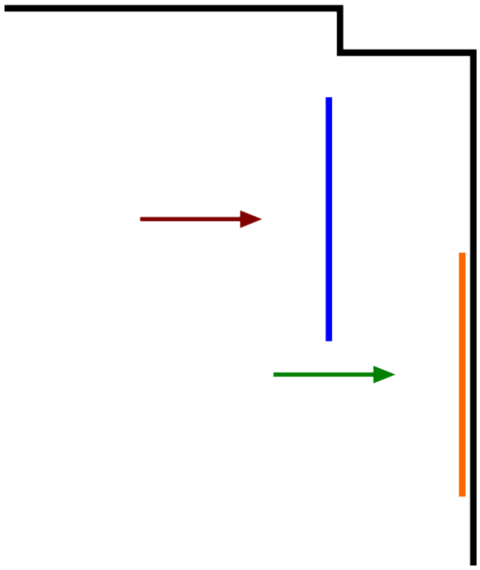

Figure 2: If only range and angle representation of lines is used for weighting, both red and green particles have the same weight because they coincide with lines at same angle and distance (red with the small line on top). If segments are checked, the blue measurement of the red particle cannot belong to the small line on top and its weight will be smaller than the green particle whose orange measurement agrees completelly with the given map.

5 Lab report

Write a brief report (maximum 2 pages) explaining your solution and problems faced. Include the final code.