Lab3 - Particle Filter

1 Introduction

This lab is designed to implement a Particle Filter (PF) or Montecarlo Localization (MCL) algorithm to localize a two dimensional robot (Turtlebot) in a given map. The bagfile (idealized) needed is the same as the one 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 Laboratory study work

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

3 Laboratory implementation work

The folder named `Lab3_particlefilter` contains the code for running this lab. Your solution of the previous lab `lab2_splitandmerge` will be called to provide the lines input for the particle filter localization. You can test the code by running:

```
roslaunch lab3_particlefilter particle_filter.launch
```

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

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). (The function *get_polar_line* can be very useful).

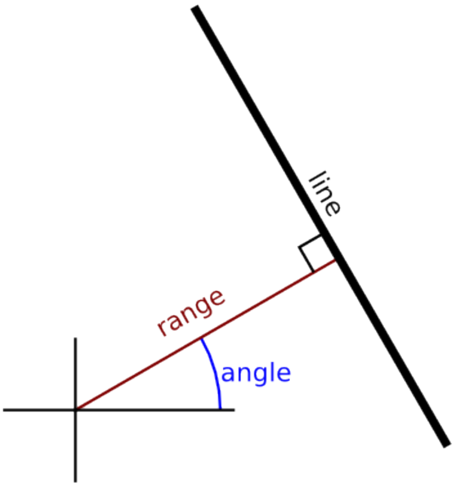

Figure 1: Representation of a line in (range, angle).

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. We will consider that range and angle uncertainties are independent so, they can be multiplied to obtain the weight for each measured line.

Lines measured can be compared to all lines in the map and take the best results. Remember that map lines are represented on the world frame and the measured lines in the robot frame (each particle own frame). When multiple lines are sensed, we consider them independent to compute a final weight for each particle.

Resampling Systematic resampling will be used in this lab.

4 Optional

Undergraduate students Move the Turtlebot around the lab and record a rosbag containing the data extracted from the `sensor_msgs/LaserScan`, the odometry `nav_msgs/Odometry`. Place the rosbag in the folder `engr4200_basics/bags/`. Test your Particle Filter with this new data (You will need to modify the file: `launch/lab2_splitandmerge.launch`).

Graduate students 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. This will discern more properly which measurements can be associated with which map features.

5 Lab report

Write a brief report (maximum 2 pages) explaining your solution and problems faced. Include the final code in the zip file (not in the report).