LAB 3: Particle Filter

Introduction

In this lab, we will implement a **Particle Filter** method for the Differential Drive Mobile robot, using the input of the wheel encoders and measurements with respect to an a priori known map of two dimensional cartesian features.

Cloning the PR_LAB3 Repository

In this section, we will guide you through the process of cloning the “PR_LAB3” repository from GitHub.

1. **Open a Terminal or Command Prompt:** Depending on your operating system (Windows, macOS, or Linux), open a terminal or command prompt.

2. **Navigate to the Desired Directory:** Use the ‘cd’ command to navigate to the directory where you want to store the repository. For example:

   ```
   $ cd Documents
   ```

3. **Clone the Repository:** Run the following command to clone the “PR_LAB3” repository from GitHub:

   ```
   $ git clone https://github.com/IFROS-MIRS/PR_LAB3.git
   ```

4. **Verify Cloning:** Once the cloning process is complete, navigate to the cloned repository’s directory using the ‘cd’ command:

   ```
   $ cd PR_LAB3
   ```

   You should be inside the cloned repository. Check the contents to verify that everything has been cloned currently.

That’s it! You’ve successfully cloned the “PR_LAB3” repository onto your computer. You can now work on the contents of the repository or use them for your laboratory exercise or project.

Updating with your solutions of LAB1, LAB2

Once you have cloned the repository, you need to update the python files that you have already programmed during the previous sessions. This includes the files:

1. DifferentialDriveSimulatedRobot.py
2. DR_3DOFDifferentialDrive.py
3. Pose3D.py

Note: DR_3DOFDifferentialDrive.py is not explicitly used but some parts will be used!

PF: Part 1 (Prediction)

In this part of the exercise we are going to implement the Prediction step of the particle filter. To do it we need to follow the following steps:

1. Program the method "*Prediction*()" within the MCLocalization class.
2. Program the method *MotionModel*() within the *PF_3DOF_DR* class.
3. Program the method *GetInput*() within the *PF_3DOF_DR* class.
    - Note that we don't just use the GetInput of the *DR_3DOFDifferentialDrive* since we need to return 2 elements, the input vector **uk** and its covariance **Qk.**
4. Program the main function to test the Prediction of the Particle Filter:
    a. Create array of a few particles distributed around the initial position x0 with probability P0
    b. Instantiate the *PF_3DOF_DR* object with the given particles
    c. Call the LocalizationLoop to see how the particles evolve over time as the robot moves

PF: Part 2 (Update)

In this part, we are going to implement the Update step of the particle filter. Here, particles are going to be "weighted" based on their likelihood of being the true state of the system given the sensor measurements. To implement it, we need to program the following methods:

1. Program the method *GetMeasurements()* within the *PF_3DOF_MBL* class. Use the *ReadRanges()* of the *DifferentialDriveSimulatedRobot* that you programmed in LAB2.
2. Program the *Update()* method within the *PFMBLocalization* class.
3. Program the *Weight()* method within the *PFMBLocalization* class.
4. Program the *Resample()* method within the *PFMBLocalization* class.
5. Program the main function to test the full Map Based Localization - Particle Filter:

a. Create array of a few particles distributed around the initial position x0 with probability P0
b. Instantiate the *PF_3DOF_MBL* object with the given particles
c. Call the LocalizationLoop to see how the particles evolve over time as the robot moves