ROB530-HW5-Localization

For this assignment, we will use ROS and Python3 to execute the robot localization task.

Dependencies

System

The system dependency preparation depends on your local system. Basically, you need to install Ubuntu to complete this homework. We recommend `Ubuntu 20.04`.

- Linux (Ubuntu)
  - There is no preparation needed. Continue to install ROS :)
- Windows
  - You can set up a dual-boot system with Ubuntu.
  - You can use `Windows Subsystem for Linux` to install Ubuntu. You can check instructions on the official website or the recitation.
  - You can use virtual machine to install Ubuntu. The VirtualBox is free.
- Mac
  - You can use `Bootcamp` to install set dual-boot system with Ubuntu.
  - You can use virtual machine to install Ubuntu. The VirtualBox is free. The Parallels Desktop is also good.

ROS

You also need to install ROS (Robot Operating System) after installing Ubuntu. `ROS Noetic` is matched with the recommended `Ubuntu 20.04`. You can find the detailed installation instructions on ROS Wiki. You can also watch or check slides of this recitation.

Python Packages

These packages are required. You can install them by typing `pip install $package name$`.

- NumPy
- SciPy
- PyYAML
- Matplotlib

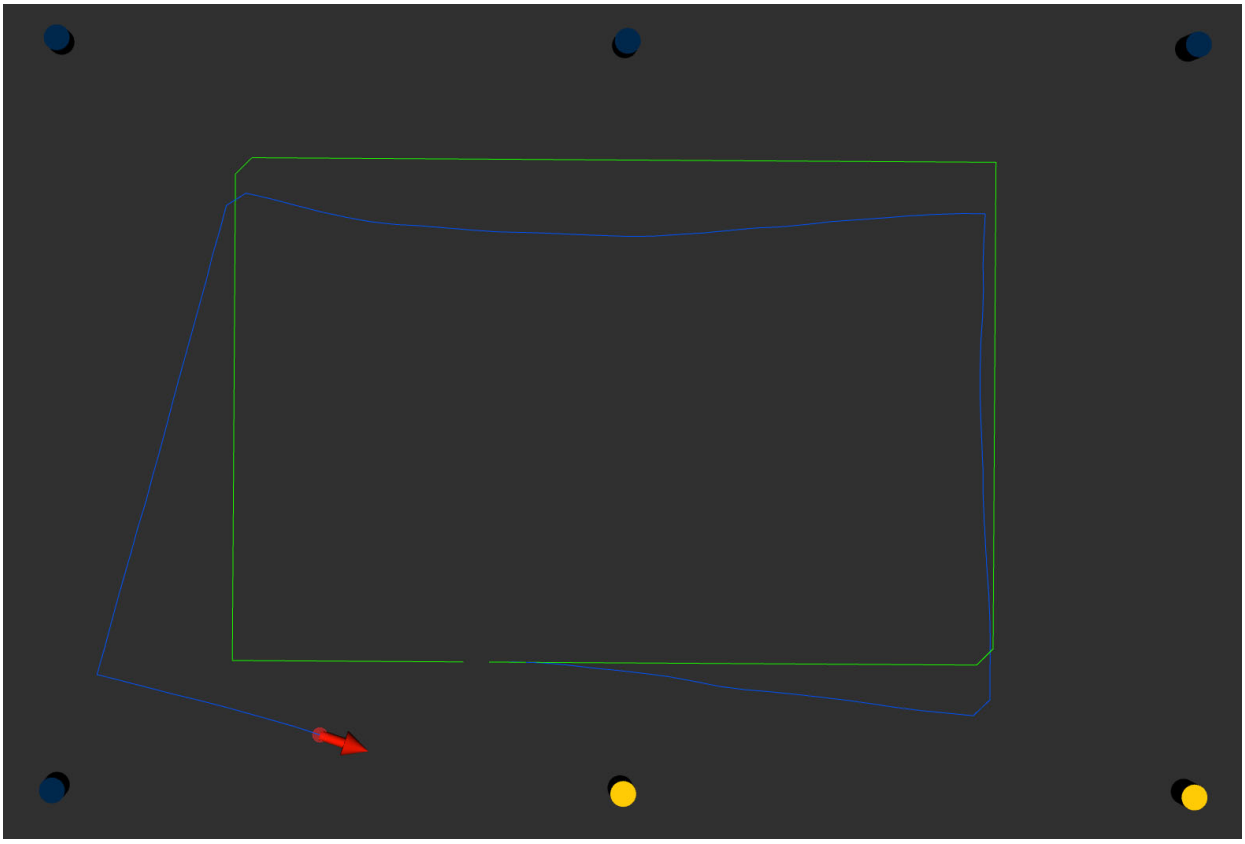
Test Your Setup

We provide a dummy filter which you can run to test if you have set up your environment correctly.

1. Open a terminal, run `roscore`.
2. Check `config/settings.yaml`, ensure the `filter_name` is set to `test`.
3. Open a new terminal, run `rviz`.
4. We open a visualization config file. In your rviz, click `file -> open config`, choose `rviz/default.rviz` in the homework folder.
5. Open a new terminal, run `python3 run.py`.

6. You should be able to see your a robot moving in `rviz`.

You should expect to see the visualization shown below. In this figure, `green path` represents command path without action noise, which is the path we want our robot to follow. `blue path` represents the exact path that the robot moves due to action noise. The `red ellipse` and the `red arrow` represent the filter prediction pose for the robot.



Start Working

Now you have everything ready. You can start reading the assignment instructuons and start implementing your filter. Write your code and adjust the config settings before testing.

Configurations

Parameters can be modified in `config/settings.yaml`.

You will only need to modify `filter_name` and `Lie2Cart`.

- `filter_name`: The filter you would like to run. Options include: `EKF`,`UKF`, `PF`, `InEKF`, and `test`.
- `Lie2Cart`: Set to `True` if you finish implementing the extra points question 2.E.

Files you need to implement

- Implement all four filters.
  - `filter/EKF.py`
  - `filter/UKF.py`
  - `filter/PF.py`
  - `filter/InEKF.py`

- Extra points
  - In `utils/util.py`, finish
    - `func()`
    - `lieToCartesian()`
    - `mahalanobis()`