Localization Lab 2

Preparation

1. Download `LocalizationLab2.zip` from Hippocampus.
2. Create a folder for the lab.
3. Decompress the zipped file into the lab folder.

Lab contents

This lab continues to explore the use of the Kalman filter in localization problems. It illustrates a frequently used technique when some parameter which is important for localization is:

- Difficut to measure, or
- Slowly evolves over time.

A classical example of such a parameter is the wheel radius of a tyre. Indeed, the radius of a type evolves over time due to various phenomena: temperature, and hence pressure evolution, slow loss of pressure, wear... Now tyre radius is an important parameter in situations where odometry is used in addition to exteroceptive sensors such as GPS.

A classical approach consists in identifying those parameters online, by adding them to the state vector. In practice, observability problems may arise, which are not deal with in the present lab (the interested reader may refer to the chapter of the book entitled "Localization and observability").

Of course, it is necessary to have an evolution model for the parameter, to become part of the evolution equation of the system. For parameters which are known to evolve very slowly, it is classical to use a constant model. The discrete evolution equation of the parameter then writes $p_{k+1} = p_k$. The state noise and initial noise will allow the parameter to evolve at each iteration.

Work to be done

Have the teacher validate each of the following steps:

- Define a state vector and an evolution equation for the same localization problem as in lab 1, with online identification of the radii of the wheels of the robot.
- Write the expressions of the matrices A and B of the Kalman filter.
- Considering that the robot is equipped with the same sensor as in lab 1, what does the measurement equation become?
- Write the new matrix C of the Kalman filter.

Programming the filter with online radius estimation:

- Using the results established above, complete the Matlab programs.
- Set initial noise values for the radii and set the state noise to zero. Set `sigmaWheels` to any non zero value. Do the same for the initial noise on x, y and θ. Then execute the program and print the matrix P during execution (just remove the ";" at the end of the line which calculates P). What do you observe?
- Form this observation, infer a way to tune the noises (other than the initial noises). You must obtain a very simple way of tuning the noises. Explain your method in a brief and clear way to the teacher for validation.
- Tune the noises according to your methodology.
- Compare the results to those obtained in lab 1. For the case at hand, what is the interest of online radius estimation?