ENMT482 Turtlebot Lab & Assignment 3 Brief

The lab and the assignment will be done in pairs. It would be to your advantage to ensure that at least one of you is comfortable with the Linux command line and programming.

Turtlebot Lab

The lab is in the Automation Lab, downstairs in the ECE wing.

Tasks:

1. Get set up with a Turtlebot and one of the lab computers. You may use your personal computer if you like (ask me how), but in previous years this has gone badly for many students.
2. Teleoperate around the lab/corridor, using Rviz to view the sensor outputs.
3. Use `gmapping` to build a map of the area.
4. Using that map along with `amcl` and the navigation stack, drive the Turtlebot around the lab/corridor using nav goals.

See the lab instructions for more details.

Assignment, Part 1

I have replicated the example in Lecture 1 using a Turtlebot robot. It has an assortment of range sensors pointing at a wall located at $x = 0$, and can be commanded to move back and forth along the x-axis up to about $x = 3$.

There are six sensors:

- sonar1 is an HC-SR04 sonar rangefinder (0.02–4m)
- sonar2 is a MaxBotix HRUSB-MaxSonar-EZ MB1433 sonar rangefinder (0.3–5m)
- ir1 is a Sharp GP2Y0A02YK0F infrared rangefinder (0.15–1.5m)
- ir2 is a Sharp GP2Y0A41SK0F infrared rangefinder (4–30cm)
- ir3 is a Sharp GP2Y0A21YK infrared rangefinder (10–80cm)
- ir4 is a Sharp GP2Y0A710K0F infrared rangefinder (1–5m)

Note that aside from sonar2, all of these sensors are Chinese knockoffs and thus their actual performance characteristics may not match their specifications.

Your task is to use the measurements from these sensors along with the commanded speed to estimate the robot's position. You will be provided with three datasets as CSV files, each containing true position, commanded velocity, and the sensor measurements. Additionally, you will be provided with a test dataset which does not contain the true position, which you will use for your report.

Some Python example code will be provided, but you may use another language if you prefer (although we may not be able to help).

Assignment, Part 2

I have replicated the example in Lecture 9 using a Turtlebot robot, with fiducial markers for the beacons. You can see these markers placed around the Automation lab and the hallway, along the route we used for the lab. An image of one of these markers can be used to estimate the six-degree-of-freedom transformation between the robot and the marker, but for our purposes I've just kept x, y and rotation. The variance of the estimate is approximately proportional to the square of the distance between the robot and the marker.

Your task is to implement a particle filter which uses the beacon observations and either odometry or commanded velocity/rotation rate to estimate the robot's location. You will be provided with one dataset as a CSV file, containing a "true" position from SLAM, estimated positions from odometry, the commanded forward speed and rotation rate, and the ID and position of any beacon observations. When two beacons are visible to the camera, there will be two rows with the same time and position but different beacon observations. The co-ordinate system used follows the ROS standard (x forward, y left). Again, some Python example code will be provided.

If you wish, you may use the true position to initialise your particles, but you must not use it after that. For extra credit, make your implementation robust to unknown starting positions.

In the video on Learn, I'm using the range-bearing sensor model from the lecture notes, and my motion model applies the difference between consecutive odometry positions to each particle with some additional noise.

The SLAM algorithm used is RTABMAP, which is generally a bit better than `gmapping` but gets confused around the identical desks. In the example code I've just dropped the bits where it jumps around; it's pretty good the rest of the time.

Report and Code submission

Your code is due at the end of Week 6, along with one short report (four A4 sides max, excluding optional appendix) and a peer assessment each (one A4 side max).

The report should include:

- A title page with your names and group number (and a link to your git repository, if you have one).

For part 1:

- An explanation of your motion and sensor models.
- An explanation of the estimation approach or approaches you took.
- A description of what you thought worked well about your estimation approach, and what you could do to improve it.
- Plots of how close your estimate was to the true position, for the datasets with true position.
- A plot of your estimate and its variance for the test dataset.

For part 2:

- A brief explanation of your motion and sensor models.
- A description of what you thought worked well about your estimation approach, and what you could do to improve it.
- Plots of your estimated trajectory alongside the position from SLAM.

Additionally, you should include the map of the Automation Lab and corridor you generated using `gmapping` in the lab, and your observations regarding `gmapping`'s performance.

The peer assessments should include:

- A summary of what you (personally) did.
- How you would divide 30 bonus marks between yourself and your partner.
- Your suggestions for how we could improve the assignment.

All submissions are through Learn. Please submit the report and peer assessment as PDF files. You may submit your code by either including a link to the Gitlab project for your group's git repository in your report, or uploading a zip file of your code alongside your report. Make sure you add Matthew Edwards (mje) and Michael Hayes (mph) to the "Members" list of your project so we can actually see it.