Institute for Aerospace Studies
UNIVERSITY OF TORONTO

Final Design Project

The Galbraith Memorial Mail Robot

ROB301 Introduction to Robotics
2022

1 Introduction

This course has exposed you to a number of challenges in the field of robotics. From dead-reckoning locomotion and PID control to Kalman filtering and localization, these techniques have been developed to allow robots to "see" and "do" things in the real world. In the Final Design Project for ROB301, you are asked to design and test a control system to enable a Turtlebot 3 Waffle Pi robot to simulate mail delivery. Yes, this will be another mail-delivery robot but the requirements are quite different than those you faced in Lab IV.

Your robot will be required to deliver mail on a closed-loop path to any one of several offices. You may think in terms of the quadrilateral hallway on any floor of the Faculty of Applied Science and Engineering's Galbraith Building, where offices line the inner side of the loop. As a consequence, and in recognition of the Faculty's first Dean, John Anderson Galbraith (1846–1914), we are calling this project, "The Galbraith Memorial Mail Robot." (Galbraith was, in fact, the first professor appointed to the School of Practical Science, "Skule," the predecessor of the Faculty.)

A topological map of the hallway will be provided to you, which will consist of a series of colored regions on the floor representing the locations of the offices connected by black tape to provide a guide for your robot. The offices (numbered 2 to 12, inclusive) are unique but the colors are not; in fact, there will be only four unique colors in all. As a result, you are expected to use Bayesian localization fused with the line-following control developed in Lab III to facilitate your design.

The final proof-of-concept demonstration will require your robot from an arbitrarily chosen starting point to deliver three pieces of mail to three different offices, also arbitrarily chosen.

2 Objective

The overall objective of this project is

- ▷ *To design a control system, based on Bayesian-localization techniques, for the Turtlebot 3 Waffle Pi robot to deliver mail to arbitrarily chosen stations on a closed-loop route.*

As stated, the line-following control algorithm developed in Lab III will be indispensable to this work.

3 Requirements

3.1 Equipment & Software

As usual in the ROB301 laboratory sessions, you will be required to use the Turtlebot 3 Waffle Pi robot. As a reminder from Lab I, the following commands should be run in separate terminal windows sequentially to start the ROS program properly (you should keep these terminal windows open as they are continuously running the program):

`$ roscore` [Remote PC]

`$ roslaunch turtlebot3_bringup turtlebot3_robot.launch` [TurtleBot]

`$ roslaunch camera camera.launch` [TurtleBot]

The robot's camera is to be used as a color sensor both to distinguish the different color patches and to detect the path marked with tape. It will output two values, one being a line sensor (through the topic `/line_idx`), which outputs the index with the least light intensity in the camera image array (corresponding to the location of the black tape), and the other being an RGB value, indicating the mean pixel intensity in the current image (published via the `/mean_img_rgb`).

Lastly, a starter code script (`final_project.py`) for the project is provided in the Catkin Workspace within the `final_project/nodes` directory. You can run this script by entering the following command in a terminal:

`$ rosrun final_project final_project.py` [Remote PC]

3.2 Map

The map for the mail route will be as shown in Figure 1. A colored rectangular patch indicates the location of an office. The offices, of which there are 11, are uniquely numbered (from 2 to 12) but there are only four colors (blue, green, yellow, orange); the same color can therefore indicate different offices. The color patches are connected by black tape, which can be used to guide the robot on the route. Note, however, that the spacing of offices will not necessarily be uniform. The given map must be treated as a topological map. You may find a character array representation of the map to be useful.

As with any complicated task, it will be helpful to subdivide your controller design into various subsystems which are responsible for different tasks. Some of these subsystems have already been completed for you. In particular, starter code will be supplied that reports the RGB color code the camera is seeing on the floor. It also tells you the location of the most prominent line in the field of view of the camera. In addition, however, it will be necessary

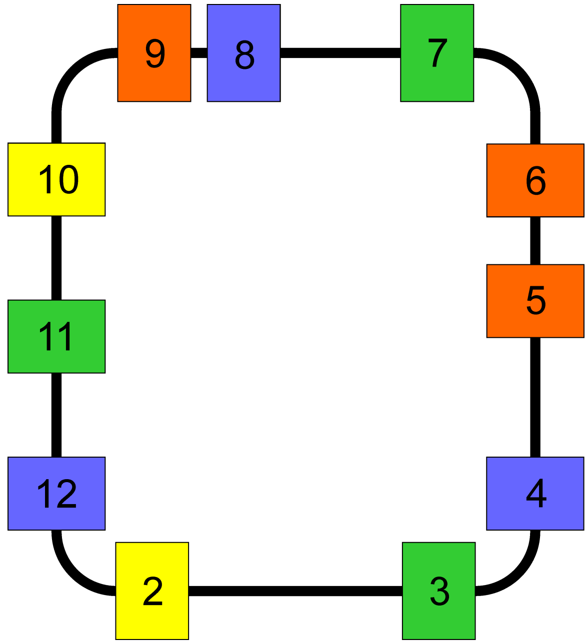

Figure 1. Sample topological map

- ▷ To develop a system that processes the camera data to determine if the robot is at the doorway of an office (a color patch) or between offices along the hallway (the black tape)
- ▷ To localize your robot based on some model of the robot, sensor, and a map of the environment, and
- ▷ To navigate and move to the desired offices.

The robot is to mimic delivery of a parcel of mail by stopping at the desired office, rotating 90° to the left, pausing and then rotating back to continue.

4 Project Deliverables and Milestones

This project requires a number of deliverables. These deliverables are established to correspond to specific subsystems of the control system. Designing, coding, debugging and testing individual subsystems is significantly easier than debugging the system as a whole. Each deliverable will be required to be demonstrated to the TA; there will be a performance mark based on the completion of each subsystem.

The specific deliverables are as follows.

4.1 Deliverable 1: Bayesian-Localization Code

Code for and simulation of a generic Bayesian-localization algorithm suitable for the project design.

The scenario that is to be simulated is specified in the Attachment. This is to be completed in Python (no ROS is required) and it will serve as a starting point for your localization framework. A simulation run of the code is to be presented to the TA. Only one completed assignment is required per group.

Milestone 1: Due within 30 minutes of the first lab session **[14/16 Nov 2022]**.

4.2 Deliverable 2: Full-Route Execution

Demonstration that the robot can complete, without localization, the full route of a given practise course.

In particular, the robot should be able to start at any office location, move ahead until the black line is detected, follow the line until the robot arrives at the next door and so on until the full route has been traversed.

Milestone 2: Due by the end of the first lab session **[14/16 Nov 2022]**.

4.3 Deliverable 3: Bayesian-Localization Node

Completion of the Bayesian-localization node and demonstrate that the state probability converges to the correct office once your robot has passed a few offices.

As there are multiple offices of the same color, the state probability will not converge immediately, so the robot would likely need to pass a few offices before becoming confident about its state estimate.

Milestone 3: Due by the end of the second lab session **[21/23 Nov 2022]**.

4.4 Deliverable 4: Proof-of-Concept Demonstration

Proof-of-concept demonstration of the full control system on the provided test course.

At the midpoint of the third and final lab session all work on the design, coding and testing of the control system will cease and the demonstrations will begin. The demonstrations will be conducted according the following protocol:

1. The final map topology (arrangements of color patches) of the route will be announced to the class. *All groups will then have an opportunity to input the given topology into their controller*.

2. Each group will be called up in turn to perform their proof-of-concept demonstration.

3. Three random offices will be chosen for the group. *The group will have the opportunity to input the offices into their controller*.

4. A random starting point will be selected for the group's robot and a member of the group will position the robot at the starting point.

5. The robot will then be commanded to initiate mail delivery.

6. Each group will have two attempts to execute the task. On the second attempt, if the robot goes off course relative to the black line, the robot may be replaced laterally such that the black line comes back into camera view. *A small penalty will be applied to the demonstration mark for every manual correction that occurs.*

Milestone 4: Due by the end of the third lab session **[28/30 Nov 2022]**.

4.5 Deliverable 5: Final Report

A final report, one per group, summarizing the design and the results.

A written final report, of no more than 5 pages, is to be submitted electronically. The report should include an introduction, brief description of the robot platform (sensors used, etc.), overview of solution strategy, technical details on the design methodology (control, estimation, etc.), summary of your demonstration performance, list of potential improvements for the future, and end with a conclusion highlighting your findings. In an appendix, please include any code that you used for your project.

The report will constitute one-third of the total 15% of the final grade allocated to the final design project.

Milestone 5: Due by **5 Dec 2022, 23h59**.

5 Closing Remark

At the end of this project, you will be able to consider yourself an initiate roboticist! Congratulations!

6 Additional Resources

1. Introduction to ROS, ROB301 Handout, 2020.
2. Robotis e-Manual, `http://emanual.robotis.com/docs/en/platform/turtlebot3/overview/`.
3. "SSH: Remote control your Raspberry Pi," The MagPi Magazine, `https://www.raspberrypi.org/magpi/ssh-remote-control-raspberry-pi/`.
4. Official ROS Website, `https://www.ros.org/`.
5. ROS Wiki, `http://wiki.ros.org/ROS/Introduction`.
6. Useful tutorials to run through from ROS Wiki, `http://wiki.ros.org/ROS/Tutorials`.
7. ROS Robot Programming Textbook, by the TurtleBot3 developers, `http://www.pishrobot.com/wp-content/uploads/2018/02/ROS-robot-programming-book-by-turtlebo3-developers-EN.pdf`.

Attachment

DATA for DELIVERABLE 1

For Deliverable 1, use the topological map shown in Figure 1. Following the format introduced in the Notes on "The Concept of Bayesian Localization," we suggest starting with the state model given in Table 1 and the measurement model in Table 2.

Table 1. State model $p(x_{k+1}|x_k = X, u_k)$

x_{k+1} | $u_k =$	-1	0	$+1$
$X - \chi$	0.85	0.05	0.05
X	0.10	0.90	0.10
$X + \chi$	0.05	0.05	0.85

These models should in fact be derived based on testing. While the motion of the robot may be quite close to deterministic, the measurement model may leave much to be desired. The camera reading may not be that reliable. Testing would reveal what the probabilities should likely be. (Note that in Table 2, "nothing" is to be interpreted as the data do not clearly indicate any of the four given colors.)

In the proof-of-concept demonstration, your robot will start at a random location. Therefore, the initial probability distribution should be uniform.

Table 2. Measurement model $p(z_k|x_k)$

z_k | $x_k \sim$	blue	green	yellow	orange
blue	0.60	0.20	0.05	0.05
green	0.20	0.60	0.05	0.05
yellow	0.05	0.05	0.65	0.20
orange	0.05	0.05	0.15	0.60
nothing	0.10	0.10	0.10	0.10

Simulate the trajectory of the robot and its state estimation assuming the actions in Table 3 are taken and the given sensor measurements are made.

Where did the robot begin? End? Be prepared to show the entire evolution of the state estimator.

Table 3. Actions and measurements

k	u_k	z_k
0	1	–
1	1	orange
2	1	yellow
3	1	green
4	1	blue
5	1	nothing
6	1	green
7	1	blue
8	0	green
9	1	orange
10	1	yellow
11	1	green
12	–	blue