

# CSCE 439/839: Robotics: Algorithms and Applications, Spring 2020

## Lab 2

Started: Monday, Feb. 24  
Checkpoint1 (Up through Sec. 2): Weds, March 4  
Checkpoint2 (Up through Sec. 4): Mon, March 9  
Due Date: Mon, March 16

## 1 Overview

In this lab, you will work with the reflectance sensor to create a map of the environment. You will also attach a range sensor, characterize it, and then implement a variety of algorithms that utilize the range finder.

Make sure to read through the entire lab before starting. This lab has checkpoints. In lab on the date of the checkpoint you will be responsible for showing the instructor your progress. You do not have to have the section written up yet (although that might not be a bad idea), but you need to be able to demonstrate all the tasks in the section.

Before starting you should read through the whole lab. Some parts can be done in parallel, while some sections rely on the completion of previous sections. You should discuss your plan of attack for the lab in your group and decide how you will work together and divide the work. Everyone, however, is responsible for knowing about all sections of the lab. In addition to completing the lab report, on the due date, you will demonstrate what you accomplished to the instructor.

Also, the points for this lab do not add up to 100, but each lab is worth the same amount as any other lab even if the points are not equal.

## 2 Reflectance Sensor Mapping (25 pts.)

*This is the first checkpoint. For this checkpoint demonstrate your mapping algorithm and the output.*

Each group has a 5-channel reflectance sensor: <https://www.pololu.com/product/3577>. On the pololu website, there is a link to sample code for the reflectance sensor. Modify the code on the Balboa and the receiving ROS node to obtain the raw readings from each of the 5 sensors.

**Question:** Describe what you had to do to modify the Balboa code and ROS code to read the sensor. Also describe how you verified that it was working properly.

Create a new node that takes the raw sensor readings and decides if each of the sensors is over a line or not. Define any thresholds you need as ROS parameters that you define in your launch file. You can use the attached page for testing line detection.

**Question:** Characterize the sensor by determining aspects like how thin of a line it can detect and how it works with different grayscale values.



The next step is to convert these raw ADC readings into ranges. I would suggest that you create a new ROS node that subscribes to the ADC message and publishes a calibrated range message. The datasheet has a graph indicating the conversion between range and voltage. However, this may not be completely accurate for your sensor and you will also need this information in your ROS node to convert between voltage and distance.

**Question:** Do an experiment to characterize the voltage-to-range conversion of your sensor. Report on your results and include a plot showing the response.

**Question:** Do some experiments to characterize the smallest objects that the IR range finder can find at various distances and report on the results. Use these experiments to determine the beam width of the IR sensor and report on how you determined this and the results.

## 4 Reactive Control (25 pts.)

*This is the 2nd checkpoint. For this checkpoint, demonstrate the reactive controller that can follow an object being moved around.*

Your next task is to develop a reactive controller that tries to maintain a fixed distance from an obstacle in front of your robot. To do this, create a new ROS node that subscribes to the range messages and publishes control commands to move it to maintain a set distance

There are a number of approaches you can use to try to maintain a fixed distance from an object. You can use thresholds and fixed values (bang-bang), a PID controller, or other methods. Regardless, you should use your distance PID controller from the prior lab to control the motion and the angular PID should still be running so that the rotation stays consistent. You should test your controller by moving something (e.g. a book) further away from the robot to see that it chases it.

**Question:** How did you implement distance control? How well does it work? Include a plot of the distances obtained from the range finder as well as control outputs as you move an object around in front of the robot. Again, there is only one question in this section, but the characterization and descriptions should be significant to get full points.

## 5 Tangent Bug (839 Only, 25 pts.)

Now that you are able to do basic mapping, you should implement a variant of the tangent bug algorithm<sup>1</sup>. In class, we discussed the Bug1 and Bug2 algorithms for navigating an unknown environment. These algorithms assumed that the bugs/robots could only detect an object if they bumped into it. Our robots are far more advanced in that they have IR range finders that can measure distances to nearby objects. Tangent bug looks ahead for obstacles and if there is one in the path to the goal, it instead moves to the furthest detected point along the obstacle boundary.

Unfortunately, our robots do not have good location information, so we will implement a variant of tangent bug. The idea is that you should give your robot a line vector for it to travel along. If it detects an obstacle along this path, it should scan to find the closest angle to the target angle that is not obstructed. You may want to ignore anything further than 50cm away for this implementation since the room is relatively small compared to range that can be detected by the range finders.

**Question:** Describe your implementation and develop some small obstacle courses to test your implementation. How well does it work? Characterize your implementation and system experimentally. Again, while there is only one question in this part of the lab, I expect that your report will have a significant amount of discussion on this section. I'm leaving much of the implementation decisions up to you, so I am also leaving it up to you to decide interesting and appropriate experiments to characterize your robot and algorithm.

---

<sup>1</sup>You can read the original article: I. Kamon, E. Rimon, and E. Rivlin, "TangentBug: A Range-Sensor-Based Navigation Algorithm," The International Journal of Robotics Research, vol. 17, no. 9, pp. 934-953, 1998. here: <http://ijr.sagepub.com/content/17/9/934.abstract>

## 6 To Hand In

You should designate one person from your group as the point person for this lab (each person needs to do this at least once over the semester). This person is responsible for organizing and handing in the report, but everyone must contribute to writing the text. You should list all group members and indicate who was the point person on this lab. Your lab should be submitted by handin, <http://cse.unl.edu/~cse439/handin/>, before the start of class on the due date. Include a pdf of your report and your source code in handin. **Name your file: csce439-CSEUSERNAME-2.pdf or csce839-CSEUSERNAME-2.pdf** depending on which class the majority of your group is in. The CSEUSERNAME should be the CSE server username of the person submitting the report.

Your lab report should have an introduction and conclusion and address the various questions (highlighted as *Question:* ) throughout the lab in detail. It should be well written and have a logical flow. Including pictures, charts, and graphs may be useful in explaining the results. There is no set page limit, but you should make sure to answer questions in detail and explain how you arrived at your decisions. You are also welcome to add additional insights and material to the lab beyond answering the required questions. The clarity, organization, grammar, and completeness of the report is worth **15 points** of your lab report grade.

In addition to your lab report, you will demonstrate your system and what you accomplished up to this point to the instructor at the beginning of lab on the due date. This is worth **15 points** of your overall lab grade. You do not need to prepare a formal presentation, however, you should plan to discuss and demonstrate what you learned and accomplished in all sections of the lab. This presentation should take around 5 minutes.

*Question:* Please include your code with the lab report. Note that you will receive deductions if your code is not reasonably well commented. You should comment the code as you write it, do not leave writing comments until the end. **Include all the code in ONE zip or tgz file.** For the ROS nodes you should include the full content of the directory so that it can compile with out errors. Similarly, for any Arduino code, make sure to include all of the code and any non-standard libraries.

*Question:* Include an `rqt_plot` of your final system and comment on your overall system architecture.

*Question:* For everyone in your group how many hours did each person spend on this part and the lab in total? Did you divide the work, if so how? Work on everything together? Give details on the hours each group member worked (e.g. keep a list of how many hours per day each person worked and on what).

*Question:* Please discuss and highlight any areas of this lab that you found unclear or difficult.



