

CS 491/691X – Topics: Introduction to Robotics

Instructor: Monica Nicolescu

Lab 3– Handout

During the first 20 minutes of the lab you will have time to test and refine your “tunnel traversing” programs. After that time we start the contest. Make sure that you have the robots ready to run, as we have only 5-6 minutes during which each team could compete with their robot.

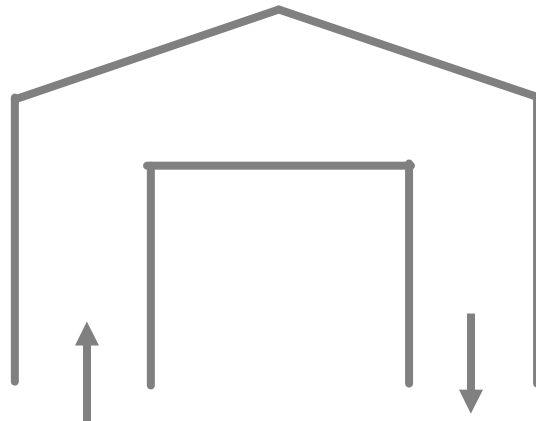
Tunnel contest

In this lab, you will use contact/touch sensors on your robots and write programs to avoid obstacles and collision. For this contest, you will write a program for your robot to go through a pre-designed tunnel and not get stuck in it.

- Make sure that your robots are in their initial configuration (Handy Bug 9645 + the custom changes in Labs 1 and 2). For reference look in the book (pages 28-50).

Contest Rules:

- In this contest, there is no winner or loser. Every group's robot should make through the tunnel in a reasonable amount of time (a few minutes).
- The order in which the robots will run will be decided at the beginning of the contest and no other changes are allowed to your programs after the contest has started.
- Your robot should back up whenever it bumps to a wall (obstacle).
- After backing up, it should turn in an angle not greater than 45 degrees.
- You should minimize the number of bumps to the opposite wall after your robot backs up from a particular wall.
- Keep in mind that the angle in which your robot will start going through the tunnel is not known to you prior to the contest. Your robot should perform satisfactorily regardless of this angle.
- Every time your robot reverses its direction in the tunnel, you will get negative points.
- The tunnel shape is like the figure on the right:



Programming tips:

- Use Appendix E (Interactive C Reference) of your book for syntaxes, built-in functions, etc.
- You can get ideas from the obstacle avoidance exercise in your book (page 72-73).
- Try using functions as much as possible. Write basic functions of your robot like turning left or right, left-sensing or right-sensing, etc as different functions in your code so that you will be able to use them in future.
- Use user buttons of the Handy Board to start and stop your programs.
- Debug your robot so that it goes as straight as possible when it is supposed to do that.

- Test your robot for turning and backing up to see it works as desired.
- Debug your bumpers so that your robot can sense left or right bumps correctly and effectively.