Control of Mobile Robotics

CDA4621

Fall 2022

Lab 3

Navigation with Distance Sensors

Total: 100 points

Due Date: 10-10-2022 by 11:59pm

The assignment is organized according to the following sections: (A) Requirements, (B) Objective, (C) Task Description, (D) Task Evaluation, and (E) Lab Submission.

A. Requirements

A.1 Physical Robot

Robot: "Robobulls-2018". **Programming**: Python
Basic Files: tof.py ("SamplePrograms.zip")

A.2 Robot Simulator

Simulator: Webots. **Robot**: "e-puck". **Programming**: Python
Basic Files: "Lab3_epuck.zip"

B. Objective

This lab will teach you about (1) Distance Sensors, and (2) PID. You will learn how to apply a PID controller to: (a) readings from front distance sensor to reduce speed as the robot gets closer to a front wall, and (b) readings from side distance sensors to reorient the robot as it gets closer to side walls.

B.1 Distance Sensors

There are 3 distance sensors configured in the robot: left, front, and right. It's important to understand the measuring range of the particular distance sensors.
Note that in general, measurements change depending on type of surface and reading angles.

B.1.1 Physical Robot

"Robobulls-2018" uses Adafruit VL53L0X Time of Flight (ToF) Distance Sensor. Read the datasheet for details[1].

B.1.2 Robot Simulator

Webot's "e-puck" uses a "generic" distance sensor type having a single ray. The distance sensor is defined by several parameters and includes a lookup table for its range[2]:

```
lookupTable [ 0 0 0, 1.27 1.27 0]
```

The above lookup table consists of two rows having 3 numbers each: [0 0 0] and [60 1e+03 0]). First row describes the minimum range and second row describes maximum range. First number in each row represents distance in meters. Second number in each row represents the

[1] https://learn.adafruit.com/adafruit-vl53l0x-micro-lidar-distance-sensor-breakout/downloads
[2] https://cyberbotics.com/doc/reference/distancesensor

corresponding return value for that range. Third number in each row indicates the standard deviation of the noise. The above lookup table describes distance sensors of 0 to 60 meters with return values of 0 to 1,000 (1e+03), and 0 noise.

B.2 PID

You will program a PID controller to control distances to walls for each of the 3 sensors. You will use only the "P" component. Note that distances to walls will be indirectly controlled by controlling motor velocities, as shown in Figure 1. Velocities should be set proportional to distance error.

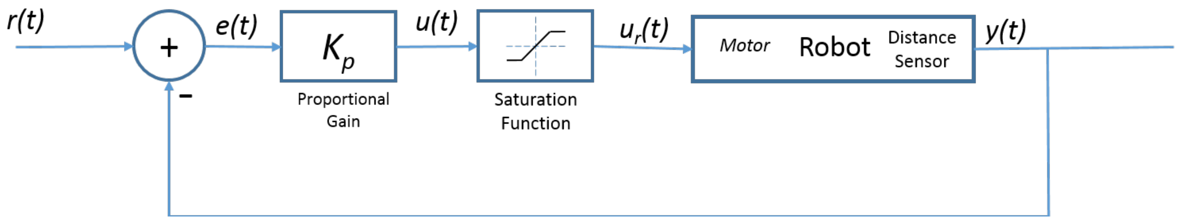

Figure 1: PID control of the robot velocity proportional to its desired distance to the wall.

The following equations summarize the diagram in Figure 1:

$$e(t) = r(t) - y(t) \quad \text{(Eq. 1)}$$
$$u(t) = K_p * e(t) \quad \text{(Eq. 2)}$$
$$u_r(t) = f_{sat}(\, u(t)) \quad \text{(Eq. 3)}$$
$$u_r(t) = f_{sat}(\, K_p * e(t)\,) \quad \text{(Eq. 4)}$$
$$u_r(t) = f_{sat}(\, K_p \left(r(t) - y(t)\right)\,) \quad \text{(Eq. 5)}$$

where:

$r(t)$ = desired distance to the wall
$y(t)$ = distance from robot to the wall
$e(t)$ = distance error
K_p = proportional gain or correction error gain
$u(t)$ = control signal corresponding to robot velocity
f_{sat} = saturation function (see explanation below)
$u_r(t)$ = control signal corresponding to the saturated robot velocity

C. Task Description

The lab consists of the following tasks:

- Task 1 – Motion with PID –Distance Sensors ('Lab3_Task1.py')
- Task 2 – Wall following – Maze ('Lab3_Task2.py')

C.1 Task 1 – Motion with PID - Distance Sensors

The robot should use K_p proportional gain control to move towards the end wall and control its stopping distance. Robot should stay as much as possible in the middle between the two side walls without hitting them, but it should not use the controller to avoid hitting the side walls.

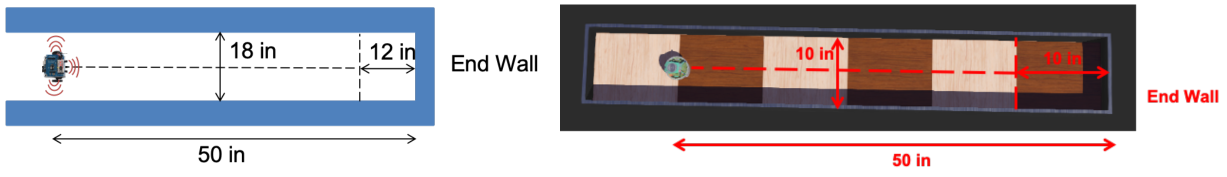

Figure 2: Robot starts 50 inches (1.27 m) away from the end wall. (Left) Physical robot values. (Right) Webots values. Note that in Webots we have set each tile to 0.508 meters width, each containing 4 colored squares.

a) Physical robot should stop at the exact 12-inch mark from the end wall using the front sensor and should keep a distance between 3 and 15 inches from side walls using the two side sensors. See Fig 2 (left).

b) Webots robot should stop at the exact 10-inch mark from the end wall using the front sensor and should keep a distance between 3 and 7 inches from side walls using the two side sensors. See Fig 2 (right).

Test the program using 6 different values of K_p(0.1, 0.5, 1.0, 2.0, 2.5, 5.0). Find the one K_p value that you think works best. Task should run for no more than 30 sec. During evaluation, the TA will start the robot at different distances from the end wall. You need to continuously print all front distance measurements on the screen.

C.2 Task 2 – Wall Following - Maze

The robot should apply the best K_p proportional gain control from Task 2 to the two side sensors to follow the maze shown in Figure 3 (Left – Physical robot, Right – Webots). You do not need to use the front sensor controller. If the robot reaches any end wall it should make a 90-degree and continue navigation. If no 90-degree turns are possible, it should make a 180-degree turn, and continue wall following in the opposite direction. The robot should navigate no further away from any wall than the distance specified in Task 2. Note that depending on the original orientation the robot my end up following a different path. Also note that navigation does not require visiting every single square. The robot can start at any grid cell with any orientation and should run for no more than 1 minute (or less if it already returned to its original starting location). During evaluation, the TA will start the robot at different initial position, orientation, and decide whether to follow left or right walls. Note that it may be simpler to follow one wall at a time.

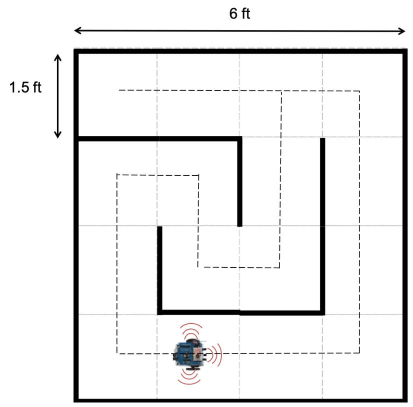

Figure 3 (Left) Physical robot maze wall-following task.

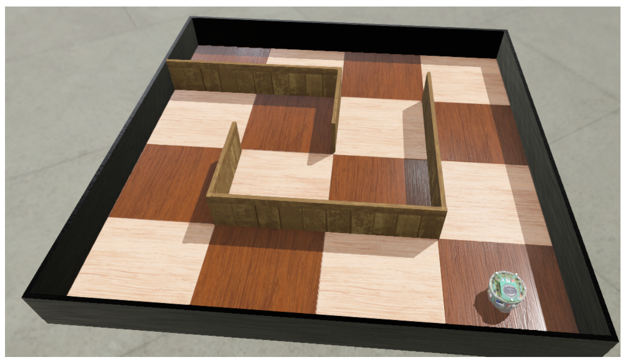
Figure 3 (Right) Webots maze wall-following task.

D. Task Evaluation

Task evaluation is based on: (1) program code, (2) report, including a link to a video showing each different task, and (3) task presentation of robot navigation with the TA. Note that all functions and tasks to be implemented are required in future labs.

D.1 Task Presentation (90 points)

The following section shows the rubric for the different tasks:

- Task 1 (45 points)
  - Robot stabilizes at specified distance from end wall (10 points)
  - Robot stabilizes at specified distance from side walls (10 points)
  - Robot utilizes PID to reduce speeds proportionally to the front wall (10 points)
  - Robot utilizes PID to updates orientation proportionally to distance from side walls (10 points)
  - Robot restabilizes when placed further or closer to the wall (5 points)
  - Robot hits walls (-5 points)
- Task 2 (45 points)
  - Robot follows left or right walls accordingly (20 points)

- Robot makes left or right turns accordingly (15 points)
- Robot follows end of walls accordingly (10 points)
- Robot average travel speed less than 2 inches per second except for corners (-5 points)
- Robot constantly exceeds 3 inches closer or further away from any wall (-5 points)
- Robot hits walls (-5 points)

D.2 Task Report (10 Points)

The report should include the following (points will be taken off if anything is missing):

1. Mathematical computations for all kinematics. Show how you calculated the speeds of the left and right servos given the input parameters for each task. Also, show how you decide whether the movement is possible or not. (4 point)
2. Video uploaded to Canvas showing the robot executing the different tasks. You should include in the video a description, written or voice, of each task. You can have a single or multiple videos. Note that videos will help assist task evaluations. (3 point)
3. Conclusions where you analyze any issues you encountered when running the tasks and how these could be improved. Conclusions need to show an insight of what the group has learnt (if anything) during the project. Phrases such as "everything worked as expected" or "I enjoyed the project" will not count as conclusions. (3 point)

D.3 Task Presentation

Task presentation needs to be scheduled with the TA. Close to the project due date, a timetable will be made available online to select a schedule on a first come first serve basis. Students must be present at their scheduled presentation time. On the presentation day, questions related to the project will be asked, and the robot's task performance will be evaluated. If it is seen that any presenter has not understood a significant portion of the completed work, points will be deducted up to total lab points. Students that do not schedule and attend presentations will get an automatic "0" for the lab.

NOTE for physical robot presentation: During presentations, robot calibration time is limited to 1 min. It is important that all members of the team attend and understand the work submitted.

E. Lab Submission

Each student needs to submit the programs and report through Canvas as a single "zip" file.

- The "zip" file should be named "yourname_studentidnumber_labnumber.zip"
- Videos should be uploaded to the media folder in Canvas. Name your files "yourname_studentidnumber_labnumber_tasknumber".
- The programs should start from a main folder and contain subfolders for each task.
- The report should be in PDF and should be stored in the same "zip" file as the programs.