Objective

This lab requires the use of proportional gain to control the speed of the robot based on its distance from the wall. Additionally, using proportional gain and lidar scanners to detect, avoid, and follow walls. The tasks to achieve this objective are listed and defined as follows:

Task 1A – Motion with PID

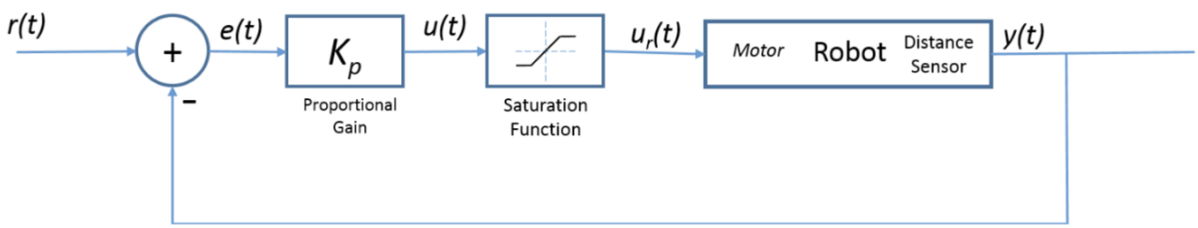

Figure 1: Flowchart describing the steps for PID motion.

In this task, the robot should use K_{pf} proportional gain control applied to its front sensor to move towards the end wall. To find the proportional gain control, it requires a few steps:

1. Find the distance error, $e(t)$:
    a. The distance error can be calculated by subtracting the current front distance sensor reading, y(t), by the target distance, r(t).
    b. $e(t) = y(t) - r(t)$
2. Now, we can obtain the control signal, $u(t)$. The control signal is the calculated velocity which is directly proportional to the error. To calculate $u(t)$, we must multiply it by a K_p value. This value will be our correction error gain.
    a. $u(t) = K_p \times e(t)$
3. Finally, we must pass this control signal into a saturation function that will test the extremes. If the control signal exceeds the max speed of the robot, the saturated robot velocity will become the max speed. If it exceeds the min speed of the robot, the saturated robot velocity will become the min speed. Otherwise, the value of $u(t)$ will remain unchanged. Here's the final equation for the saturated robot velocity:
    a. $u(t) = f_{sat}[K_p(r(t) - y(t))]$

Task 1B – Wall Avoidance

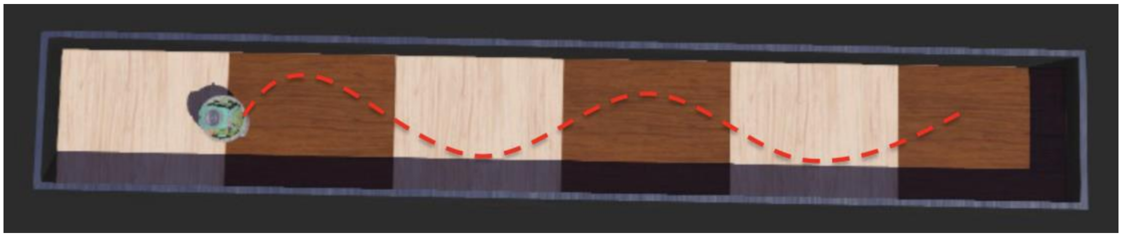

Figure 2: Robot path to avoid walls using PID.

Continuing task 1, the robot must avoid any walls using its left and right LiDAR scanner readings. If put into an orientation which moves the robot towards the wall, it will follow a similar path as shown in figure 2 to avoid the wall.

Using the saturated velocity that was calculated in Task 1A, we will adjust the robot's left and right motors accordingly to avoid walls. To achieve this, the steps are as follows:

1. Assign a distance from the wall the robot should maintain. Let's call this d_{wall}. If we set this variable to 0.40, for example, we would want the robot to maintain 0.40 meters away from any given wall.

2. Now, using d_{wall} and the robot's left and right LiDAR scanner readings, we can move the robot away from walls accordingly. From now on, let's refer to the left distance reading as "d_{left}" and the right distance reading as "d_{right}".

   a. If $d_{left} < d_{wall}$, we want to add proportional gain control to the right wheel. This slows down the right motor proportionally to the distance between d_{left} and d_{wall}.

   b. If $d_{right} < d_{wall}$, we do the same as explained above but for the left wheel.

3. Essentially, with this under a loop and correct K_p values set; the robot should avoid the walls.

Task 2A – Wall Following

Wall following builds up from the previous concepts described above. The only difference here is it must follow either the left or right wall chosen by the user.

Instead of "pushing" away from walls as done in Task 1B, this requires "pushing" and "pulling" constantly to the selected wall. This can be achieved as follows:

1. If the chosen wall is RIGHT:
    a. And if $d_{right} < d_{wall}$, this means the robot is getting too close to the wall and it must push away. In this case, the right motor velocity will be higher.
    b. Otherwise, the robot is getting too far from the wall, and it must pull back to it. In this case, the left motor velocity will be higher.

2. If the chosen wall is LEFT:
    a. And if $d_{left} < d_{wall}$, this means the robot is getting too close to the wall and it must push away. In this case, the left motor velocity will be higher.
    b. Otherwise, the robot is getting too far from the wall, and it must pull back to it. In this case, the right motor velocity will be higher.

Task 2B-- Rotation

When the robot approaches a corner, it must rotate 90-degrees counter-clockwise if it is following the right wall and 90-degrees clockwise for the left wall. To detect the corner, I used two LiDAR scanner readings. The front reading, d_{front}, and the left or right reading depending on the wall that is being followed. The robot will rotate until d_{front} exceeds 1.0 meters. In most cases, this will be a 90-degree rotation. The benefit of this is that the robot will rotate until it finds clearance however, the downside is that you can't set the rotation to the angle you want.

Conclusion

This lab has put into application a good understanding of wall following with PID, more specifically, proportional gain control. With Task1A, the PID calculations were not too difficult however, I started to encounter issues with wall avoidance until I figured out the concept behind it and setting the correct control values. I also encountered a lot of issues regarding wall following with the robot getting too close or too far from the walls, sometimes hitting the walls, and even getting lost. With some correction to the code and control values, it is working much better than before however, it is not perfect and unfortunately, still fails in certain scenarios. Additionally, the rotate function does not use encoders to rotate by a set angle as I found encoder readings difficult to work with as you must constantly keep track of distance.