

# Individual Assignment: Differential Drive and LiDAR

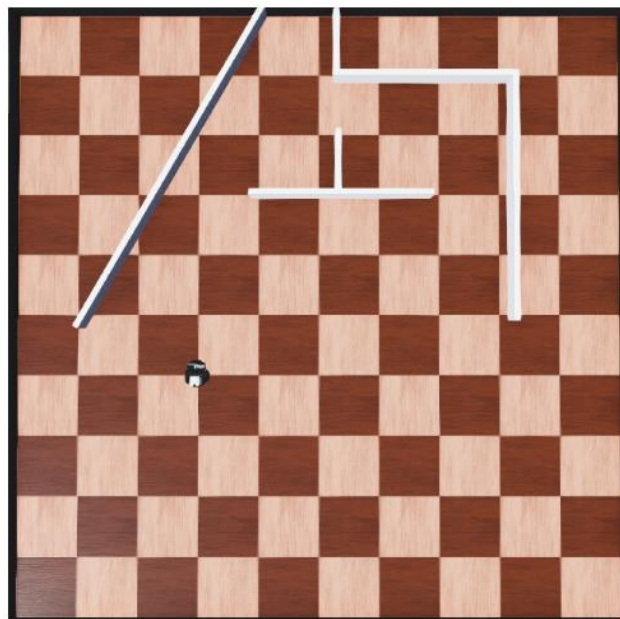
## Assignment Overview

In this assignment, you will use the Webots simulator to control a Pioneer 3-DX robot. First, you are asked to implement a differential drive controller using the formulas you learned in this module. Second, you have to use the robot's LiDAR to find the closest surface and navigate to it. Finally, you will devise and implement a wall-following algorithm. Before starting with the actual assignment, you should read and follow the instructions in the Preliminaries section in order to familiarize yourself with the simulator.

## Preliminaries (1 hour)

If you have not yet installed the [Webots simulator](#), please do so. We recommend the latest version, 2022b, however, you can also use version 2022a if that works better on your system. Then, follow the [official tutorial](#) (only Tutorial 1) to learn the basics of the simulator. Since you are supposed to use Python for implementing the assignment, please select the corresponding code option in the tutorial.

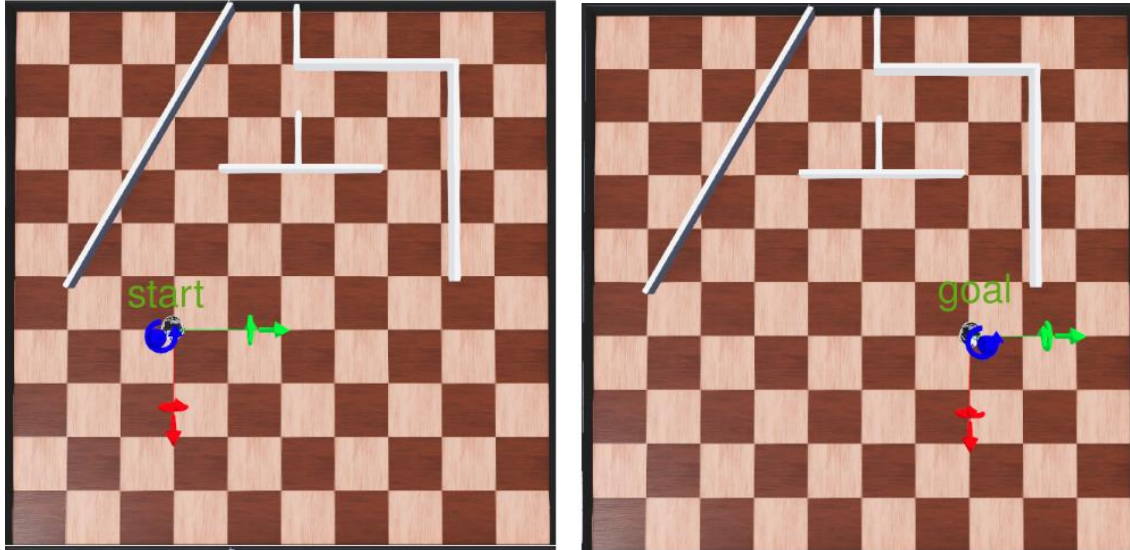
Then, download module2\_assignment.zip (or module2\_assignment\_2022a.zip if you are using 2022a), unpack it, and open module2\_assignment/worlds/module2\_assignment.wbt via File->Open World. You should see a rectangular arena with a Pioneer robot and several walls as shown below:



## Differential drive (3 hours, 10 points)

Implement a differential drive controller to move the robot from the start pose to the goal pose. In other words, the robot should be translated by 4 meters along the y axis in world coordinates, as shown below.

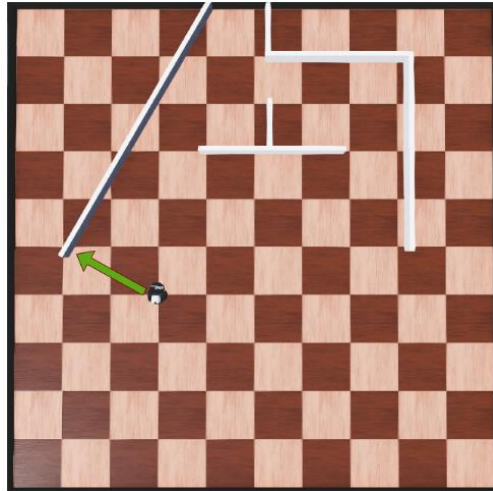
start pose :  $x_s = -1\text{m}$  ,  $y_s = 2\text{m}$  ,  $\theta_s = \pi$  , Goal pose :  $x_g = -1\text{m}$  ,  $y_g = -2\text{m}$  ,  $\theta_g = \pi$



A trivial solution requires three steps to reach the goal, and one step to stop the robot once the goal is reached. Here, a step means setting or changing the pair of wheel velocities. However, it is also possible to reach the goal pose using two steps, and one step for halting. Just reaching the goal will grant you 7 points, by reaching it in two steps you will earn 10 points. For your code, please create a new controller called "diff\_drive", you can use the provided "proxy\_controller" as a template. Note that the wheel speed is measured in radians per second, i.e., a velocity of  $2\pi$  means that the wheel will make one turn per second. The maximal wheel velocity of the Pioneer3-DX is 12.3, the wheel diameter is  $D = 0.195\text{m}$ , and the distance between the wheels is  $L = 0.325\text{m}$ .

## Closest surface (2 hours, 7 points)

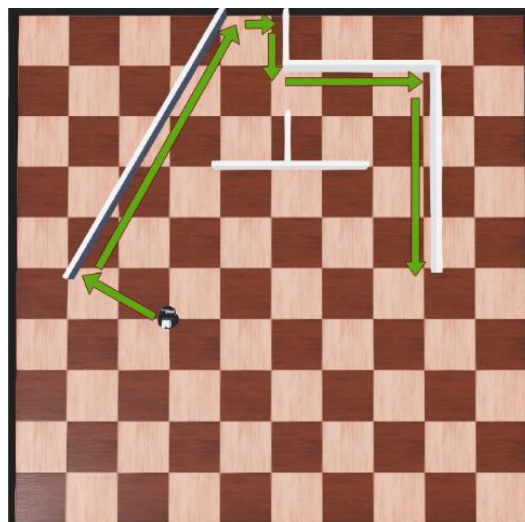
In the second part of the assignment, you should use the robot's LiDAR to find the wall point closest to the robot's start position and then move the robot to that point. For your information, the following image indicates the closest point:



However, in your code this point should be determined from LiDAR data. To that end, please create a new controller called "closest\_point". As a template, you can again use the "proxy\_controller" which also contains a code snippet for reading LiDAR measurements. The LiDAR has a  $180^\circ$  FoV, a scan contains  $n = 181$  measurements, and the position of the LiDAR in the robot coordinate frame is  $x = 0.08\text{m}$ ,  $y = 0\text{m}$ ,  $z = 0.21\text{m}$ . The robot should halt 50cm in front of the wall, measured from the center of the robot. Please drive the robot directly to the goal point and do not use the two-step algorithm from the previous part.

## Wall following (2 hours, 3 points)

In the last part, please implement a wall-following algorithm in a new controller called "wall\_following". In the first step, the robot should drive to the closest wall, here you can use your results from the previous part. Then, it should drive along that wall as



## Differential Drive and LiDAR Assignment

Differential drive (three steps) The robot reaches the goal pose using at most three steps and halts at the goal in an additional step.	7points
Differential drive (two steps) The robot reaches the goal pose using two steps and halts at the goal in an additional step.	3points
Closest point (detection) The point on a wall closest to the robot's start position is determined from the LiDAR data.	5points
Closest point (moving) The robot moves towards the detected point and halts 50cm before the wall (the distance is relative to the robot center).	2points
Wall following The robot follows the wall as depicted in the assignment description.	3points