SC627-Motion Planning & Coordination of Autonomous Vehicles

Assignment-4

Danish Behnal (190010018)
Instructor: Prof. Arpita Sinha
8 April 2022

Robot Network in Balancing

The goal of this assignment was to make a set of robots randomly spawned in the Gazebo world become equidistant from each other (in axis). We control the velocity command for the robots in this assignment.

Implementation

The problem statement mentions that each robot can access the robot pose data for a robot before it and one after it. So we define 3 arrays

bot, left_bot, right_bot which will hold the the pose data for the three robots.

We then move on to define 3 functions: ***callback_odom(), callback_left_odom(), callback_right_odom().*** Each of the three functions receive data from *odom, left_odom, right_odom* topics about the x-coordinate, y-coordinate and the yaw value for the appropriate robots.

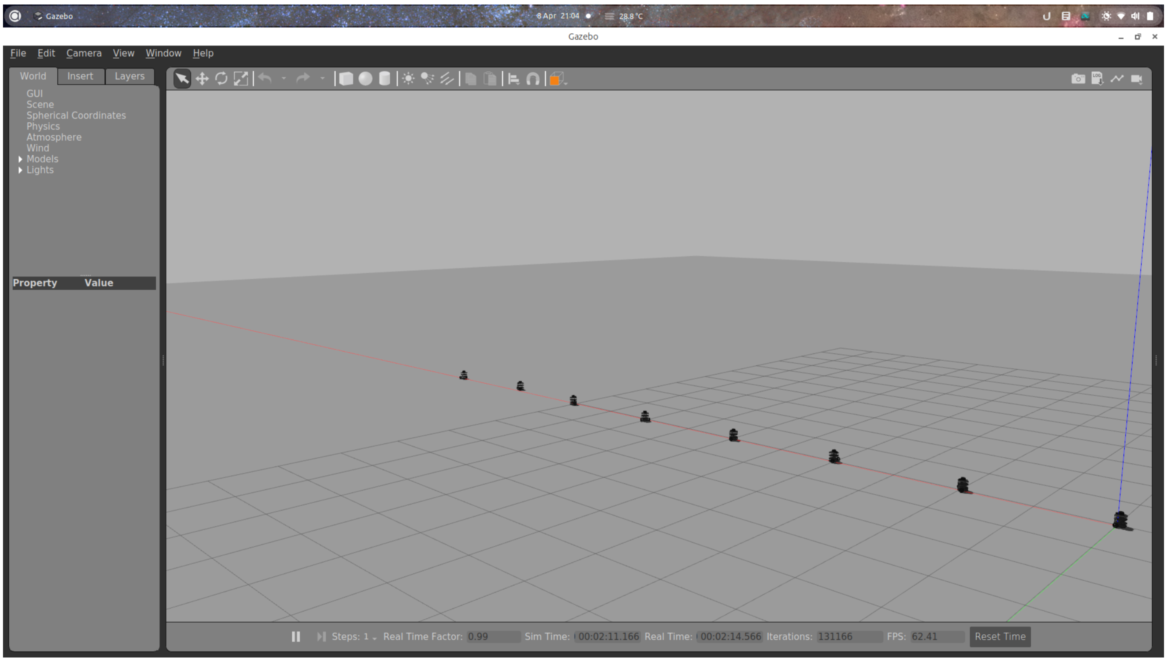
In the assignment, it is mentioned to replace the while condition with the true terminating condition of when the balance is achieved. I left it as it is because we have to manually terminate the process in the terminal anyways. It is a crude method but essentially serves the same purpose. I also removed the 10 degrees maximum deviation constraint because the turtlebot is capable of deviations of more than 10 degrees and it was also affecting the balancing of the robots.

I used this basic equation to command the velocity to the robots:

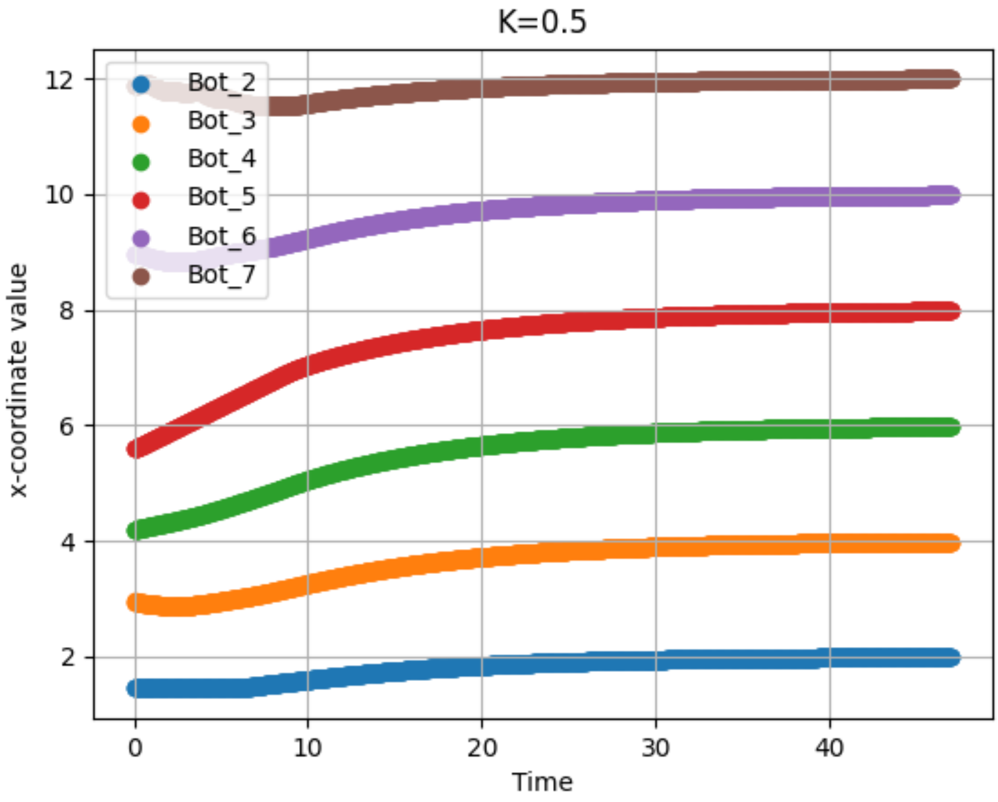
$$v = K * [(x_l - x) + (x_r - x)]$$

Where K is the gain, x_l is the position of left robot and x_r is the position of the right robot and x is the position of robot in focus.

Results:



Steps: 1 Real Time Factor: 0.99 Sim Time: 00:02:11.166 Real Time: 00:02:14.566 Iterations: 131166 FPS: 62.41



Plot for the all moving Robots