SC 627
Assignment 4: Balancing Robots

Hitesh Kandala | 180070023

Implementation Details

Here to balance the bots equidistantly, we need to control their velocities in correspondence to the distance between their left and right bots. For this as well, we could easily ignore the velocity in y-direction and operate on only velocity in x-direction. So, the velocity for the current robot, v_{curr} is given by:

$$v_{curr} = k * [(x_{right} - x_{curr}) - (x_{curr} - x_{left})] \tag{1}$$

Here, k is a hyper-parameter. Now that we have defined the velocity update equation for each bot, we need to define callback functions properly i.e. callback odom, callback left odom, callback right odom. For all three, we get data from nav_msgs, which gives us position, orientation, linear vel. Using the euler_from_quaternion function, we get the orientation of bot. As it is a linear motion in x-direction, we can ignore the angular component in the velocity convert function to allow the robots to move back and forth only in one direction. Therefore, the velocity convert function takes the corresponding inputs and returns the linear and angular velocities of which we only use the linear velocity and publish it to cmd_vel. Now that we have defined all the required publisher and subscriber functions, we go on to write the main loop where the termination condition is as follows, the velocities of current bot, its left bot and its right bot, all of them should have their magnitudes of velocities less than some epsilon defined in the code and can be tuned. Also since at the start the velocities are zero so we also add a condition that after some steps for eg. 1000, which I have used in the code, the above described termination condition will take effect and not before.

Simulation results

We get the following results upon simulation:

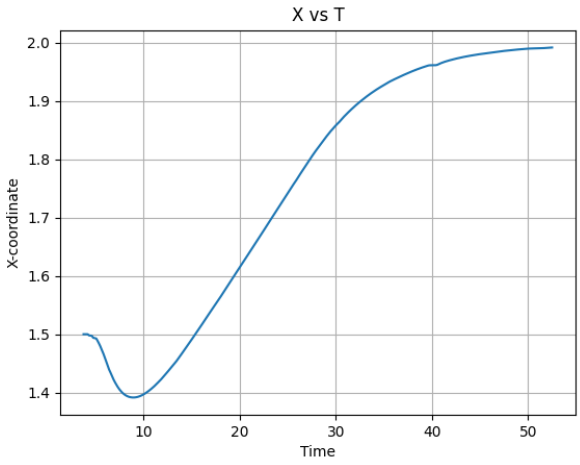

Figure 1: Robot 2: X vs Time

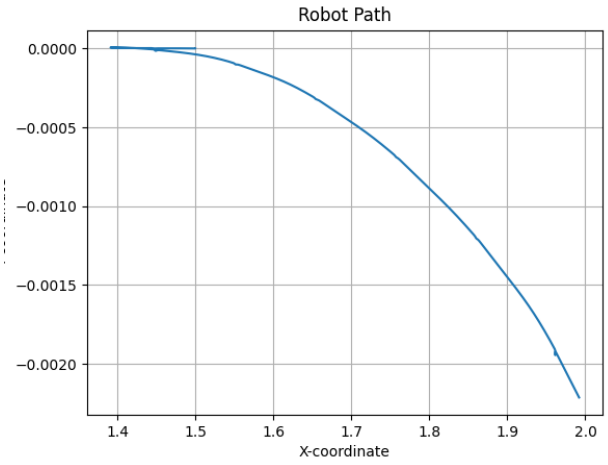

Figure 2: Robot 2: Path Plot

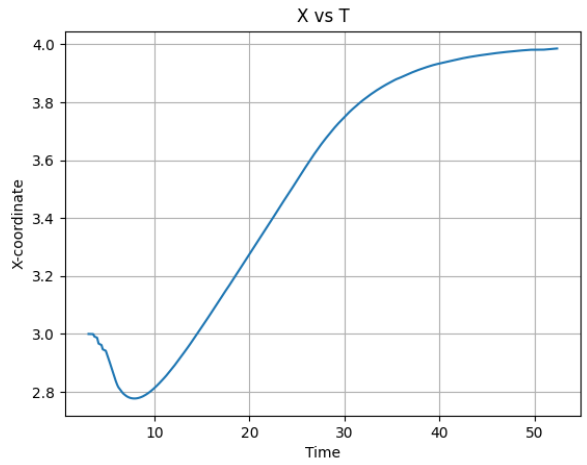

Figure 3: Robot 3: X vs Time

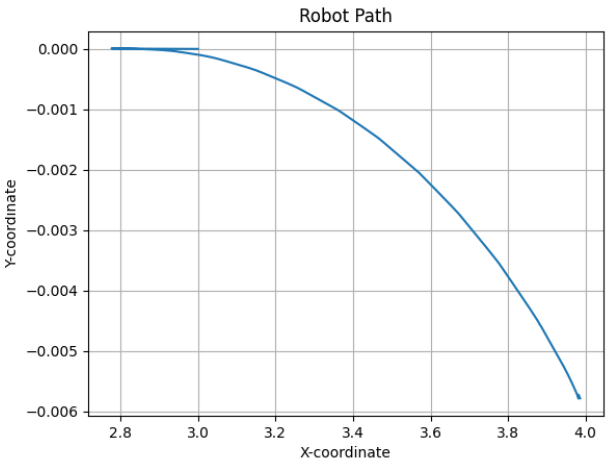

Figure 4: Robot 3: Path Plot

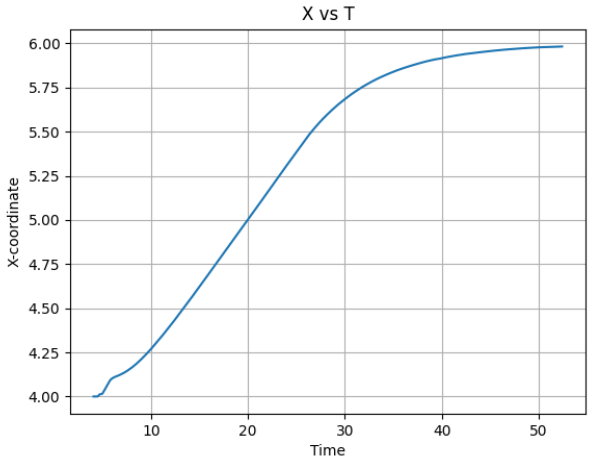

Figure 5: Robot 4: X vs Time

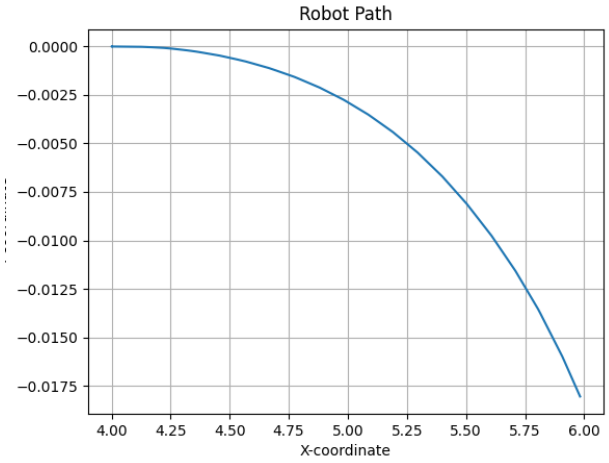

Figure 6: Robot 4: Path Plot

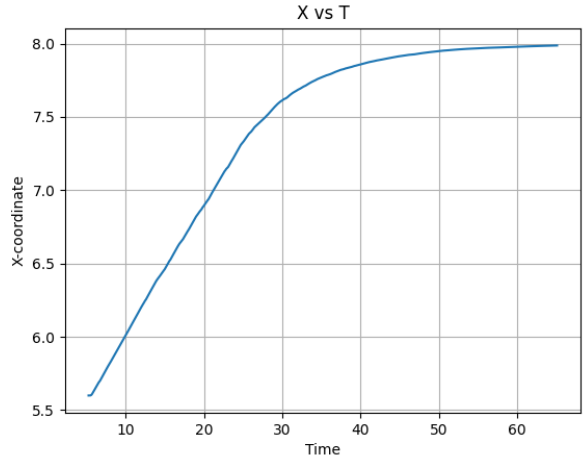

Figure 7: Robot 5: X vs Time

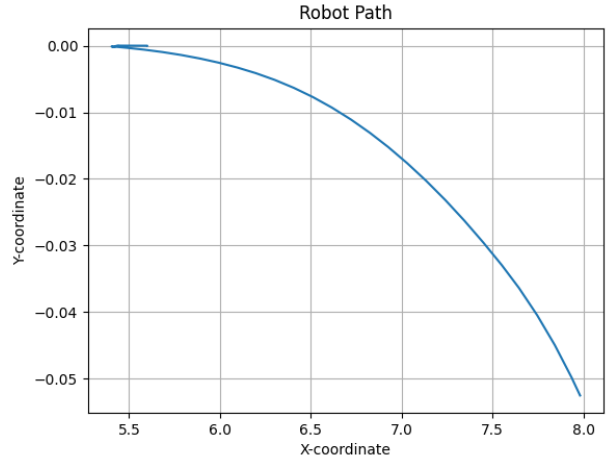

Figure 8: Robot 5: Path Plot

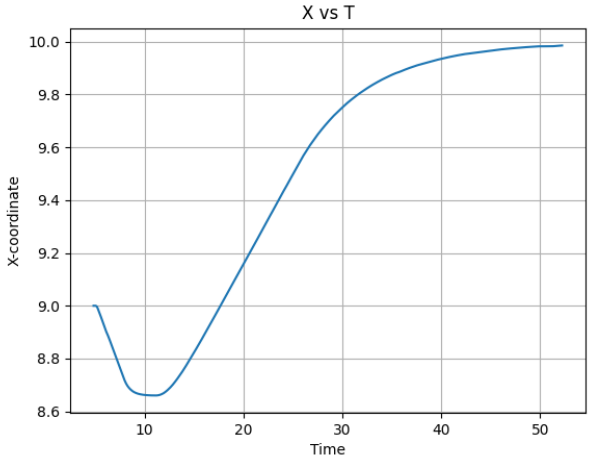

Figure 9: Robot 6: X vs Time

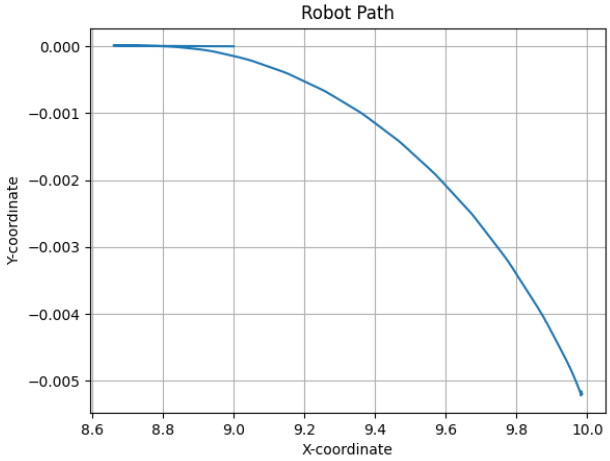

Figure 10: Robot 6: Path Plot

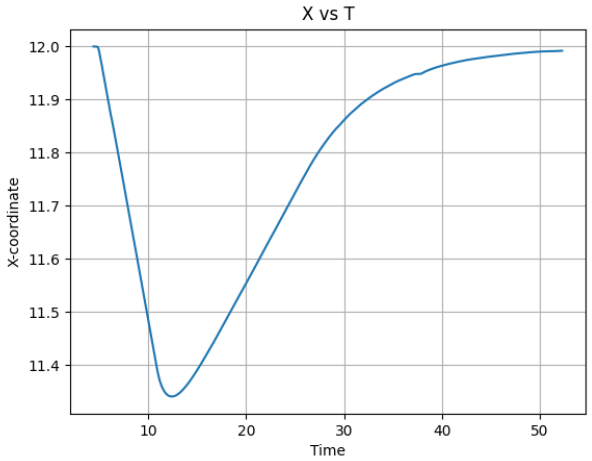

Figure 11: Robot 7: X vs Time

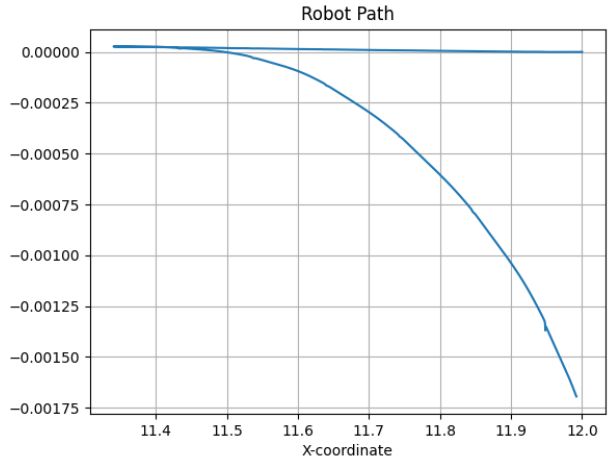

Figure 12: Robot 7: Path Plot