MTE 544 Report

Stephen Wang	Ajit Rakhra	Howard Dong
x2274wan	a3rakhra	h53dong
20821034	20832703	20847639

Station 1, Robot 4 on Fri Oct 20 at 8:30 AM

Navigate to Point

Figure 1 and 2 show how the robot made "U turns" before finally navigating to the goal points using both the linear and angular controller. Comparing the trajectory and state variables of the robot motion under P and PID control, PID control provides a better trajectory. Both trajectories look similar, but PID control produced less overshoot. Note that we made the robot goal criterion very tolerant because the robot couldn't stop when the goal is reached otherwise.

To tune the steering controller parameters, we began by setting K_i and K_d terms to 0, which effectively turned the PID controller into a P controller. P control produced Figure 1. We then added K_i; the robot settled quicker but oscillated about a straight line. We continued to increase K_d until that oscillation is gone, producing Figure 2.

The linear controller was straightforward to tune. We observed that the overall trajectory was not impacted when both motors were saturated to the maximum speed of Turtlebot 3, so we left K_i and K_d to be small.

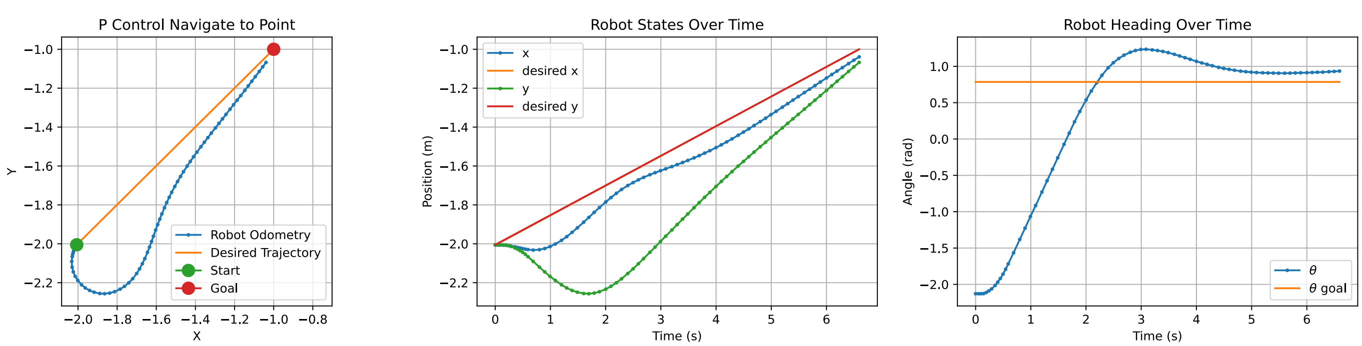

Figure 1: The three plots above show the robot driving to a point under the P controller. From left to right, the 2D trajectory, position states and theta states are plotted with the corresponding desired values.

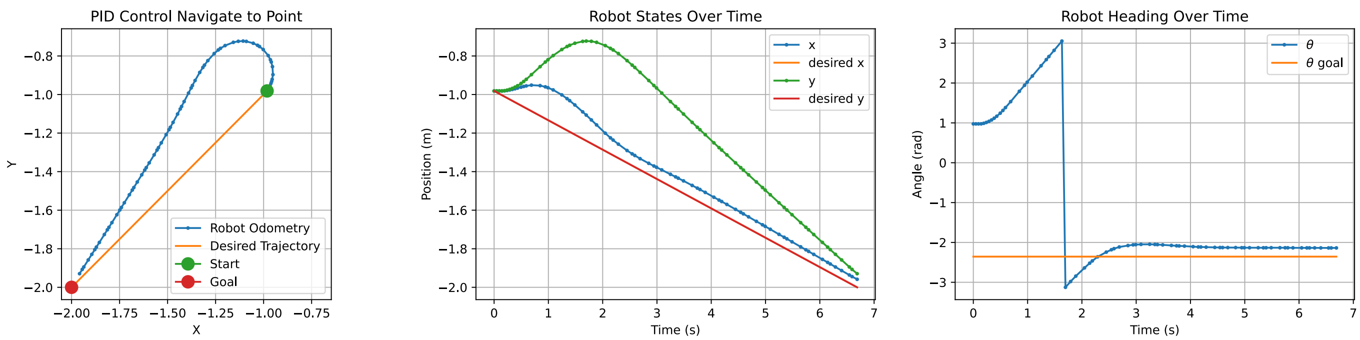

Figure 2: The three plots above show the robot driving to a point under the PID controller. From left to right, the 2D trajectory, position states and theta states are plotted with the corresponding desired values.

Follow Reference Trajectory

Two reference trajectories were generated and the robot followed them, shown in Figure 3 and 4.

We observed that trajectories densely sampled from the reference functions, at 100 points, the robot oscillated uncontrollably; changing the PID parameters did not reduce the oscillations. We theorized that that there was a very significant amount of latency between the controller and the robot, ie the plant. Thus, we ensured that there were no more than 21 points in the reference trajectory. The resultant odometry plots, as shown below, are satisfactory, with the parameters tuned from point-to-point navigation.

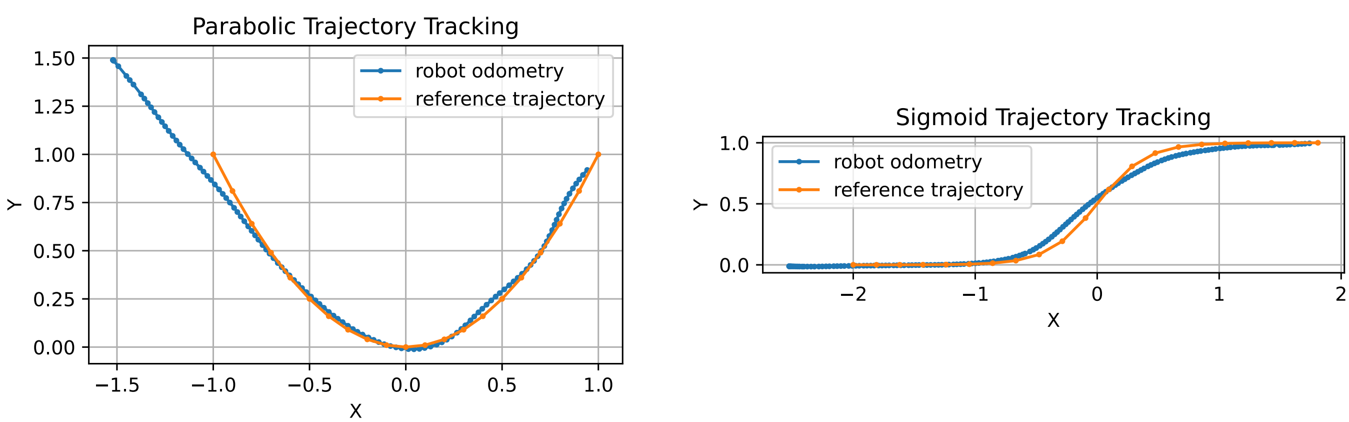

Figure 3: The two plots above show the robot trajectory under the PID controller. The left plot shows the trajectory of parabolic tracking and the right plot shows the trajectory of sigmoid tracking. The blue line shows the robots trajectory, and the orange line shows the reference trajectory that the robot used for the motion.

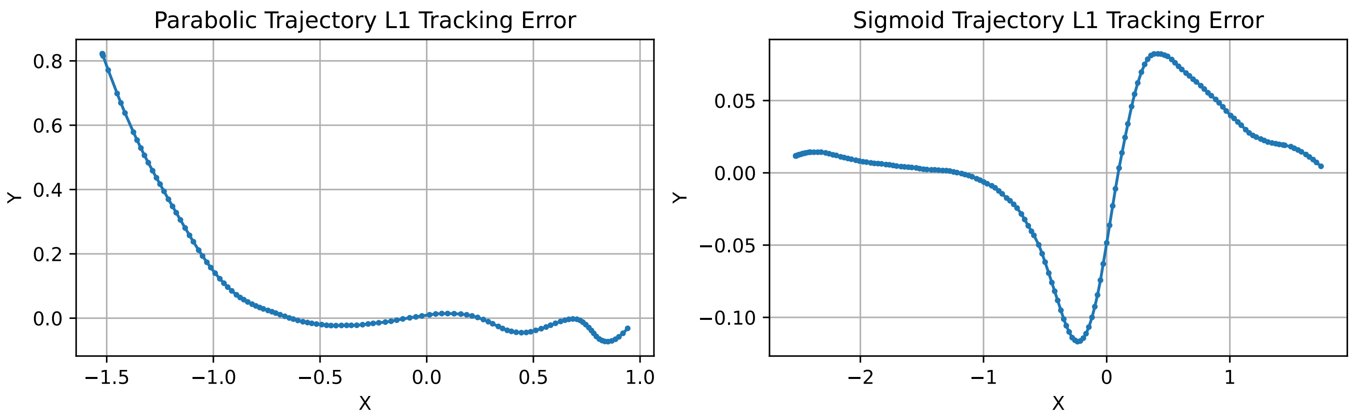

Figure 4: The two plots above show the robot trajectory L1 error relative to the reference trajectory. From left plot shows the error of parabolic tracking and the right plot shows the error of sigmoid tracking.

The sigmoid trajectory illustrates the limitation of using PID for steering control. The reference signal is exclusively based on (up to, if at the end) the third next point in the trajectory, without considering the possible curvature that intermediate points would create. This resulted in an odometry trajectory that conforms to the general shape of the function but it looks like $\frac{1}{1+e^{-3x}}$ rather than $\frac{1}{1+e^{-5x}}$.

Depending on the application, this may not be consequential. However, if the trajectory was planned to optimize a very specific objective, failing to follow the trajectory exactly would produce a suboptimal path.