Homework 1: PD Controller Tuning in ROS2

RAS 568: Space Robotics and AI

Wilson Kerr

Final Parameter Values

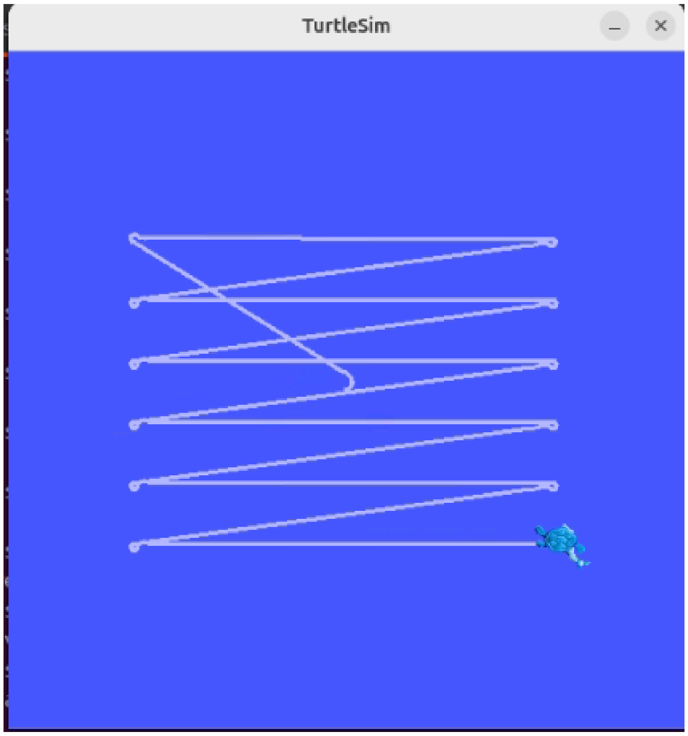
The tuning of the PD controller was performed manually by adjusting each parameter individually using the rqt_configure tool until the turtlesim appeared to be close to the desired pattern shown in class. The values were further tuned in by adjusting the values in the python script and running it the sim in full in order to minimize the average cross-track error. Increasing the Kp_linear value leads to quicker convergence with the desired trajectory but can have issues with overshoot and oscillations when too high. Increasing Kd_linear lowers the convergence rate and leads to a slower response but lessens overshoot and oscillatory behavior. Increasing Kp_angular leads to quicker angle correction but when too high leads to over correction and oscillatory behavior. Increasing Kd_angular caused the turtles path to oscillate across the path between waypoints.

$Kp\_linear\ =\ 7.0$
$Kd\_linear\ =\ 0.4$
$Kp\_angular\ =\ 14.0$
$Kd\_angular\ =\ 0.0$
$Spacing\ =\ 1.0$

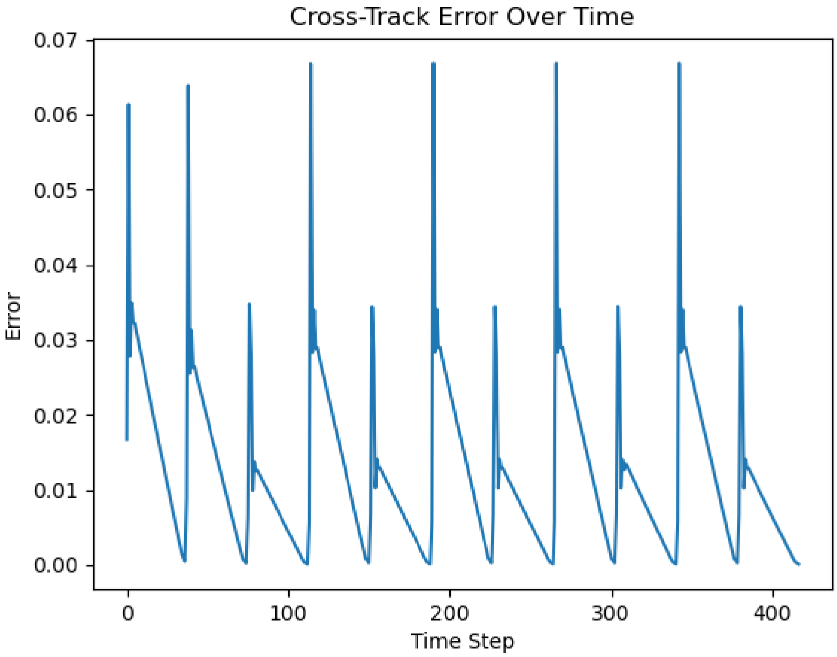
These final parameter values were acquired by increasing each value individually and observing the average cross-track. When the cross-track error increased after adjusting the value of a parameter, the value used before was used to then further adjust the other parameters. This was the repeated until all of the parameters were optimized.

Performance Metrics

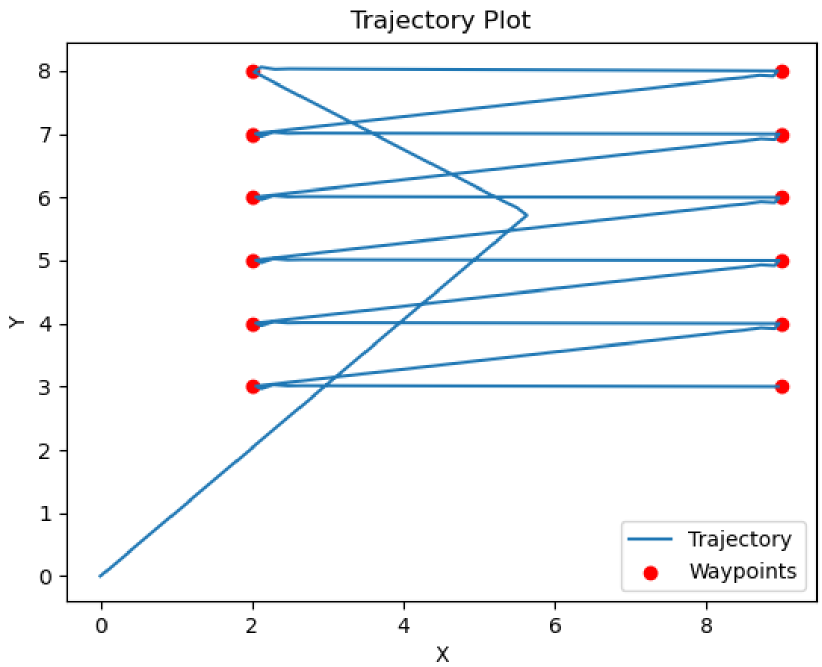
The optimal parameter values found resulted in the following trajectory.



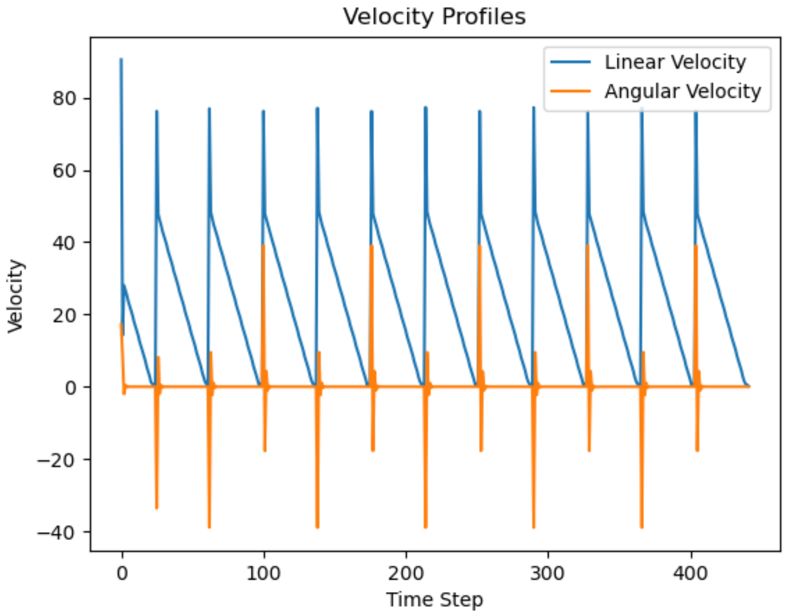
The main performance metric that we care about is cross-track error. The average cross-track error for the found optimal values was 0.013 with maximums close to 0.08.



The cross-track error over time plot shows that the error is oscillatory. This is due to the error increasing while turning after reaching a waypoint



The trajectory plot shows the smooth motion of the turtle as it makes its way from waypoint to waypoint. There is some oscillatory motion and turning overshoot at the waypoints after turning. This shows why the cross-track error spikes at regular intervals.



The velocity profiles show the turtle only making angle changes at the waypoints.

Discussion of Tuning Methodology

Manually tuning the parameters is an inefficient and tedious process for controller design. On more complex systems it could be impossible due to time limitations. This is why better optimization methods are important.

Challenges and Solutions

My biggest challenge was getting ROS2 to run on my windows 11 pc. After a few hours of tinkering with it I could get the basic turtle sim to run, but I would get errors while trying to build the boustrophedon navigator package. My solution to this was installing Linux. This was simple and I had little to no issue with the program running on Linux. From there was just the learning hurdle utilizing ROS2, which became easier overtime after use.