EBS 221 HW 2

Benoit Rouchy and Kevin Oghalai

1. Part a was implemented in code and is tested in subsequent sections
3. This was implemented and is shown in the next section
4. The implemented closed loop control is shown below



One thing that was unexpected is that the robot does not follow the path perfectly. Through testing, this was determined to be an issue with the pure pursuit controller, as it’s goal point did not update for significant periods of time, since each goal point was fairly spread out(0.1 radians apart). By decreasing the distance between points on the path(goal points every 0.1 radians), this consistent error was able to be decreased to nearly 0. Shown below is a plot of what that looks like.



1. The cross track error for the points spaced every 0.1 radians is plotted below in blue. There is a steady state error and jagged behavior. The steady state error is caused by goal points that are spaced too wide, and the jagged behavior is caused by the error increasing when the vehicle is in between two goal points. The error rapidly increases, then decreases as the robot gets further from the point that is behind it, eventually getting closer to a point in front, at which the distance to the closest point starts to rapidly decrease. The plot below shows that the jagged behavior was able to be alleviated by interpolating between goal points, so that the cross track error is measured as being the closest distance to the target path, rather than the closest distance to a target point.



1. Shown below is the results when goal points are more closely located together, with interpolation between closest points. This leads to roughly a steady state error of 0 with smooth behavior.

