Translate Waypoints to Commands

After determining the waypoints, these were then converted to linear distance in millimeters and angular displacement in radians to be sent to the Arduino. To determine the depth and angle, we looped over the waypoints found in the trapezoidal trajectory and found the difference between them. From there, were took the Euclidean distance to determine the linear displacement between the current and the next waypoint. We found that our camera was often providing us with a depth measurement, or z-coordinate, that was greater than what we expected, thus, we reduced the displacement by 20% by multiplying the resulting displacement by 0.8. This corrected both our displacement error as well as augmented the turn radius between waypoints to be over the deadband and increase turn accuracy on carpeted surfaces.

To determine the turn radius, we used the equation below, where we centered our turns such that the origin was straight ahead at the robot's $\frac{\pi}{2}$ axis and corrected the angles by subtracting the previous heading.

$$\theta_{\text{Calc}} = \text{atan2}(\text{Diff\_z}, \text{Diff\_x})$$

$$\theta_{\text{Center}} = \theta_{\text{Calc}} - \frac{\pi}{2}$$

$$\theta_{\text{Correct}} = \theta_{\text{Center}} - \theta_{\text{Prev}}$$

From there, the Arduino was fed a turn command with the calculated angle first, then a displacement command.

Prototyping