Remote Control Robot Car
Jimmy Huang, John Greaney-Cheng, Rohan Variankaval

Overview

• Created user interface to remotely control robots integrated with ROS (Robotic Operating System)
  ○ Installed ROSARIA libraries for odometry data
  ○ Built Flask 1.1 web server to stream robot camera footage, control robot movements, and measure odometry

• Improved Pioneer 3-DX robot through hardware and software upgrades
  ○ Implemented waypoint path following through Hermite Spline Interpolation
  ○ Eliminated variance in drift via replacement of back wheel with ball bearing
  ○ Calibrated robot through ROSARIA parameters: DriftFactor, Ticksmm, and RevCount

Spline Path Following

• Generate Spline Path with Hermite Spline Interpolation
  ○ Given a set of waypoints, creates parametric differentiable path connecting points

• Follow waypoints through velocity commands
  ○ Calculate forward and angular velocity through path derivative and curvature of path respectively

• Pure Pursuit waypoint following uses odometry to adjust robot movement back to generated path

Pioneer 3DX CLI

• Command Line interface to control the Pioneer 3DX and view odometry/battery information
  ○ Uses ncurses for nice UI
  ○ Accessible via ssh terminal

Webserver

• Users can view and control ROS topic information in browser
  ○ Uses Flask 1.1
  ○ Accessible using ssh tunnels and browser SOCS proxy
  ○ Allows for the creation of GUIs without remote desktop

• Can send commands to Pioneer through browser
  ○ See through RealSense Camera attached to Pioneer
  ○ Move Pioneer with buttons on site

Pioneer 3DX Calibration

• Discovered need for calibration after closed-loop test (move Pioneer in square repeatedly to see drift)

• Adjusted parameters within ROSARIA libraries
  ○ ticksmm - distance accuracy
  ○ driftFactor - minimize veer left/right when moving in a straight line
  ○ revCount - turning accuracy

• Minimized variance by replacing backwheel with ball bearing roller wheel
  ○ Designed and 3D printed connector between wheel and Pioneer