15-112 Project Proposal

20 Nov 2019

Eric Jenny

Description

Differential Drive Simulation

The project is a simulation of a differential drive robot. It is a 2D, top-down simulation using models of motor voltage-speed-torque, inertia, friction, and collisions with fixed objects. The goal is to produce a simulation which is easily adaptable to test different control methods and algorithms, by including visuals and a lot of user control and feedback.

Competitive Analysis

Many differential drive simulations are built using the Gazebo platform, which although is a very adaptable and supported platform, has a large learning curve and is more complicated than necessary for simple, flat environments. The goal is that this simulation is easier to use for those learning to code and experimenting with controls, path following, and path planning algorithms.

Structural Plan

- Main class: top level class controlling which modes are active
    - User Driven Mode: Keyboard control (up/down/right/left) to drive the robot around the environment.
    - Autonomous path following mode: Animation of the robot following a pre-specified path using a given path planning and following aproach
    - Path selection mode: User input mode to specify points the robot must go to (used alongside autonomous path following mode)
- Robot class: model for differential drive robot
    - Motor model
    - Model for acceleration and velocity of individual sides
- Controls class:
    - Various path planning, path following, and driving algorithms
    - Ability to incorporate user-written python files containing other algorithms

Algorithmic Plan

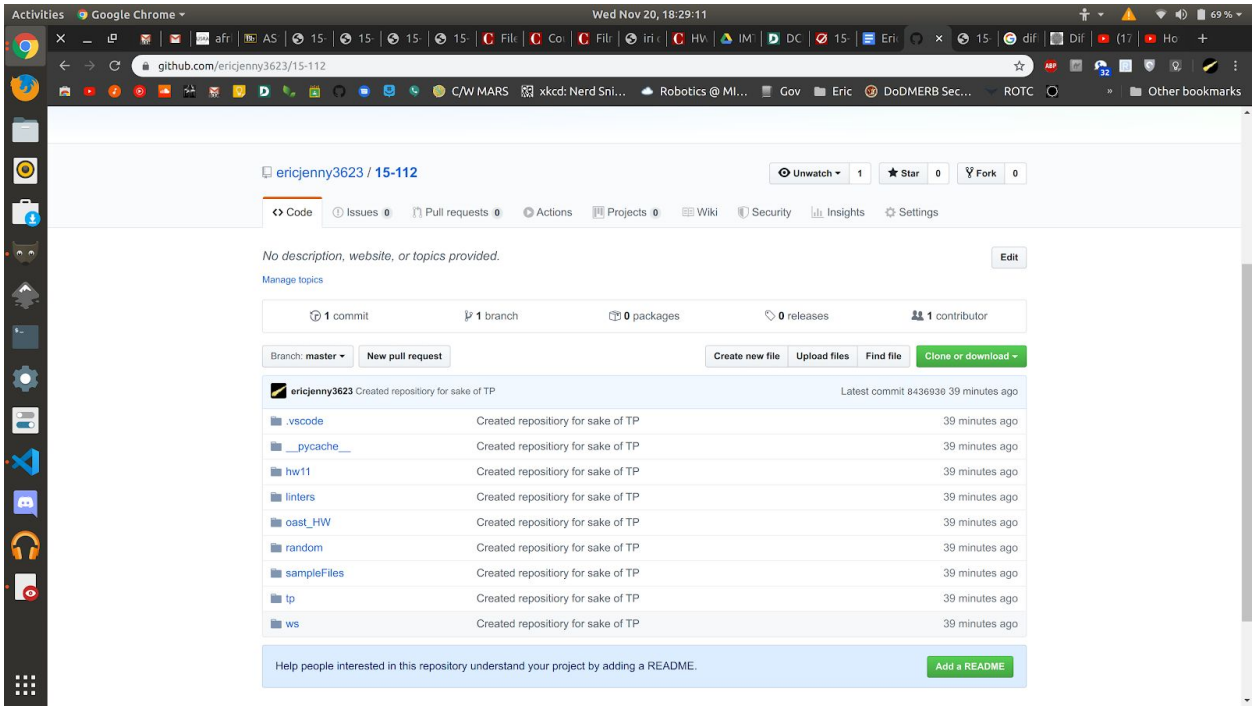
-

Timeline Plan

1. Nov 20 (TP1): Working primitive differential drive model, user controlled driving using keyboard
2. Nov 23: Working autonomous path following given prespecified inputs
3. Nov 26 (TP2): Working path planning (obstacle detection and avoidance)
4. Nov 30: UI Features (Graphs, replay ability, user-written code import)

Version Control Plan

Github will be used as version control with changes commited hourly and synced daily.



Module List

No modules currently planned to be used.

TP2 Design Update

Implementing collision detection has turned out to be a harder problem than expected, and might not be feasible to finish by TP3. Therefore, for TP3, more UI features will be implemented such as saving waypoints, coefficients, and other data in JSON files, and implementing more controls for the user such as variable sliders and menu selections.