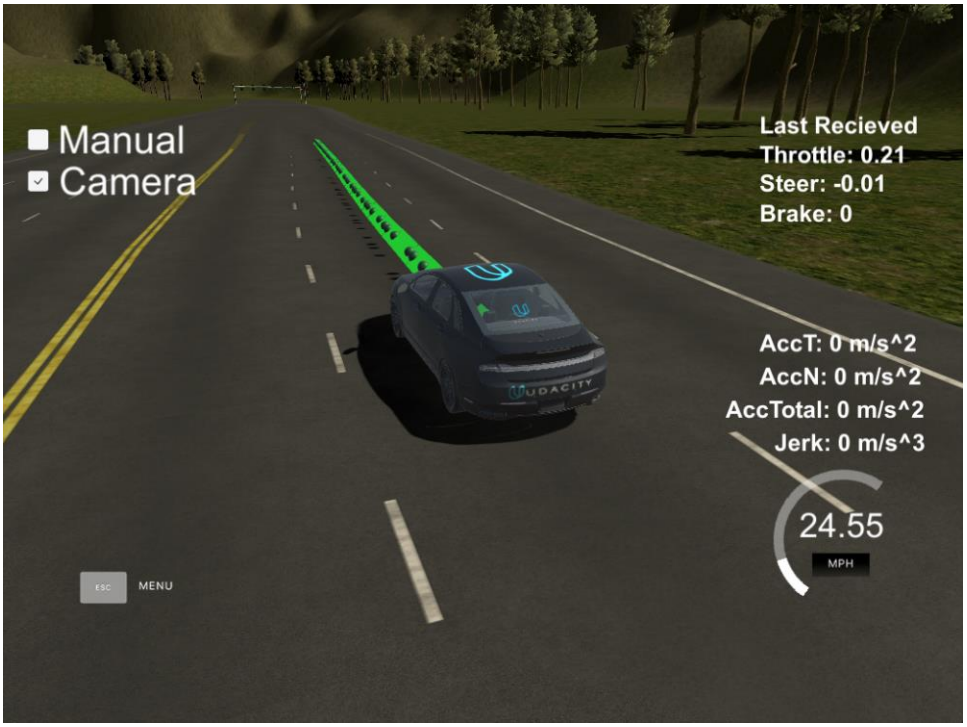
Project Overview

This is the project repo for the final project of the Udacity Self-Driving Car Nanodegree: Programming a Real Self-Driving Car.

Individual Submission

Name	Udacity Account Email
Muraliprasath Arumugam	murali.inst@gmail.com

Description



Waypoint Updater

This node publishes waypoints from the car's current position to some x distance ahead. If an upcoming stop light is detected, the velocity of the waypoints will be adjusted and the car decelerates or accelerates depending on the light state.

It was implemented by defining the following 4 states and changes between those states:

State	Description
Start Acceleration	This state determines how much acceleration is needed for the car to reach the target velocity.
Continue Acceleration	This state continues with the acceleration and keeps the target velocity if reached.
Start Deceleration	This state determines how much deceleration is needed for the car to stop at the next stop line.
Continue Deceleration	This state continues with the deceleration and keeps zero speed if reached.

Drive-By-Wire Node / Twist Controller

Drive-By-Wire Node

This node represents a drive by wire controller. It receives current and requested steering/velocities, calculates throttle, brake and steering commands and publishes them to the vehicle.

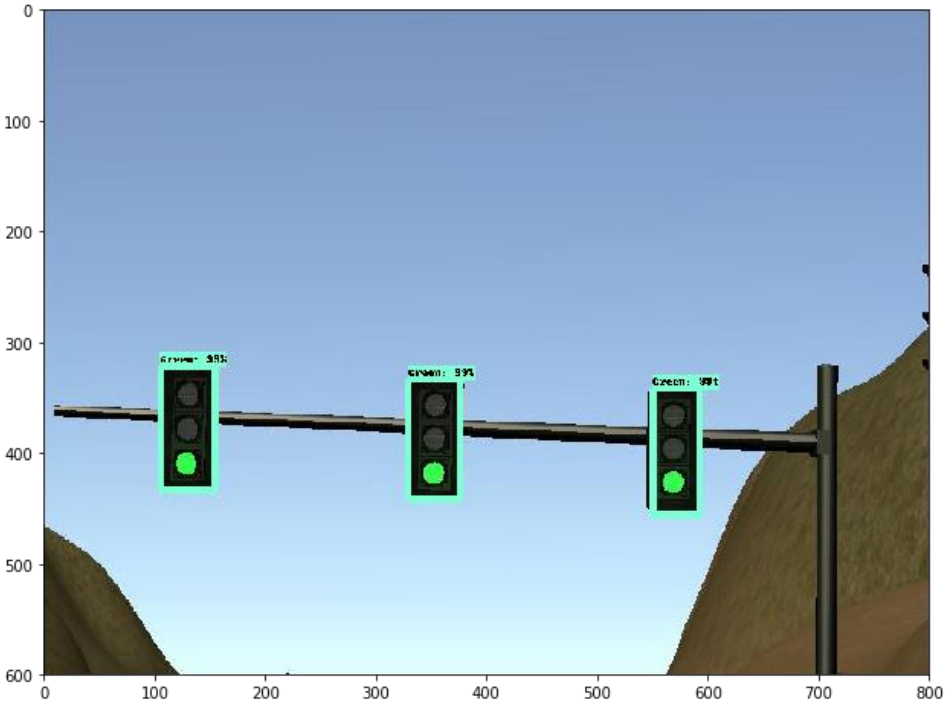
Twist Controller

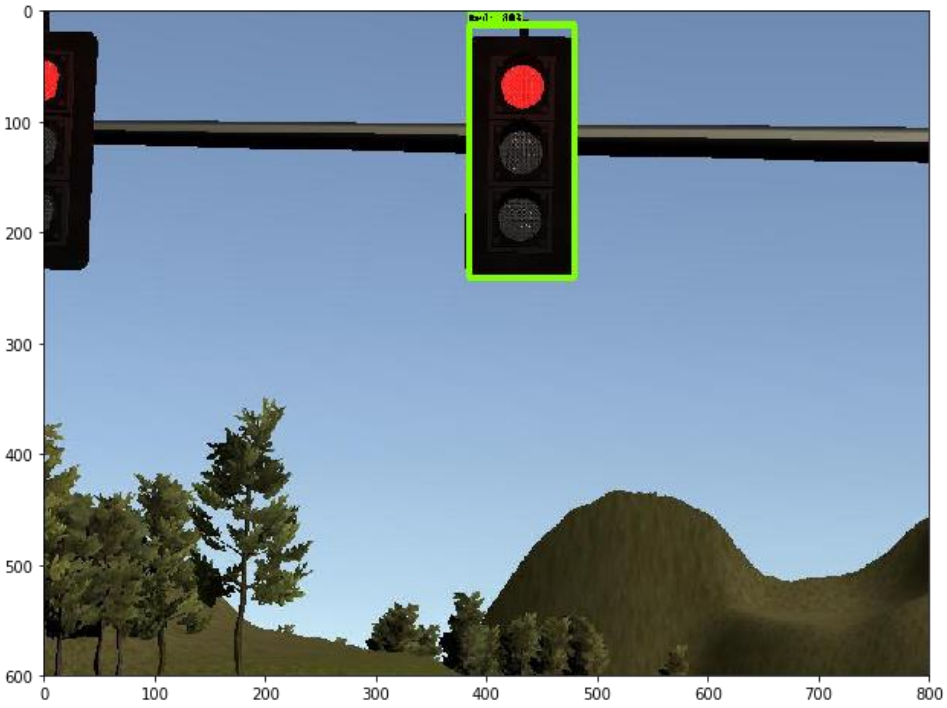
This controller is responsible for acceleration and steering. The acceleration is controlled via PID controller. Steering is calculated using YawController which simply calculates needed angle to keep needed velocity.

Traffic Light Detection / Classification

This node is responsible for detecting upcoming traffic lights and classify their states (red, yellow, green). Tensorflow's Object Detection API was used to detect and classify

the traffic lights in the images provided by the camera.





Result

1. Car running in the simulator successfully runs full track inside a lane.

2. Car deaccelerates whenever it sees the RED light and stops at 0 speed
3. It accelerates back to the normal speed without Jerk gradually
4. When the Red light is detected it slows down and during a slow down if the green is sensed it again pace back to the normal speed.

Flaws:

1. Model does not detect the first Red light at the start.