**Interim Report**

Predict Steering Angles in Self-Driving Cars and System Integration with ROS Nodes

**CT-509 Distributed Systems   
NED UET Fall 2018**

**By: SIDRAH YOUSUF  
ROLL NO. IS005**

**CLASS: MS (IS)**

**Abstract**

In future, the idea of automation will be implemented in innovative ways and grow fast as growing technology this will be more autonomous and furious when we are talking about automobiles. Major implementation is the ability to making the car self-driving and automates by an auto-system.  
This will not only make life easier as well as simpler and safer.

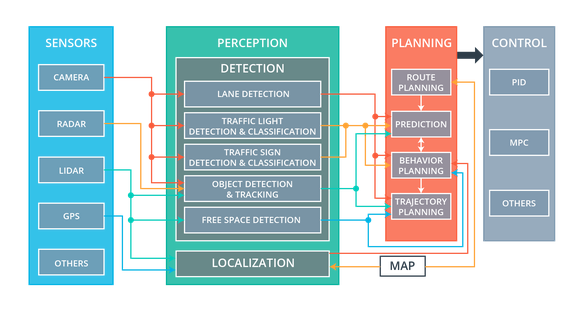
We have a dataset approximately 40GB which contains a set of images with different angles predicted and set by the human or any user of the car by real experience. These data measurement has been compiled by placing the camera behind the car’s windshield and captured the particular images from different angles.

This idea will be very useful and effective in many ways like the sensor for autonomous car will predict each angle in different directions and also its safer like it will not dependent upon driver’s mood.

There are multiple levels of prediction by different technologies like Neural Network Implementation and hybrid models etc.

**Introduction**

The self-driving frameworks in independent vehicles have four major components: sensors, discernment computer program, arranging program, and control computer program. Inside each of the computer program components, different modules execute concurrently to handle endless sums of information collected from the sensor frameworks. This handling in the long run comes about in control yields which control, quicken, and brake the vehicle at the proper minutes to induce the vehicle securely and effectively to its goal whereas giving a charming ride for its travelers.



**Technologies Used**

Python

ROS

Tensorflow

Python Imaging Library

OpenCV

**System Integration with ROS:**

**Autonomous Vehicle Systems**

In autonomous car steering prediction and integration each vehicle component has a specific task:

**Sensors** are the optional one for monitoring the hardware parts and collect the information about whole data about its environment this data is in raw form. Specifically include some special sensor functioning like RADAR, cameras, LIDAR, and even GPS sensors.

**Perception** frameworks (counting location and localization components) prepare crude sensor information into significant organized data almost the environment around a vehicle. The discovery components are mindful for path line discovery, activity sign and light discovery and classification, protest location and following and free space location. Localization components utilize sensor and outline information to decide the vehicle’s exact area.

**Planning** frameworks utilize the yield from the discernment frameworks to make driving behaviors and to arrange brief and long-term ways for the vehicle to take after. Arranging components incorporate high-level course arranging from begin to goal on a outline, forecast of what other objects on the street might do, behavior arranging to choose what particular activities the vehicle ought to take following, and direction era which plots the exact way the vehicle ought to take after.

**Control**

frameworks guarantee that the vehicle takes after the way given by the arranging framework and send commands to the vehicle’s equipment for secure driving. These incorporate sending speeding up, braking, and controlling commands. Information by and large streams from beat to foot, beginning with sensor frameworks, at that point moving through discernment, arranging, and at last to control systems.

**ROS Node**

The most DBW ROS hub performs the taking after setup upon being created:

* accepts a number of vehicle parameters from the vehicle configuration
* implements the dbw\_node ROS theme distributers and subscribers
* the four ROS subject endorsers (twist\_cmd, current\_velocity, dbw\_enabled, is\_decelerating) allot different occurrence factors, utilized by the Control class, once extricated from the subject messages
* creates a Controller occurrence to oversee the specific vehicle control
* enters a circle which gives the foremost later information from point endorsers to the Controller instance.

The circle executes at a target rate of 50Hz (any lower than this and the vehicle will naturally cripple the DBW control interface for security). The circle checks in the event that the DBW hub is empowered, and all fundamental information is accessible for the Controller, at that point hands the fitting values (current and target direct speed, target precise speed, and whether the vehicle is endeavoring to decelerate) to the Controller. Once the Controller returns throttle, brake, and controlling commands, these are distributed on the comparing ROS interfacing.

**Controller**

A Controller course oversees the computation of throttle, brake, and controlling control values. The controller has two fundamental components: speed control and directing control. The Controller, upon initialization, sets up a Yaw Controller occasion for computing controlling estimations, as well as three Moo Pass Channel occasions for throttle, brake, and steering.

**Speed control**

At each control ask, the taking after steps are performed:

* Compute the timestep from the final control ask to the current one
* Compute the direct speed mistake (the contrast between target and current direct velocity)
* Reset PI control respectability on the off chance that the vehicle is halted (includes a zero target and current straight speed); more on the trustworthiness.

Following, the crude throttle and brake values are computed. The essential design:

* adds variable throttle in the event that the vehicle is quickening or vehicle is abating down but not essentially sufficient to discharge the throttle entirely
* adds variable braking in case the vehicle is traveling as well quick relative to the target speed (and essentially discharging throttle will not moderate down quick enough)
* adds steady braking in the event that the vehicle is abating down to a stop. Once the raw throttle and braking values are computed, the raw braking value is sent through a low pass filter to prevent rapid braking spikes.

On the off chance that the coming about esteem is as well little (underneath 10Nm), the braking esteem is decreased to zero; else, the throttle is decreased to zero. Typically to anticipate the brake and throttle from inciting at the same time. At long last, the throttle esteem is sent through a partitioned moo pass channel to avoid fast throttle spikes.

**Steering control**

The target direct speed, target precise speed, and current straight speed are sent into the Yaw controller. This controller computes a ostensible controlling point based on a straightforward kinematic bike show. At long last, this directing esteem is sent through its possess moo pass channel to smooth out last controlling commands.

**Results**

For the final test, the working set is in under process of implementation. But angle prediction attain at some extent.

**References for Current work Status:**

1. Molla, Tesfamichael. (2018). Self-Driving car. 10.13140/RG.2.2.36042.82885.
2. Tian, Yuchi & Pei, Kexin & Jana, Suman & Ray, Baishakhi. (2018). DeepTest: automated testing of deep-neural-network-driven autonomous cars. 303-314. 10.1145/3180155.3180220.
3. Yadav,Neha & MA,Amherst & MA,Amherst.(2017).Predict Steering Angles in Self-Driving Cars
4. Tian, Yuchi & Pei, Kexin & Jana, Suman & Ray, Baishakhi. (2018). DeepTest: automated testing of deep-neural-network-driven autonomous cars. 303-314. 10.1145/3180155.3180220.
5. Du,Shuyang & Guo,Haoli & Simpson,Andrew.(2017).Self-Driving Car Steering Angle Prediction Based on Image Recognition
6. Eraqi, Hesham & Moustafa, Mohamed & Honer, Jens. (2017). End-to-End Deep Learning for Steering Autonomous Vehicles Considering Temporal Dependencies.
7. John, Vijay. (2018). Vision-based Steering Angle Prediction by the Fusion of Depth and Intensity Deep Features.

**WebLinks used in project\_middle part compilation:**

1. <https://towardsdatascience.com/teaching-cars-to-drive-using-deep-learning-steering-angle-prediction-5773154608f2>
2. <https://medium.com/udacity/challenge-2-using-deep-learning-to-predict-steering-angles-f42004a36ff3>