Analysis of vehicle driving behavior on special road segments using context-specific information

Abstract

Typical telematics and fleet-management systems today use embedded systems attached to the vehicle and get their driving data from their diagnostics port to identify the action of driver and grade it to provide feedback based on their quality of driving to efficiently handle the vehicle and also their driving behavior.

Today’s insurance companies provide embedded devices or the customer’s smartphone to analyze basic driving parameters such as speed, rpm, GPS location to understand driver’s braking, acceleration and distance travelled over a period and use it to assess quotes for insurance premium.

But most of the solutions above do not consider of context specific information in the cases of fixed-route scenarios whose details can be understood better in the first place and use it to grade the driver’s performance for the trip more efficiently.

In this experiment, a driver’s behavior on a pre-defined route is analyzed on different perspectives by also taking into account of the road context, such as turns, straight road segments, traffic lights, stop signs etc. and graded accordingly and providing a score to reflect their behavior in each segment of the road as well as a complete score for their trip

Introduction

Vehicle driving behavior analysis involves combination of hardware and software technologies which help us get real-time feedback about the vehicle as well be able to profile a driver’s behavior. Some of the core technologies involved are mentioned below

# Controller Area Network (CAN)

A Controller Area Network (CAN bus) is a robust vehicle bus standard designed to allow microcontrollers and devices to communicate with each other in applications without a host computer. It is a message-based protocol, designed originally for multiplex electrical wiring within automobiles to save on copper, but can also be used in many other contexts.[1]

CAN bus technology has been in development since its conception by Robert Bosch in 1983. The more widely used version of CAN, CAN 2.0 which are further classified into CAN 2.0A and CAN 2.0B based on the identifier width supported.

The latest CAN standard is the CAN-FD or Flexible Data Rate introduced in 2012 which is backward compatible with the CAN 2.0 standard and can also co-exist as a separate bus with the CAN 2.0

CAN bus is one of five protocols used in the on-board diagnostics (OBD)-II vehicle diagnostics standard. The OBD-II standard has been mandatory for all cars and light trucks sold in the United States since 1996[1]

# On-Board Diagnostics port (OBD)

On-board diagnostics (OBD) is an automotive term referring to a vehicle's self-diagnostic and reporting capability. OBD systems give the vehicle owner or repair technician access to the status of the various vehicle subsystems. [2]

The OBD-II standard provides list of standardized DTCs. As a result of this standardization, a single device can query the on-board computer(s) for these parameters in any vehicle.

Manufacturers may also add custom data parameters to their specific OBD-II implementation, including real-time data requests as well as trouble codes.

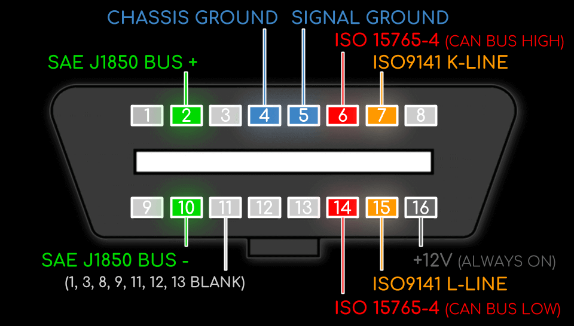


Figure 1: OBD 2 Port Pinout[3]

Typically, queries with particular PID values to get real-time data is performed to know the vehicles instantaneous speed and other engine parameters

While it did not meet the OBD-II requirements for U.S. vehicles prior to 2003, as of 2008 all vehicles sold in the US are required to implement CAN as one of their signaling protocols.

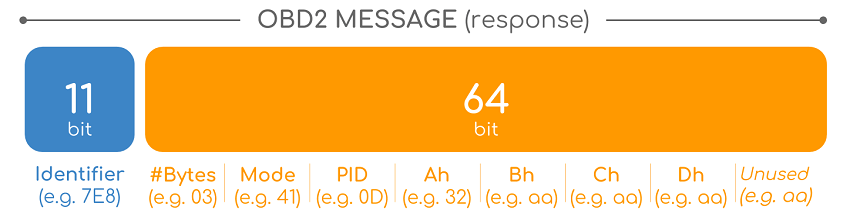


Figure 2: OBD2 Message format [3]

For OBD2 messages, the identifier is standard 11-bit and used to distinguish between “request messages” (ID 7DF) and “response messages” (ID 7E8 to 7EF).

There are 10 modes as described in the SAE J1979 OBD2 standard. Mode 1 shows Current Data and is e.g. used for looking at real-time vehicle speed, RPM etc. Other modes are used to e.g. show or clear stored diagnostic trouble codes and show freeze frame data.

For each mode, a list of standard OBD2 PIDs exist - e.g. in Mode 01 PID 0D is Vehicle Speed

# Telematics and Usage-based Insurance (UBI)

Telematics is a method of monitoring a vehicle. By combining a GPS system with on-board diagnostics it's possible to record – and map – exactly where a car is and how fast it's traveling, and cross reference that with how a car is behaving internally.[4]

Telematics is used in applications such as tracking vehicles, heavy-duty container trucks used for transportation and fleet management.

Fleet management is the management of a company's fleet. Fleet management includes the management of ships and or motor vehicles such as cars, vans and trucks. Fleet (vehicle) Management can include a range of functions, such as vehicle financing, vehicle maintenance, vehicle telematics (tracking and diagnostics), driver management, fuel management, health and safety management and dynamic vehicle scheduling.[5]

Usage-Based Insurance (UBI) is a type of auto insurance that tracks mileage and driving behaviors. UBI is often powered by in-vehicle telecommunication devices (telematics)-technology that is available in a vehicle that is self-installed using a plug in-device or already integrated in original equipment installed by car manufactures. It can also be available through mobile applications.

The basic idea of UBI is that a driver's behavior is monitored directly while the person drives, allowing insurers to more closely align driving behaviors with premium rates.[6]

Related works

Several works have been done before in the field of vehicle driving behavior analysis from just analyzing the driving behavior in general to focusing on certain aspects of driving such as turns, lane shifting etc.

Most of the works use data from various sources such as the data from OBD port, smartphone sensors, external sensors attached to embedded systems placed on the vehicle.

Some of the distinguishing works from various projects and research are highlighted below

# Software Solution for Monitoring and Analyzing Driver Behavior

The motive of [7] is that by giving to the user a reliable driving score, he can relate to it and be aware that there might be some problems in his actions, so he becomes more attentive

The data is collected using an external device connected to the OBD port of the vehicles. Speed, engine rpm, gear position is collected form the vehicles directly. GPS position and vehicle rotation are collected using sensors from the external device mounted to the vehicle

The data from all the sensors is mapped to a weighted graph called State Graph. This graph is used to detect anomalies by studying the weights of the edge

An Arduino based logger, Freematics is used for communicating with the car using OBD2 port.

Each trip receives a total score obtained from four different scores: acceleration, breaking, left corners, right corners

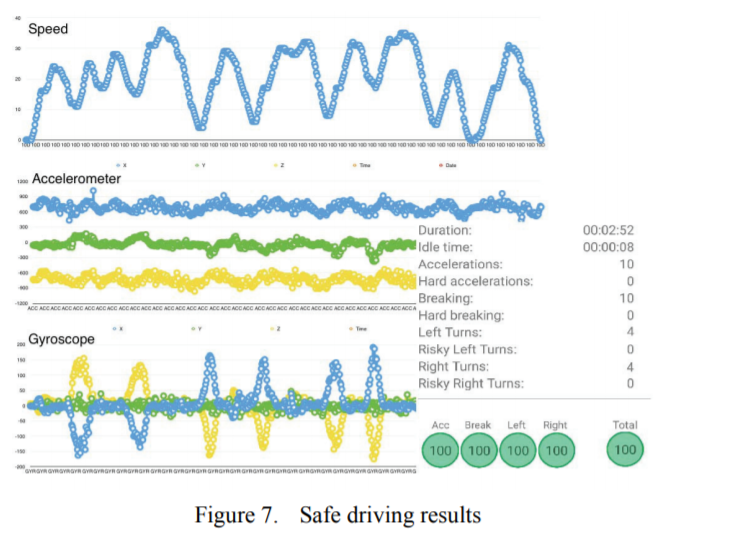
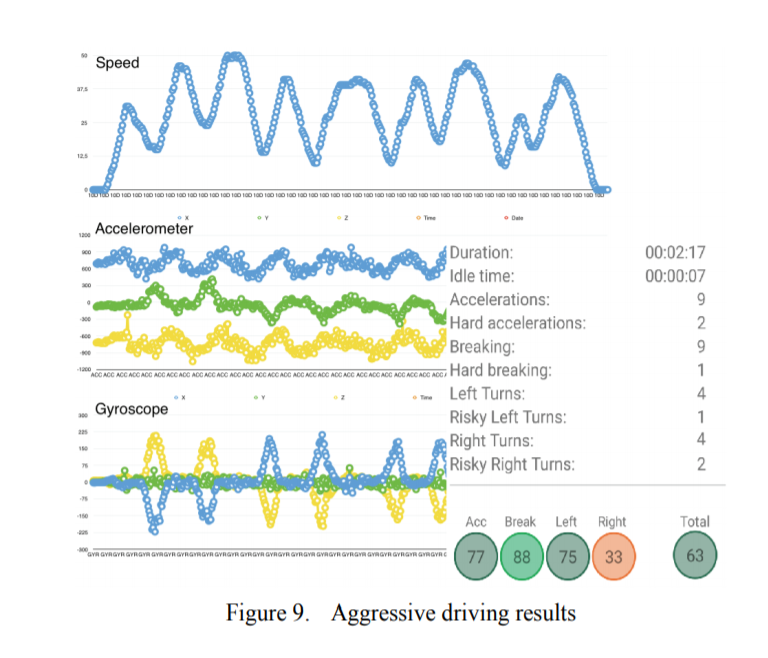


Figure 4: Aggressive driving results

Figure 3: Safe driving results

# **Invisible Sensing of Vehicle Steering with Smartphones**

In [8], they developed a vehicle steering detection middleware called *V-Sense* which can run on commodity smartphones without additional sensors or infrastructure support. Instead of using cameras, the core of *V-Sense/*senses a vehicle's steering by only utilizing non-vision sensors on the smartphone. Algorithms were designed and evaluated for detecting and differentiating various vehicle maneuvers, including lane-changes, turns, and driving on curvy roads

V-Sense performance was tested on a Samsung Galaxy S4 with a 1.6GHz quad-core processor running Android 4.4.1 KitKat OS. Over 40 hours of test were conducted and tried to cover different environments both in a parking lot and real roads.

The cars involved in the test were a 2010 Mitsubishi Lancer and a 2006 Mazda 6.

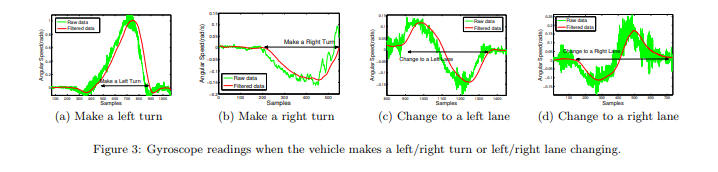


Figure 5: Gyroscope readings during different turns

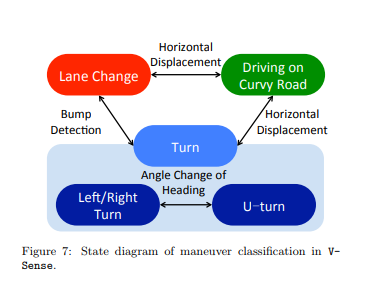


Figure 6: V-Sense state diagram for maneuver classification

# Modeling speed profiles of turning vehicles at signalized intersections

[9] Considers that turning vehicles need special attention in the context of the safety evaluation and improvement of signalized intersections.

By using empirical data of vehicle trajectories collected at signalized intersections in Japan, a model is developed and presented, which provides stochastic speed profiles of free-flowing left- and right turning vehicles. The speed profiles are sensitive to intersection layout and the vehicle speed and position at the beginning and ending of the maneuver

Based on the observations, a model describing the speed profile of unimpeded (free flowing) turning vehicles was developed. The speed profile is divided into two parts, an inflow part and an outflow part, the boundary defined by the moment the vehicle reaches the minimum speed. The acceleration of both parts was found to follow a parabolic shape

A polynomial of third degree for the speed as a function of the time as shown in fulfills this requirement. Different coefficients are chosen for the inflow and the outflow

Figure 7: 3rd degree polynomial used to fit the speed profile

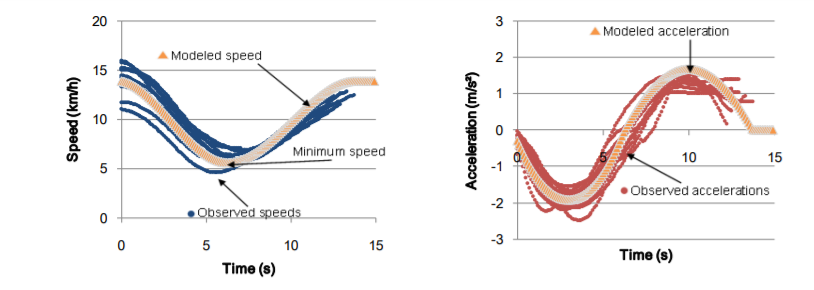


Figure 8: Speed and acceleration over time

For the regression analysis the trajectory data had first to be classified into the inflow and outflow part of the individual speed profiles and cleaned for outliers by visual inspection. The polynomial speed function was then fitted to each trajectory (separately for inflow and outflow).

They found that their coefficients showed an influence of entering of speed of vehicle, approach angle etc. Their speed profiles were typically applicable to free-flowing vehicles, i.e. when not influenced by external disturbances like other vehicles

Implementation

# Objective

Our objective is to create a software which will be able to analyze driving behavior of a person in a fixed route and provide them a driving score for each segment of the road as well as a holistic report which helps to highlight the bad and good driving aspects.

This software is designed by keeping in mind to be used in places where fixed routes are assigned to drivers and multiple trips are made in the same route.

By getting the context-information of the route beforehand, we believe that we gain an advantage over other similar implementations which try to generalize a driver’s environment and provide the same analysis.

# Hardware Setup

The Hardware setup consists of the following devices, viz. Raspberry Pi 3+, PiCAN 2 module and a

GPS module. Raspberry Pi is the mini-computer responsible for the processing needs.

The PiCAN 2 shield attached to the GPIO pins of the Pi serves as an interface for the Pi to connect to the OBD II port of the car to log CAN data. GPS module is attached to the Pi to obtain the location of the car.



Figure 9: Hardware setup

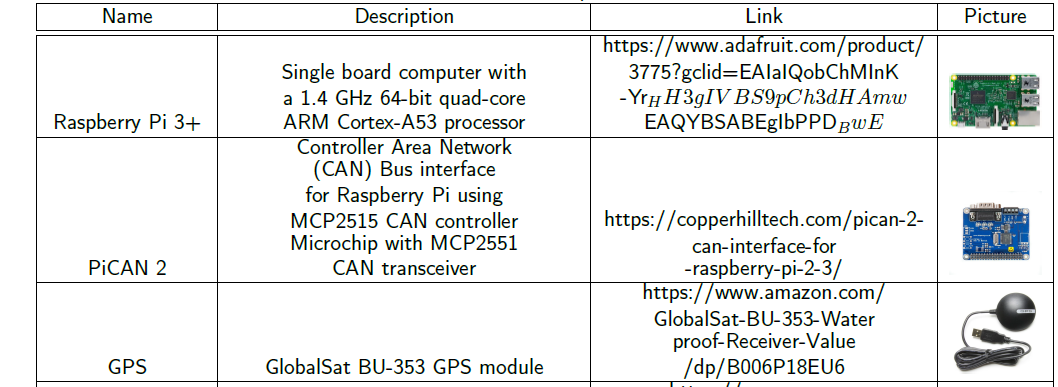


Figure 10: List of hardware components

The vehicle used for testing was a 2011 model Scion xB which supported CAN via its OBD 2 port. The raspberry pi was powered using the Car’s 12 V socket which in turn powered the GPS

## Other hardware considered:

As an alternative for Raspberry pi, few other telematics devices were also considered which could provide better software and hardware support in the long run.

One of them tested hands-on was the Reliagate 10-12 model from Eurotech

The ReliaGATE 10-12 is a low power gateway suitable for demanding use cases: it supports a 6 to 36V power supply, two protected RS-232/RS-485 serial ports, two CAN bus interfaces, three noise and surge protected USB ports, and four isolated digital interfaces.[10]

It is powered by a TI AM335x Cortex-A8 processor with 1 GB of RAM and 4GB of eMMC support.

The ReliaGATE 10-12 features a wide range of connectivity capabilities: it integrates an internal LTECat 1 cellular modem with dual Micro-SIM support, Wi-Fi, Bluetooth, and two Fast Ethernet ports; an optional internal GPS provides precise geolocation capabilities.



Figure 11: Reliagate 10-12 model from Eurotech

The ReliaGATE 10-12 comes with Everyware Software Framework (ESF), a commercial, enterprise-ready edition of Eclipse Kura, the open source Java/OSGi middleware for IoT Edge Gateways.

Distributed and supported by Eurotech, ESF supports ready-to-use field protocols (including Modbus, OPC-UA, S7), MQTT connectivity, web-based visual data flow programming and deep configuration. ESF is also integrated with Everyware Cloud (EC), Eurotech IoT Integration Platform (separately available), enabling advanced diagnostics, provisioning, and full remote device access and management.

Steps were taken to understand the hardware and software setup of the device. But unfortunately, due to lack of technical support across internet and the manufacturer, we were not able to build custom plugins for test on the device.

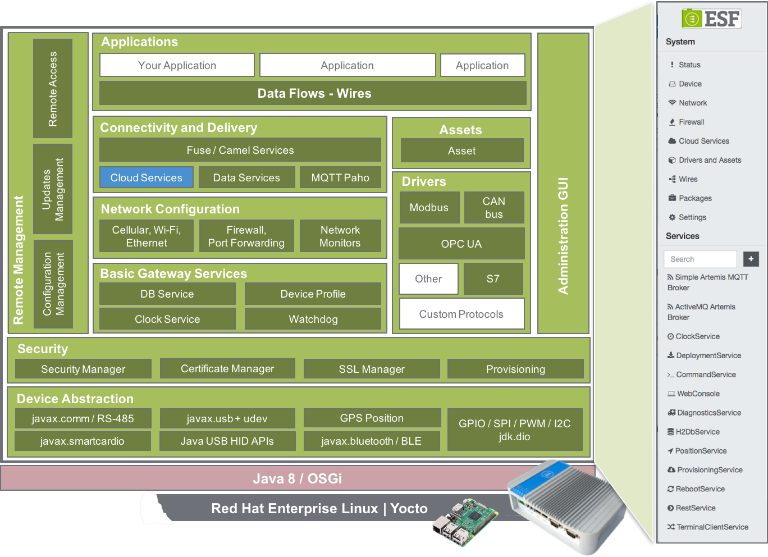
Also, the device does not run Python 3 which is required to port raspberry pi scripts to this device.

Figure 12: ESF Multi-level framework

Issues were found when building and deploying custom plugins and couldn’t solve in sufficient time.

While the device serves as a better replacement in terms of hardware build, the software side of the device lacks enough documentation and support to continue developing plugins for it.

# Software Setup

Python version 3.x was used for both data collection using OBD port as well as for the grading engine.

For the raspberry-pi based data logger, following python libraries were used:

* Python-can – for CAN library
* Gps3 – to interface with gpsd services

Further the log files are planned to be stored, graded and retrieved using a cloud service and a web-based GUI is intended to display various results.

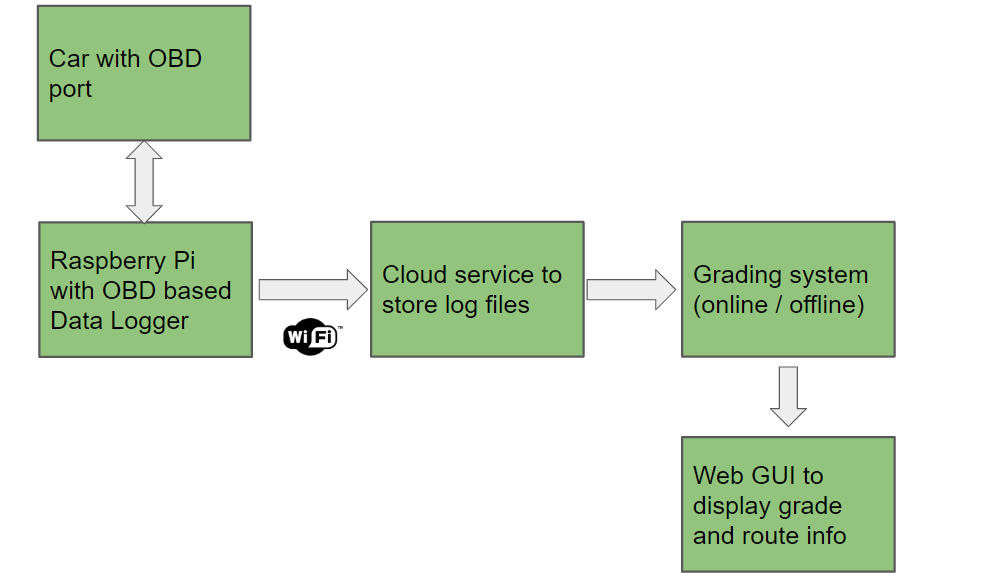


Figure 13: Overall setup in plan

# Test Route

The test route was chosen by keeping in mind to contain different type of road elements such as stop signs, stop lights, left and right turns, roundabouts and straight segments of road.

The test route is located at Ames, Iowa with a lap of about 8 miles long.

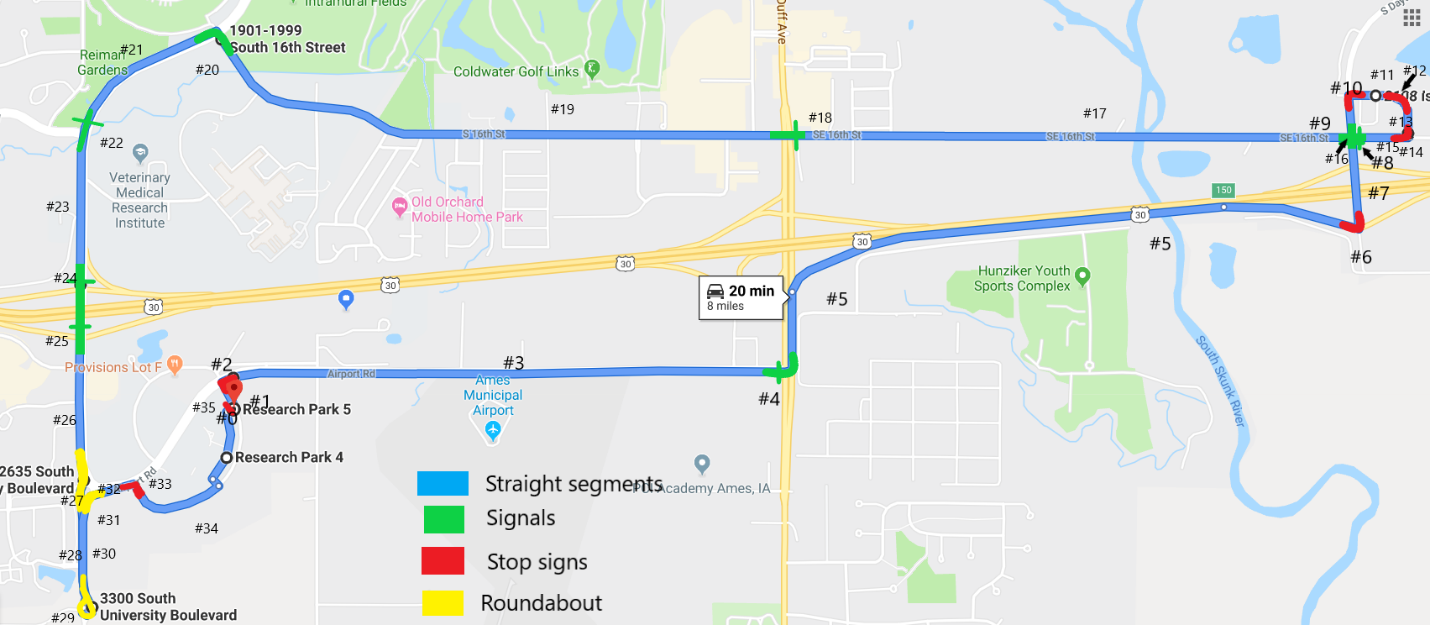


Figure 14: Route tested for driving behavior analysis

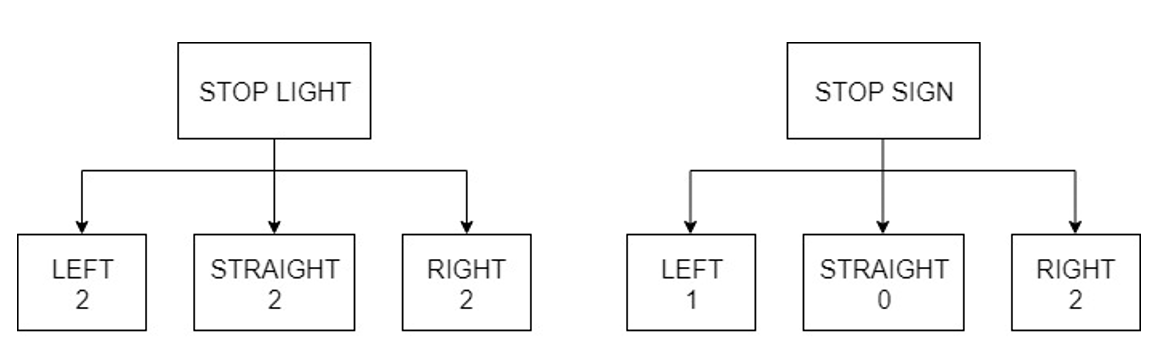


Figure 15: Left and right turns from stop sign and stop lights

There are total 35 segments in the test route and they are as described below:

|  |  |
| --- | --- |
| Segment number | Description |
| 0 | Depot to road |
| 1 | straight to airport rd. stop |
| 2 | airport rd stop sign |
| 3 | airport rd |
| 4 | airport rd. signal |
| 5 | highway |
| 6 | highway end stop-sign left |
| 7 | straight on bridge |
| 8 | signal on bridge |
| 9 | straight to issac newton drive |
| 10 | issac newton drive free right |
| 11 | straight |
| 12 | 2nd free right on issac newton drive |
| 13 | straight |
| 14 | stop sign right to S 16th |
| 15 | straight |
| 16 | Signal |
| 17 | Straight |
| 18 | signal # near old chicago |
| 19 | Straight |
| 20 | signal left |
| 21 | Straight |
| 22 | Signal |
| 23 | Straight |
| 24 | signal |
| 25 | Signal |
| 26 | Straight |
| 27 | roundabout |
| 28 | Straight |
| 29 | roundabout |
| 30 | Straight |
| 31 | roundabout |
| 32 | Straight |
| 33 | right turn to S loop drive |
| 34 | Straight |
| 35 | Stop sign at bus depot |

# Grading

## Thresholding

The first step in order to grade is to set the thresholds in which acceleration or deceleration is considered bad to take off penalty.

A bad acceleration or deceleration level can be depended on many criteria such as type of vehicle, human perception, levels which may seem dangerous on roads and can vary according to the environment of the road on which the driver is driving.

For e.g. Verizon in [11] states different thresholds for hard accelerations and hard braking also differentiated for different class of vehicles based on their size as light/medium and heavy for their telematics device.

Adaptive acceleration threshold based on context like vehicle traffic, signal lights are difficult to predict just based on driving parameters unless external sensors are used.

Hence, we decided to follow a regression-based thresholding approach, where we obtain acceleration and deceleration thresholds which we consider bad at a given speed based on the driving data we collected.

The acceleration/decelerations were plotted for different speeds and a closely fitting trendline was obtained with help of Microsoft Excel’s graphing capabilities.

Figure 16: Acceleration regression for regular segments Figure 17: Acceleration regression for turn

Speed based acceleration thresholds - applicable to all categories

* Here adaptive speed - based thresholding is derived exponential regression equations
* Thresholds for turn / straight segments are different. Stricter for turns

Similarly, for deceleration

Figure 17: Deceleration regression for regular segments Figure 17: Deceleration regression for turn

Based on this, acceleration or deceleration thresholds are determined dynamically based on the vehicle speed as harsh accelerations are bad at higher speeds than in lower speeds.

For e.g. for heavy vehicles, jackknifing is a common problem at higher speeds which tends to make the vehicle unstable [12]

For turns, it is recommended to keep the turning speed around 15-25 mph [13]

References

[1] <https://en.wikipedia.org/wiki/CAN_bus>

[2] <https://en.wikipedia.org/wiki/On-board_diagnostics>

[3] <https://www.csselectronics.com/screen/page/simple-intro-obd2-explained/language/en>

[4] <https://www.techradar.com/news/car-tech/telematics-what-you-need-to-know-1087104>

[5] <https://en.wikipedia.org/wiki/Telematics>

[6] <https://www.naic.org/cipr_topics/topic_usage_based_insurance.htm>

[7] Găvrută, Alex, Marius Marcu, and Răzvan Bogdan. "Software Solution for Monitoring and Analyzing Driver Behavior." 2018 IEEE 12th International Symposium on Applied Computational Intelligence and Informatics (SACI). IEEE, 2018.

[8] Dongyao Chen, Kyong-Tak Cho, Sihui Han, Zhizhuo Jin, and Kang G. Shin. 2015. Invisible Sensing of Vehicle Steering with Smartphones. In Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '15). ACM, New York, NY, USA, 1-13.

[9] Wolfermann, Axel, W. K. Alhajyaseen, and Hideki Nakamura. "Modeling speed profiles of turning vehicles at signalized intersections." 3rd International Conference on Road Safety and Simulation RSS2011, Transportation Research Board TRB, Indianapolis. 2011.

[10] <https://www.eurotech.com/en/products/iot/multi-service-iot-edge-gateways/reliagate-10-12>

[11] <http://tracknet.accountsupport.com/wp-content/uploads/Verizon/Hard-Brake-Hard-Acceleration.pdf>

[12] <https://en.wikipedia.org/wiki/Jackknifing>

[13] <http://www.drivingtips.org/steering-a-car.html>

[14] turn speeds and crashes within right-turn lanes <https://static.tti.tamu.edu/tti.tamu.edu/documents/0-4365-4.pdf>

[] Chen, Shi-Huang, Jeng-Shyang Pan, and Kaixuan Lu. "Driving behavior analysis based on vehicle OBD information and adaboost algorithms." Proceedings of the International MultiConference of Engineers and Computer Scientists. Vol. 1. 2015.