

## Background

With recorded vertical acceleration of different cars on a given road, we can accurately cancel it using feedforward control. For that purpose, we can map the road data in terms of road velocity and store the data in the server. However, instead of actually driving different cars on the same road and record their verticle accelerations, it is more efficient to train a neural network model for each car to predict the acceleration whenever some new road data is available through crowd-sourcing. In this assignment, you are asked to train such a model with given data.

## Sample Data

You can find two folders in this repository, they are training data and testing data, respectively. Each folder contains multiple CSV files, and each of them was extracted from a real drive session of the exact same car with same suspension controller settings.

Each CSV file is organized as follows:

```
| Time stamp | Vehicle speed in (km/h) | Road vertical velocity (m/s) | Vertical acceleration |
```

Each row contains data collected at the corresponding time stamp. The sampling frequency is approximately 100 Hz.

## Requirements

- Train a neural network that takes road velocity (m/s) as input and generates vertical acceleration ( $\text{m/s}^2$ ) as output.
- Feel free to choose any machine learning packages you are familiar with (e.g., TensorFlow, Keras, PyTorch, MATLAB DL Toolbox, CNTK, etc.)
- Please use one of the four programming languages: MATLAB, Python, Javascript, C/C++.
- Create a new branch for your work. Please provide code, figures, and summaries of your work.

## Hints

- It is possible to normalize velocity and acceleraiton using speed. For example, if you set a nominal speed to 20m/s. Then, velocity can be linearly normalized and acceleration can be quadratically normalized.
- Very low speed data may not be useful since the vehicle dynamics are different than mid or high speed. Also, large yaw rate indicates narrow turning, we are less interested in that data.
- RNN (LSTM) is preferred but a feedforward network may also work.