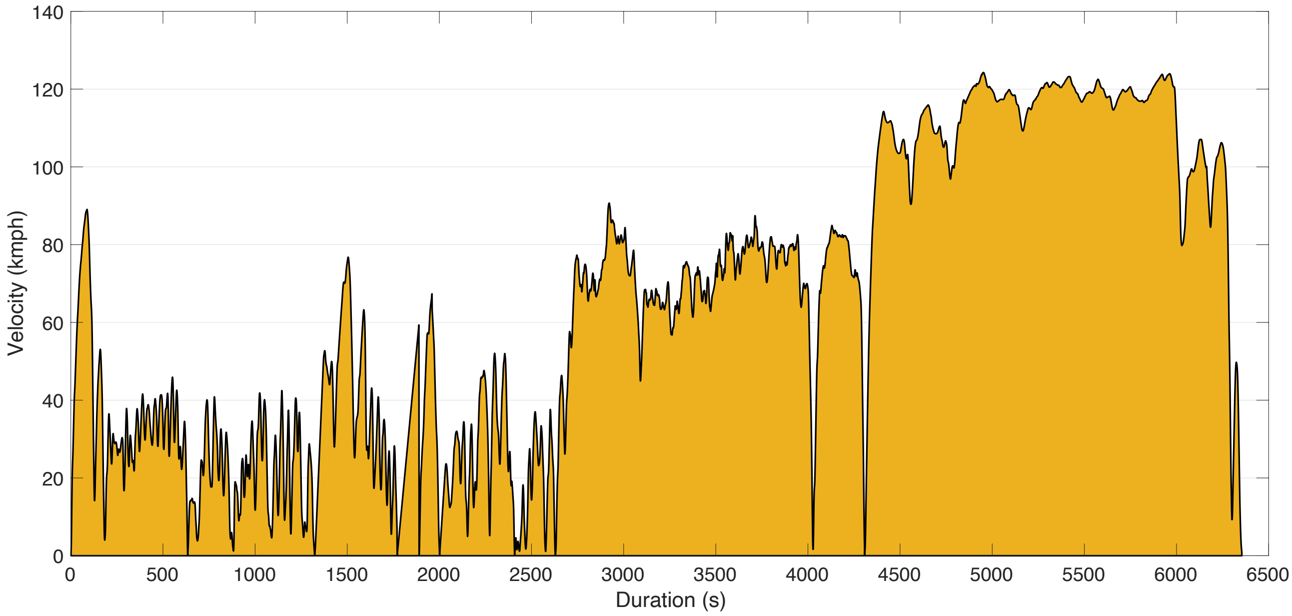
CHALMERS
UNIVERSITY OF TECHNOLOGY



Algorithm to generate Synthetic Driving Cycle using Real driving data

Master's thesis in Automotive Engineering

SUBRAMANYA NAGARAJ

DEPARTMENT OF MECHANICS AND MARITIME SCIENCES
DIVISION OF COMBUSTION AND PROPULSION SYSTEM

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2020
www.chalmers.se

Master's thesis in automotive engineering

Algorithm to generate Synthetic Driving Cycles using Real driving data

SUBRAMANYA NAGARAJ

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mechanics and Maritime Sciences
Division of Combustion and Propulsion System
Chalmers University of Technology
Gothenburg, Sweden 2021

Algorithm to generate Synthetic Driving Cycles using Real driving data
SUBRAMANYA NAGARAJ

Master's Thesis 2020:68
Department of Mechanics and Maritime Sciences
Division of Combustion and Propulsion System
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Synthetic driving cycle developed from Real driving trips.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2021

Algorithm to generate Synthetic Driving Cycles using Real driving data
Master's thesis in Automotive Engineering
SUBRAMANYA NAGARAJ
Department of Mechanics and Maritime Sciences
Division of Combustion and Propulsion System
Chalmers University of Technology

Abstract

The growth of technology has led to much increase in pollution levels. The European Union has enforced strict rules for car manufacturers to reduce the emission levels for vehicles. The regulation of the European Union includes a test for Real Driving Emissions. The automobile manufacturers are forced to test their vehicles for Real Driving Emissions. The available driving cycles like WLTC or NEDC lack real-world driving characteristics. This makes it is highly essential to develop a driving cycle by using real driving data. An algorithm is created to produce a driving cycle delivering the parameters within the Real Driving Emissions test parameters.

In this master thesis, micro-trip based construction model is applied for the vast data collected from real driving trips. The process includes use of unsupervised learning algorithm by utilizing k-means clustering technique to group the data. The statistical CH index is used to evaluate the performance of clustering and the trips are filtered with the Real Driving Emissions parameters before deploying D-optimal design to maximize the created design matrix from the filtered data. The micro-trips are selected in a ratio of 7:1:1 with urban, rural and motorway sections to stay within the required duration limits. The selected micro-trips are combined to form complete driving cycles, and are simulated using a simulation model constructed by using QSS toolbox in Simulink. The model comprises a normal IC engine with manual transmission, capable enough to determine the fuel consumption.

The developed driving cycles are analyzed and their parameters are compared with real driving emission test criteria. The results show that the cycles are valid. The results of the simulation are dependent on the engine operating points. The transmission model needs to be calibrated and evaluated with the real scenario to increase accuracy. The regression analysis carried forward from the simulations, predicts the relation of VA_{pos} with fuel consumption. The aggressiveness of the cycle tends to increase fuel consumption. Hence, it helps to understand the variation in fuel consumption based on the driving cycle parameters

Keywords: Micro trip based construction model, K-means clustering, CH Index, D-optimal design, Driving cycle, fuel consumption.

Acknowledgements

The satisfaction and euphoria that accompany the successful completion of this master thesis would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crowned our efforts with success.

I take this opportunity to express my gratitude to academic examiner David Sedarsky along with technical supervisors Jonas Sjöblom and Sten Karlsson for their guidance and valuable inputs during the entire thesis work. I am fortunate to have them supervising me by provinding invaluable support in the form of sharing technical content and ideas whenever in need and helping me to understand the concepts required for development of Synthetic driving cycle by characterizing real driving trips.

Finally, I would like to acknowledge my gratitude to my parents for their love and support during the entire duration of master studies and thank my friends for supporting me when I was in need to continue further.

Thank you very much!

Subramanya Nagaraj, Gothenburg, June, 2021

Nomenclature

BCV	Between-the-Cluster Variance
BSFC	Brake Specific Fuel Consumption
CH	Calinski-Harabasz
cm	Centimeter
CO_2	Carbon dioxide
CV	Co-Variance
DB	Davies-Bouldin
DC	Driving Cycle
ECE	Economic Commission of Europe
ED	Euclidean Distance
EU	European Union
EUDC	Extra Urban Driving Cycle
GLS	Generalised Least Squares
GPS	Global Positioning System
HCCI	Homogeneous Charge Compression Ignition
HEV	Hybrid Electric Vehicles
ICE	Internal Combustion Engine
kW	Kilowatt
MVEG	Motor Vehicles emission Group
NEDC	New European Driving Cycle
NO_x	Oxides of Nitrogen (NO, NO_2)
OLS	Ordinary Least Squares
PCCI	Premixed Charge Compression Ignition
RDE	Real Driviing Emissions
RLS	Recursive Least Squares
RPA	Relative Positive Acceleration
RPM	Revolutions per minute
SI	Silhouette Coefficient
WCV	Within-the-Cluster Variance
WLTC	World harmonized Light vehicles Test Cycle
WLTP	World harmonized Light vehicles Test Procedure

Contents

List of Figures

List of Tables

1
Introduction

As technology is growing, the advances in the area of combustion engines have grown rapidly. The emissions from automobiles have been devastating, leading to an increase in global warming, adverse climatic changes, etc. To reduce the effects on the environment due to the emission, certain conditions based on the parameters causing the harmful emissions are induced on the automotive manufacturers in the form of a speed chart or driving cycle. The vehicles are tested according to these driving cycles before launching them in the market. From the year 2017, European Union introduced real-driving emissions test procedure on passenger cars and light commercial vehicles. This procedure aimed to control the pollution caused by vehicles, and this will apply to all new cars from 2021[1]. Hence, it is very important to understand the real-world driving behavior using real-world driving data.

In this report, data collected from privately driven vehicles in Sweden is evaluated. The evaluated data is processed to create driving cycles as per the regulations imposed on real driving emission procedures. This is done at the Chalmers University of Technology as a final project for a master's degree in Automotive Engineering.

1.1 Background

Road traffic causes a dramatic increase in emission levels. The spike is boosted by increasing car users, especially over the years. Over the last decade, an approximate 21% increase in the number of vehicles has caused air pollution to rise to a significant level due to emissions[2],[3]. The emission of greenhouse gases during the 21st century has led to a prediction of a 3°C rise in the global temperature[4]. Climatologists predict an increase of sea level, leading to a high risk of flood situations. From the early stages, several norms to control various emission from vehicles have been imposed by the European Union. The stringent rules to improve the air quality by reducing the emission of NO_x serves as a challenge for car manufactures globally. The study of Hooftman et al.,[5] states that nearly 46% of NO_x is by the automobile sector globally, and 80% of those are from the combustion of diesel in cars, buses, trucks. The strong drift of the European Union towards pollution control forced the combustion researchers to experiment with different advanced technologies such as homogeneous charge compression ignition (HCCI), premixed charge compression ignition (PCCI)[6]. The various dramatic changes in the rules framed by European Union to cut down the air pollution have also led to introduction of electrified vehicles. Thus, a new era with penetration of Hybrid electric vehicles began [7].

Since 1992, the goal of the European Union to reduce CO_2 and NO_x emissions led to few important changes in the certification of vehicles[8]. Changes included utilization of driving cycles like the New European Driving Cycle (NEDC), and lately the more representative World harmonized Light vehicle Test Procedure (WLTP) for vehicle certification.

The driving cycles like NEDC, WLTC are created for assessing the emission levels and fuel economy of lightweight/passenger cars[9]. This is even referred to as Motor Vehicles Emission Group Cycle (MVEG)[10]. The driving cycles are used for type approvals using a chassis dynamometer. The NEDC cycle includes 2 segments of cycles, wherein one cycle-ECE 15(fig:7.1) is repeated 4 times and concluded with a high-speed cycle-EUDC(fig:7.2).The overall NEDC cycle is shown in fig:1.1. The NEDC involves constant accelerations and decelerations. The WLTC test cycles are based on the regions and used in Europe for vehicle type approval[11]. The test cycle has 2 divisions, based on the maximum speed of the vehicle. The cycle has 4 segments based on the vehicle speed: Low, Medium, High, and Extra High (fig:1.2).

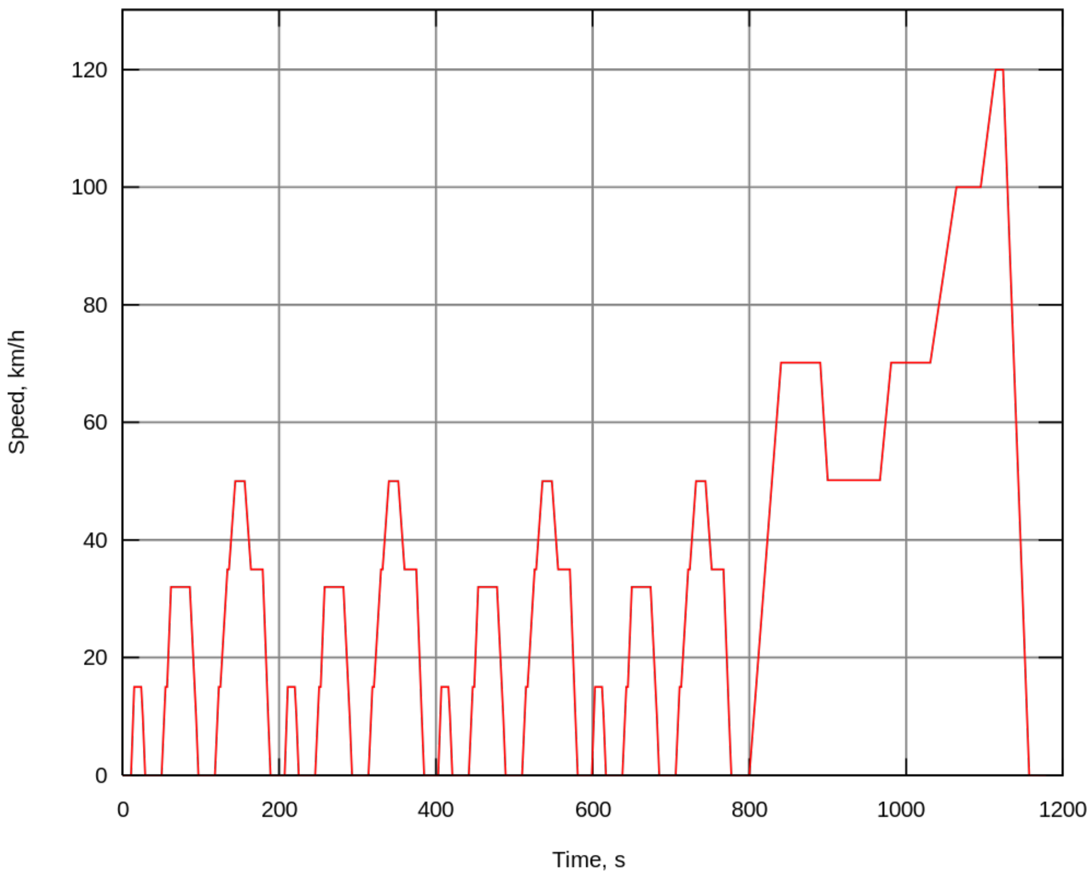


Figure 1.1: *New European Driving Cycles (NEDC)*

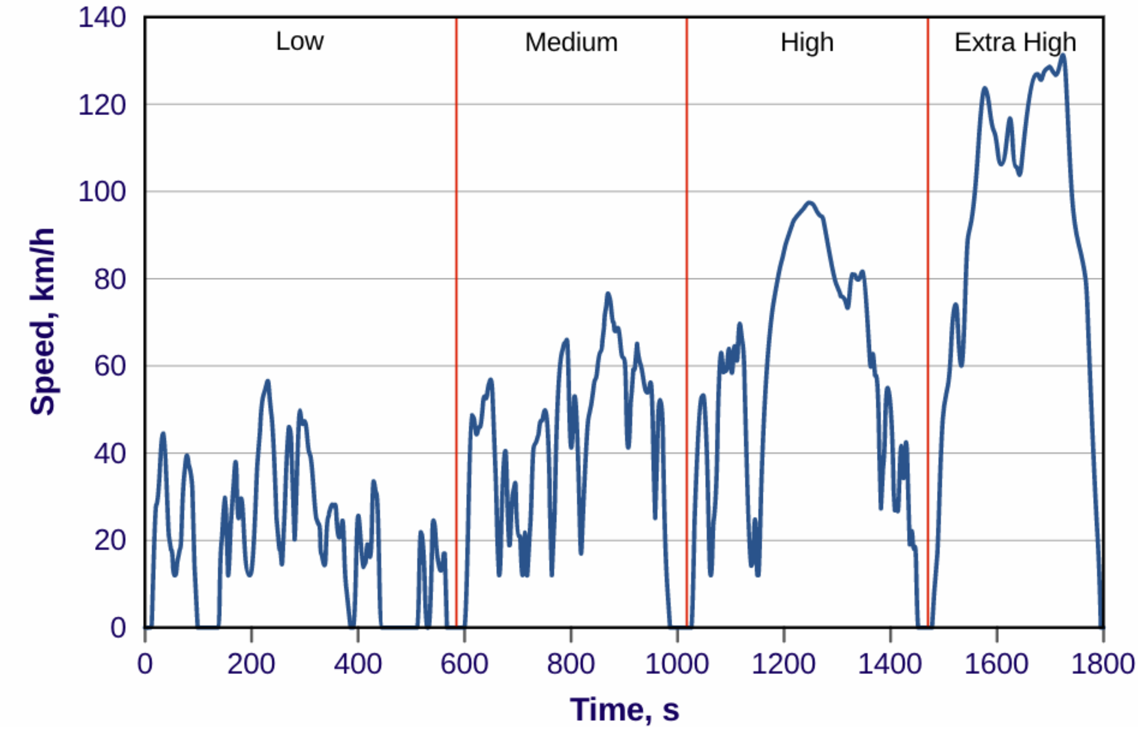


Figure 1.2: *World harmonized Light vehicles Test Cycle*

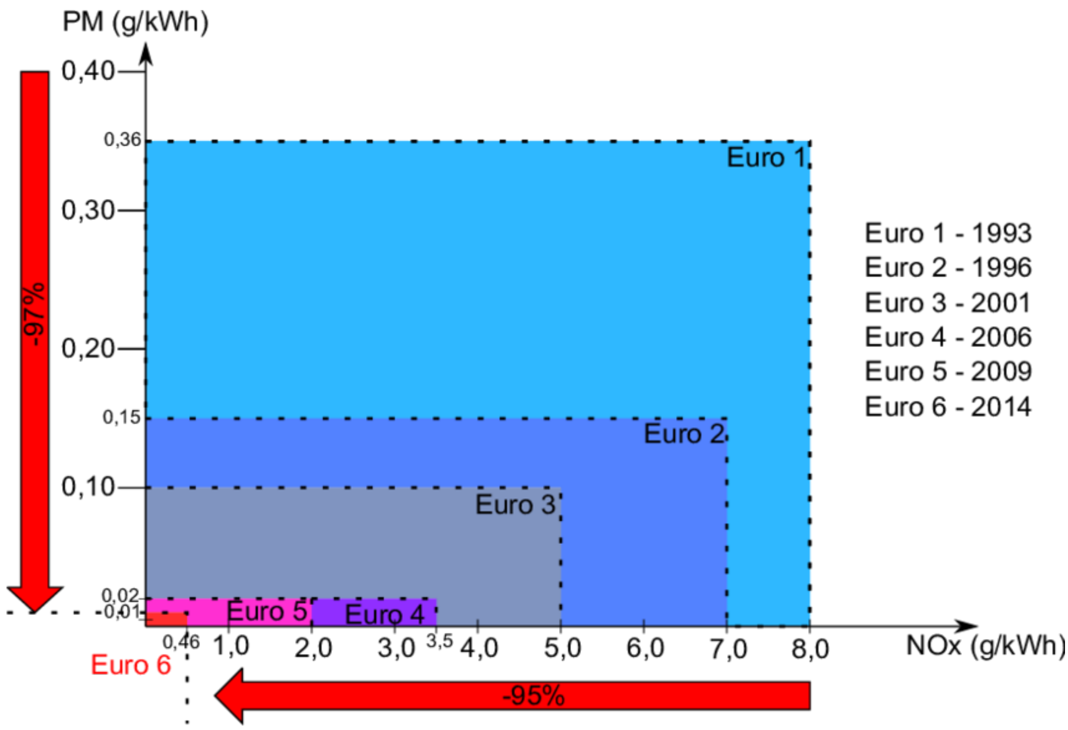


Figure 1.3: *Euro emission norms*

However, there is a significant amount of difference in NO_x emission between the type approval test done using the representative cycles and Real-world situations. For instance, a study from Chen et al.,[12] depicts that, the shift from Euro-1 to

Euro-6 (fig:1.3) to reduce the emission of NO_x has led to no significant reduction in emissions from European diesel cars. Hence, it is very important to test the vehicles under real-world driving conditions. Real driving emissions test poses as a hurdle to manufacturers, as it is difficult to re-create real-world environmental changes, traffic situations, the behavior of fellow drivers, etc.

A bad representativeness of the test cycle with the real driving conditions may lead to major errors in the estimated emissions and fuel consumption[13]. Based on the study by Lujan et al[8], the emission of NO_x is highly dependent on the share of a trip in the urban section, as well as the aggressiveness and driving behavior. This makes it very much important to have test cycles to evaluate the engine performance based on real driving characteristics[14]. The real driving data includes the driving behavior of fellow drivers, aggressiveness, dynamics of driving[15].

1.2 Aim and Objectives

This thesis focuses on the development of an algorithm of a synthetic driving cycle from the data of real driving trips logged through GPS from 378 privately driven Swedish cars. The synthetic driving cycle shall involve the desired cycle characteristics within the boundaries of the Real driving emission test protocol. The synthetic driving cycle is developed based on statistical methods. In more detail, the aim of the thesis was

- To develop a list of trip parameters that affect the emission and performance of the vehicle.
- To analyze real driving trips and categorize them through a statistical approach.
- To group the trip segments and optimize in the most possible flexible way for obtaining synthetic driving cycles.
- To simulate the driving cycles for estimating fuel consumption and compare the trip parameters with RDE boundary conditions.
- To present the algorithm, capable to develop a driving cycle from the provided trip data.
- To make recommendations and create guidelines for the developed driving cycle.
- To present the method used in the algorithm and the driving cycle in the form of a master thesis report.

1.3 Outline

The structure of this thesis report is as follows: Chapter-2 describes the theory for the algorithm development. The methods used are introduced in Chapter-3. Chapter-4 contains all the essential results and a general discussion related to the work and conclusions including possible future studies in Chapter-5.

2
Theory

It is important to assess various ways to approach the driving cycle construction and ways to process the available data. Hence, in this section, a brief overview is provided of theoretical knowledge regarding different forms of approach to construct a driving cycle, and techniques of data processing .

2.1 Construction models

The development or construction of the driving cycle is very important to determine emission levels of vehicles under real-traffic driving conditions. A driving cycle describes the change in the speed of a vehicle throughout the driving. The construction methods are classified based on the approach to develop the driving cycle.

There are few extensively used methods[16] in the development of driving cycle:

- Micro-Trip based construction.
- Segment based cycle construction.
- Modal based cycle construction.
- Pattern classification cycle construction.

2.1.1 Micro-Trip based construction model

The common approach in cycle construction is Micro-trip based cycle construction[16]. It involves selecting several micro-trips, which yields in better classification of driving patterns, bounded by a start and stop[17]. In this method of construction, a set of micro-trips from the real driving data, which can represent the driving pattern closely, is selected[18]. This method is generally based on specific speed, acceleration, and duration of constraints. The method involves dividing the trips based on the trip characteristics and assigning them into several bins[19]. This method is highly suitable for its 'Stop-Go' situation, to evaluate the emission and fuel consumption under traffic conditions. The selection of several trips for a driving cycle is supposed to meet the required target parameters. It involves filtering the trips based on the target parameter as a constraint with the least possible tolerance. The reason being the analogy - "lower the tolerance - higher the representativeness of trip". In a study from Gangamuwa et al.,[19], there are several ways to select the micro-trips; quasi-random method, random selection, incremental method, statistical methods like Fourier series, time series analysis, polynomial curve fitting technique.

2.1.2 Segment based cycle construction model

Segment-based cycle construction is based on the specific type of roadway, several stops, traffic conditions to represent the real traffic conditions, and physical characteristics of the road[19]. The method involves dividing the trips according to the condition of traffic and physical characteristics based on the target parameters and the mode of the trip can start with any speed and end with any speed. This method of construction is highly recommended for the construction of a driving cycle to a particular type of road due to the fewer number of stops. For better representativeness, it requires identification of various road categories like highly congested, residential, highway situations based on their average speeds which increases the difficulties to match the level of acceleration and speed of various consecutive trips while chaining them together due to stratified data[20]. Further, this method is much suitable to develop driving cycles for expressways. It lacks in adjacent starts and stops. Hence, it is not suitable to measure the emissions level[19].

2.1.3 Modal based cycle construction model

Modal-based cycle construction is based on a specific frequency of driving. It involves the process of dividing the pattern of driving into several dynamic patterns of acceleration, deceleration, cruising, and idling components[19]. The generated snippets of patterns using the Markov process are selected by assuming the maximum likelihood through means of clustering a particular event of the modal pattern. The selected trips are chained to form a driving cycle through a transition matrix based on the probabilities of successive modal events. The generation of the driving cycle requires a higher probability of several modal events. Since this method is required for a larger number of data, it is highly suitable for the regional data population[18].

2.1.4 Pattern classification cycle construction model

This model of cycle construction is focused on the kinematic sequence of the trip. The group of trips is divided into several classes by a statistical approach[19]. The kinematic sequence is selected based on maximum likelihood estimation based on succession probability. This form of approach is highly statistical and European driving cycles are constructed based on this form of approach. The selected kinematic sequences are connected to form a certain driving cycle. The form of the driving cycle entirely depends on the form of the selected kinematic sequence. There are certain drawbacks to this approach. It requires more information to classify and divide the kinematic sequences and is a time consuming approach.

2.2 Data processing

A fundamental base for any form of data analytical process is to consider the quality of the primary data. There are several ways to process the data using algorithms based on the availability of classifiers and predictor; ways which can be grouped into Supervised learning algorithm and a Unsupervised learning algorithm. A supervised learning algorithm mainly requires certain classifiers for grouping the data. Regression analysis and Naive Bayes are some of the algorithms helpful under Supervision learning. On the other hand, Unsupervised learning algorithms do not need pre-determined classifiers to group the data[21]. The grouping of data is based on the similarities between the data. K-means, Spectral and Hierarchical clustering algorithms are some of the prominent Unsupervised learning algorithms making it possible to distinguish the groups clearly.

2.2.1 Supervised learning algorithms

Data processing by the support of certain classifiers is termed as Supervised learning algorithms. The most abundantly used method is Regression analysis[21]. It is a classical approach for variables possessing linear variability. Depending on the number of variables, the method is sub-classified as Simple and Multiple regression methods. A linear function is modeled by the aid of a dependent and independent set of variables. Considering regression coefficients (r) and an error parameter((ϵ), the general form of regression model is written as:

$$y = r_0 + r_1 x_1 + r_2 x_2 + ... + r_n x_n + \epsilon = (x_i^T r + \epsilon) \tag{2.1}$$

To obtain a full-rank regression model, certain approximation methods have to be considered and are much necessary for estimating the regression coefficients. The methods of approximations include Ordinary Least Squares [OLS], Generalized Least Squares [GLS], Recursive Least Squares [RLS]. The method used to determine the regression coefficients are different, still, the results appear to be the same. The minimization approach to minimize the sum of squares of variable differences using a cost function of order 2 is utilized in the OLS model. While the GLS model tends to reduce the covariance between the error residuals. The RLS model follows the same strategy as of OLS model with an addition of an assumption variable. The models for the variable matrix (X) and co-variance (C) are written as:

OLS Model:

$$E(r) = \sum_{i=1}^{n} (y_i - x_i^T r)^2 \tag{2.2}$$

GLS Model:

$$E(r) = (X^T C^{-1} X)^{-1} X^T C^{-1} y \tag{2.3}$$

RLS Model:

$$E(r) + \lambda_r = \frac{(y - Xr)^2}{n} + \lambda_r, \lambda > 0 \tag{2.4}$$

Another method to classify the dependant variable directly is the Logistic Regression Method. This method is advantageous for a known categorical independent variable. It is also possible to use logistic regression for the classification of multiple variables. Logistic regression is based on the estimation of logarithmic odd values. In the below equation, 'x' represents the dependent variables, 'p' is the probability of the dependent variable.

$$ln(\frac{p}{1-p}) = w_0 + w_1x_1 + w_2x_2 + ... + w_nx_n \tag{2.5}$$

2.2.2 Unsupervised learning algorithms

The method of Unsupervised learning is applicable during the absence of predetermined classifiers[21]. Unsupervised learning is focused on grouping the data according to the available input features and similarities. It is called as Clustering technique. The most performed methods are discussed below.

K-means clustering technique is the most popular method and groups the available data into so called k-groups[22]. A highly efficient approach to make clusters of a large volume of data is based on its kinematic segments[23]. It tries to group the data to have large variation between the clustered groups by assigning random data points, where the data sets are clustered based on the similarity of the classifying feature[24]. This is made possible by using a k-means minimizing function (C_n) which reduces the variance between the clusters.

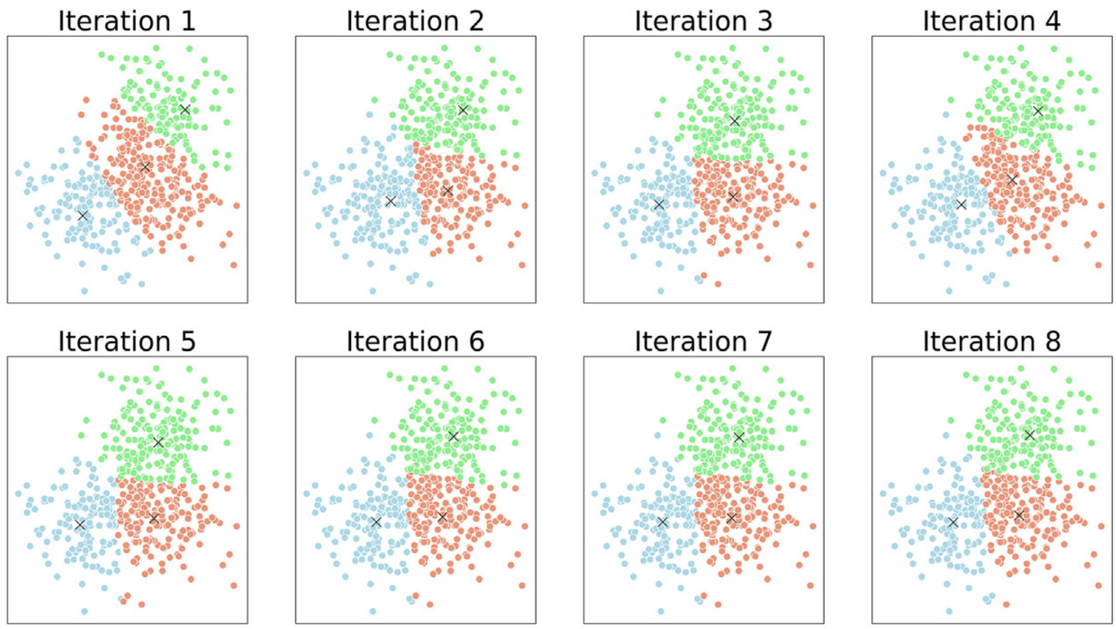


Figure 2.1: *The k-means clustering with 3 clusters*

$$C_n = \sum_{i=1}^{k} min(|| C_j - \mu_i ||^2) \tag{2.6}$$

$$C_n = C_1, C_2, C_3, ..., C_k \quad \forall \quad k \leq n \tag{2.7}$$

A general procedure to perform k-means clustering is as follows:

- Select total number of clusters (k).
- Set the total number of iterations(i).
- Compute Euclidean distance and centroid (C_j) for each cluster (k).
- Initiate the minimization function (C_n).
- Repeat the process to attain convergence or reach maximum iterations.

Spectral clustering is an indirect mode of grouping by projecting the data in various spatial dimensions. Spectral clustering uses the concept of Graph theory and minimizes the similarities between the clusters. It is done by determining the affinity matrix using the Euclidean distance of the data and converting the affinity matrix to the Laplace matrix. To keep the process simple and easy, eigenvalues of the Laplace matrix are determined to reduce the spatial dimension. The obtained eigenvalues are clustered into several clusters. The feature of the Laplace matrix eigenvalues tends to be smooth at highly dense areas. This makes it easy to recognize and group the data. Considering, A_{ij} as the affinity matrix, the diagonal matrix of same order, the Laplace matrix L, can be computed as:

$$A_{ij} = \sum_{i=1}^{n} \sum_{j=1}^{n} || x_i - x_j ||^2) \tag{2.8}$$

$$L = D - W \tag{2.9}$$

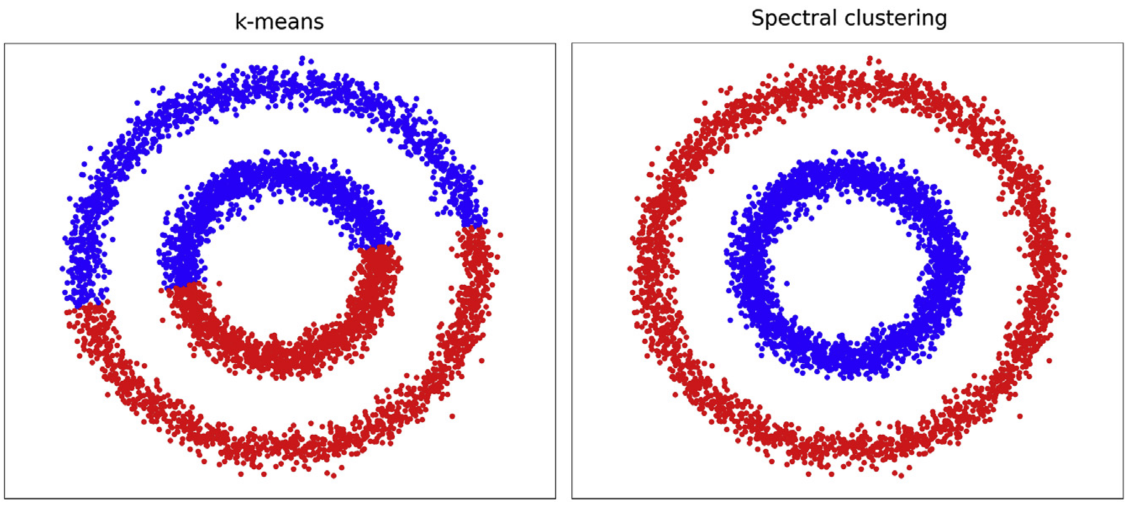


Figure 2.2: *Cluster methods*

2.3 Performance Evaluation

One of the most important aspects after processing the data is to determine the performance aspect of the applied method. When following the strategy of Supervised learning, the performance is evaluated through the method of Cross-validation and Train/Test split strategy. The performance of Unsupervised learning methods is dependant on the number of clusters, and hence, it is required to determine the most optimal number of clusters in an Unsupervised learning strategy[21]. Harabasz(CH) index, Silhouette coefficient, and Davies - Bouldin (DB) index are the most effective methods to determine the optimal number of clusters.

2.3.1 Supervised learning

The Train/Test split strategy is the simplest performance evaluation method for Supervised learning. In this method, the provided data is partitioned rationally into the training set and testing set. The Cross-validation methods are of different types, such as, k-fold cross-validation, repeated k-fold cross-validation, leave-one-out cross-validation, stratified k-fold cross-validation, leave P-out cross-validation. They are similar to each other, though. The k-fold cross-validation process involves dividing the data-set equally and training the model by using k-1 folds, repeating the procedure until all the folds are tested and performance scores are evaluated on each fold. The Stratified k-fold cross-validation is a systematic variation of k-fold cross-validation. It involves maintaining the ratio between the target groups and the same process is followed as k-fold cross-validation. Similarly, repeating the k-fold validation process for pre-defined n-times yields the 'Repeated k-fold cross-validation' method.

Leave-one-out cross-validation involves the process of formulating subsets and testing them by dropping one sample at a time and repeating the process until all samples are tested. For example, if the data set has n-samples, the supervised model shall be trained for (n-1) samples. This makes it a complex approach of computation due to the occurrence of variance in every iteration due to varying subsets. Further, accuracy tends to be the performance factor in the above-said methods. Accuracy is directly related to the performance of the supervised model. Care should be taken since accuracy is not a strong classifier to differentiate between the methods.

2.3.2 Unsupervised learning

Estimating the optimal number of clusters is the main parameter in the performance evaluation of the unsupervised model. There are various methods to determine the optimal number of clusters based on the strategy, and the most popular methods are discussed in this section.

2.3.3 Silhouette coefficient

The Silhouette coefficient is the measure of similarity of observation in its cluster group in comparison with the other cluster group. It is represented as :

$$SI_c = \frac{D_c - d_i}{max(D_c\,, d_i)} \tag{2.10}$$

Here D_c is the average distance between objects in the set of the cluster (C). This is termed as 'Distance within the clusters'. Average distance of an object from the nearest cluster group d_i is termed as 'Distance between the clusters'. The performance is evaluated on the value of the Silhouette coefficient. The higher the Silhouette coefficient - the higher the number of distinguished cluster groups. The coefficient ranges between -1 to +1: +1 denotes well-separated cluster groups and -1 means the opposite.

2.3.4 Davies - Bouldin (DB) index

The coefficient of similarity, when measured as the average distance between the centroids (C_i) of clusters (k) is termed as Davies - Bouldin index, or in short, DB index. For a well-separated cluster set, the DB index lies close to 0, representing a greater distance between the cluster centroids. DB index is given as:

$$DB_c = \frac{1}{k}\sum_{i=1}^{k} max(C_i) \tag{2.11}$$

2.3.5 Calinski-Harabasz (CH) index

A method to evaluate the performance by considering 'within-the-cluster variance(WCV)' and 'between-the-cluster variance (BCV)' is Calinski-Harabasz index or CH index. Considering the clusters k, BCV symbolizes the size of the variation between the clusters. For n_i samples in a dataset with C_i cluster centroids and mean distance between the samples μ, BCV can be written as:

$$BCV(k) = \sum_{i=1}^{k} n_i \mid\mid C_i - \mu \mid\mid^2 \tag{2.12}$$

Similarly, considering the Euclidean distance between the data sample x and cluster centroid C_i, WCV, depicting the variation between the data samples in every cluster group, can be determined as.

$$WCV(k) = \sum_{i=1}^{k} \mid\mid x_i - C_i^2 \mid\mid \tag{2.13}$$

By definition of CH index, considering total n number of samples with k number of cluster groups, higher CH index represents well-separated cluster groups. It can be written as:

$$CH = \left(\frac{n-k}{k-1}\right)\frac{BCV(k)}{WCV(k)} \tag{2.14}$$

In brief, due to the possibility of processing the available data in numerous ways using an unsupervised model, it is always a difficult task to choose a procedure to process and evaluate the performance. Based on the study conducted by the authors[25], for a moderate set of clinical data to compare the clustering algorithms, it is said to be entirely dependent on the quantity of data set. Various algorithms exhibit several different properties, making it difficult to draw a fair conclusion, although the authors value using the k-means algorithm and evaluating the performance with CH index[26] [27].

2.4 Simulation model

In this work, QSS toolbox in the Simulink is used to create a simulation model of a vehicle system. The QSS toolbox is extremely helpful to determine the fuel consumption of vehicle powertrain. The toolbox consists of various masks or blocks, required for modeling in a quasi-static approach. In the quasi-static approach, forces are computed within the masks from the provided velocity and acceleration data. The toolbox consists of a library involving various blocks to model Electric vehicles, Hybrid vehicles, Conventional vehicles with various forms of powertrain systems.

The important elements of the QSS library are:

- Driving cycle
- Controller
- Vehicle
- Gear system
- Energy converter
- Energy buffer
- Energy source

These elements have multiple blocks and are used in simulation models based on the type of vehicle. Brief overviews about the blocks useful in this work are focused in this section.

2.4.1 Vehicle model

A simple conventional vehicle model system was created in Simulink using the QSS toolbox. The model is presented in fig:2.3. The model contains the Driving cycle, Vehicle model, Wheel model, Transmission system, IC Engine and Fuel source.

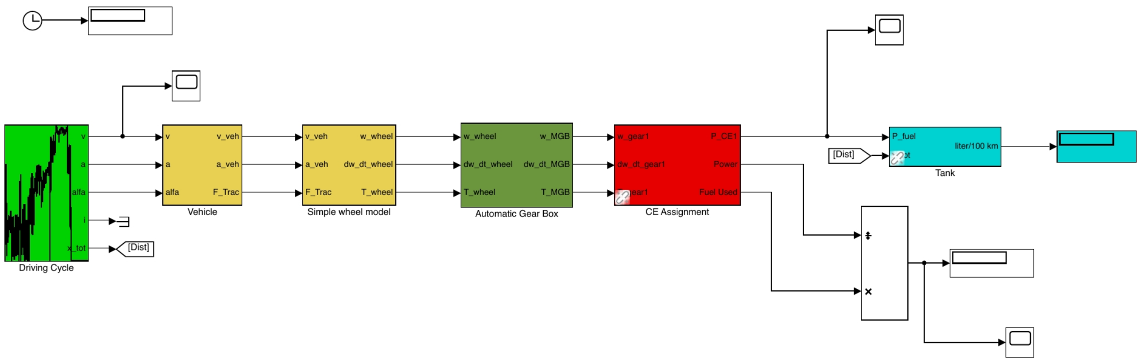


Figure 2.3: *Simulation model with IC engine in Simulink*

The QSS model uses a quasistatic approach. It requires input of velocity, acceleration and distance. The block or mask Driving cycle (fig:3.2) is an input section for the rest of the simulation model. The driving cycle should be present in the QSS toolbox database in-order to select the cycle for simulation.

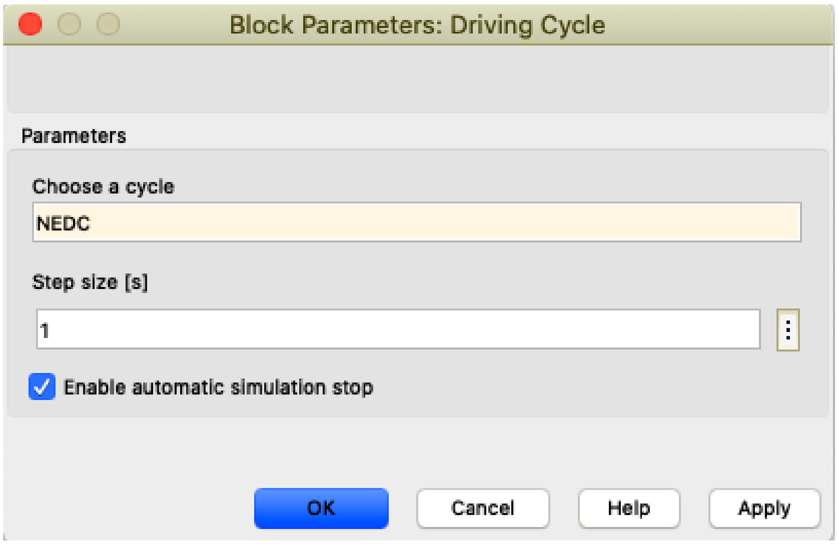


Figure 2.4: *Driving cycle block*

The vehicle block in the model requires the physical data of the vehicle. That includes weight, the frontal area of cross-section, drag and rolling resistance coefficients. Also, the diameter of the vehicle wheel is provided in the wheel block of the model. These parameters are essential for the model to compute the driving force required. The transmission is an essential part of the transfer of the power from engine to wheel. Hence, the model has to be provided with gear ratios of the powertrain in the transmission or Geax box block. The type of engine and the maximum power of the engine is provided in the Combustion engine block. In the final block, the type of fuel is provided to determine fuel consumption on simulation.

3
Methodology

This chapter focuses in detail the used methodology. It includes sections on the used real driving data, an approach using a construction model, various statistical computations to initiate data processing, D-optimal design approach, acquisition of the constructed driving cycle, and finally the simulation of the driving cycle in Simulink. Matlab is used for computation and writing the algorithm.

3.1 Real driving data

The used driving data set is obtained from *'The Swedish car movement data project'*[28]. The project involved gathering and analyze a large amount of data regarding the patterns of privately driven vehicles within Sweden. The data were collected using GPS equipment between June 2010 and Sept 2012. The cars were chosen by a random stratified selection from the Swedish vehicle registry. The data from the project is confined to the use of type-1 passenger cars of model year 2002 or younger, registered in Västra Götaland county and Kungsbacka municipality. The stratification was performed on the properties urban/rural, age, weight, fuel, private/company car.

3.1.1 Data logger and data handling

The data logger unit utilized a GPS logger combined with a GSM modem available from Host Mobility, which operates at 12VDC. The signals logged from this unit are:

- Timestamp (current and last valid)
- Position (latitude, longitude and altitude)
- Velocity (speed and direction)
- Used satellites (identity)
- Dilution of precision (pdop, hdop, vdop)
- Over-the-air-provision OTAP

The data has been divided into trips and stored together with statistics on trip and vehicle level. Filtered trip data in the form instant power at the wheels for a standardized vehicle derived for estimating the potential for brake energy regeneration [26] makes up the input of real driving data to this study.

3.2 Approach to construction model

There are multiple ways to construct a driving cycle from the available real-world driving data. As mentioned in section 2.1, the models are differentiated through the process of driving cycle construction using micro-trips, driving segments, modal frequency, and driving patterns. The real driving data collected at various locations contain dis-similar driving patterns and irregular driving routines. This makes it easier to choose a micro-trip-based model to construct the driving cycle.

3.2.1 Micro-trips segmentation

A trip or driving pattern containing one-start and one-stop makes is here defined as a micro-trip. The used real driving trips may contain long and continuous driving patterns with multiple starts and stop as shown in fig:3.1. The real driving trips are therfore segmented into numerous short-duration micro-trips involving one start and one stop, which contains phases of acceleration, deceleration, cruising, etc.

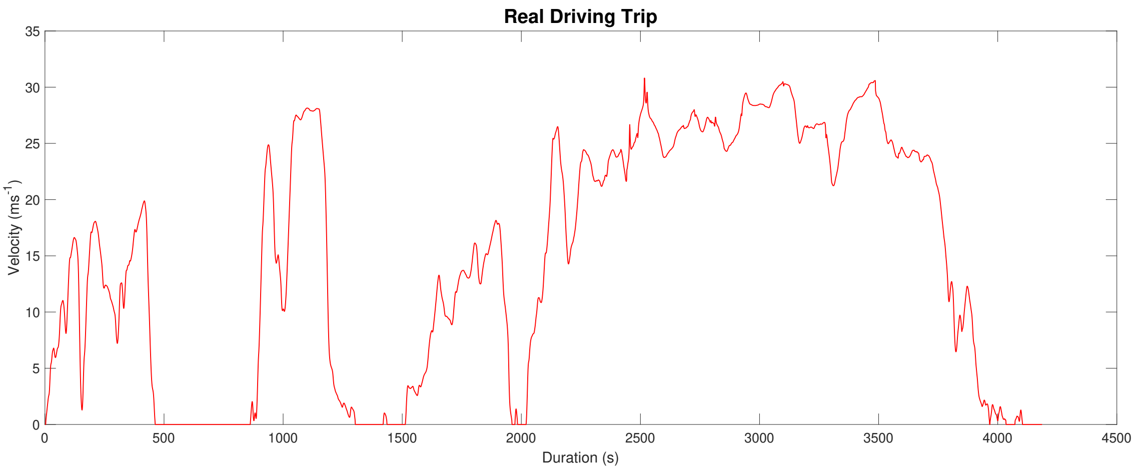


Figure 3.1: *Real driving trip*

3.2.2 Micro-trips calibration

The vehicles was logged with a sampling rate of 0.4 seconds. Some of the micro-trips are prone to errors[29], for instance, due to the start-up delays in the GPS system. Errors may include sudden rise or fall of vehicle velocity beyond the feasible limits. Such micro-trips are grouped and are interpolated linearly to fill the gap in the micro-trips data. This is much necessary to avoid discarding several trips that would contain important characteristic features of real driving. It is highly important to make it possible for testing in a test rig or virtual simulation.

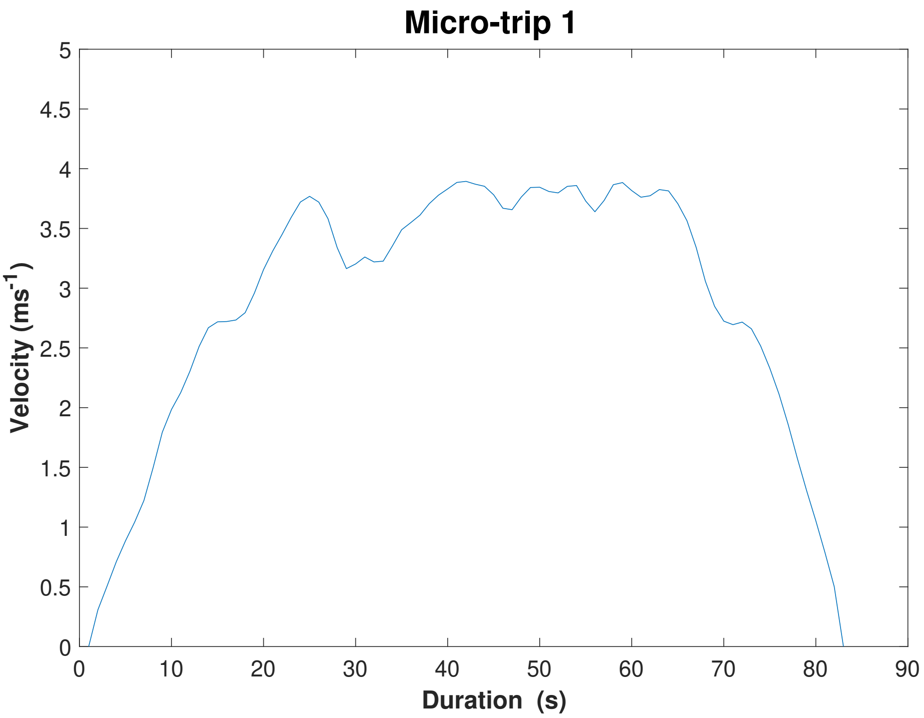


Figure 3.2: *Example of segmented micro-trip*

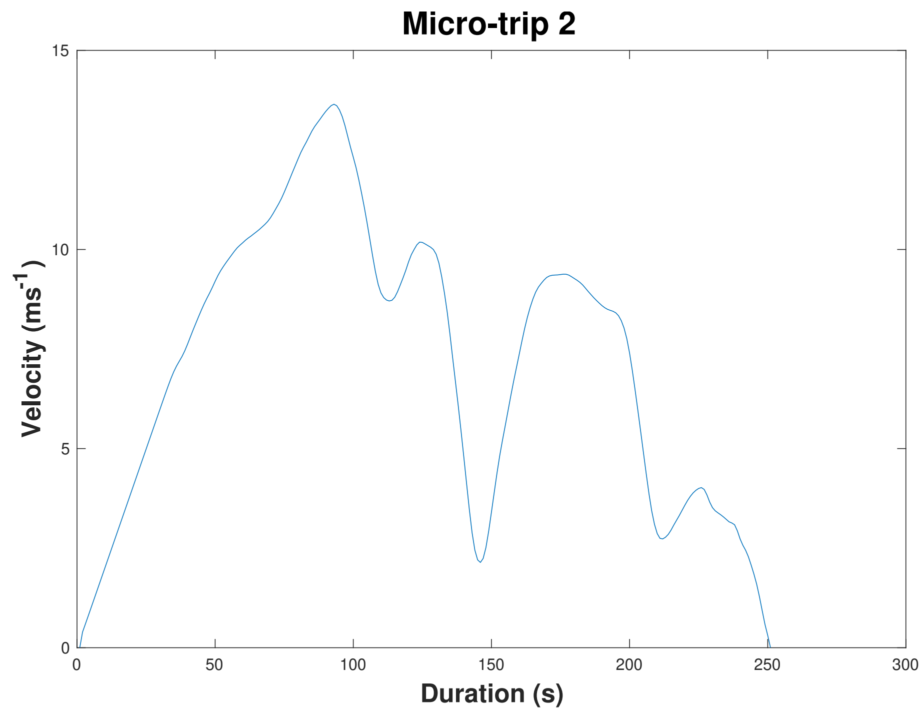


Figure 3.3: *Example of segmented micro-trip*

3.3 Data processing

The generated micro-trips are utilized to group them based on the similarities. Due to the absence of classifiers to group the data, an unsupervised learning algorithm method is applied using the *k-means* clustering technique. However, it is important to compute vital necessary trip parameters prescribed by the European Union on the test procedure for Real Driving Emissions (RDE) for light commercial vehicles in Europe[1].

3.3.1 Trip parameters

Here it is focused on few vital parameters like trip distance, the average speed of trip, and 95^{th} percentile of VA_{pos}. VA_{pos}, which is the product of velocity and positive acceleration. It describes the aggressiveness of the trip. These trip parameters are computed for all the micro-trips and form the input features for the clustering procedure. The available data is vast. It includes very long trips (fig:7.3 and 7.4) and very short trips. Long trips greater than 3000 seconds and short trips less than 500 meters of distance traveled are excluded for further process.

3.3.2 k-means clustering

k-means clustering is the most popular method of grouping the data based on their similarities and the one used here. The procedure of *k-means* clustering is followed as explained in section 2.2.2. The computed trip parameters: trip distance, the average speed of trip, and 95^{th} percentile of VA_{pos}, are the input features for clustering.

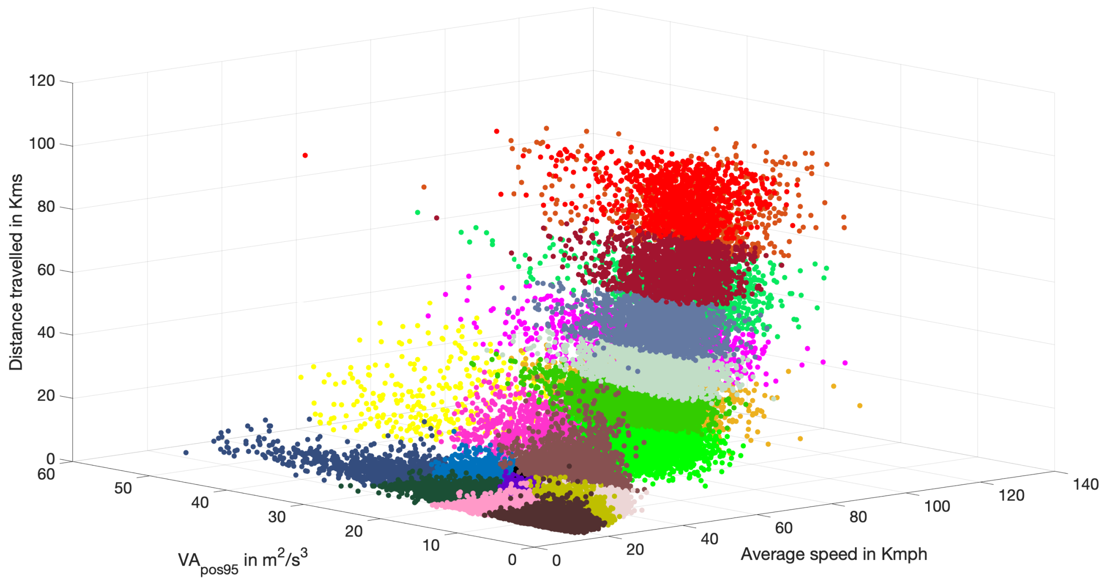


Figure 3.4: *k-means clustering with 22 clusters*

A 3-dimensional *k-means* clustering is deployed in MatLab. Due to the vast set of data, the iteration is limited to 10. The clusters are increased from few to many to determine an optimal number of clusters k. The *k-means* clustering data of various clusters are further shown in the appendix.

3.3.3 Cluster performance evaluation

The performance of the clusters is evaluated based on CH Index. CH index is obtained by determining BCV and WCV using the formulae 2.12 and 2.13. TThe value of BCV tends to increase as the number of clusters increase. Similarly, on increasing the number of clusters, the value of WCV decreases. Fig: 7.5 in the appendix, shows the variations of BCV and WCV for various clusters. From the value of BCV and WCV for every cluster k, the CH index is estimated using the formula 2.14. The cluster with a high CH index tends to be the best cluster with good data separation. Variation in CH index for various clusters is shown in fig: 3.5.

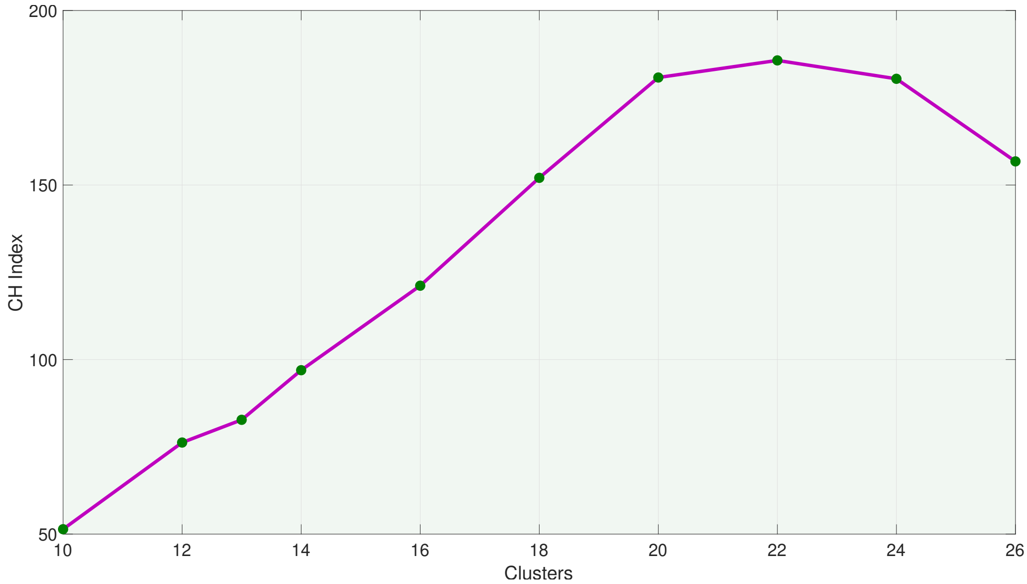


Figure 3.5: *Variation of CH index for various clusters*

3.3.4 Cluster grouping

From the evaluation of cluster performance, the cluster with the best CH index is selected. To develop a Driving cycle from Real-world driving data, it is necessary to consider the RDE requirements on the trip parameters for the test cycle. The trip parameters requirements are presented in the table: 3.1

Parameter	**Segment**	**Value**
Distance	Urban	>16 km
	Rural	>16 km
	Motorway	>16 km
Average Speed	Urban	15 - 40 $\mathrm{km\,h^{-1}}$
	Rural	60 - 90 $\mathrm{km\,h^{-1}}$
	Motorway	>100 $\mathrm{km\,h^{-1}}$
95^{th} VA_{pos}	Urban	<18.7 $\mathrm{m^2\,s^{-3}}$
	Rural	<24.3 $\mathrm{m^2\,s^{-3}}$
	Motorway	<26.6 $\mathrm{m^2\,s^{-3}}$

**Source : Test procedure for RDE by European Union[1]*

Table 3.1: *Trip Parameters for Test cycle*

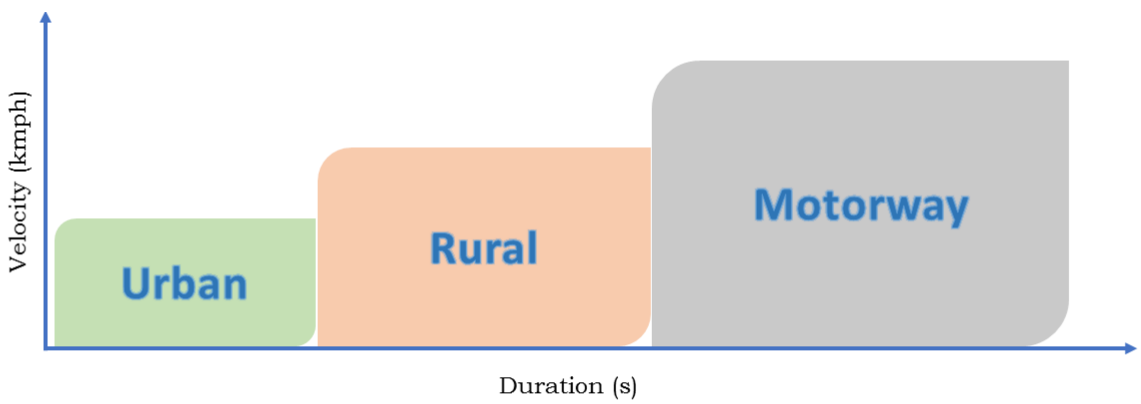


Figure 3.6: *Proposed driving cycle sequence*

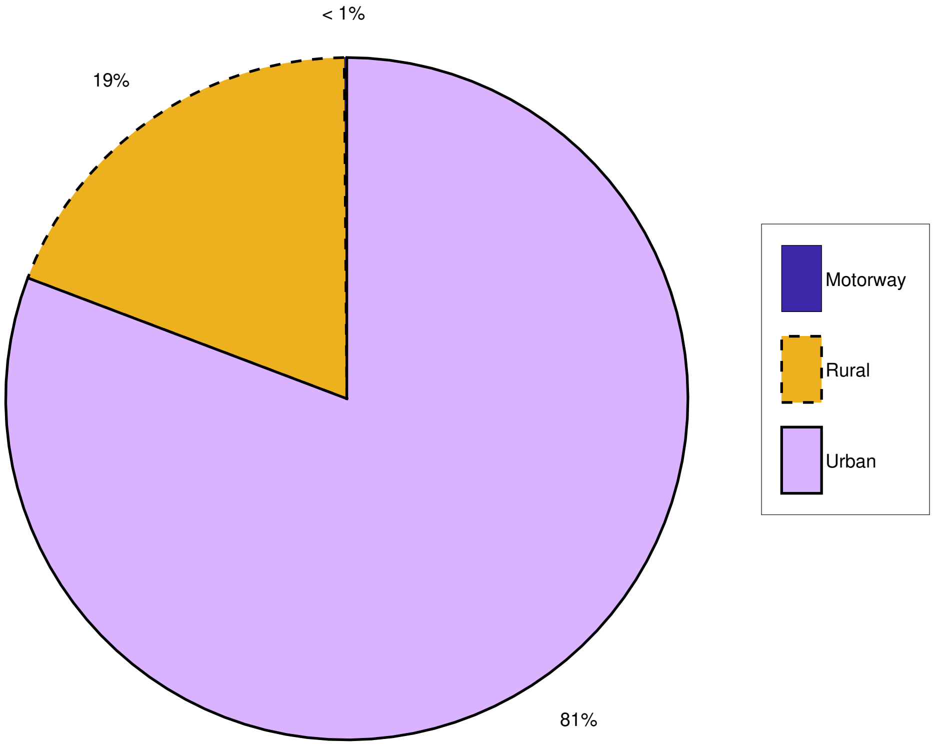


Figure 3.7: *Micro-trips in segments*

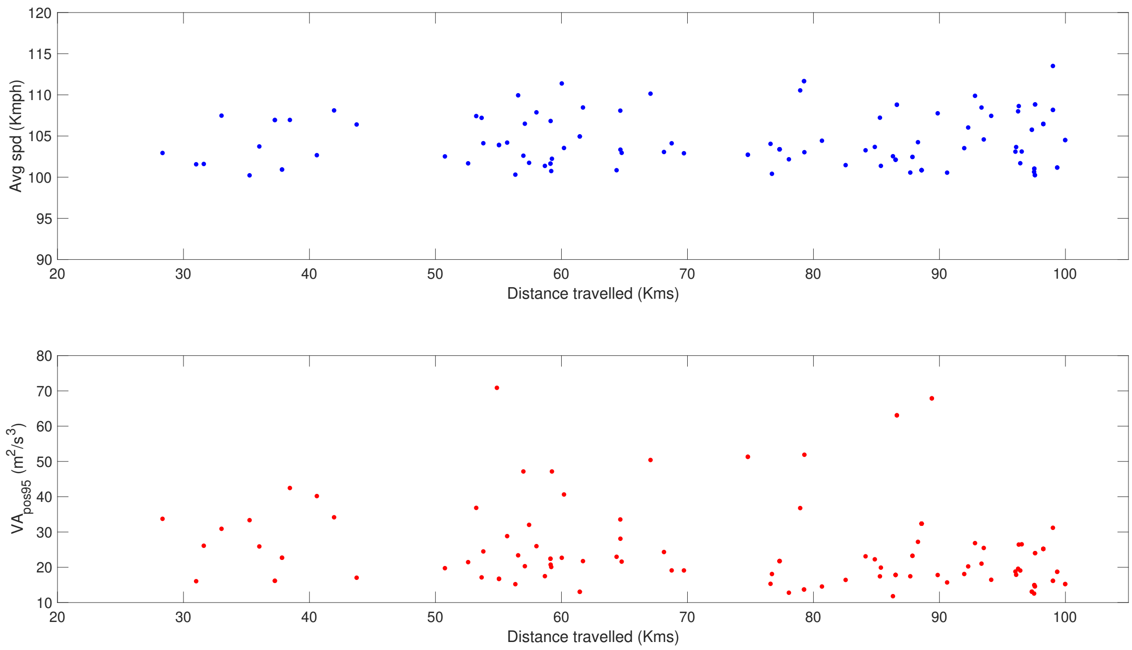


Figure 3.8: *Motorway group*

In the RDE test procedure, the test cycle should contain Urban, Rural, and Motorway segments (fig: 3.6). Based on the average speed of the micro-trips, the clustered data are grouped into such segments. fig: 3.7 shows the percentage of trips in each segment. For the motorway segment, variations between the micro-trips are presented in fig: 3.8. The variation graphs for the other two segments are depicted in fig: 7.6 and 7.7

3.4 Cycle development

This work aims at developing an algorithm, that could potentially create a driving cycle, according to the RDE test cycle conditions.Various conditions prescribed by the European Union for the driving cycle are presented in the table: 3.2. A vast amount of micro-trips are present in the Urban segment (fig:3.7). It is therefore important to select those trips that are within the RDE test conditions.

Parameter	**Segment**	**Value**
Cycle duration	-	90 to 120 minutes
Distance	Urban	>16 km
	Rural	>16 km
	Motorway	>16 km
Average Speed	Urban	15 - 40 $\mathrm{km\,h^{-1}}$
	Rural	60 - 90 $\mathrm{km\,h^{-1}}$
	Motorway	>90 $\mathrm{km\,h^{-1}}$
95^{th} VA_{pos}	Urban	<18.7 $\mathrm{m^2\,s^{-3}}$
	Rural	<24.3 $\mathrm{m^2\,s^{-3}}$
	Motorway	<26.6 $\mathrm{m^2\,s^{-3}}$
RPA	Urban	>0.13 $\mathrm{m\,s^{-2}}$
	Rural	>0.06 $\mathrm{m\,s^{-2}}$
	Motorway	>0.03 $\mathrm{m\,s^{-2}}$
Maximum Speed	-	<160 $\mathrm{km\,h^{-1}}$
Stop Percentage	-	6 to 30% Urban duration

**Source : Test procedure for RDE by European Union[1]*

Table 3.2: *Conditions for Real Driving Emission Test cycle*

3.4.1 D-Optimal Design

A computer algorithm could be used to optimize and choose the data based on desired criteria. The D-Optimal design tends to reduce the parametric variations concerning desired parameters in the specified model. The D-Optimal design is carried out in Matlab using its inbuilt algorithm 'Candexch'- representing row exchange algorithm to determine an optimal design. A total of 3 design matrices comprising of Average speed, 95^{th} VA_{pos}, RPA, Maximum speed, are used. Distance are created from the grouped trips in Urban, Rural, and Motorway segments. It is essential to set constraints for optimization of data, hence, the design matrix is constrained by the parameters from table:3.2 specifically.

The created design matrix of each segment is filtered based on the design constraints and are updated. The figure 3.9 presents the total percentage of micro-trips in each segment after filtering the data. Then D-Optimal design is deployed on the design matrices.

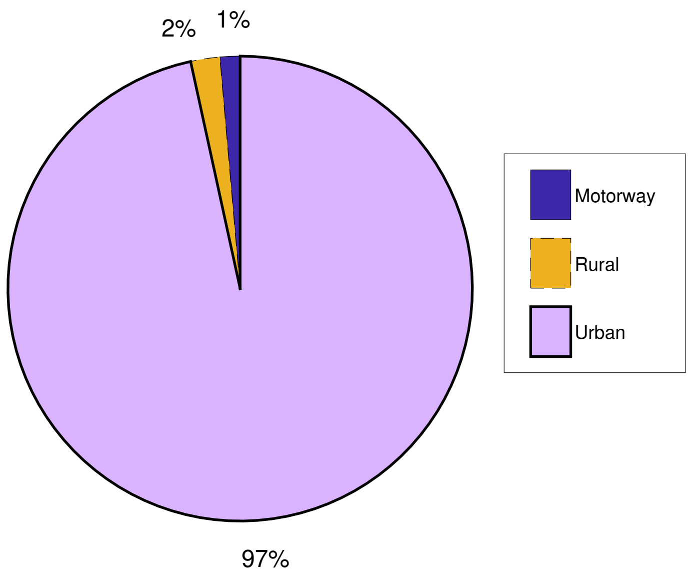


Figure 3.9: *Filtered micro-trips in segments*

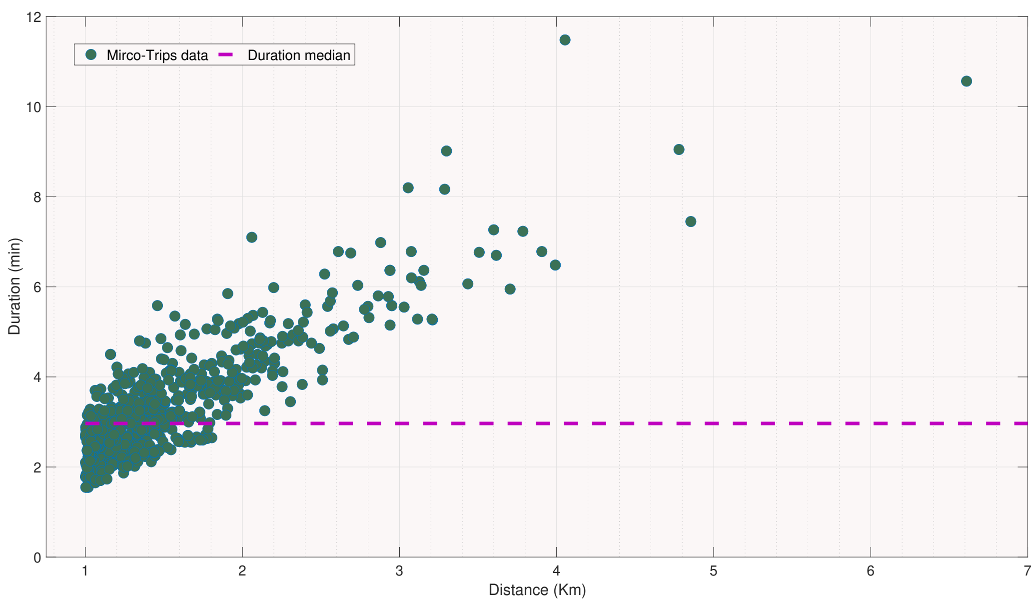


Figure 3.10: *Median duration of micro-trips in urban segment*

It is necessary to select and combine a number of micro-trips to create a driving cycle. The final driving cycle should have a total duration of between 90 to 120 minutes. Hence, the number of trips from each segment is determined based on the median micro-trips duration. From the fig:3.10 it can be deduced that the median duration of the micro-trips in the urban segment is roughly 3 minutes. From the fig:7.8 and fig:7.9, we get the median duration of approximately 31 and 35 minutes for the rural and motorway segments, respectively. Based on the duration constraint only one micro-trip can be chosen from the rural segment and motorway segment. A total of 7 trips is possible to select from the urban segment. Hence, the trips ratio is 7:1:1 for urban, rural, and motorway segments. The D-optimal design optimizes the design matrices of each segment and selects the mentioned number of micro-trips from each design matrix to create a driving cycle.

3.5 Driving cycle formation and Simulation

The optimized micro-trips obtained from the D-optimal design are joined together. The process of joining the micro-trips is based on the sequence illustrated in fig:3.6. A total of 10 driving cycles are developed from the process using the algorithm. Further, the thesis also strengthens the ability to simulate the derived driving cycle by using a simulation model. The developed driving cycles are used to obtain the required inputs for the simulation models in Simulink - velocity, time, acceleration. .

3.5.1 Simulation settings

The simulation model requires certain data regarding gear ratios and maximum power generated by the combustion engine at its rated rpm. For the simulation, the technical features of the Volvo XC40 are chosen. These features are listed in table:3.3.

Engine / **Parameter**	**Gasoline**	**Diesel**
Maximum Engine Speed (rpm)	6000	4500
Maximum Power Output (kW)	120	110
Rated Engine Speed (rpm)	5500	3750
First Gear	3.583	3.583
Second Gear	2.048	2.048
Third Gear	1.310	1.310
Fourth Gear	0.919	0.919
Fifth Gear	0.69	0.69
Sixth Gear	0.578	0.578
Reverse Gear	3.333	3.333
Final Drive	4.563	4.056
Payload (kg)	1900	1900

**Source : Volvo cars home page[30]*

Table 3.3: *Settings for Engine and Gear box block in simulation models*

Engine **Parameter**	**Gasoline**	**Diesel**
Area of cross-section (m^2)	2.56	2.56
Drag co-efficient	0.34	0.34
Rolling resistance co-efficient	0.01	0.01
Wheel diameter (m)	0.4572	0.4572

**Source : Volvo cars home page[30]*

Table 3.4: *Settings for Vehicle block in simulation models*

Figure 3.11: *Screenshot of gearbox block settings in simulation model*

The upshift and downshift speeds in the gearbox settings are calculated based on the engine rpm (fig:3.11) . The equation used to determine upshift speed is provided in eq:3.1. The calculation is based on the rated engine speed (N_e), wheel radius (r), gear ratios (i_g), and final drive ratio (i_f). The required driving cycle is selected in the driving cycle selection block to proceed with simulation. Data of several other settings are mentioned in table:3.4. A complete data of simulation settings are presented in appendix-2, for reference. Also, for the diesel engine model another set of gear ratios are used

$$V = \frac{N_e * 2\pi * r}{i_g * i_f} \tag{3.1}$$

3.5.2 Simulation Outputs

As explained in section:2.4 each driving cycle is selected manually from the library for simulation. The output from the simulation is noted manually for each driving cycle on each model separately. The fuel consumption and BSFC are used to determine the efficiency of the engine. Along with fuel consumption and BSFC, it is also possible to extract the information on engine operation points with BSFC iso-plots.

4
Results

The developed cycles are analyzed for the trip parameters and compared with RDE test cycle parameters. Those results are presented in this section, along with the outputs from the simulation model, when using the developed cycles.

4.1 Developed driving cycles

The developed driving cycles are derived by combination of optimized trips of urban, rural, and motorway sections, as discussed in the section:3.4. A total of 10 driving cycles are created from the real driving data.

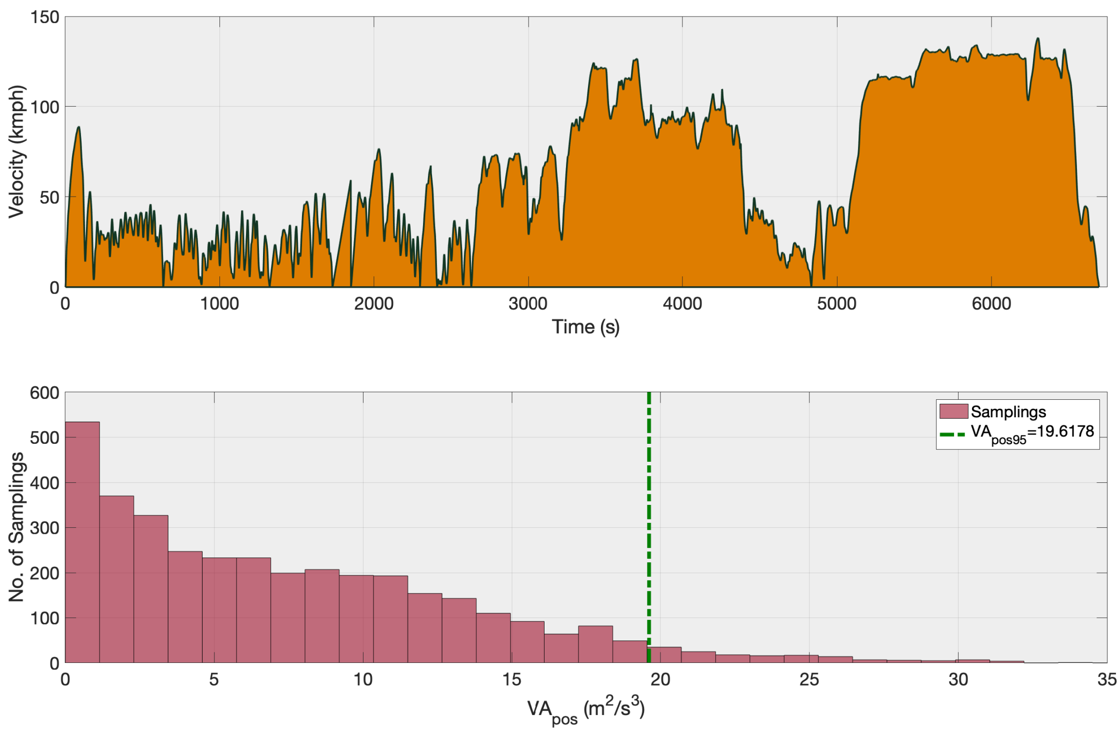


Figure 4.1: *Synthetic Driving Cycle - 1*

In figure:4.1, one of the developed cycles is presented. The combinations of micro-trips are optimized to possess the required cycle criteria of the RDE test cycle. The multiple starts and stops in the urban section tend to possess driving in city limits. The urban section emulates stops due to signals, pedestrians crossing, etc. The rapid acceleration and deceleration in urban sections are a result of several interferences in the path, such as lane additions, vehicle overtakes, etc. The rural and motorway sections possess a long driving path with less change in vehicle velocity. The combination of urban, rural and the motorway section resembles a real driving scenario of the vehicle starting from home - moving across residential road way and joining the motorway at the exit. It is important to compute the trip parameters and verify them. It is also important to understand the aggressiveness of the driving cycle. As per the RDE requirement, the cycle should not be too smooth. The second subplot in figure:4.1 represents the 95^{th} VA_{pos} of the cycle developed and presented in the first subplot.

The tables:4.1 and 4.2 provides the values of trip parameters computed from the 10 developed cycles. The tabular columns show the comparison with the values of RDE test cycle requirements. The trip parameters obtained by the developed cycles lie within the boundary conditions of RDE requirements, making it useful for testing and analysis of a vehicle's RDE emission. The aggressiveness value of the cycles in motorway segments tends to be lower than other segments. At high speeds, the change in velocity is lower than at other speeds.

Parameters	**Segment**	**RDE**	**Cycle-1**	**Cycle-2**	**Cycle-3**	**Cycle-4**	**Cycle-5**
Distance (km)	Urban	>16	21.89	17.04	21.89	20.91	21.89
	Rural	> 16	44.49	43.49	32.64	40.04	44.46
	Motorway	> 16	52.59	56.56	61.70	52.59	56.56
Average Speed ($km\,h^{-1}$)	Urban	15 - 40	30.09	28.23	30.09	29.38	30.09
	Rural	60 - 90	72.60	65.18	69.98	73.39	72.68
	Motorway	>100	101.68	109.9	108.46	101.68	109.9
95^{th} VA_{pos} ($m^2\,s^{-3}$)	Urban	< 18.7	15.89	16.37	15.89	16.11	15.89
	Rural	< 24.3	18.55	17.24	19.43	20.16	18.55
	Motorway	< 26.6	15.79	17.87	17.64	15.79	17.87
	Total	-	19.61	19.56	19.75	20.12	20.175
RPA ($m\,s^{-2}$)	Urban	> 0.13	0.144	0.146	0.144	0.14	0.144
	Rural	> 0.06	0.064	0.074	0.064	0.066	0.064
	Motorway	> 0.03	0.034	0.036	0.031	0.034	0.036
Max Speed ($km\,h^{-1}$)	-	<160	138.17	134.48	124.26	138.17	134.42
Stop Percentage %	-	6 - 30	18.32	22.08	18.32	19.92	18.32
Duration (min)	-	90 - 120	111.58	107.28	105.9	105.6	111.41

Table 4.1: *Parametric comparison of developed driving cycle with RDE - 1*

Parameters	Segment	RDE	Cycle-6	Cycle-7	Cycle-8	Cycle-9	Cycle-10
Distance (km)	Urban	> 16	21.82	21.89	18.57	20.91	18.57
	Rural	> 16	44.33	38.10	42.75	33.62	38.64
	Motorway	> 16	56.34	52.59	52.59	52.59	61.70
Average Speed ($\mathrm{km\,h^{-1}}$)	Urban	15 - 40	30.09	30.09	28.78	30.1	28.78
	Rural	60 - 90	78.77	75.78	75.55	67.35	71.12
	Motorway	>100	100.31	101.68	101.68	108.46	101.68
95^{th} VA_{pos} ($\mathrm{m^2\,s^{-3}}$)	Urban	< 18.7	15.87	15.87	15.91	16.117	15.91
	Rural	< 24.3	18.96	19.19	19.98	18.80	18.32
	Motorway	< 26.6	11.78	15.79	15.79	15.79	17.64
	Total	-	18.59	19.57	20.04	19.56	19.61
RPA ($\mathrm{m\,s^{-2}}$)	Urban	> 0.13	0.144	0.144	0.145	0.144	0.145
	Rural	> 0.06	0.061	0.064	0.065	0.066	0.066
	Motorway	> 0.03	0.03	0.034	0.034	0.031	0.034
Max Speed ($\mathrm{km\,h^{-1}}$)	-	<160	124.29	138.17	138.17	124.26	138.17
Stop Percentage %	-	6 - 30	18.32	18.32	20.66	19.19	20.62
Duration (min)	-	90 - 120	111.28	105.017	103.86	102.81	105.61

Table 4.2: *Parametric comparison of developed driving cycle with RDE - 2*

4.2 Simulation results

The developed cycles are used to generate required input data for simulation. A car model is simulated with two different combustion engines as mentioned in section:3.5.1. The total fuel consumption is determined, along with a BSFC plot. The BSFC contour plots in figure:4.2 and 4.3 represent gasoline engine and diesel engine, respectively. The BSFC plots depict the engine operating points needed for achieving the force required for propulsion. In turn, the operating points determine the fuel consumption of the vehicle to complete the driving cycle.

The fuel consumption and BSFC value of each cycle simulation for both the models are tabulated in table:4.3. The simulation model is optimized with up-shifts and down-shift speed limits for one driving cycle. The strategy of this optimization is to run the engine at points facilitating low fuel consumption, in other words, trying to run the engine at its 'sweet spot'. Varying the up and down-shift speeds will bring a change in the engine operating point and results in different fuel consumption. Hence, the speeds are (manually) optimized so as to obtain the required maximum torque in the cycle.

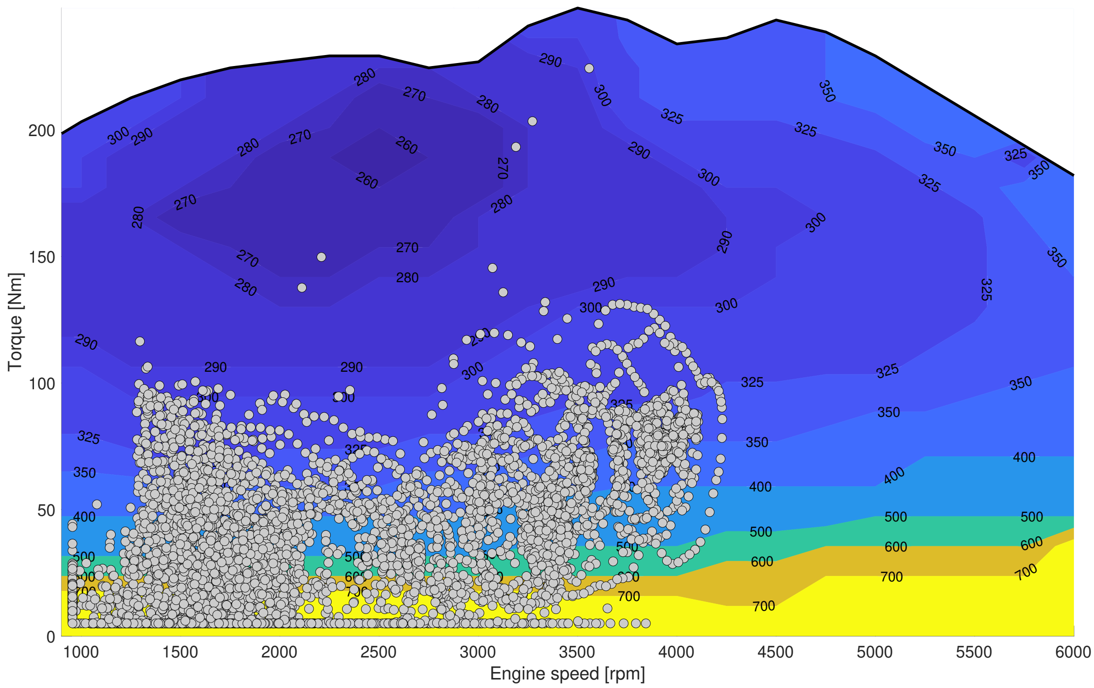


Figure 4.2: *BSFC plot of Gasoline engine for Cycle - 1*

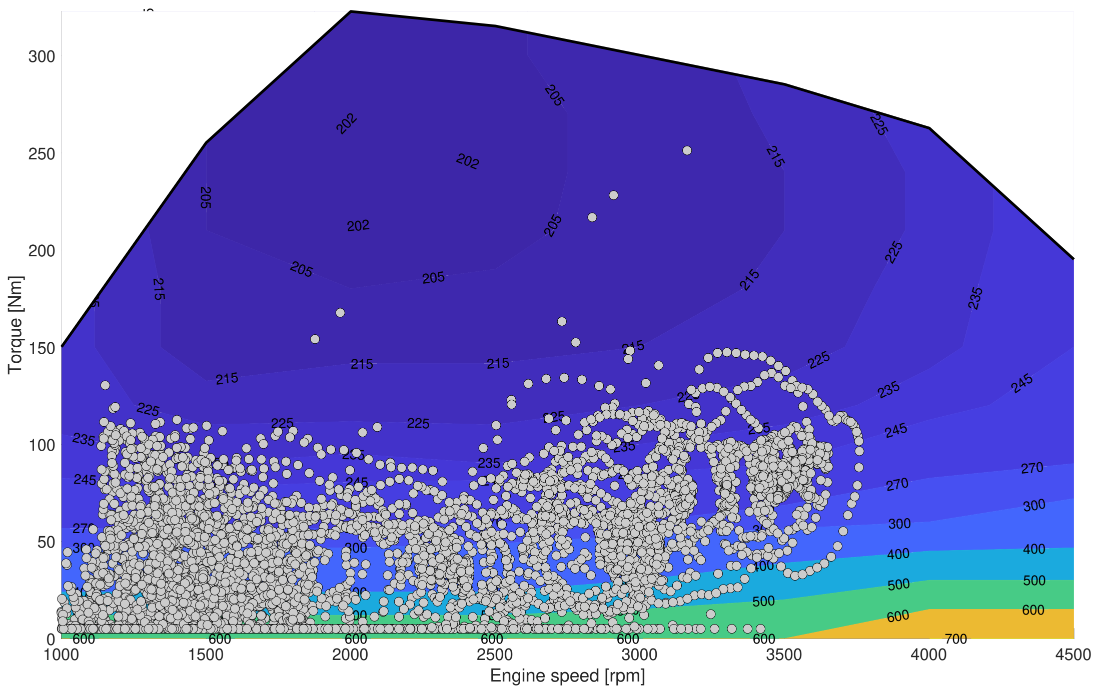


Figure 4.3: *BSFC plot of Diesel engine for Cycle - 1*

Cycle	**Gasoline**		**Diesel**	
	Fuel Consumption (ltrs/100km)	BSFC ($\mathrm{g\,kW^{-1}\,h}$)	Fuel Consumption (ltrs/100km)	BSFC ($\mathrm{g\,kW^{-1}\,h}$)
Cycle-1	10.79	259.4	6.612	192.2
Cycle-2	10.49	248	6.395	182.9
Cycle-3	10.38	243.1	6.311	17.9
Cycle-4	10.74	245.9	6.583	182.5
Cycle-5	10.78	266.9	6.567	196.9
Cycle-6	10.51	258.7	6.39	190.8
Cycle-7	10.66	242.8	6.581	181.1
Cycle-8	10.98	251.6	6.749	187.4
Cycle-9	10.54	228.6	6.452	169.1
Cycle-10	10.43	249.5	6.353	184.2

Table 4.3: BSFC and fuel consumption from simulation

4.3 Regression analysis

The relation between the mean VA_{pos} and specific fuel consumption have been determined using a 1-degree linear regression. The figure:4.4 depicts the relation between VA_{pos} and fuel consumption. The linear regression coefficients for Mean-VA_{pos} are tabulated in table:4.4 for the equation:

$$Y = A * X + B \tag{4.1}$$

where Y is the Fuel consumption, and X is the mean VA_{pos}.

Parameter	**Gasoline**	**Diesel**
A	0.307	0.215
B	8.2733	4.84

Table 4.4: *Regression co-efficients*

The fuel consumption is directly dependent on the VA_{pos} or aggressiveness: Higher aggressiveness, leads to higher fuel consumption.

For comparison, the figures:7.40 and 7.41 show the relation of fuel consumption with $95^{th}\mathrm{VA}_{pos}$, but the fit of this regression was worse. Further, linear regression was carried out for RPA and the Mean velocity of each cycle. The correlations of RPA to Mean velocity for various sections are shown in figures: 7.42, 7.43, 7.44, 7.45.)

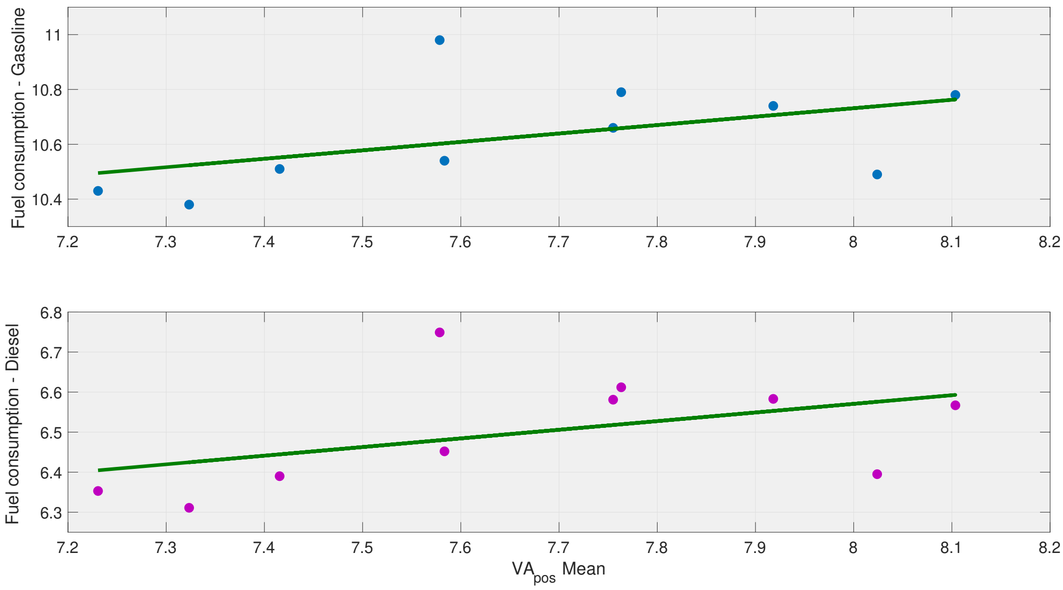


Figure 4.4: *Fuel consumption dependency on Mean-VA_{pos}*

4.4 Discussions

The obtained driving cycles from the algorithm show favorable values when compared with the RDE test cycle requirements. The tables:4.1 and 4.2 show the trip parameters in comparison with RDE requirements. The trip distance segments are unevenly distributed due to the absence of short trips in the used data. From the figure:4.1, the value of 95^{th} VA_{pos} reduces with an increase in velocity of the vehicle. As the velocity of the vehicle increases, the changes in velocity tends to decrease. Due to the reduction in change of velocity, the overall value of 95^{th} VA_{pos} starts to decrease.

The results from the simulation for fuel consumption tabulated in table:4.3, shows a large deviation between gasoline and diesel engine. This is because the engine lacks calibration to operate at sweet spot, resulting in higher BSFC for a gasoline engine than that of diesel engine. Calibration of the up-shift and down-shift speed required for gear transmission will result in lower fuel consumption. The engine used for simulation has different power for gasoline and diesel, since, original data of the vehicle is used in simulation settings. Also, these simulations helps to understand the variation in fuel consumption based on the driving cycle behaviour. Fuel consumption tend to increase with mean-VA_{pos}. Trips with higher aggressiveness tend to have higher fuel consumption than the trips with low aggressiveness.

5 Conclusion

Real-world driving data obtained from 378 Swedish cars were used. The data comprised filtered velocity data derived from ´GPS measurement. An algorithm was written to generate RDE driving cycles from the data.

The micro-trip-based construction model is found to be effective to analyse the variation of aggressiveness between cycles. The calibration of data in micro-trips to remove the GPS errors uses the linear interpolation method. This makes it too passive for and far from reality. The use of the unsupervised model in this thesis is found to be highly efficient. It helped in grouping the data based on the trip parameters to comply with RDE test procedures. The CH-index method to determine the best number of clusters resulted in grouping the trips to yield better between the cluster variation and with-in-cluster variation. The number of clusters found to be optimal to classify the data groups based on their parameters.

The importance to filter the data based on the RDE requirements is facilitated by D-optimal design. It includes the filtration of data and the creation of a design matrix for the filtered data. The D-optimal design led to maximization of the design matrix to select the best possible trip relating closely to the required trip parameters in the RDE test procedure. The number of micro-trips for each segment of the driving cycle is determined to have a total trip duration within the guidelines. Hence, a ratio of 7:1:1 between Urban, rural, and Motorway micro-trips is chosen. The selected micro-trips are joined to have a complete driving cycle from the real driving data.

The simulation model developed in Simulink represents normal IC engines combined with a manual gearbox. The driving cycles generated are simulated using the model to determine the specific fuel consumption. This results in a considerably higher fuel consumption in the gasoline engine than in the diesel engine. This is mainly due to the engine operation points, though. Hence they should be properly calibrated as much as close to reality to obtain better results in this part.

The results from a regression analysis focus on the relationship between the mean value of VA_{pos} and fuel consumption. The fuel consumption increases with the mean VA_{pos}. Due to the lack of well-spread data concerned to 95^{th} VA_{pos}, it is a bit hard to determine its relation with fuel consumption.

The thesis is aimed at generating an algorithm to create a driving cycle from the real driving data, including calibration, selection, and joining of trips. It begins with data selection, trip segmentation, clustering, selection of best cluster, grouping the clustered data in segments of urban, rural, motorway sections. D-optimal design for each section is deployed and the algorithm completes with the generation of the final driving cycle.

An analysis of the driving cycle is done separately based on the focus of the analysis aspect. Hence, for future studies, it will be interesting to use a Supervised learning algorithm to generate a driving cycle and visualize the changes in the developed driving cycle. Using a more advanced model for simulation would be much more helpful to determine highly satisfactory results in terms of fuel consumption.

Bibliography

[1] *Real-driving emissions test procedure for exhaust gas pollutant emissions of cars and light commercial vehicles in Europe.* ICCT, 2017.

[2] Claßen, Johannes, Stefan Pischinger, Sascha Krysmon, Stefan Sterlepper, Frank Dorscheidt, Matthieu Doucet, Christoph Reuber, Michael Görgen, Johannes Scharf, Martin Nijs, and Silja Thewes: *Statistically supported real driving emission calibration: Using cycle generation to provide vehicle-specific and statistically representative test scenarios for Euro 7.* International Journal of Engine Research, 21:146808742093522, July 2020.

[3] OICA: *Organisation Internationale des Constructeurs d'Automobiles.* Technical report.

[4] Esteves-Booth, A, T. Muneer, J Kubie, and H Kirby: *A review of vehicular emission models and driving cycles.* Proceedings of The Institution of Mechanical Engineers Part C-journal of Mechanical Engineering Science - PROC INST MECH ENG C-J MECH E, 216:777–797, August 2002.

[5] Hooftman, Nils, Maarten Messagie, Joeri Van Mierlo, and Thierry Coosemans: *A review of the European passenger car regulations – Real driving emissions vs local air quality.* Renewable and Sustainable Energy Reviews, 86:1–21, 2018, ISSN 1364-0321.

[6] Chen, Guisheng, Lei Di, Yinggang Shen, Wei Zhang, and Bin Mao: *Strategies for emissions control in heavy-duty diesel engines to achieve low-emissions combustion with a high efficiency.* Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 230(5):593–608, 2016.

[7] Guardiola, Carlos, Benjamín Pla, Pau Bares, Edward Chappell, and Richard Burke: *Improving CO2 emission assessment of diesel-based powertrains in dynamic driving cycles by data fusion techniques.* Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 235(2-3):362–372, 2021.

[8] Luján, José Manuel, Carlos Guardiola, Benjamín Pla, and Varun Pandey: *Impact of driving dynamics in RDE test on NOx emissions dispersion.* Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 234(6):1770–1778, 2020.

[9] Grüner, Jan and Stefanie Marker: *A Tool for Generating Individual Driving Cycles - IDCB.* SAE International Journal of Commercial Vehicles, 9, September 2016.

[10] *New European Driving Cycle (NEDC)* . DieselNet, 2013.

[11] *Worldwide Harmonized Light Vehicles Test Cycle (WLTC).* DieselNet, 2018.

[12] Chen, Yuche and Jens Borken-Kleefeld: *Real-driving emissions from cars and light commercial vehicles – Results from 13 years remote sensing at Zurich/CH.* Atmospheric Environment, 88:157–164, 2014, ISSN 1352-2310.

[13] Huertas, José, Michael Giraldo, Luis Quirama, and Jenny Díaz-Ramírez: *Driving Cycles Based on Fuel Consumption.* Energies, 11:3064, November 2018.

[14] Anida, I.N., Idrus Salimi Ismail, JS Norbakyah, W.H Atiq, and Salisa Abdul Rahman: *Characterisation and development of driving cycle for work route in Kuala Terengganu.* International Journal of Automotive and Mechanical Engineering, 14:4508–4517, September 2017.

[15] Evangelos Tzirakis, Alexios Kyriakidis and Fanourios Zannikos: *Methodologies for driving cycle development, using on-Road data from Athens.* Transport Research Arena Europe - Conference, 2008.

[16] Xiao, Zhang, Zhao Dui-Jia, and Shen Jun-Min: *A Synthesis of Methodologies and Practices for Developing Driving Cycles.* Energy Procedia, 16:1868–1873, 2012, ISSN 1876-6102. 2012 International Conference on Future Energy, Environment, and Materials.

[17] Anida, I. and Salisa Abdul Rahman: *Driving cycle development for Kuala Terengganu city using k-means method.* International Journal of Electrical and Computer Engineering (IJECE), 9:1780, June 2019.

[18] Czégé, Levente, Attila Vámosi, and Imre Kocsis: *Review on Construction Procedures of Driving Cycles.* International Journal of Engineering and Management Sciences, 5:266–285, April 2020.

[19] Uditha Galgamuwa, Loshaka Perera, Saman Bandara: *Developing a General Methodology for Driving Cycle Construction: Comparison of Various Established Driving Cycles in the World to Propose a General Approach.* Journal of Transportation Technologies, 5, 2015.

[20] Dai, Zhen, Deb Niemeier, and Douglas Eisinger: *Driving cycles: a new cycle-building method that better represents real-world emissions.* January 2008.

[21] *Chapter 7 - Machine learning and data analytics.* In Pezoulas, Vasileios C., Themis P. Exarchos, and Dimitrios I. Fotiadis (editors): *Medical Data Sharing, Harmonization and Analytics*, pages 227–309. Academic Press, 2020, ISBN 978-0-12-816507-2.

[22] Nanjundan, Sukavanan, Shreeviknesh Sankaran, C. R. Arjun, and G. Paavai Anand: *Identifying the number of clusters for K-Means: A hypersphere density based approach.* CoRR, abs/1912.00643, 2019.

[23] Zou, Sijia, Xinru Zhu, Linjun Xu, and He Zhu: *Construction of vehicle driving cycle in Fuzhou.* Journal of Physics: Conference Series, 1453:012146, January 2020.

[24] Maulik, Ujjwal and Sanghamitra Bandyopadhyay: *Genetic algorithm-based clustering technique.* Pattern Recognition, 33(9):1455–1465, 2000, ISSN 0031-3203.

[25] Craenendonck, Toon van and H. Blockeel: *Using Internal Validity Measures to Compare Clustering Algorithms.* In *ICML 2015*, 2015.

[26] Mark Ming-Tso Chiang, Boris Mirkin: *Intelligent Choice of the Number of Clusters in K-Means Clustering: An Experimental Study with Different Cluster Spreads.* Journal of classification, 27:3–40, 2010.

[27] Douglas Steinley, Michael J. Brusco. *Psychological Methods*, 16(3):285–297, 2011.

[28] Karlsson, Sten: *The Swedish car movement data project Final report.* Chalmers University of Technology, 2013.

[29] Björnsson, Lars Henrik and Sten Karlsson: *The potential for brake energy regeneration under Swedish conditions.* Applied Energy, 168:75–84, 2016, ISSN 0306-2619.

[30] *Volvo cars technical data.* 2020.

6
Appendix I

Symbol	**Meaning**	**Unit**
Acc_{avg}	Average Positive Acceleration	$\mathrm{m\,s^{-2}}$
Dcc_{avg}	Average Negative Acceleration	$\mathrm{m\,s^{-2}}$
A_{avg}	Average Acceleration	$\mathrm{m\,s^{-2}}$
V_{avg}	Average Speed	$\mathrm{m\,s^{-1}}$
P_{avg}	Average Power	$\mathrm{kW\,kg^{-1}}$
T_{acc}	Duration of Positive acceleration	s
T_{dcc}	Duration of Negative acceleration	s
T_{trip}	Duration of Trip	s
T_{idle}	Duration of Idle	s
V_{max}	Maximum Speed	$\mathrm{m\,s^{-1}}$
K	Number of Clusters	–
T_{cycle}	Obtained Cycle Duration	s
T_{target}	Required Cycle Duration	s
σ_{acc}	Standard deviation of Positive acceleration	$\mathrm{m\,s^{-2}}$
σ_{dcc}	Standard deviation of Negative acceleration	$\mathrm{m\,s^{-2}}$
Σ_{stop}	Total Stops	–
Σ_{trips}	Total trips	–
δ_{error}	Relative error	%
Δ_{error}	Absolute error	–
ϕ_A	Calculated value	–
ϕ_E	Expected value	–

Table 6.1: Acronyms

7

Appendix II

7.1 Model testing cycles

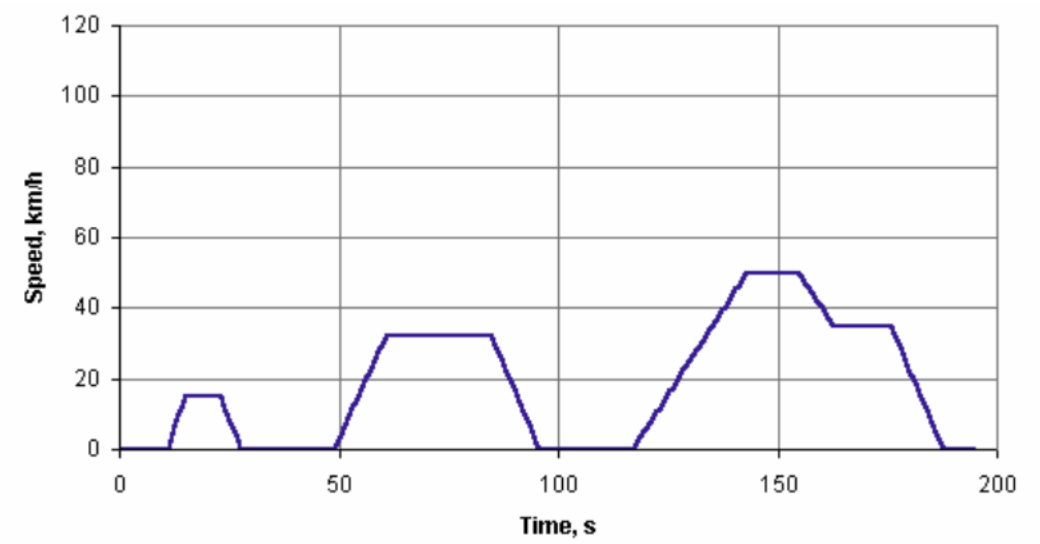


Figure 7.1: *ECE 15 Cycle*

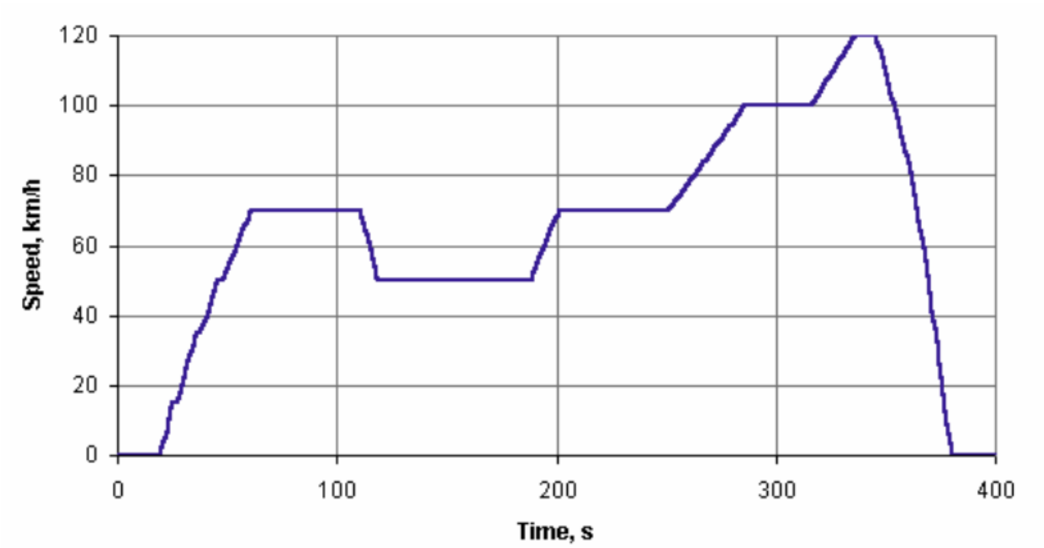
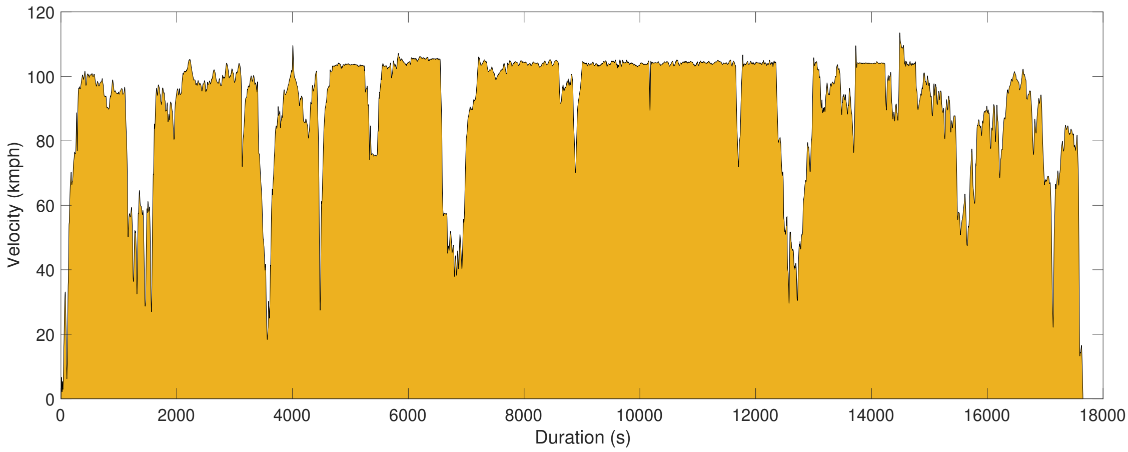


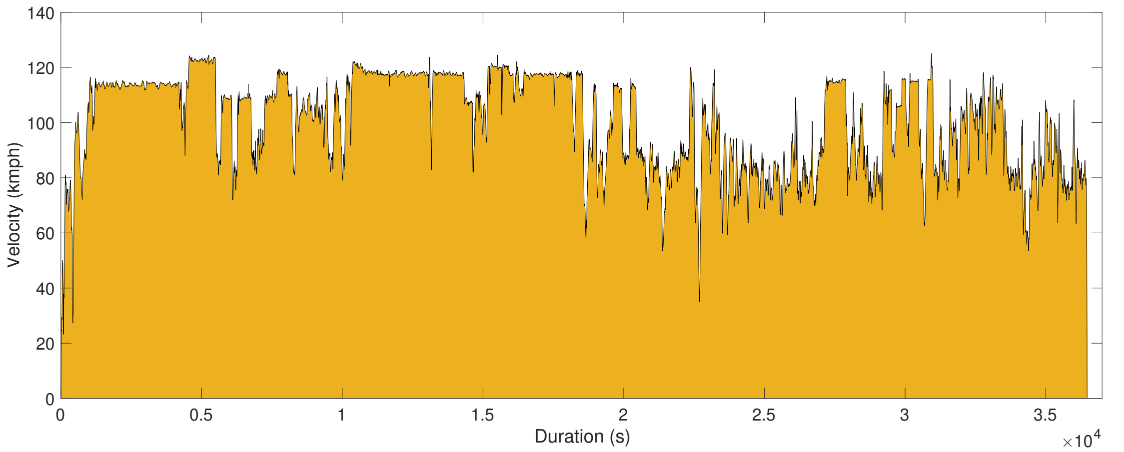
Figure 7.2: *EUDC Cycle*

7.2 Long haul trips



A sample of long duration trip, removed from the accumulated real driving data. The trip shows the driving pattern longing for several hours without a stop.

Figure 7.3: *Long haul micro-trip - 1*



A sample of long duration trip, removed from the accumulated real driving data. The trip shows the driving pattern longing for several hours without a stop.

Figure 7.4: *Long haul micro-trip - 2*

7.3 Cluster variations

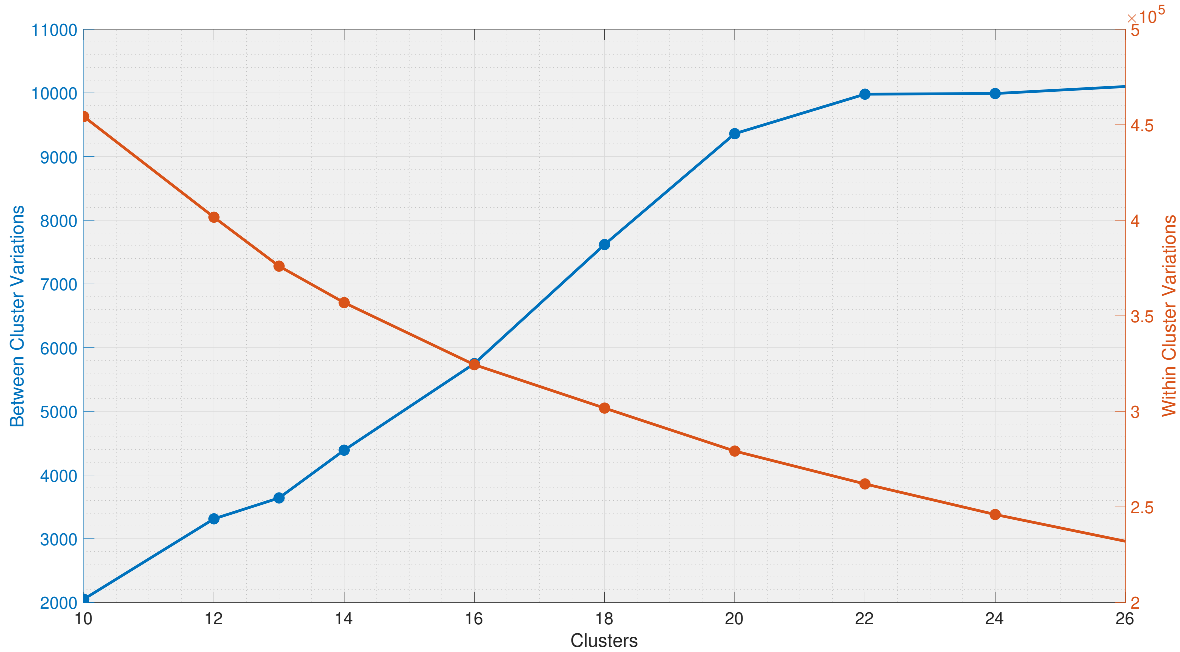


Figure 7.5: *Between and Within Cluster Variations*

7.4 Micro-trips grouped data

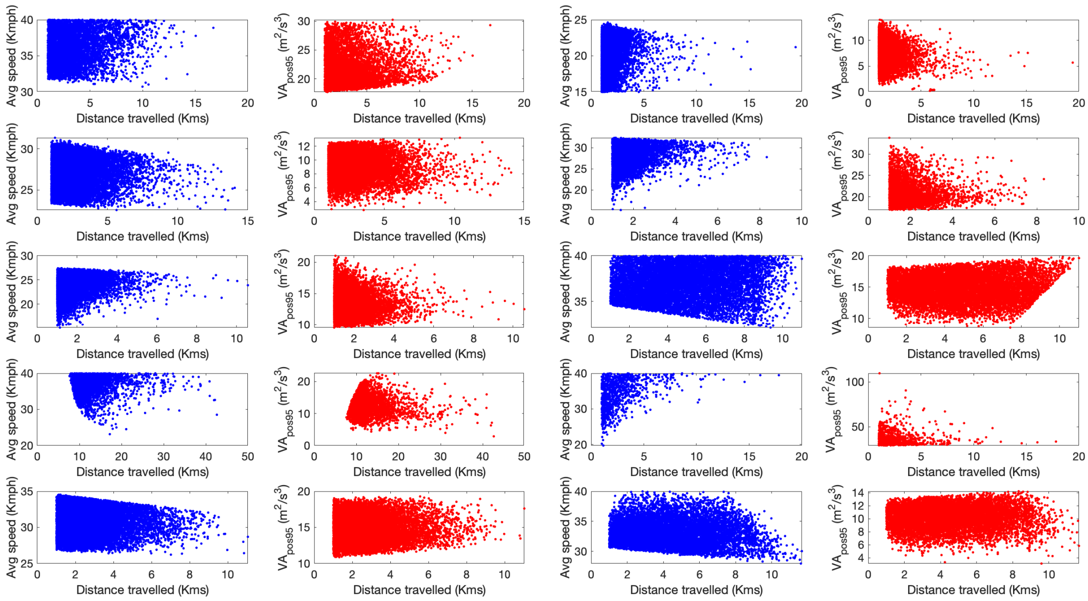


Figure 7.6: *Urban Group*

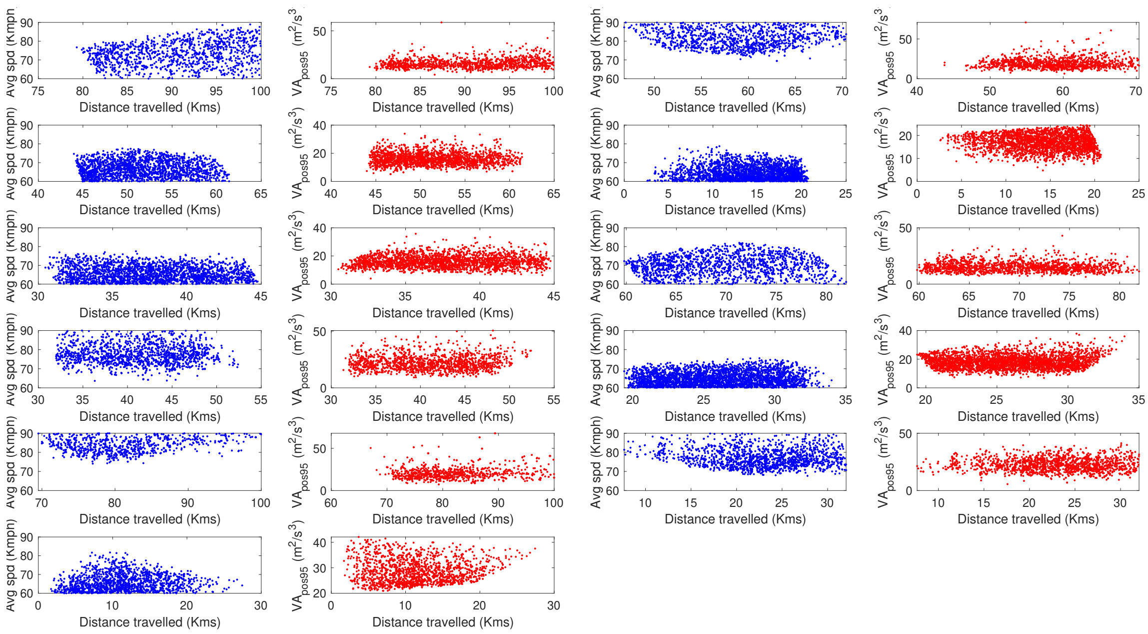


Figure 7.7: *Rural Group*

7.5 Design constraints

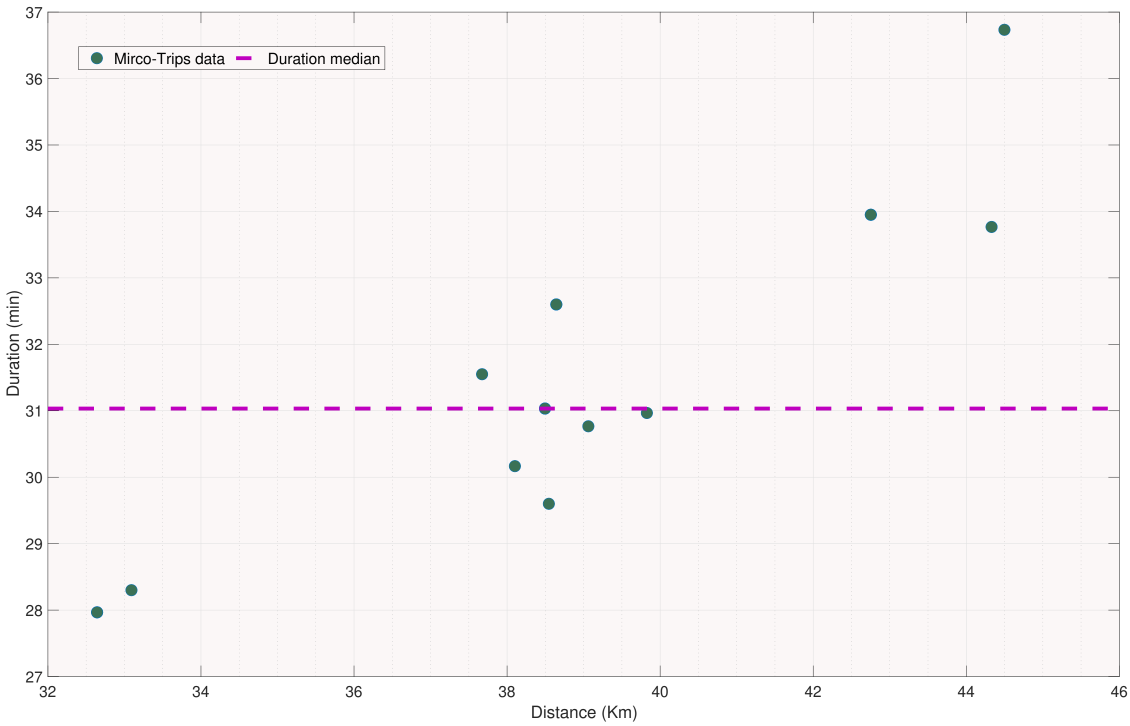


Figure 7.8: *Median duration of micro-trips in rural segment*

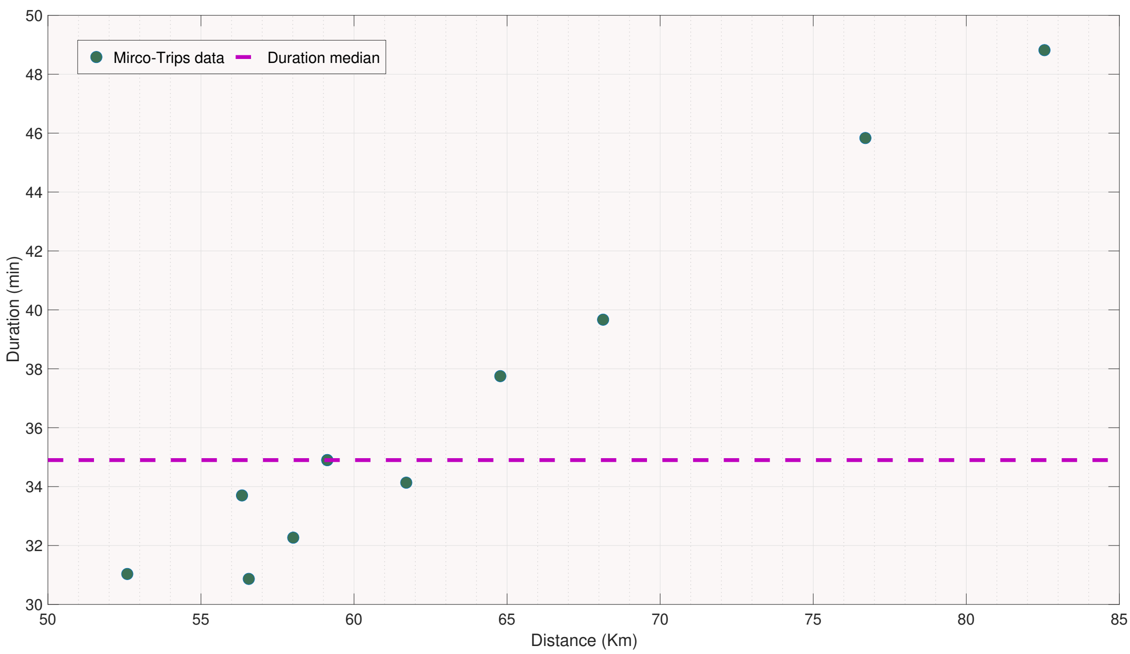


Figure 7.9: *Median duration of micro-trips in motorway segment*

7.6 Simulation settings

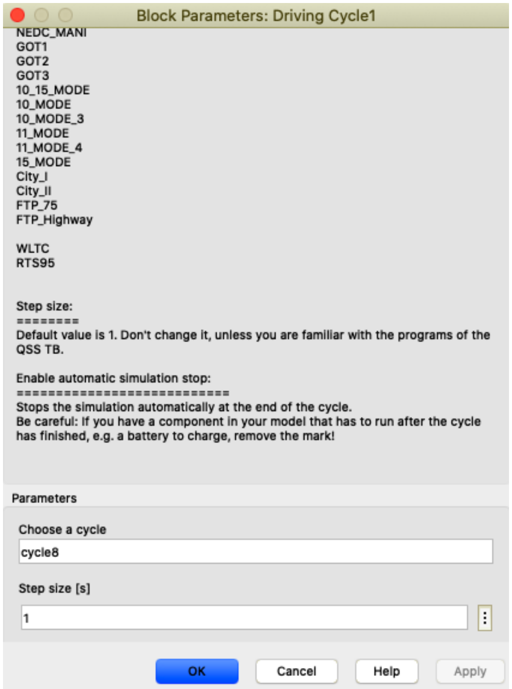


Figure 7.10: *Screenshot of driving cycle block settings in simulation model*

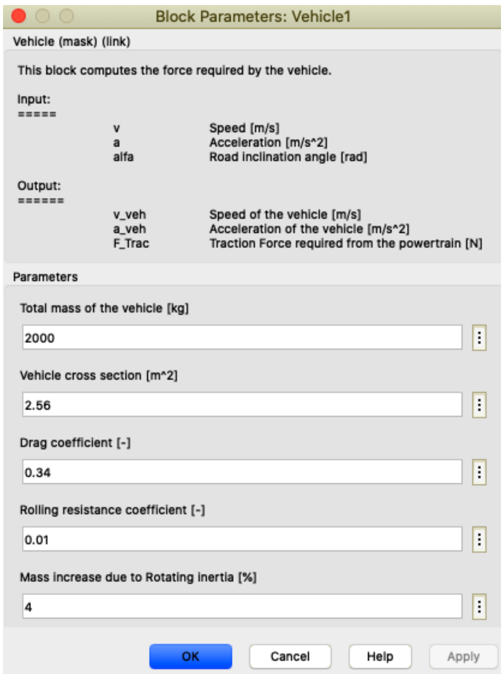


Figure 7.11: *Screenshot of vehicle block settings in simulation model*

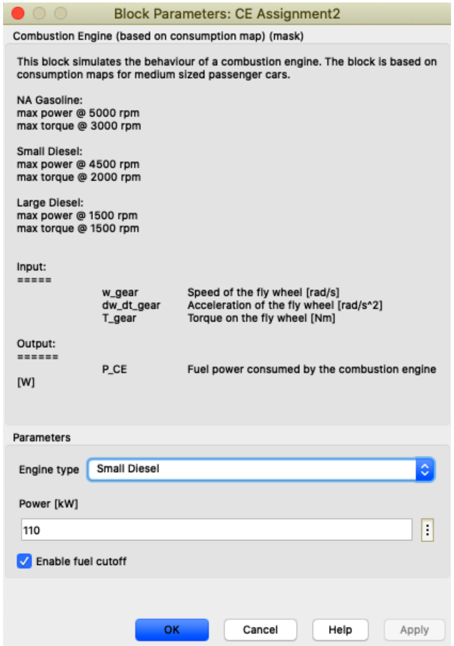


Figure 7.12: *Screenshot of engine block settings in simulation model*

7.7 Developed driving cycles

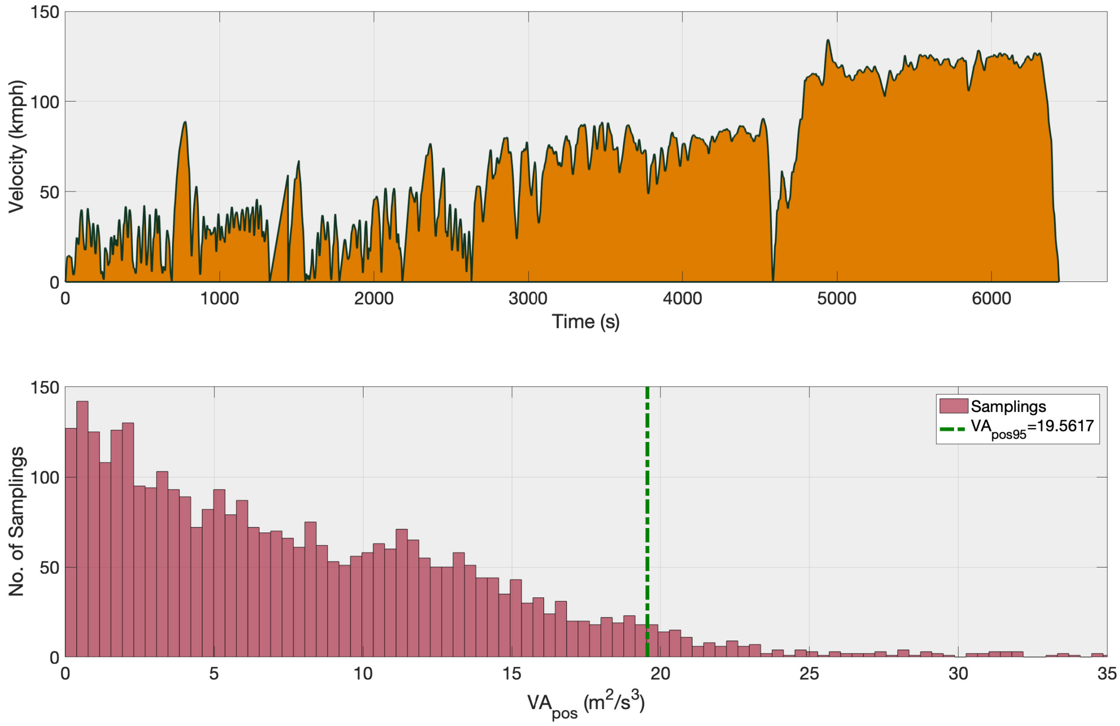


Figure 7.13: *Synthetic Driving Cycle - 2*

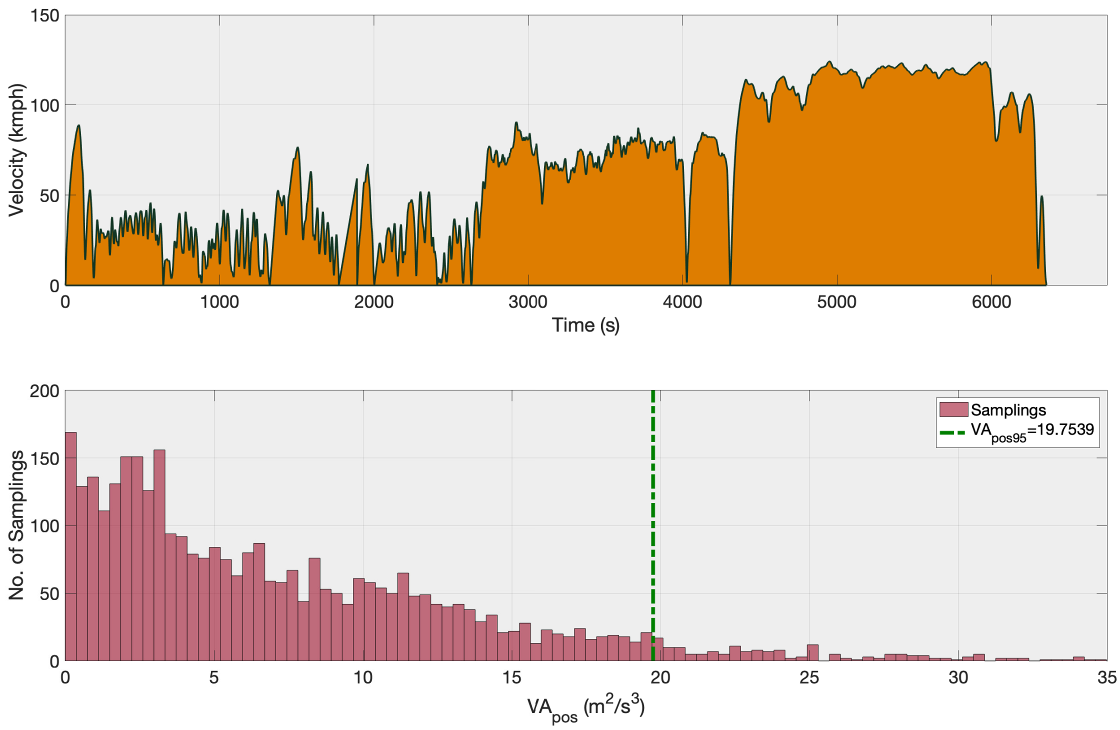


Figure 7.14: *Synthetic Driving Cycle - 3*

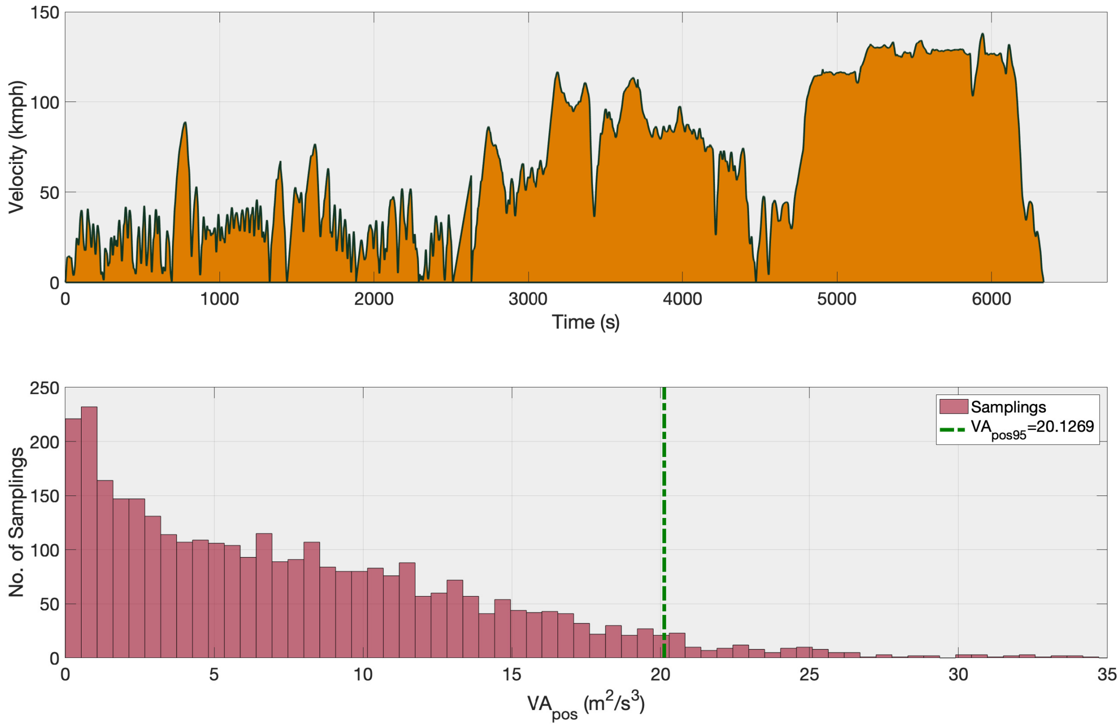


Figure 7.15: *Synthetic Driving Cycle - 4*

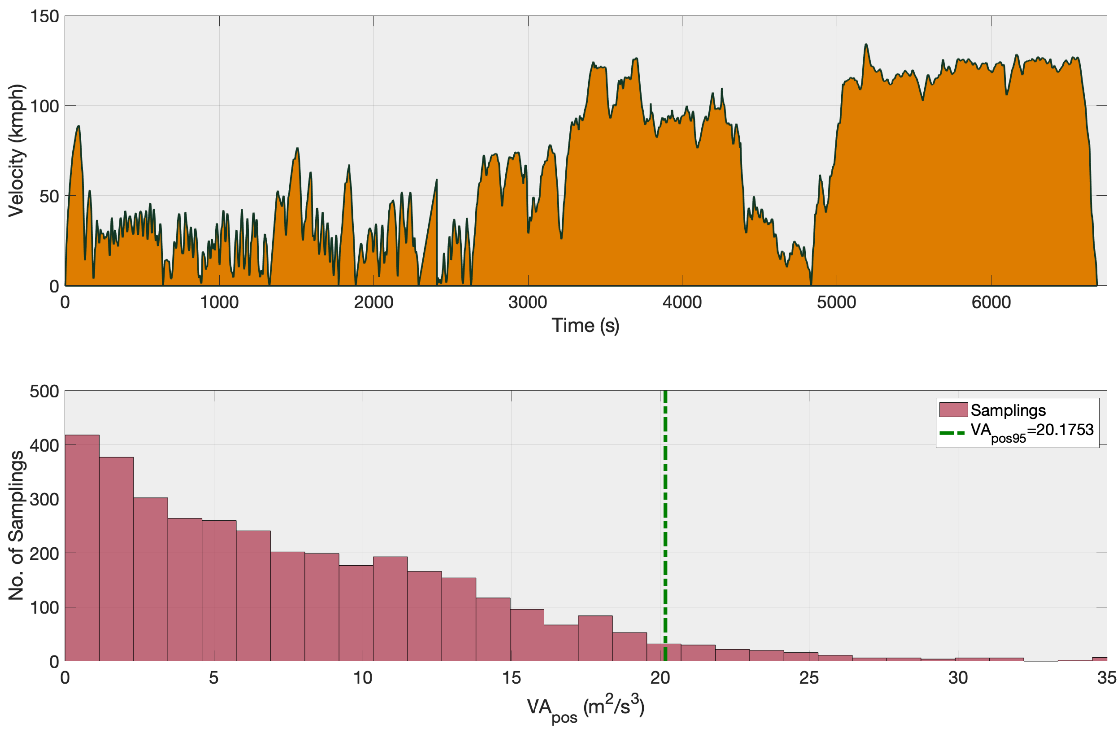


Figure 7.16: *Synthetic Driving Cycle - 5*

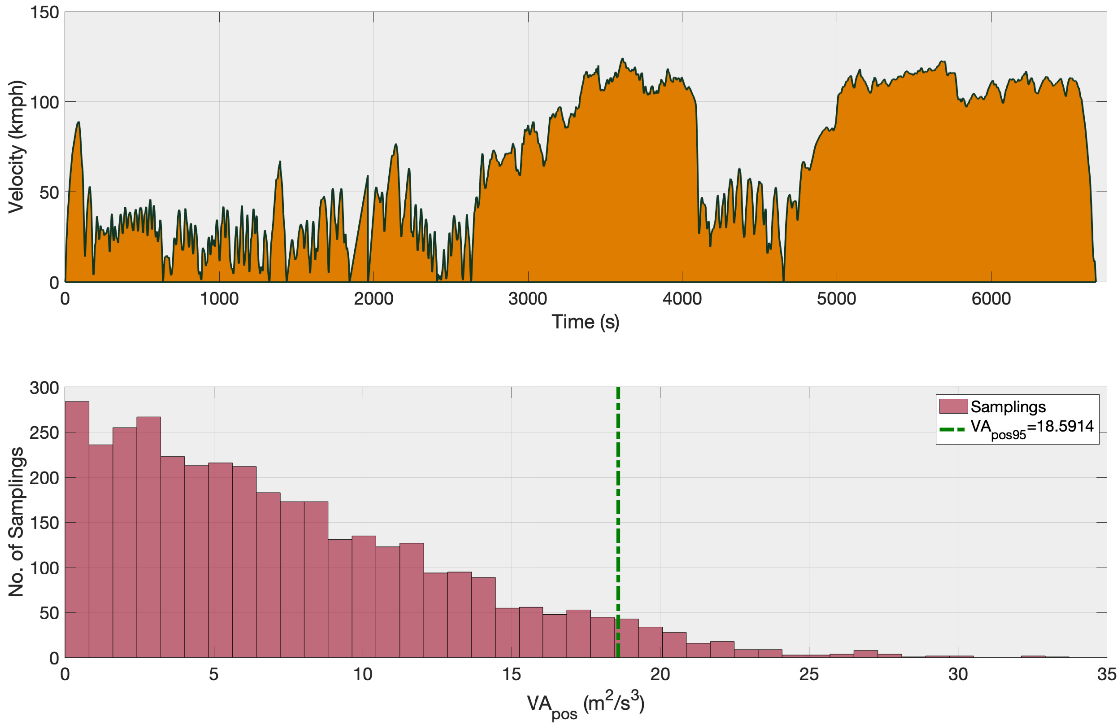


Figure 7.17: *Synthetic Driving Cycle - 6*

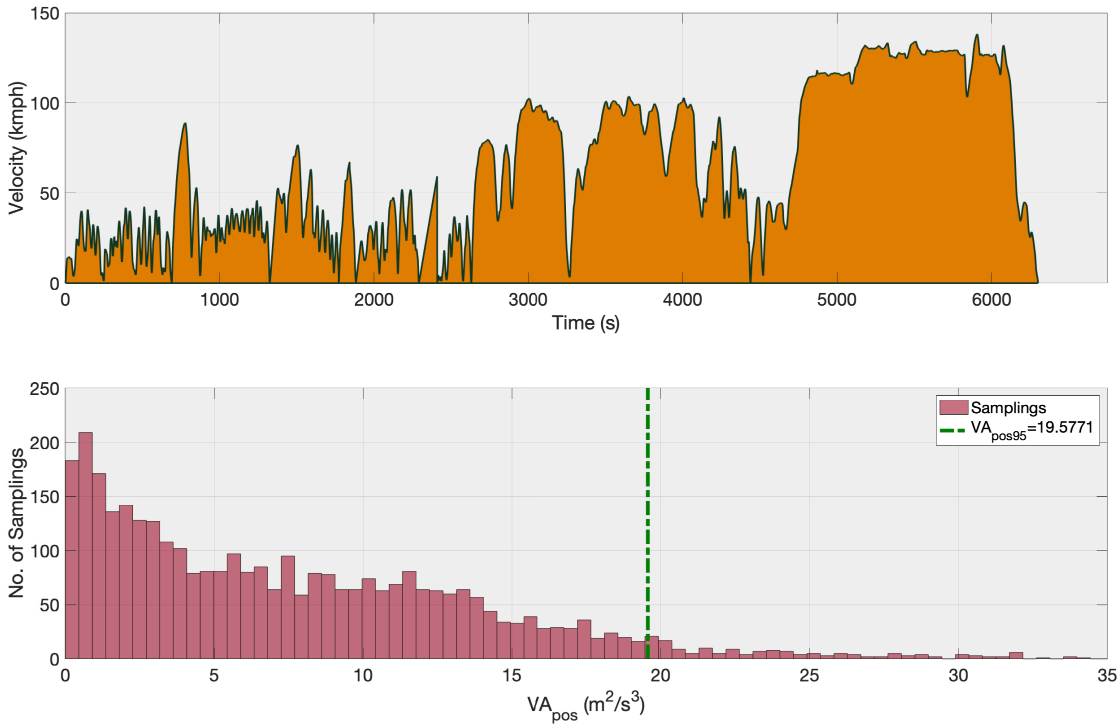


Figure 7.18: *Synthetic Driving Cycle - 7*

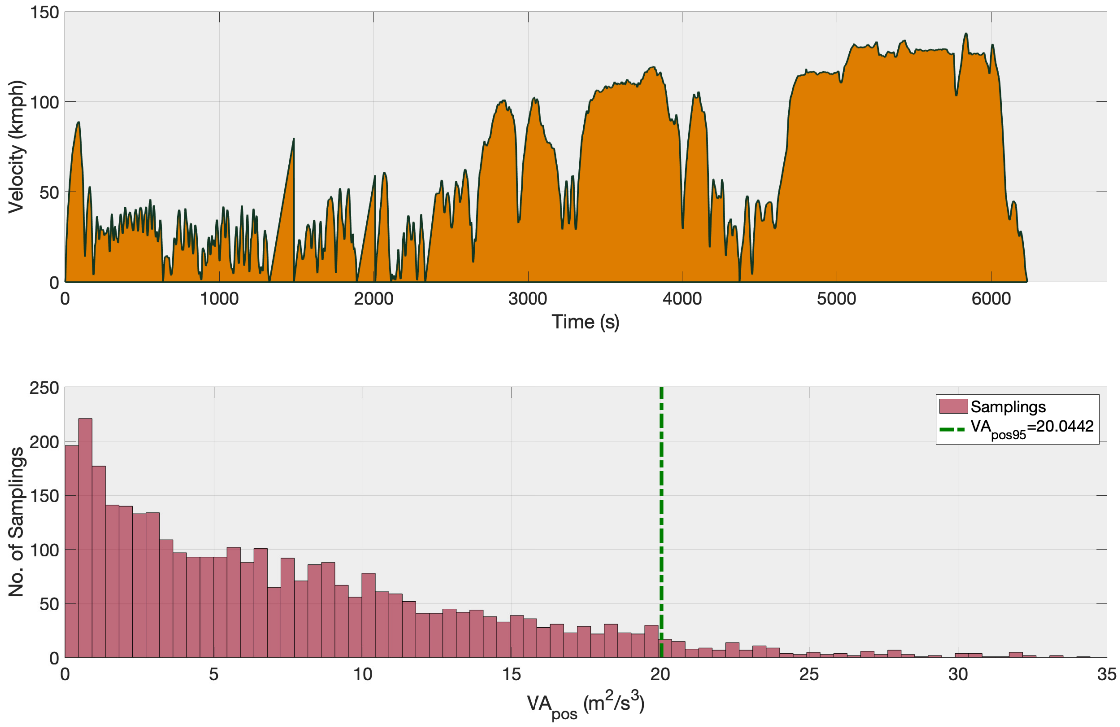


Figure 7.19: *Synthetic Driving Cycle - 8*

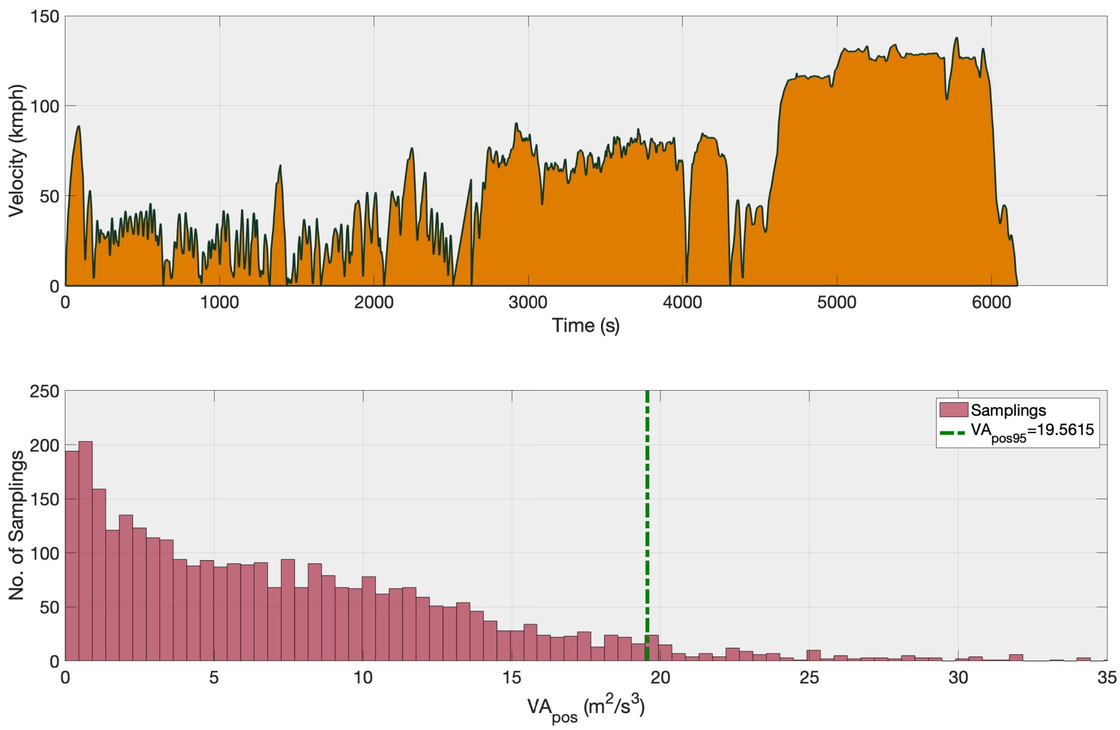


Figure 7.20: *Synthetic Driving Cycle - 9*

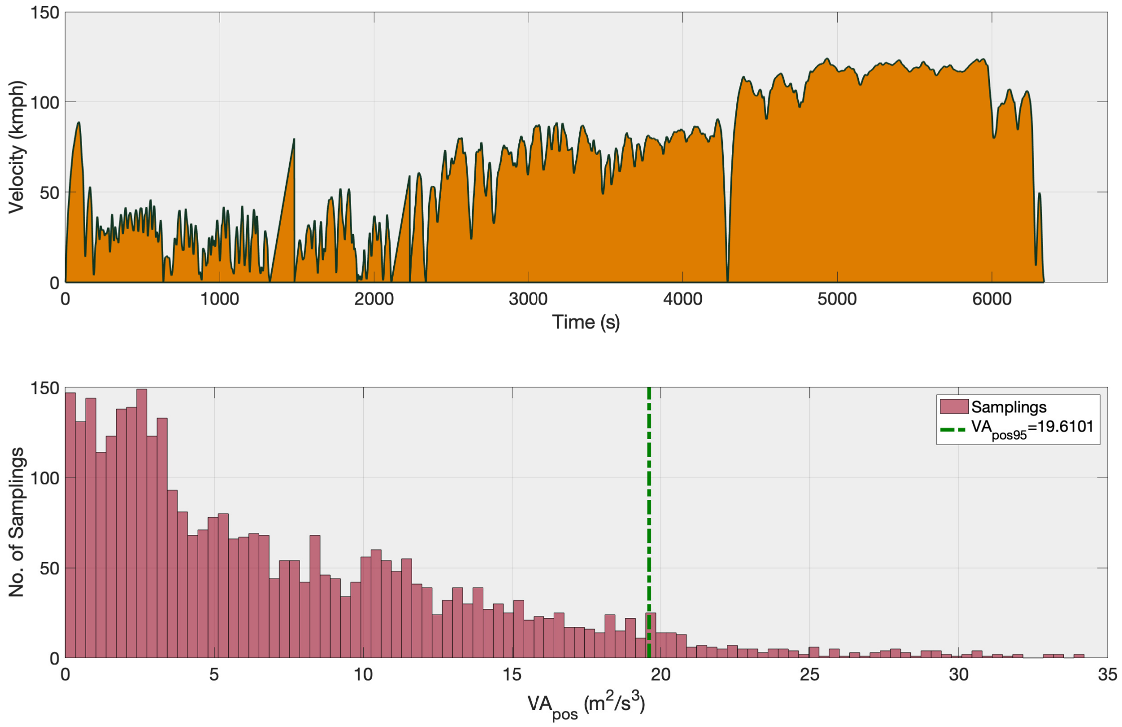


Figure 7.21: *Synthetic Driving Cycle - 10*

7.8 Simulation results - BSFC

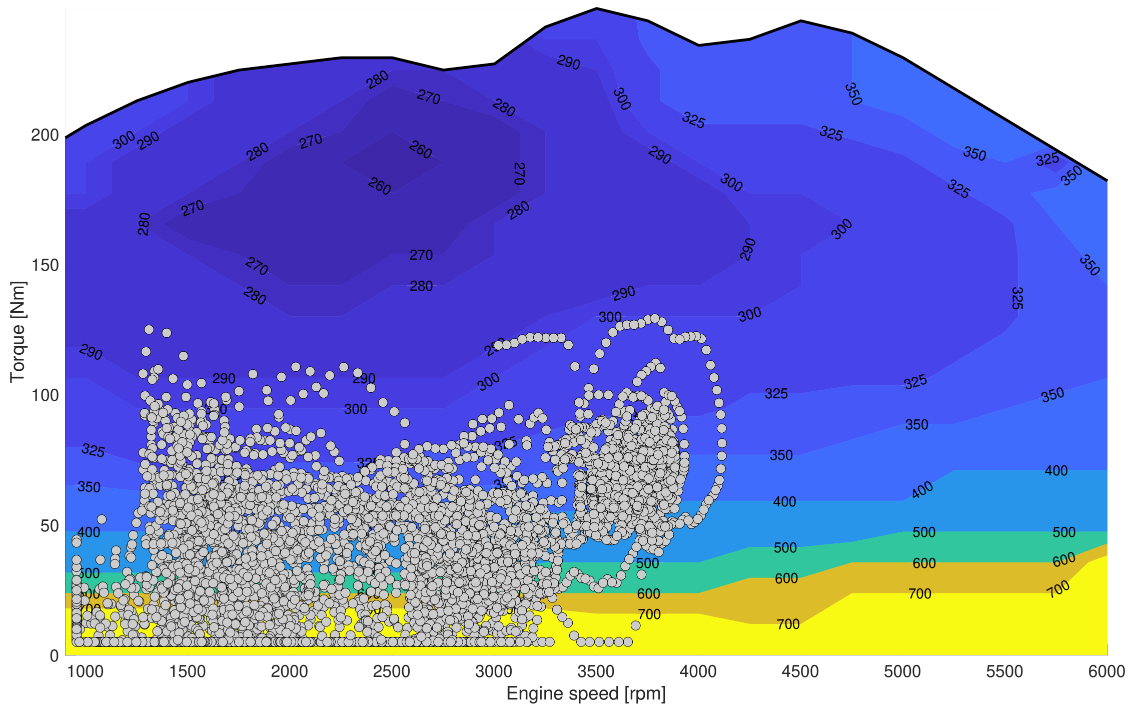


Figure 7.22: *BSFC plot of Gasoline engine for Cycle - 2*

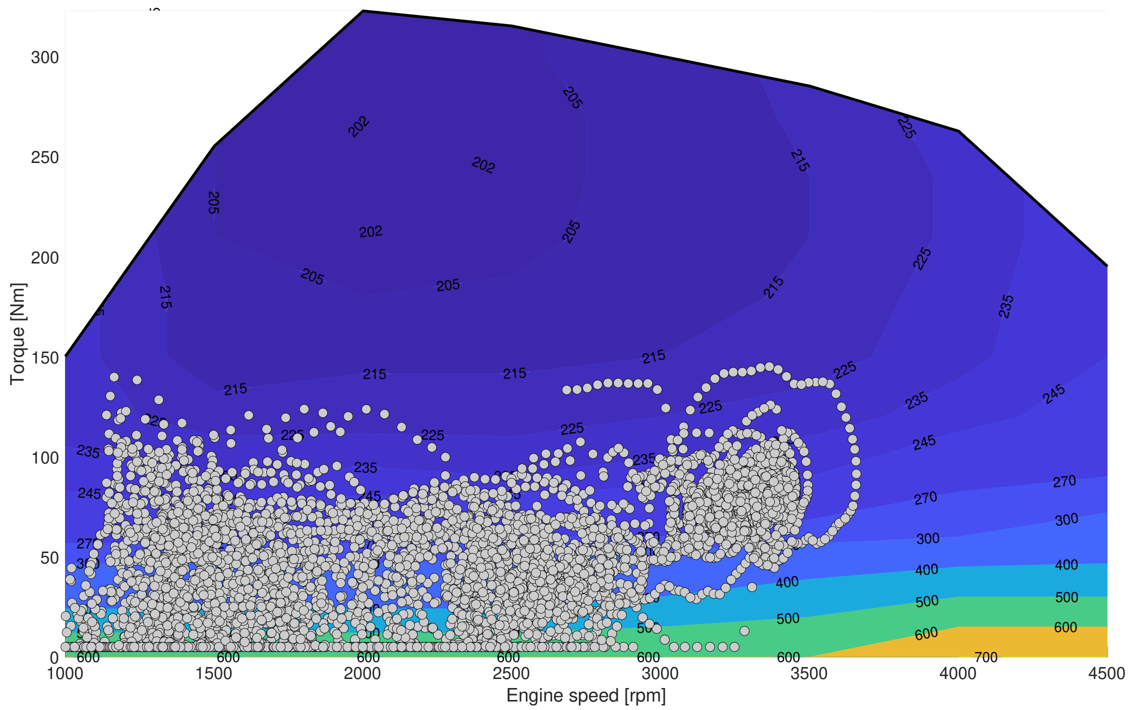


Figure 7.23: *BSFC plot of Diesel engine for Cycle - 2*

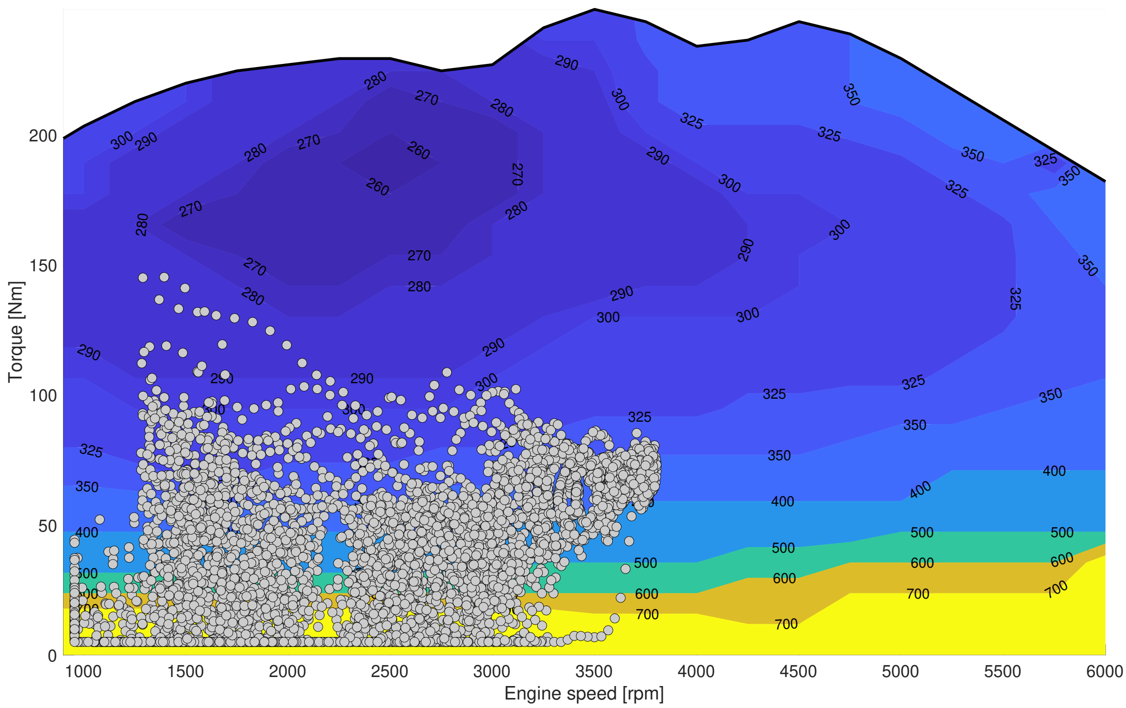


Figure 7.24: *BSFC plot of Gasoline engine for Cycle - 3*

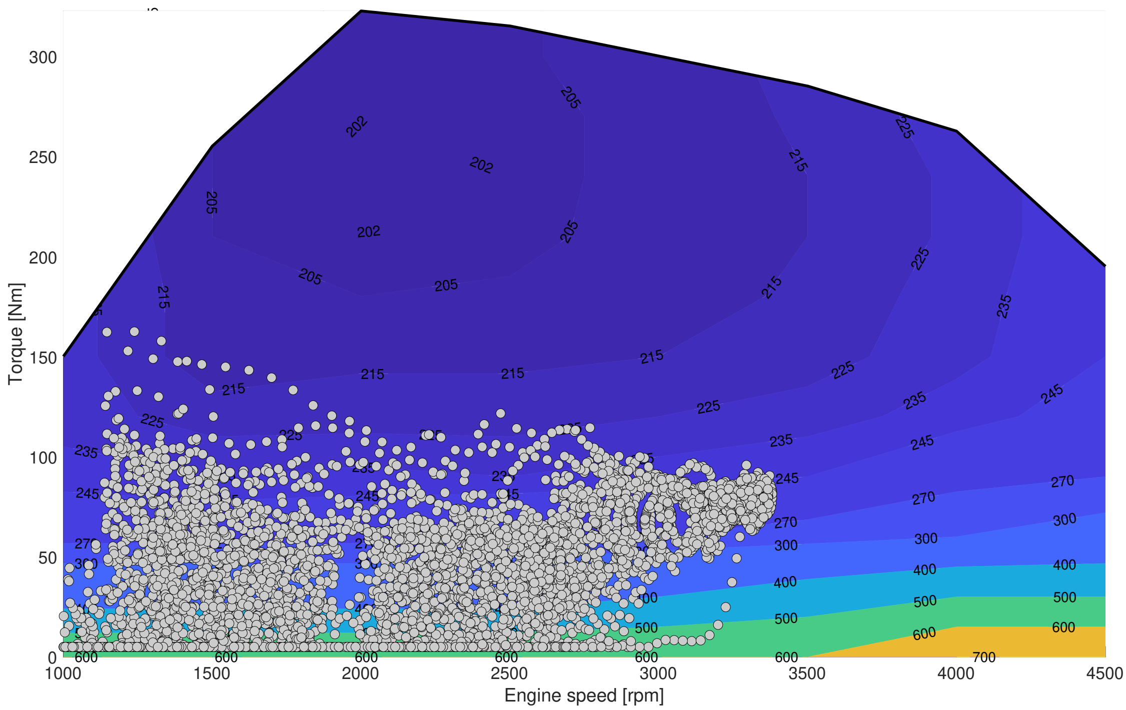


Figure 7.25: *BSFC plot of Diesel engine for Cycle - 3*

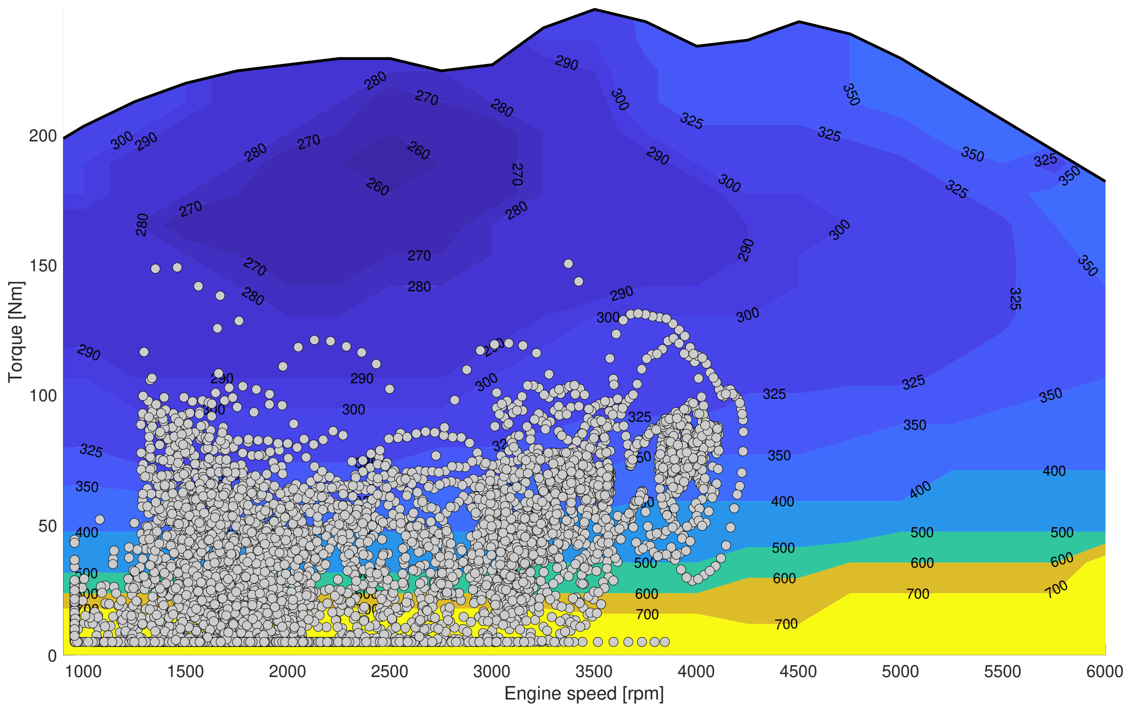


Figure 7.26: *BSFC plot of Gasoline engine for Cycle - 4*

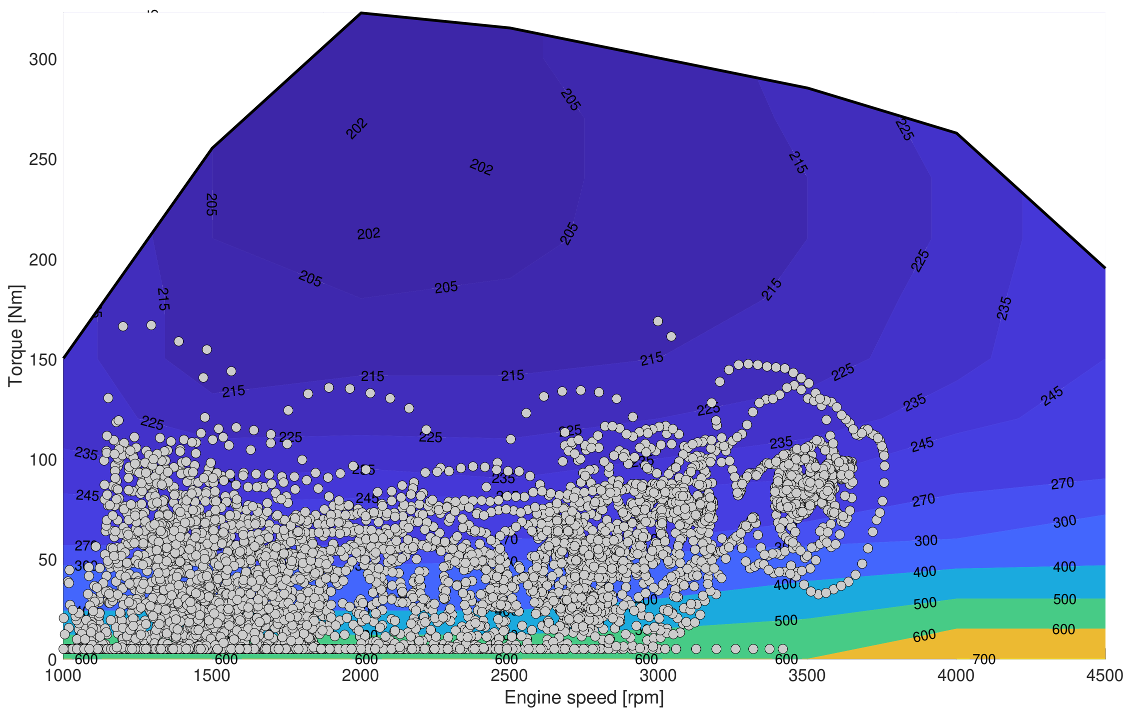


Figure 7.27: *BSFC plot of Diesel engine for Cycle - 4*

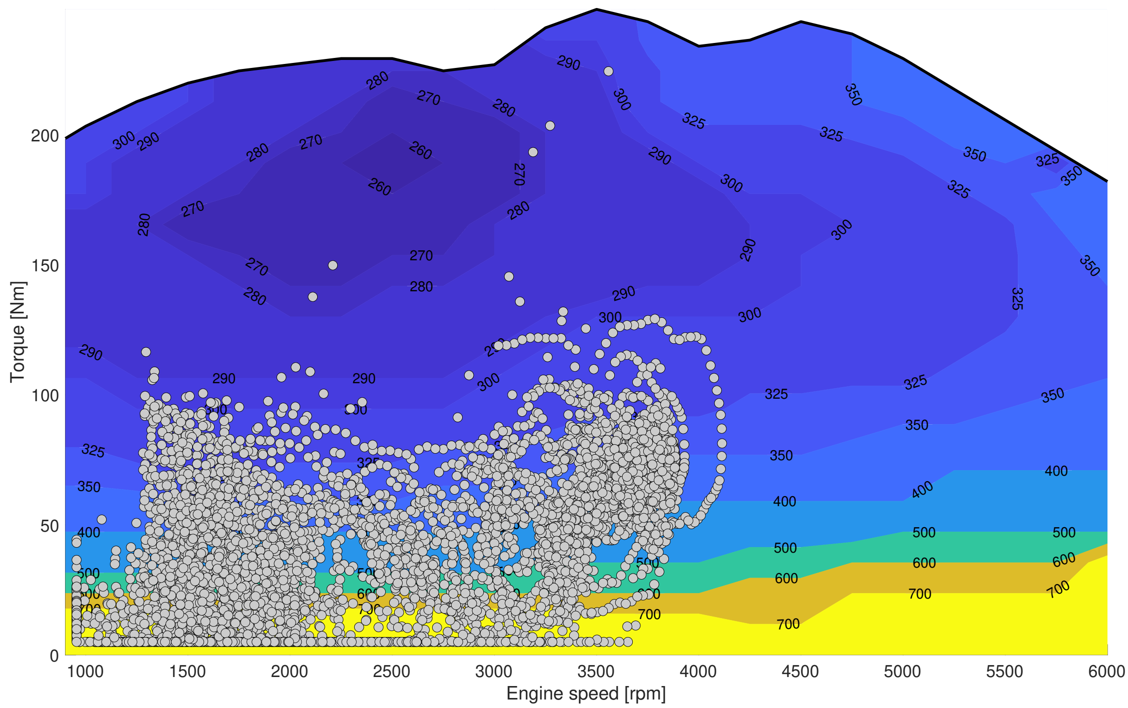


Figure 7.28: *BSFC plot of Gasoline engine for Cycle - 5*

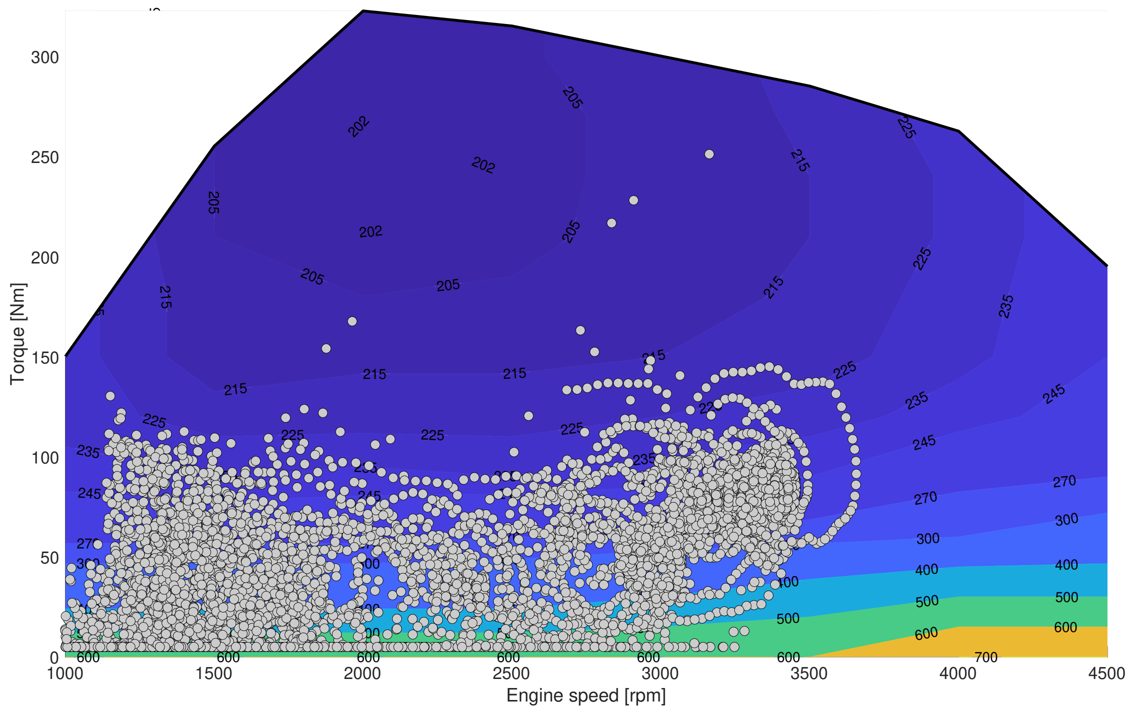


Figure 7.29: *BSFC plot of Diesel engine for Cycle - 5*

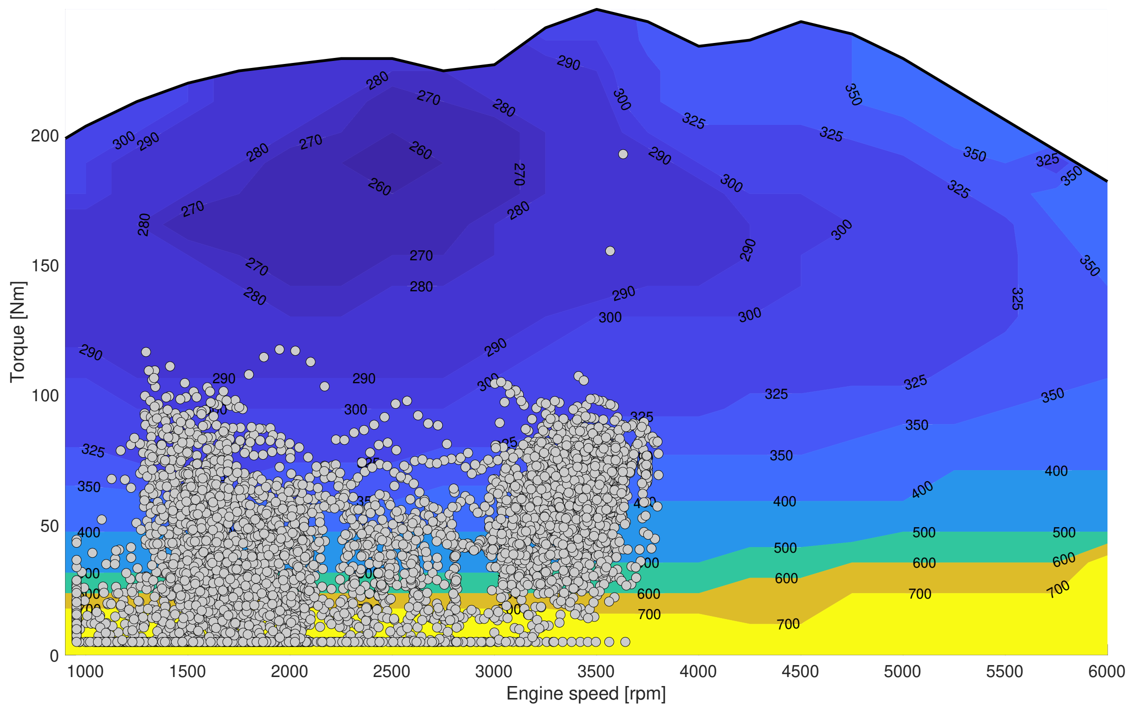


Figure 7.30: *BSFC plot of Gasoline engine for Cycle - 6*

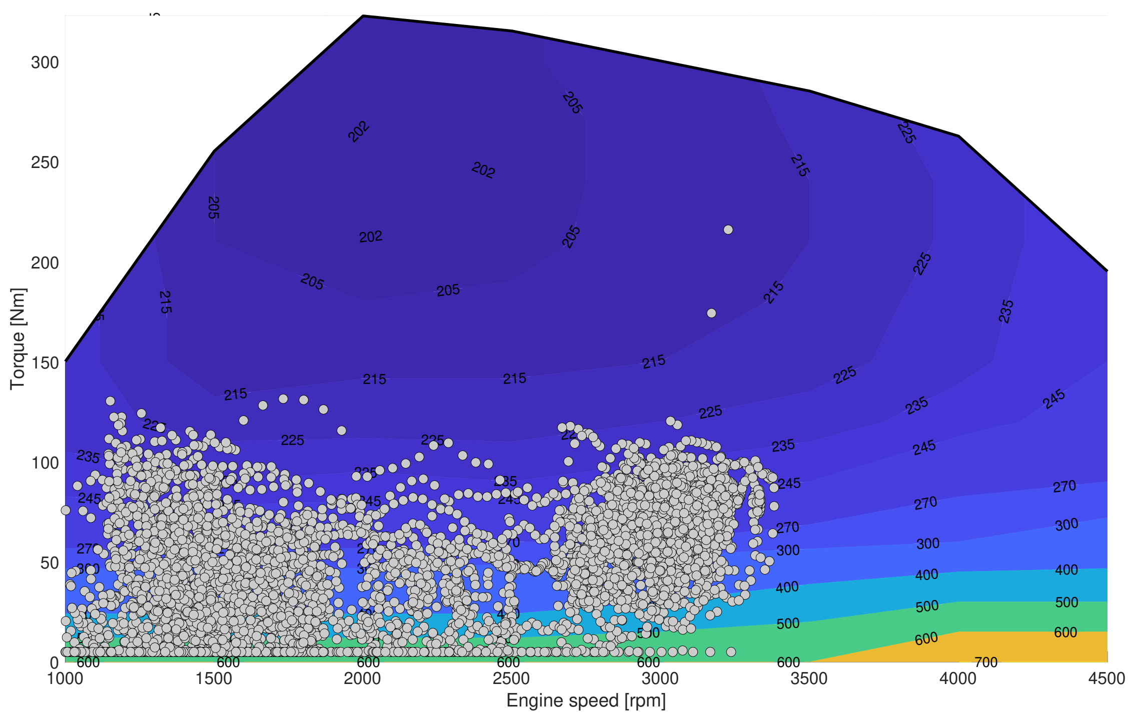


Figure 7.31: *BSFC plot of Diesel engine for Cycle - 6*

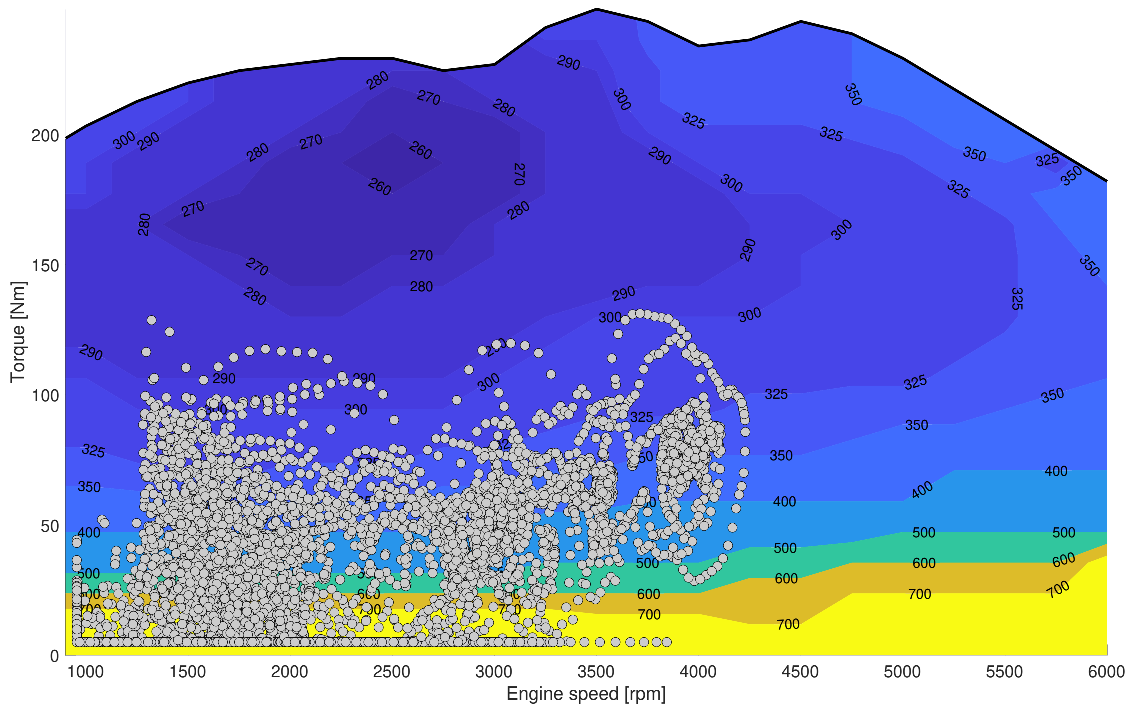


Figure 7.32: *BSFC plot of Gasoline engine for Cycle - 7*

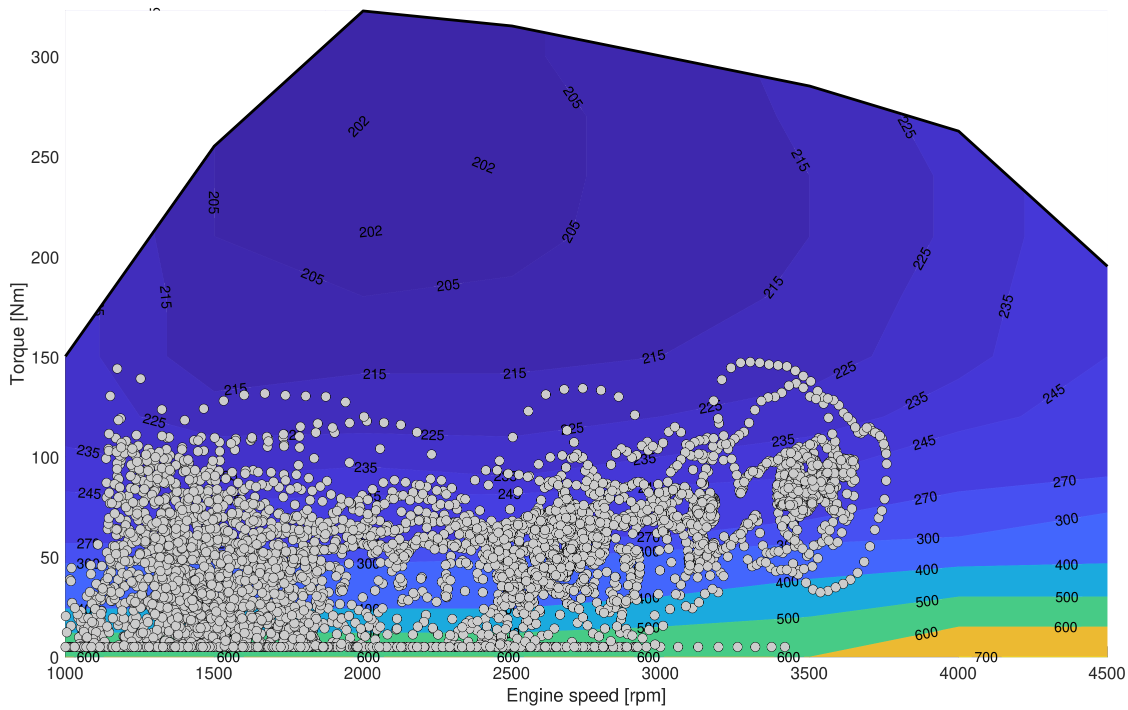


Figure 7.33: *BSFC plot of Diesel engine for Cycle - 7*

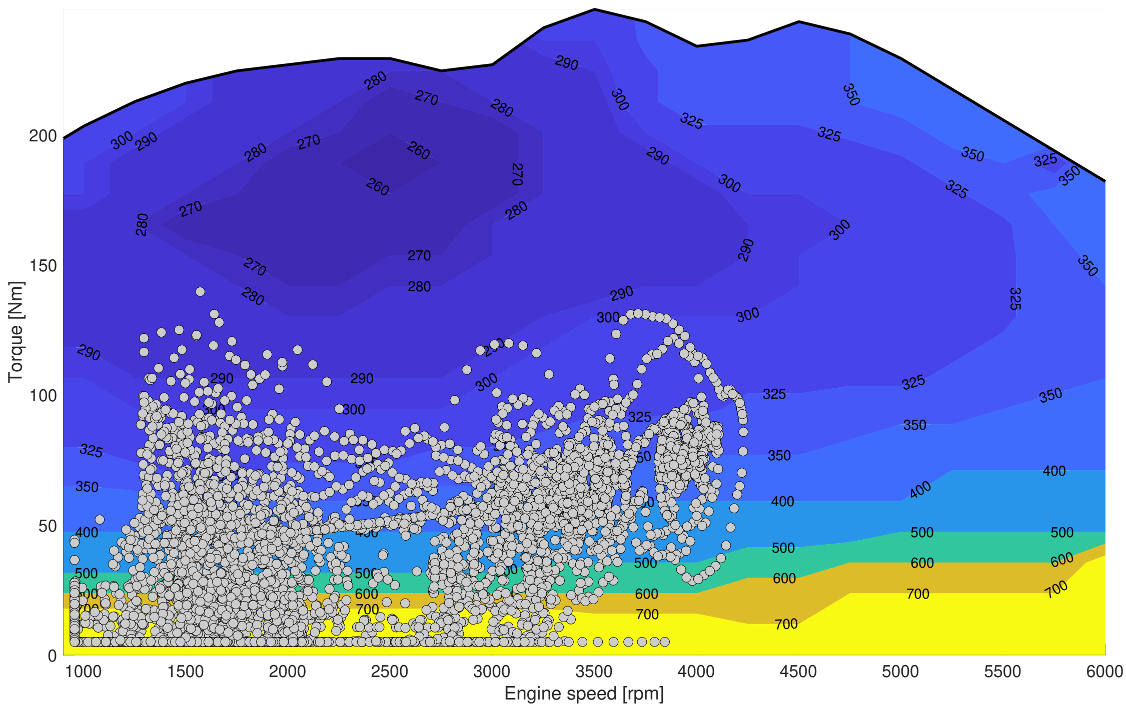


Figure 7.34: *BSFC plot of Gasoline engine for Cycle - 8*

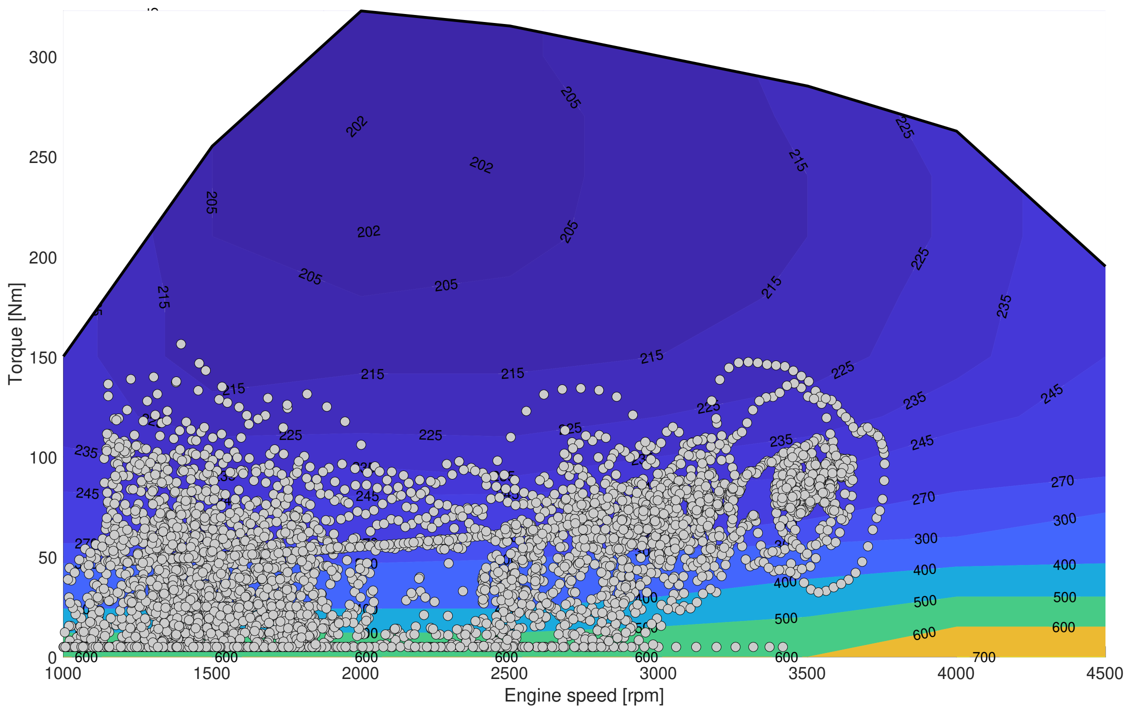


Figure 7.35: *BSFC plot of Diesel engine for Cycle - 8*

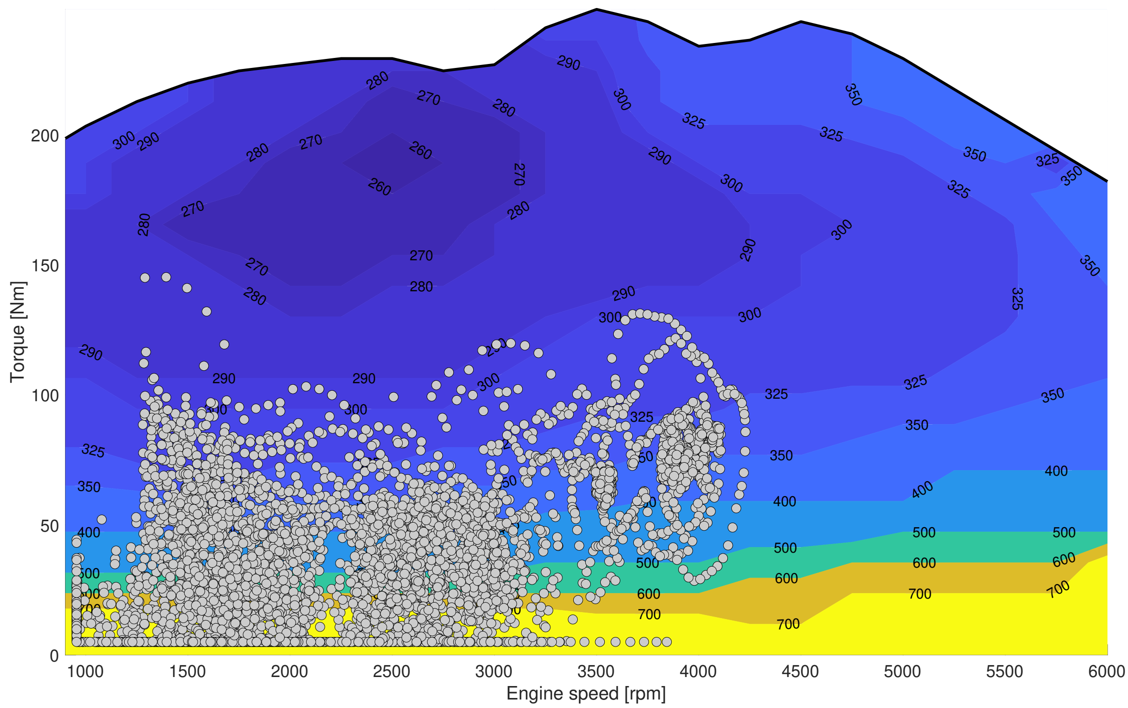


Figure 7.36: *BSFC plot of Gasoline engine for Cycle - 9*

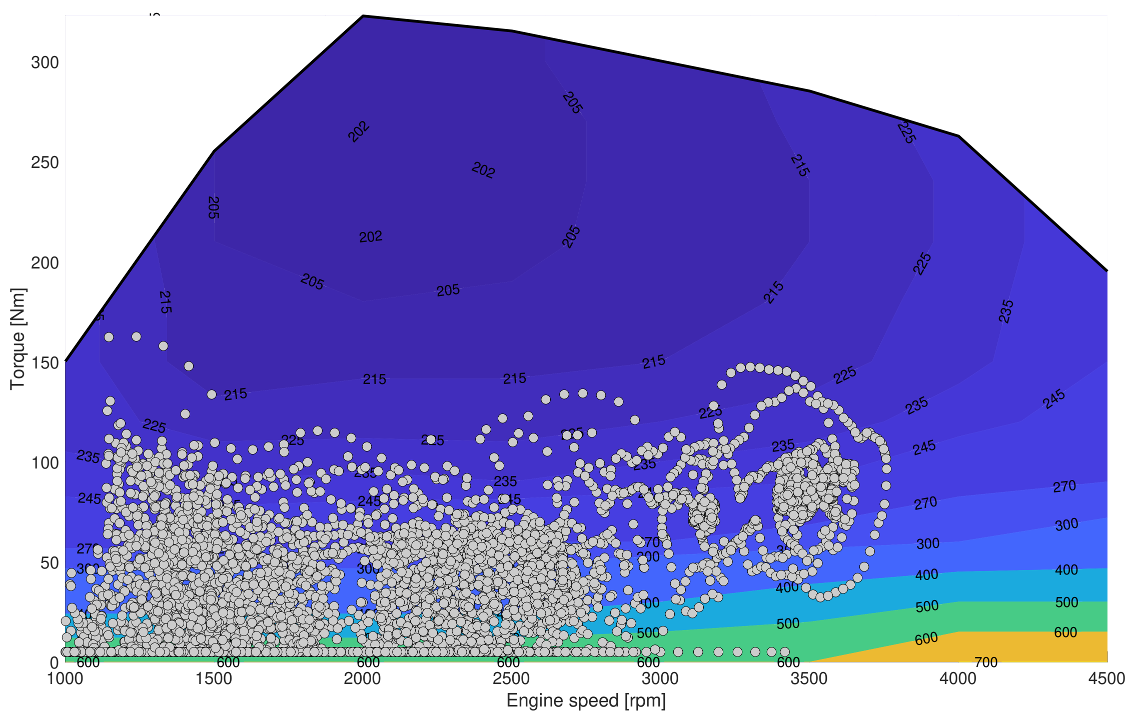


Figure 7.37: *BSFC plot of Diesel engine for Cycle - 9*

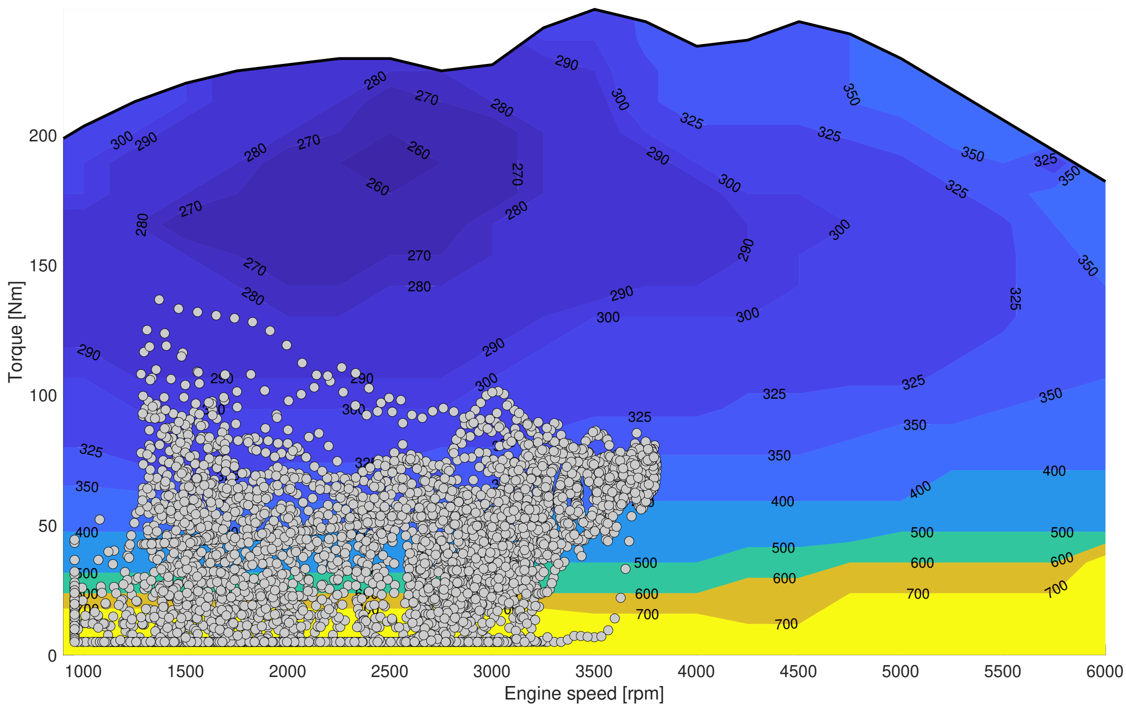


Figure 7.38: *BSFC plot of Gasoline engine for Cycle - 10*

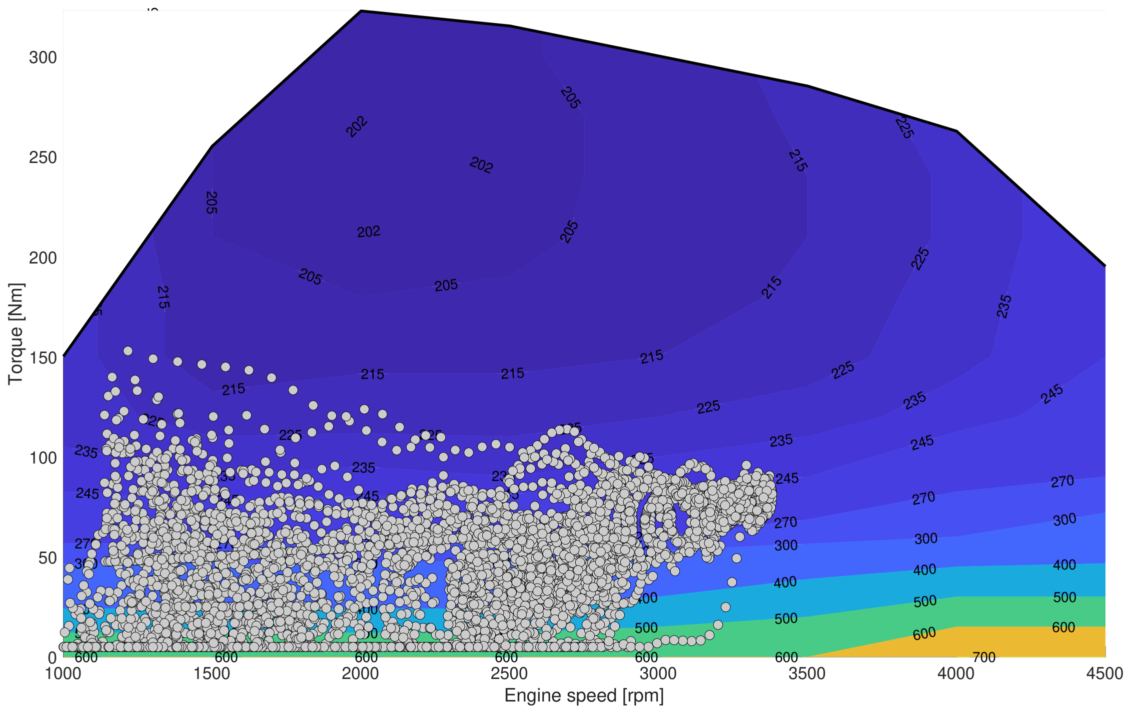


Figure 7.39: *BSFC plot of Diesel engine for Cycle - 10*

7.9 Regression analysis

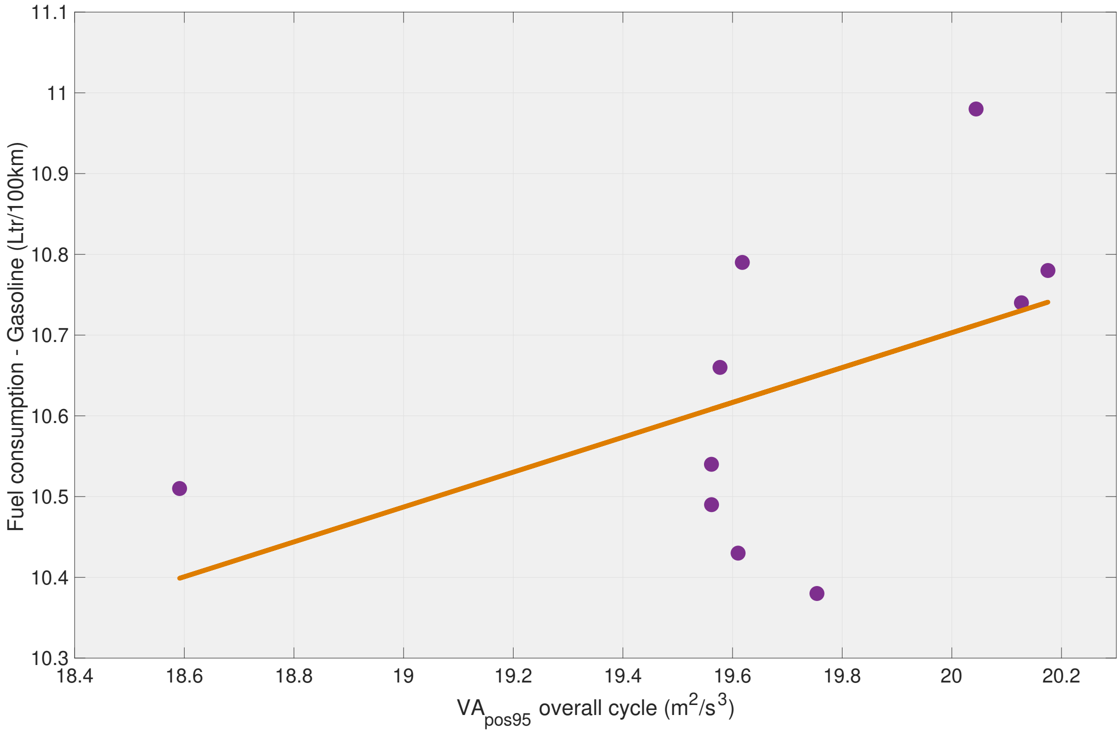


Figure 7.40: *Fuel consumption dependency on 95th VA$_{pos}$* - Gasoline

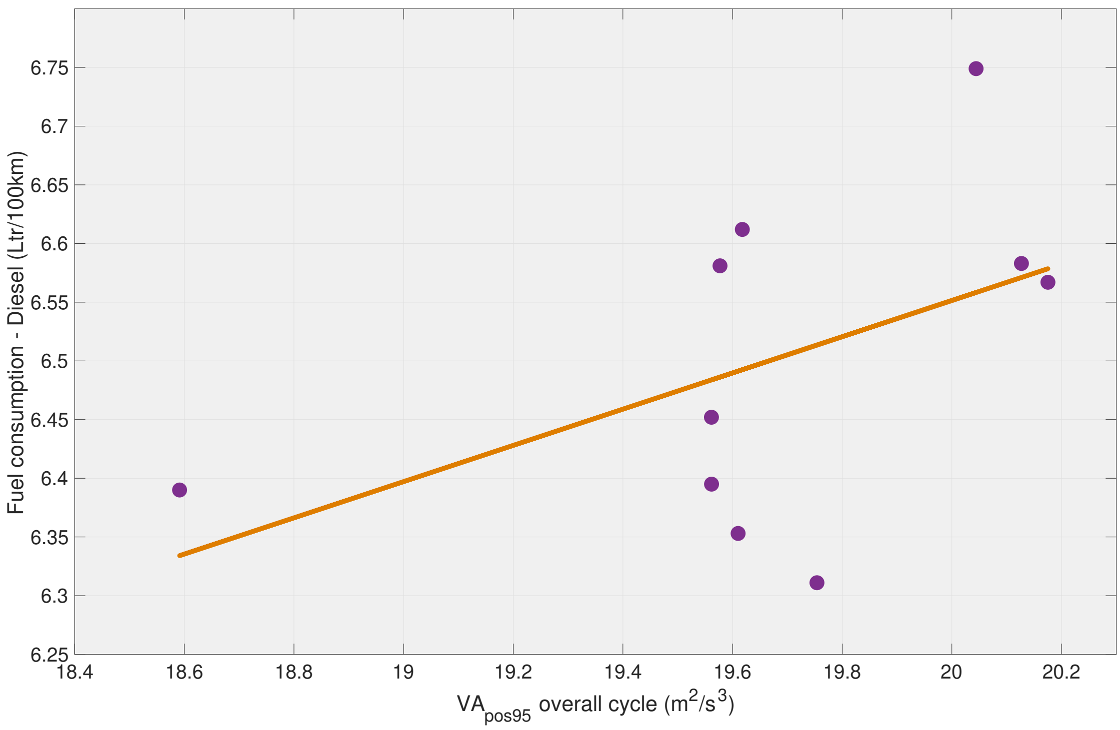


Figure 7.41: *Fuel consumption dependency on 95th VA$_{pos}$* - Diesel

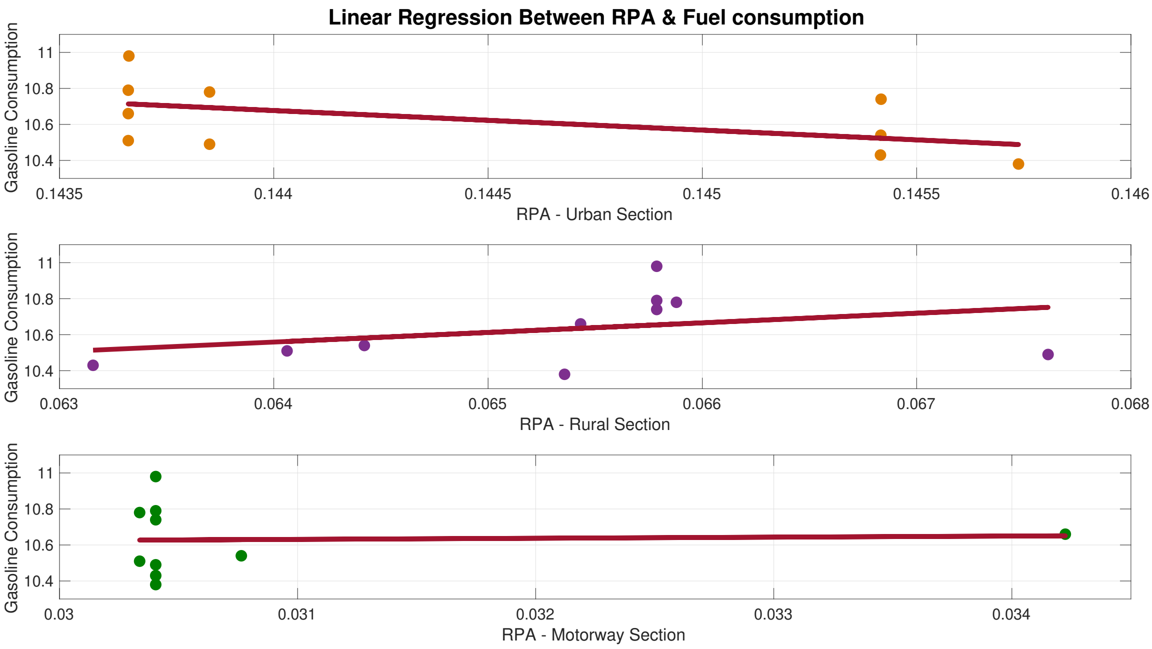


Figure 7.42: *Fuel consumption dependency on RPA - Gasoline*

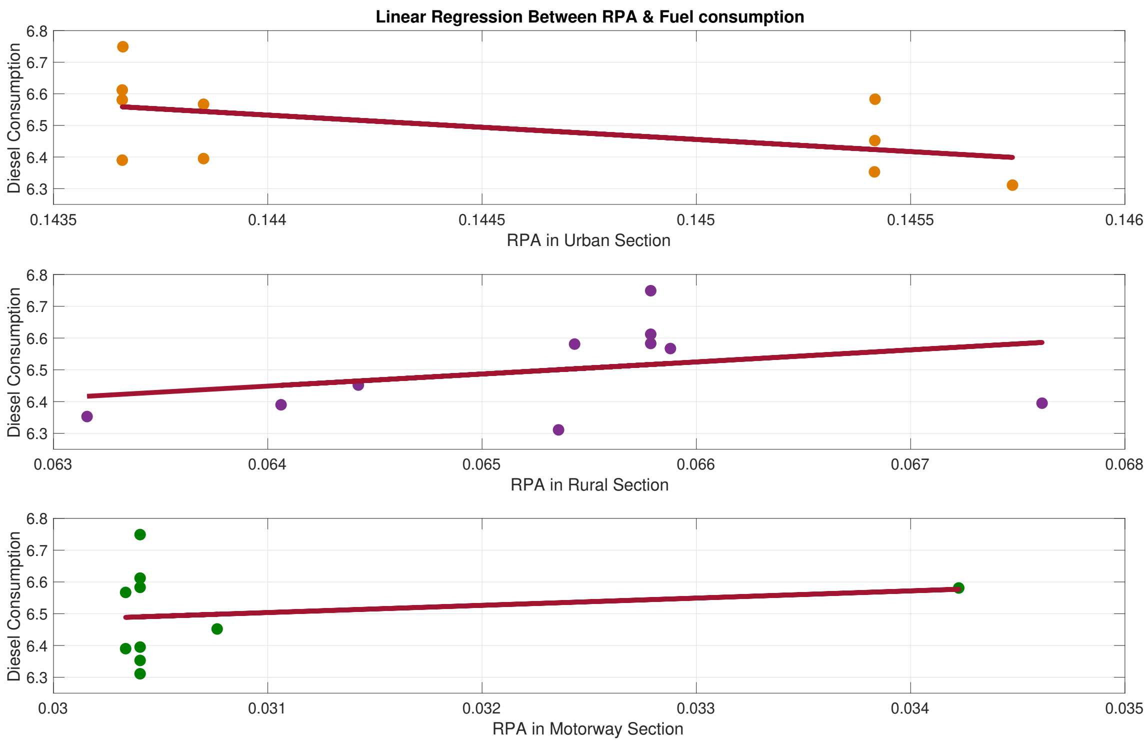


Figure 7.43: *Fuel consumption dependency on RPA - Diesel*

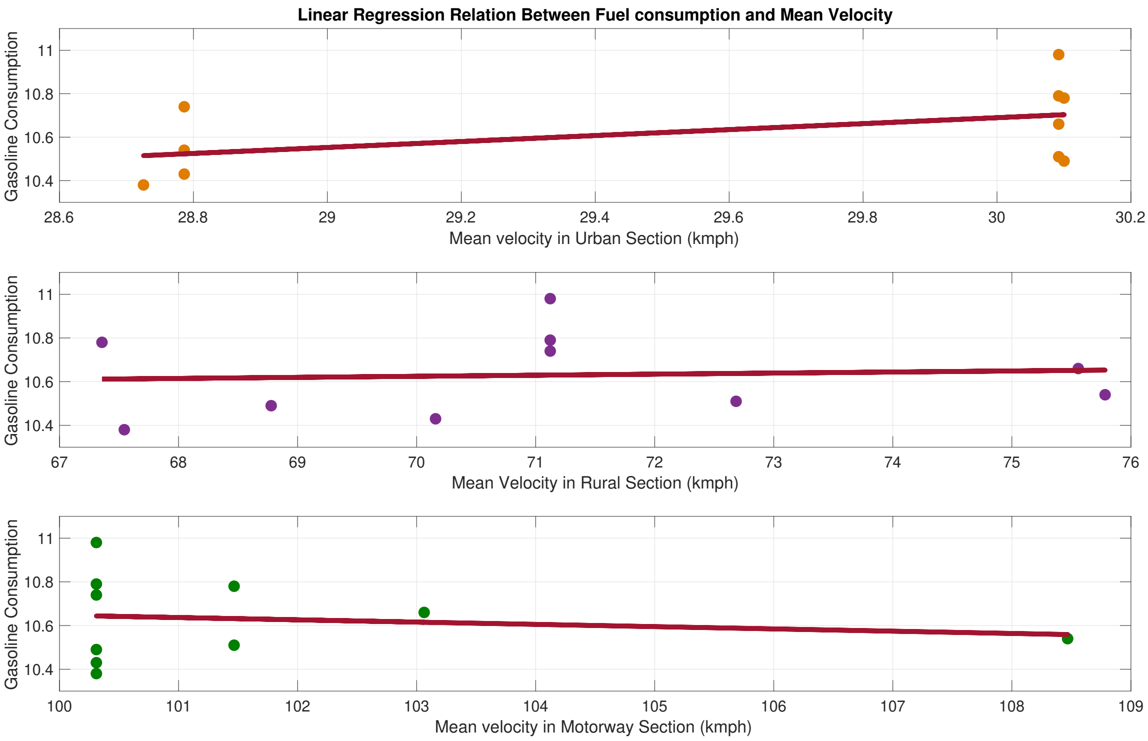


Figure 7.44: *Fuel consumption dependency on Mean Velocity - Gasoline*

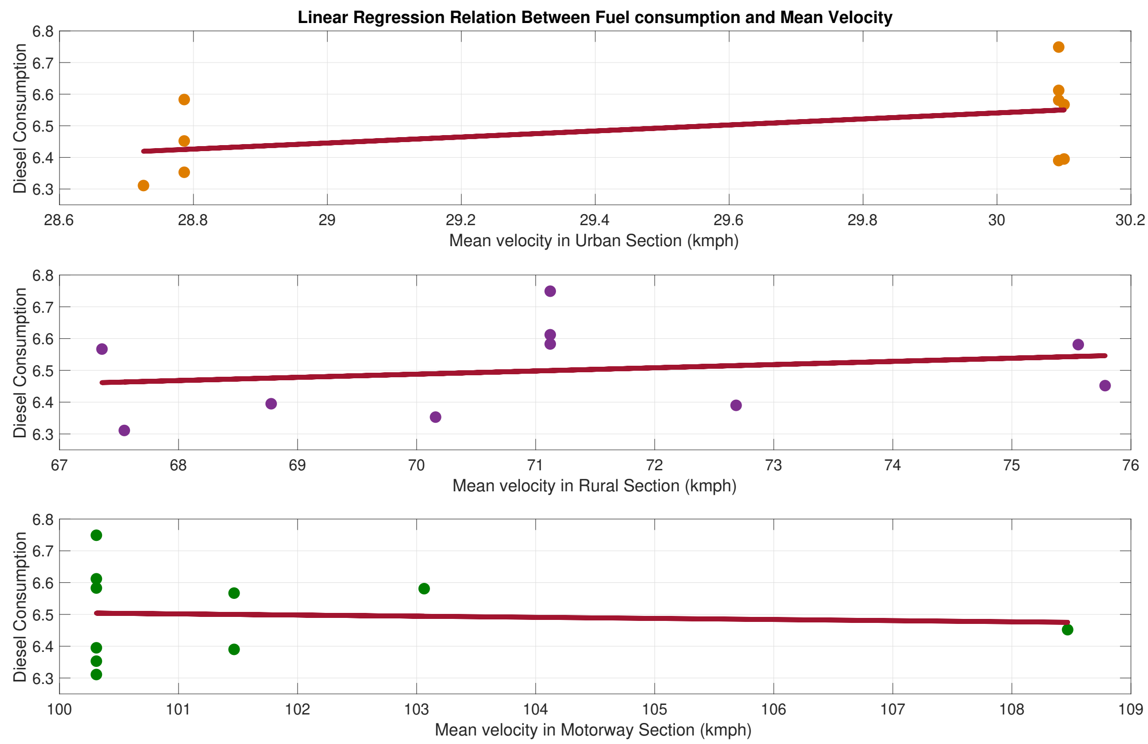


Figure 7.45: *Fuel consumption dependency on Mean Velocity - Diesel*

8 Appendix III

8.1 Data division

8.1.1 Handling original data

The data set obtained from 'The Swedish car movement data project' consists of several data logs of each trip in form of .mat files. The data files are created based on the vehicle numbers. Hence, a MATLAB script is used to extract the data from all .mat files.

```
% To load files

for k = 1:length(matFiles)  % all trips to be evaluated

  fprintf(1, 'Now reading %s\n', matFiles(k).name);
  matData(k) = load(matFiles(k).name);

for i=1:length( matData(k).MasterThesis.Speed)

          Speed{k}.Trip{i} =cell2mat(matData(k).MasterThesis.Speed(i));
          Acceleration{k}.Trip{i} =cell2mat(matData(k).MasterThesis.Acceleration(i))
              ;

end

  fprintf(1, 'Completed %s\n', matFiles(k).name);

end
```

8.1.2 K-means clustering method

The data set to begin with K-means clustering is stored as 'Grouped_Trips.mat'. This file contains cell arrays of micro trips grouped into Urban trips, Rural trips, Motorway trips, Long trips, and Extra-long trips. They are grouped based on the average speed and trip duration. The results of K-means clusters are present in 'Clustered_data.mat'. It contains a k-number of structures comprising cell arrays of velocity, average speed, VA_{pos95} and total distance of each micro trip. Since 22 is found to be the best k-value in this thesis work, the data file has 22 structure groups. The 'Clustered_data.mat' data file can be utilized for various statistical plots. A brief script for deploying k-means clustering and grouping the dataset is presented below.

```
%% K-Means clustering

Average_Spd_MT = All_Trips.Average_Speed;
Distance_fixed = All_Trips.Distance;
VA_pos_95 = All_Trips.VA_pos_95;
Microtrips_fixed = All_Trips.Speed;

x = cell2mat(Average_Spd_MT);
y = cell2mat(Distance_fixed);
z = cell2mat(VA_pos_95);
opts = statset('Display','final');
[idx,C] = kmeans([x,y,z],22,'Distance','sqeuclidean','Replicates',5,'Options',opts)
    ;

%% Grouping the clusters

for i = 1:length(idx)

    if idx(i) == 1
        Group1.Velocity{i,1} = Microtrips_fixed{i,1};
        Group1.Average_SPD{i,1} = Average_Spd_MT{i,1};
        Group1.VA_Pos_95{i,1} = VA_pos_95{i,1};
        Group1.Distance{i,1} = Distance_fixed{i,1};


    else if idx(i) == 2
        Group2.Velocity{i,1} = Microtrips_fixed{i,1};
        Group2.Average_SPD{i,1} = Average_Spd_MT{i,1};
        Group2.VA_Pos_95{i,1} = VA_pos_95{i,1};
        Group2.Distance{i,1} = Distance_fixed{i,1};

        ......
        ......
        ......
        ......

    else if idx(i) == 22
        Group22.Velocity{i,1} = Microtrips_fixed{i,1};
        Group22.Average_SPD{i,1} = Average_Spd_MT{i,1};
        Group22.VA_Pos_95{i,1} = VA_pos_95{i,1};
        Group22.Distance{i,1} = Distance_fixed{i,1};

        end
        end
        end
end

% To replace empty cells

for i = 1:length(Group1.Velocity)

  if isempty(Group1.Velocity{i,1})
    Group1.Velocity{i,1} = 0;
  end

  if isempty(Group1.Average_SPD{i,1})
    Group1.Average_SPD{i,1} = 0;
  end

  if isempty(Group1.VA_Pos_95{i,1})
    Group1.VA_Pos_95{i,1} = 0;
    Group1.Distance{i,1} = 0;
  end

end

Group1.Velocity = Group1.Velocity(cellfun(@(x) ~isequal(x, 0), Group1.Velocity));
Group1.Average_SPD = Group1.Average_SPD(cellfun(@(x) ~isequal(x, 0), Group1.
    Average_SPD));
Group1.VA_Pos_95 = Group1.VA_Pos_95(cellfun(@(x) ~isequal(x, 0), Group1.VA_Pos_95))
```

```
    ;
Group1.Distance = Group1.Distance(cellfun(@(x) ~isequal(x, 0), Group1.Distance));


for i = 1:length(Group2.Velocity)

  if isempty(Group2.Velocity{i,1})
    Group2.Velocity{i,1} = 0;
  end

  if isempty(Group2.Average_SPD{i,1})
    Group2.Average_SPD{i,1} = 0;
  end

  if isempty(Group2.VA_Pos_95{i,1})
    Group2.VA_Pos_95{i,1} = 0;
    Group2.Distance{i,1} = 0;
  end

end
Group2.Velocity = Group2.Velocity(cellfun(@(x) ~isequal(x, 0), Group2.Velocity));
Group2.Average_SPD = Group2.Average_SPD(cellfun(@(x) ~isequal(x, 0), Group2.
    Average_SPD));
Group2.VA_Pos_95 = Group2.VA_Pos_95(cellfun(@(x) ~isequal(x, 0), Group2.VA_Pos_95))
    ;
Group2.Distance = Group2.Distance(cellfun(@(x) ~isequal(x, 0), Group2.Distance));

        ......
        ......
        ......
        ......

for i = 1:length(Group22.Velocity)

  if isempty(Group22.Velocity{i,1})
    Group22.Velocity{i,1} = 0;
    Group22.Average_SPD{i,1} = 0;
    Group22.VA_Pos_95{i,1} = 0;
    Group22.Distance{i,1} = 0;
  end

end

Group22.Velocity = Group22.Velocity(cellfun(@(x) ~isequal(x, 0), Group22.Velocity))
    ;
Group22.Average_SPD = Group22.Average_SPD(cellfun(@(x) ~isequal(x, 0), Group22.
    Average_SPD));
Group22.VA_Pos_95 = Group22.VA_Pos_95(cellfun(@(x) ~isequal(x, 0), Group22.
    VA_Pos_95));
Group22.Distance = Group22.Distance(cellfun(@(x) ~isequal(x, 0), Group22.Distance))
    ;
```

8.1.3 D-Optimal design

The data set to begin D-optimal design is stored as 'Grouped_Trips_final.mat'. This file contains 3 structure groups named Urban, Rural, and Motorway. Each structure contains an array of structure groups containing cell arrays corresponding to each micro trip. The cell arrays present in each structure group are Velocity, Average speed, VA_{pos95}, Distance, Acceleration, Positive acceleration, Total relative positive acceleration (TRPA), Relative positive acceleration (RPA), and Area under the curve. A brief script to deploy D-optimal design and generate a driving by utilizing the dataset is presented below.

```
%% To deploy D-Optimal design

% D- Optimal design Urban section

for u = 1:11
    if u <= 10

a1 = Urban.Groups.Group1.Velocity;
a2 = Urban.Groups.Group5.Velocity;
a3 = Urban.Groups.Group10.Velocity;
a4 = Urban.Groups.Group11.Velocity;
a5 = Urban.Groups.Group13.Velocity;
a6 = Urban.Groups.Group17.Velocity;
a7 = Urban.Groups.Group14.Velocity;
a8 = Urban.Groups.Group15.Velocity;
a9 = Urban.Groups.Group19.Velocity;
a10 = Urban.Groups.Group22.Velocity;

Urban_Velocity = vertcat(a1,a2,a3,a4,a5,a6,a7,a8,a9,a10);

clear a1 a2 a3 a4 a5 a6 a7 a8 a9 a10

a1 = Urban.Groups.Group1.Average_SPD;
a2 = Urban.Groups.Group5.Average_SPD;
a3 = Urban.Groups.Group10.Average_SPD;
a4 = Urban.Groups.Group11.Average_SPD;
a5 = Urban.Groups.Group13.Average_SPD;
a6 = Urban.Groups.Group17.Average_SPD;
a7 = Urban.Groups.Group14.Average_SPD;
a8 = Urban.Groups.Group15.Average_SPD;
a9 = Urban.Groups.Group19.Average_SPD;
a10 = Urban.Groups.Group22.Average_SPD;

D = vertcat(a1,a2,a3,a4,a5,a6,a7,a8,a9,a10);

clear a1 a2 a3 a4 a5 a6 a7 a8 a9 a10

a1 = Urban.Groups.Group1.VA_Pos_95;
a2 = Urban.Groups.Group5.VA_Pos_95;
a3 = Urban.Groups.Group10.VA_Pos_95;
a4 = Urban.Groups.Group11.VA_Pos_95;
a5 = Urban.Groups.Group13.VA_Pos_95;
a6 = Urban.Groups.Group17.VA_Pos_95;
a7 = Urban.Groups.Group14.VA_Pos_95;
a8 = Urban.Groups.Group15.VA_Pos_95;
a9 = Urban.Groups.Group19.VA_Pos_95;
a10 = Urban.Groups.Group22.VA_Pos_95;

E = vertcat(a1,a2,a3,a4,a5,a6,a7,a8,a9,a10);

clear a1 a2 a3 a4 a5 a6 a7 a8 a9 a10

a1 = Urban.Groups.Group1.RPA;
a2 = Urban.Groups.Group5.RPA;
a3 = Urban.Groups.Group10.RPA;
a4 = Urban.Groups.Group11.RPA;
```

```
a5 = Urban.Groups.Group13.RPA;
a6 = Urban.Groups.Group17.RPA;
a7 = Urban.Groups.Group14.RPA;
a8 = Urban.Groups.Group15.RPA;
a9 = Urban.Groups.Group19.RPA;
a10 = Urban.Groups.Group22.RPA;

F = vertcat(a1,a2,a3,a4,a5,a6,a7,a8,a9,a10);

clear a1 a2 a3 a4 a5 a6 a7 a8 a9 a10

a1 = Urban.Groups.Group1.Distance;
a2 = Urban.Groups.Group5.Distance;
a3 = Urban.Groups.Group10.Distance;
a4 = Urban.Groups.Group11.Distance;
a5 = Urban.Groups.Group13.Distance;
a6 = Urban.Groups.Group17.Distance;
a7 = Urban.Groups.Group14.Distance;
a8 = Urban.Groups.Group15.Distance;
a9 = Urban.Groups.Group19.Distance;
a10 = Urban.Groups.Group22.Distance;

G = vertcat(a1,a2,a3,a4,a5,a6,a7,a8,a9,a10);

clear a1 a2 a3 a4 a5 a6 a7 a8 a9 a10

for i = 1:length(Urban_Velocity)

H(i,1) = max(size(Urban_Velocity{i,1}));

end

Total_Design = zeros(max(size(D)),5);

Total_Design(:,1) = cell2mat(D);
Total_Design(:,2) = cell2mat(E);
Total_Design(:,3) = cell2mat(F);
Total_Design(:,4) = cell2mat(G);
Total_Design(:,5) = H;

clear D E F G H

% Setting Optimal design constraints

for i = 1:length(Total_Design)

    if Total_Design(i,1) >= 15 && Total_Design(i,1) <= 40 && Total_Design(i,2) <=
        18.7 &&...
            Total_Design(i,3) >= 0.13 && Total_Design(i,4) <= 30 && Total_Design(i
            ,4) <= 240

        T(i,1) = 1;

    else  T(i,1) = 0;

    end
end

Valid_Design = zeros(max(size(Total_Design)),5);

for i = 1:length(T)

    if T(i,1) == 1

        Valid_Design(i,:)= Total_Design(i,:);

    else Valid_Design(i) = 0;

    end
end
```

```
for i = 1:length(T)

    if T(i,1) == 1

        Urban_Velocity{i,1}= Urban_Velocity{i,1};

    else Urban_Velocity{i,1} = 0;

    end
end

% To remove Zeros

Urban.Valid_Design = Valid_Design(any(Valid_Design,2),:);

Urban.Urban_Velocity_Valid = Urban_Velocity(cellfun(@(x) ~isequal(x, 0),
    Urban_Velocity));

clear T Total_Design Urban_Velocity Valid_Design

% Deploying Candexch (Candidate Exchange Method)

clear C R Test_Constraints Test

C = [ones(size(Urban.Valid_Design,1),1) Urban.Valid_Design Urban.Valid_Design.^5];

R = candexch(C,7);

close all

% Test constraints

for i = 1:max(size(R))

    Urban.Optimal_Constraints(i,:) = Urban.Valid_Design(R(i),:);

end

for i = 1:max(size(R))

    Urban.Urban_trips_Final{i,1} = Urban.Urban_Velocity_Valid{R(i),1};

end

% D- Optimal design Rural section

a1 = Rural.Groups.Group2.Velocity;
a2 = Rural.Groups.Group3.Velocity;
a3 = Rural.Groups.Group4.Velocity;
a4 = Rural.Groups.Group6.Velocity;
a5 = Rural.Groups.Group7.Velocity;
a6 = Rural.Groups.Group8.Velocity;
a7 = Rural.Groups.Group9.Velocity;
a8 = Rural.Groups.Group12.Velocity;
a9 = Rural.Groups.Group16.Velocity;
a10 = Rural.Groups.Group18.Velocity;
a11 = Rural.Groups.Group21.Velocity;

Rural_Velocity = vertcat(a1,a2,a3,a4,a5,a6,a7,a8,a9,a10,a11);

clear a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 a11

a1 = Rural.Groups.Group2.Average_SPD;
a2 = Rural.Groups.Group3.Average_SPD;
a3 = Rural.Groups.Group4.Average_SPD;
a4 = Rural.Groups.Group6.Average_SPD;
a5 = Rural.Groups.Group7.Average_SPD;
a6 = Rural.Groups.Group8.Average_SPD;
```

```
a7 = Rural.Groups.Group9.Average_SPD;
a8 = Rural.Groups.Group12.Average_SPD;
a9 = Rural.Groups.Group16.Average_SPD;
a10 = Rural.Groups.Group18.Average_SPD;
a11 = Rural.Groups.Group21.Average_SPD;

D = vertcat(a1,a2,a3,a4,a5,a6,a7,a8,a9,a10,a11);

clear a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 a11

a1 = Rural.Groups.Group2.VA_Pos_95;
a2 = Rural.Groups.Group3.VA_Pos_95;
a3 = Rural.Groups.Group4.VA_Pos_95;
a4 = Rural.Groups.Group6.VA_Pos_95;
a5 = Rural.Groups.Group7.VA_Pos_95;
a6 = Rural.Groups.Group8.VA_Pos_95;
a7 = Rural.Groups.Group9.VA_Pos_95;
a8 = Rural.Groups.Group12.VA_Pos_95;
a9 = Rural.Groups.Group16.VA_Pos_95;
a10 = Rural.Groups.Group18.VA_Pos_95;
a11 = Rural.Groups.Group21.VA_Pos_95;

E = vertcat(a1,a2,a3,a4,a5,a6,a7,a8,a9,a10,a11);

clear a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 a11

a1 = Rural.Groups.Group2.RPA;
a2 = Rural.Groups.Group3.RPA;
a3 = Rural.Groups.Group4.RPA;
a4 = Rural.Groups.Group6.RPA;
a5 = Rural.Groups.Group7.RPA;
a6 = Rural.Groups.Group8.RPA;
a7 = Rural.Groups.Group9.RPA;
a8 = Rural.Groups.Group12.RPA;
a9 = Rural.Groups.Group16.RPA;
a10 = Rural.Groups.Group18.RPA;
a11 = Rural.Groups.Group21.RPA;

F = vertcat(a1,a2,a3,a4,a5,a6,a7,a8,a9,a10,a11);

clear a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 a11

a1 = Rural.Groups.Group2.Distance;
a2 = Rural.Groups.Group3.Distance;
a3 = Rural.Groups.Group4.Distance;
a4 = Rural.Groups.Group6.Distance;
a5 = Rural.Groups.Group7.Distance;
a6 = Rural.Groups.Group8.Distance;
a7 = Rural.Groups.Group9.Distance;
a8 = Rural.Groups.Group12.Distance;
a9 = Rural.Groups.Group16.Distance;
a10 = Rural.Groups.Group18.Distance;
a11 = Rural.Groups.Group21.Distance;

G = vertcat(a1,a2,a3,a4,a5,a6,a7,a8,a9,a10,a11);

clear a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 a11

for i = 1:length(Rural_Velocity)

H(i,1) = max(size(Rural_Velocity{i,1}));

end

Total_Design = zeros(max(size(D)),5);

Total_Design(:,1) = cell2mat(D);
Total_Design(:,2) = cell2mat(E);
Total_Design(:,3) = cell2mat(F);
Total_Design(:,4) = cell2mat(G);
```

```
Total_Design(:,5) = H;

clear D E F G H

% Setting Optimal design constraints

for i = 1:length(Total_Design)

    if Total_Design(i,1) > 70 && Total_Design(i,1) < 90 && Total_Design(i,2) <=
        24.3 &&...
            Total_Design(i,3) >= 0.06 && Total_Design(i,4) <= 45 && Total_Design(i
                ,4) >= 20 && ...
            Total_Design(i,5) >= 26*60 && Total_Design(i,5) <= 38*60

        T(i,1) = 1;

    else  T(i,1) = 0;

    end

end

Valid_Design = zeros(max(size(Total_Design)),5);


for i = 1:length(T)

    if T(i,1) == 1

        Valid_Design(i,:)= Total_Design(i,:);

    else Valid_Design(i) = 0;

    end
end


for i = 1:length(T)

    if T(i,1) == 1

        Rural_Velocity{i,1}= Rural_Velocity{i,1};

    else Rural_Velocity{i,1} = 0;

    end
end

% To remove Zeros

Rural.Valid_Design = Valid_Design(any(Valid_Design,2),:);

Rural.Rural_Velocity_Valid = Rural_Velocity(cellfun(@(x) ~isequal(x, 0),
    Rural_Velocity));

clear T Total_Design Rural_Velocity Valid_Design Optimal_Constraints

% Deploying Candexch (Candidate Exchange Method)

clear C R

C = [ones(size(Rural.Valid_Design,1),1) Rural.Valid_Design Rural.Valid_Design.^4];

R = candexch(C,18);

close all

% Test constraints

for i = 1:max(size(R))
```

```

    Rural.Optimal_Constraints(i,:) = Rural.Valid_Design(R(i),:);

end

for i = 1:max(size(R))

    Rural.Rural_trips_Final{i,1} = Rural.Rural_Velocity_Valid{R(i),1};

end

clear C R

% To select rural trip from the sample

Rural.Select_Cycle = datasample(Rural.Rural_trips_Final,1);

% D- Optimal design motorway section

a1 = Motorway.Groups.Group20.Velocity;

Motorway_Velocity = a1;

clear a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 a11

a1 = Motorway.Groups.Group20.Average_SPD;

D = a1;

clear a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 a11

a1 = Motorway.Groups.Group20.VA_Pos_95;

E = a1;

clear a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 a11

a1 = Motorway.Groups.Group20.RPA;

F = a1;

clear a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 a11

a1 = Motorway.Groups.Group20.Distance;

G = a1;

clear a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 a11

for i = 1:length(Motorway_Velocity)

H(i,1) = max(size(Motorway_Velocity{i,1}));

end

Total_Design = zeros(max(size(D)),5);

Total_Design(:,1) = cell2mat(D);
Total_Design(:,2) = cell2mat(E);
Total_Design(:,3) = cell2mat(F);
Total_Design(:,4) = cell2mat(G);
Total_Design(:,5) = H;

clear D E F G H

% Setting Optimal design constraints

for i = 1:length(Total_Design)

    if Total_Design(i,1) >= 100 && Total_Design(i,2) <= 26.6 &&...
```

```
                Total_Design(i,3) >= 0.03 && Total_Design(i,5) >= 25*60 && Total_Design
                    (i,5) <= 50*60

            T(i,1) = 1;

        else  T(i,1) = 0;

        end
end

Valid_Design = zeros(max(size(Total_Design)),5);


for i = 1:length(T)

    if T(i,1) == 1

        Valid_Design(i,:)= Total_Design(i,:);

    else Valid_Design(i) = 0;

    end
end


for i = 1:length(T)

    if T(i,1) == 1

        Motorway_Velocity{i,1}= Motorway_Velocity{i,1};

    else Motorway_Velocity{i,1} = 0;

    end
end

% To remove Zeros

Motorway.Valid_Design = Valid_Design(any(Valid_Design,2),:);

Motorway.Motorway_Velocity_Valid = Motorway_Velocity(cellfun(@(x) ~isequal(x, 0),
    Motorway_Velocity));

clear T Total_Design Motorway_Velocity Valid_Design Optimal_Constraints

% Deploying Candexch (Candidate Exchange Method)

clear C R

C = [ones(size(Motorway.Valid_Design,1),1) Motorway.Valid_Design Motorway.
    Valid_Design.^2];

R = candexch(C,5);

close all

% Test constraints

for i = 1:max(size(R))

    Motorway.Optimal_Constraints(i,:) = Motorway.Valid_Design(R(i),:);

end

for i = 1:max(size(R))

    Motorway.Motorway_trips_Final{i,1} = Motorway.Motorway_Velocity_Valid{R(i),1};

end

```

```
clear C R

% To select Motorway trip from the sample

Motorway.Select_Cycle = datasample(Motorway.Motorway_trips_Final,1);

% To create a complete driving cycle on each iteration

Driving_Cycle{u,1} = vertcat(Urban.Urban_trips_Final,Rural.Select_Cycle,Motorway.
    Select_Cycle);

u = u+1;
    else
        break
    end
end

% Joining cycle code

% To find the total duration

for i = 1:length(Driving_Cycle)
    for j = 1:length(Driving_Cycle{i,1})

    X{i,1}(j,1) = (max(size(Driving_Cycle{i,1}{j,1})));

    end

end

for i = 1:length(X)
    Y{i,1} = sum(X{i,1});
end

% Join the cycles

clc
clear i
Driving_Cycle_Total = cell(10,1);

j = 1;
for i = 1:length(Driving_Cycle_Total)

    for i = 1:length(Driving_Cycle)
    Driving_Cycle_Total{i,1} = vertcat(Driving_Cycle{i, 1}{j,1}, Driving_Cycle{i,
        1}{j+1,1}, Driving_Cycle{i, 1}{j+2,1},...
        Driving_Cycle{i, 1}{j+3,1}, Driving_Cycle{i, 1}{j+4,1}, Driving_Cycle{i,
            1}{j+5,1}, Driving_Cycle{i, 1}{j+6,1},...
        Driving_Cycle{i, 1}{j+7,1}, Driving_Cycle{i, 1}{j+8,1});

    end


end
```

8.1.4 RDE qualified driving cycles

The RDE qualified driving cycles are the ultimate results of this thesis work. They are stored in the 'Driving_Cycle_Total.mat' file. This data file contains an array of driving cycles, that could be used for future work.

8.1.5 Data fit for simulation in Simulink model

A simulation model named 'ConventionalModel.slx' is created for simulating the created driving cycle for determining fuel consumption. The simulink model required 4 inputs namely, 'V_z', 'D_z', 'T_z', and 'G_z'. These inputs are obtained by using a MATLAB script, provided in this section.

```
%% To fit data for simulink

for j = 1:length(Driving_Cycle_Total)

V_z = (Driving_Cycle_Total{j,1})/3.6;

for i = 1:length(V_z)

 if i == max(size(V_z))

      D_z(i,1) = 0;

 else

      D_z(i,1) =  V_z(i,1) - V_z(i+1,1);

 end

end

for i = 1:length(V_z)

 if V_z(i,1)*3.6 >= 0 && V_z(i,1)*3.6 < 15

      G_z(i,1) = 1;

 else if V_z(i,1)*3.6 >= 15 && V_z(i,1)*3.6 < 45

      G_z(i,1) = 2;

 else if V_z(i,1)*3.6 >= 45 && V_z(i,1)*3.6 < 70

      G_z(i,1) = 3;

 else if V_z(i,1)*3.6 >= 70 && V_z(i,1)*3.6 < 105

      G_z(i,1) = 4;

      else if V_z(i,1)*3.6 >= 105

      G_z(i,1) = 5;

          end
      end
      end
      end
 end
end

T_z = (1:max(size(V_z)))';

save(['/Users/chinna/Documents/Thesis/Final Code/QSS_TB_2018b/Data/DrivingCycles/
    Europe/Cycle' num2str(j) '.mat'],'V_z','D_z','G_z','T_z')
fprintf('Saved cycle_%d ',j);

clear T_z V_z G_z D_z

end
```

DEPARTMENT OF MECHANICS ANS MARITIME SCIENCES
DIVISION OF COMBUSTION AND PROPULSION SYSTEM
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY