Institutionen för systemteknik
Department of Electrical Engineering

Examensarbete

Driving Cycle Generation Using Statistical Analysis and Markov Chains

Examensarbete utfört i Fordonssystem
vid Tekniska högskolan vid Linköpings universitet
av

Emil Torp och Patrik Önnegren

LiTH-ISY-EX--13/4670--SE

Linköping 2013

Linköpings universitet
TEKNISKA HÖGSKOLAN

Department of Electrical Engineering
Linköpings universitet
SE-581 83 Linköping, Sweden

Linköpings tekniska högskola
Linköpings universitet
581 83 Linköping

Driving Cycle Generation Using Statistical Analysis and Markov Chains

Examensarbete utfört i Fordonssystem
vid Tekniska högskolan vid Linköpings universitet
av

Emil Torp och Patrik Önnegren

LiTH-ISY-EX--13/4670--SE

Handledare: **Peter Nyberg**
ISY, Linköpings universitet

Examinator: **Erik Frisk**
ISY, Linköpings universitet

Linköping, 13 juni 2013

Avdelning, Institution
Division, Department

Avdelningen för Fordonssystem
Department of Electrical Engineering
SE-581 83 Linköping

Datum
Date

2013-06-13

Språk
Language

□ Svenska/Swedish

☒ Engelska/English

□ ____________

Rapporttyp
Report category

□ Licentiatavhandling

☒ Examensarbete

□ C-uppsats

□ D-uppsats

□ Övrig rapport

□ ____________

ISBN
—

ISRN
LiTH-ISY-EX--13/4670--SE

Serietitel och serienummer
Title of series, numbering

ISSN
—

URL för elektronisk version

http://www.ep.liu.se

Titel
Title

Körcykelgenerering med statistisk analys och markovkedjor

Driving Cycle Generation Using Statistical Analysis and Markov Chains

Författare
Author

Emil Torp och Patrik Önnegren

Sammanfattning
Abstract

A driving cycle is a velocity profile over time. Driving cycles can be used for environmental classification of cars and to evaluate vehicle performance. The benefit by using stochastic driving cycles instead of predefined driving cycles, i.e. the New European Driving Cycle, is for instance that the risk of cycle beating is reduced. Different methods to generate stochastic driving cycles based on real-world data have been used around the world, but the representativeness of the generated driving cycles has been difficult to ensure.

The possibility to generate stochastic driving cycles that captures specific features from a set of real-world driving cycles is studied. Data from more than 500 real-world trips has been processed and categorized. The driving cycles are merged into several transition probability matrices (TPMs), where each element corresponds to a specific state defined by its velocity and acceleration. The TPMs are used with Markov chain theory to generate stochastic driving cycles. The driving cycles are validated using percentile limits on a set of characteristic variables, that are obtained from statistical analysis of real-world driving cycles.

The distribution of the generated driving cycles is investigated and compared to real-world driving cycles distribution. The generated driving cycles proves to represent the original set of real-world driving cycles in terms of key variables determined through statistical analysis.

Four different methods are used to determine which statistical variables that describes the features of the provided driving cycles. Two of the methods uses regression analysis. Hierarchical clustering of statistical variables is proposed as a third alternative, and the last method combines the cluster analysis with the regression analysis.

The entire process is automated and a graphical user interface is developed in Matlab to facilitate the use of the software.

Nyckelord
Keywords

drive cycle, mean tractive force, cluster analysis, regression analysis, percentile validation, transition probability matrix

Abstract

A driving cycle is a velocity profile over time. Driving cycles can be used for environmental classification of cars and to evaluate vehicle performance. The benefit by using stochastic driving cycles instead of predefined driving cycles, i.e. the New European Driving Cycle, is for instance that the risk of cycle beating is reduced. Different methods to generate stochastic driving cycles based on real-world data have been used around the world, but the representativeness of the generated driving cycles has been difficult to ensure.

The possibility to generate stochastic driving cycles that captures specific features from a set of real-world driving cycles is studied. Data from more than 500 real-world trips has been processed and categorized. The driving cycles are merged into several transition probability matrices (TPMs), where each element corresponds to a specific state defined by its velocity and acceleration. The TPMs are used with Markov chain theory to generate stochastic driving cycles. The driving cycles are validated using percentile limits on a set of characteristic variables, that are obtained from statistical analysis of real-world driving cycles.

The distribution of the generated driving cycles is investigated and compared to real-world driving cycles distribution. The generated driving cycles proves to represent the original set of real-world driving cycles in terms of key variables determined through statistical analysis.

Four different methods are used to determine which statistical variables that describes the features of the provided driving cycles. Two of the methods uses regression analysis. Hierarchical clustering of statistical variables is proposed as a third alternative, and the last method combines the cluster analysis with the regression analysis.

The entire process is automated and a graphical user interface is developed in Matlab to facilitate the use of the software.

Sammanfattning

En körcykel är en beskriving av hur hastigheten för ett fordon ändras under en körning. Körcykler används bland annat till att miljöklassa bilar och för att utvärdera fordonsprestanda. Olika metoder för att generera stokastiska körcykler baserade på verklig data har använts runt om i världen, men det har varit svårt att efterlikna naturliga körcykler.

Möjligheten att generera stokastiska körcykler som representerar en uppsättning naturliga körcykler studeras. Data från över 500 körcykler bearbetas och kategoriseras. Dessa används för att skapa överergångsmatriser där varje element motsvarar ett visst tillstånd, med hastighet och acceleration som tillståndsvariabler. Matrisen tillsammans med teorin om Markovkedjor används för att generera stokastiska körcykler. De genererade körcyklerna valideras med hjälp percentilgränser för ett antal karaktäristiska variabler som beräknats för de naturliga körcyklerna.

Hastighets- och accelerationsfördelningen hos de genererade körcyklerna studeras och jämförs med de naturliga körcyklerna för att säkerställa att de är representativa. Statistiska egenskaper jämfördes och de genererade körcyklerna visade sig likna den ursprungliga uppsättningen körcykler.

Fyra olika metoder används för att bestämma vilka statistiska variabler som beskriver de naturliga körcyklerna. Två av metoderna använder regressionsanalys. Hierarkisk klustring av statistiska variabler föreslås som ett tredje alternativ. Den sista metoden kombinerar klusteranalysen med regressionsanalysen.

Hela processen är automatiserad och ett grafiskt användargränssnitt har utvecklats i Matlab för att underlätta användningen av programmet.

Acknowledgments

We would like to thank the division of Vehicular Systems for giving us the opportunity to carry out this master thesis by providing relevant data and support. A special thanks go to Erik Frisk and Peter Nyberg who have provided feedback and relevant expertise through the thesis.

We would also like to thank those who have proofread the report, you know who you are and it has been much appreciated.

Linköping, June 2013
Emil Torp and Patrik Önnegren

Contents

Notation

Nomenclature	
a	Acceleration
a_{res}	Acceleration resolution
A_f	Vehicle frontal area
C_d	Aerodynamic drag coefficient
C_r	Rolling resistance coefficient
d	Driving distance
m_v	Vehicle mass
T_s	Sample time
v	Velocity
$\overline{v}_{pos}$	Mean positive velocity
v_{res}	Velocity resolution

Abbreviations	
FPC	First principal component
GUI	Graphical user interface
LASSO	Least absolute shrinkage and selection operator
MTF	Mean tractive force
NaN	Not a number
NEDC	New European Driving Cycle
PCA	Principal component analysis
SAFD	Speed-acceleration frequency distribution
TPM	Transition probability matrix
UDDS	Urban Dynamometer Driving Schedule

1 Introduction

There are multiple predefined driving cycles used for environmental classification of vehicles and in the vehicle product development process in the world today. Two well known examples are the New European Driving Cycle (NEDC), seen in Figure 1.1, and the Urban Dynamometer Driving Schedule (UDDS). Development of some driving cycles are summarized in André [1996].

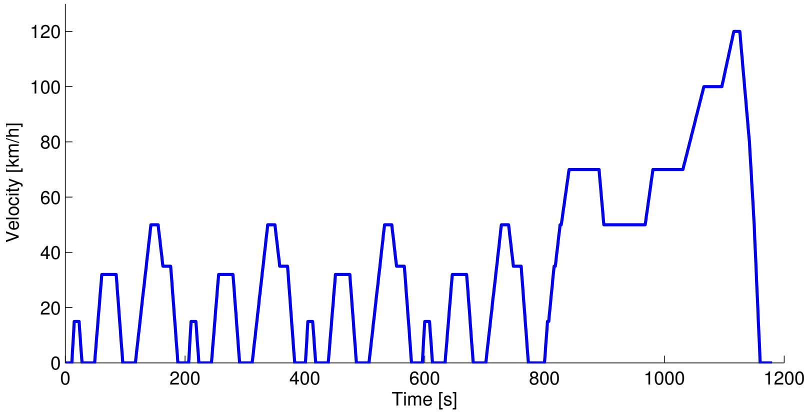


Figure 1.1: *The New European Driving Cycle (NEDC).*

However, a problem when testing vehicles with predefined driving cycles is that the risk for cycle beating is increased. This means that vehicle parameters affecting emissions and fuel consumption can be optimized for a specific cycle [Kågeson, 1998, Schwarzer et al., 2010]. But there are no guarantee that the vehicle will perform in the same way when driven in real-world traffic. A natural driving cycle is usually more aggressive than the standardized cycles [Fellah et al., 2009]. It is therefore necessary to test vehicles with natural diving cycles in order to obtain more relevant results. An example of a real-world driving cycle is seen in Figure 1.2, where it is clear that the acceleration varies more than in Figure 1.1.

The risk for cycle beating is significantly decreased when vehicles are tested against several different driving cycles. However, obtaining driving cycles through measurements can be costly, and there is much to gain if they can be generated automatically.

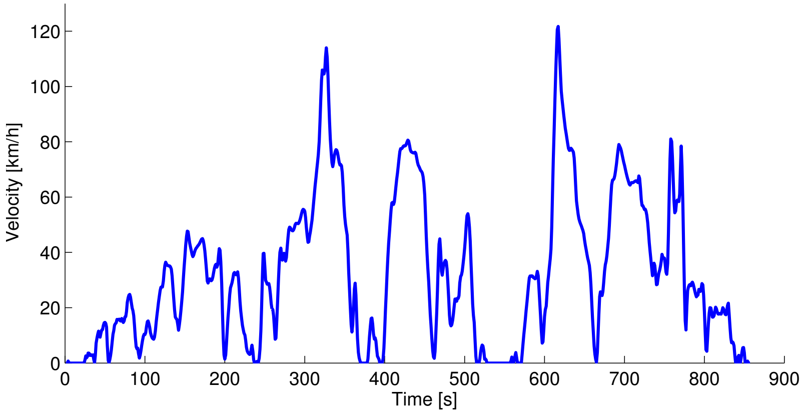


Figure 1.2: *Example of a natural driving cycle.*

A common method for construction of driving cycles is to randomly append driving segments, where a segment is a driving sequence between two stops [André, 1996]. Lin and Niemeier [2002] describes the method as a combination of 'microtrips'. A problem when randomly appending microtrips is that no consideration for differentiation in modal events (e.g. cruise, idle, acceleration and deceleration) within a segment is made [Lin and Niemeier, 2002]. Furthermore, the method has problems achieving the desired driving cycle duration [André, 1996].

Lin and Niemeier [2002] used a stochastic process to assemble small snippets of data until certain statistical criteria were met. Snippets are based on which modal event they belong to and is extracted from the measured driving cycles. The main difference between snippets and segments is that a snippet is not constrained to be a driving segment between two stops. However, due to the size of these snippets, it is still difficult to achieve the desired driving distance and at the same time obtain driving cycles that are representative for natural driving [Lee and Filipi, 2011].

Another way would be to assemble single velocity and acceleration states instead of entire snippets. One option is to generate driving cycles by using Markov chains, as described in Lee and Filipi [2011]. This includes extracting information from a database of real-world traffic, analyzing the data and to generate driving cycles from a stochastic process.

The objective of this thesis is to use the Markov chain approach when applicable and at the same time propose improvements to the algorithm.

1.1 Problem formulation

This thesis addresses the problem of synthesizing driving cycles that are representative for real-word driving cycles. All important characteristic features from a specific type of driving shall be captured in a single stochastic driving cycle. This means that the specific features must be determined, and that the generated driving cycles must be validated.

Since the process is composed of many complex steps, which can be performed in many ways, it is thus desirable to automate the process as much as possible in order to obtain a structured method.

1.2 Limitations

Since the measured driving data can be formatted differently in different studies, it is not possible to write software that handles every type of data. This is solved by defining a specification on how input data has to be formatted. The specification can be seen in Section 3.2.

Some of the statistical analysis rely on that a sufficiently large amount of real-world driving cycles are available. Most of the driving cycles available have either a very short driving distance or a low average speed. For this reason, it is hard to assure validity in driving cycles generated from categories with a long driving distance, or a high average speed.

1.3 Approach

As described above, this thesis is based on the work by Lee and Filipi [2011]. The proposed method is used as a foundation and certain parts are developed even further.

An important part of the thesis is to study what describes a representative natural driving cycle. It is investigated through statistical analysis and the results are used to validate generated driving cycles.

The methods are implemented in Matlab and an accompanying graphical user interface (GUI) is developed.

1.4 Thesis contributions

Unlike previous work in the field, this thesis propose to use a unique set of validation variables for each categorization set of real-world driving cycles in order to ensure the representativeness of the synthesized driving cycles. The characteristics of a driving cycle depends on the type of driving and the validation must therefore be different.

Another contribution is the proposed cluster analysis method to determine what represents a set of driving cycles. It uses principal component analysis to calculate the similarities between the statistical variables in each category and determines a subset from 27 proposed characteristics, depending on the real-world driving cycles. Unlike the regression analysis method proposed by Lee and Filipi [2011], the cluster analysis is well suited to be automated.

1.5 Thesis outline

Chapter 2 describes the theory used for the analysis and generation of driving cycles. The methods used to analyze provided data are described in Chapter 3. Chapter 4 contains descriptions on how driving cycles are generated and the validation of those. The results are presented in Chapter 5. The last part, Chapter 6, contains discussion of the results and Chapter 7 presents the conclusions.

2

Theory

Different methods to determine representative variables for sets of real-world driving cycles is presented. The described methods are based on linear regression analysis and hierarchical clustering of variables. The Markov chain theory to generate new driving cycles is presented in Section 2.4.

2.1 Multiple linear regression

Assume that a response variable y is observed n times together with a set of explanatory variables $[x_1, x_2, \ldots, x_j, \ldots, x_m]$, e.g. calculated for n real-world driving cycles. (The explanatory variables are also referred to as regressors.) The objective of a regression analysis is to explain as much of the variation in the response variable as possible using linear combinations of the explanatory variables, namely estimate the coefficients in the linear model

$$y = \beta_1 + \beta_2 x_1 + \beta_3 x_2 + ... + \beta_{m+1} x_m + \epsilon, \tag{2.1}$$

where ϵ is a random normally distributed stochastic variable. The estimated model can be used to predict future values of the response variable. The set of optimal equation coefficients $\hat{\boldsymbol{\beta}} = [\hat{\beta}_1, \hat{\beta}_2, \ldots . \hat{\beta}_{m+1}]$ are estimated as

$$\hat{\boldsymbol{\beta}} = \arg\min \{Q_{ols}(\boldsymbol{\beta})\} = \arg\min \left\{ \sum_{i=1}^{n} (y_i - \beta_1 - \beta_2 x_{1,i} - ... - \beta_{m+1} x_{m,i})^2 \right\}. \tag{2.2}$$

The coefficients are optimal in the sense that they minimize the squared model residuals $\boldsymbol{\epsilon} = [\epsilon_1, \epsilon_2, \ldots, \epsilon_n]^T$, as shown in [Enqvist, 2007, p. 21].

The solution is found by taking the partial derivatives of Q_{ols}, $\frac{\partial Q_{ols}}{\partial \beta_k}$, for $k = 1, 2, \ldots, m+1$. By setting each partial derivative equal to zero, a linear equation system is formed with the unknown parameters $\boldsymbol{\beta}$. Overall, the system contains $m+1$ equations and $m+1$ unknown variables and can be written on the matrix form

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}. \tag{2.3}$$

The estimated coefficients, $\hat{\boldsymbol{\beta}}$, can be derived as

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1} \mathbf{X}^T\mathbf{Y}, \tag{2.4}$$

if $det(\mathbf{X}^T\mathbf{X}) \neq 0$ and the matrices $\mathbf{Y}$, $\boldsymbol{\beta}$, $\boldsymbol{\epsilon}$ and $\mathbf{X}$ are defined as

$$\mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{m+1} \end{bmatrix}, \quad \boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}, \tag{2.5}$$

$$\mathbf{X} = \begin{bmatrix} 1 & x_{1,1} & x_{2,1} & \ldots & x_{m.1} \\ 1 & x_{1,2} & x_{2,2} & \ldots & x_{m,2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{1,n} & x_{2,n} & \ldots & x_{m,n} \end{bmatrix}. \tag{2.6}$$

If the estimated residuals,

$$\hat{\boldsymbol{\epsilon}} = \mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}}, \tag{2.7}$$

are independent identically distributed (i.i.d.) random variables, $\hat{\boldsymbol{\epsilon}} \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_n)$, then the regression model predicts the response variable. The estimated coefficients $\hat{\boldsymbol{\beta}}$ are in that case normally distributed as well, $\hat{\boldsymbol{\beta}} \sim \mathcal{N}(\boldsymbol{\beta}, \sigma^2(\mathbf{X}^T\mathbf{X})^{-1})$.

2.1.1 T-test

A T-test can be performed in order to determine whether an explanatory variable actually contributes to the estimation of the response variable. The standard error of the regression, s^2 is calculated as

$$s^2 = \frac{\hat{\boldsymbol{\epsilon}}^T \hat{\boldsymbol{\epsilon}}}{n-m-1} \sim \frac{\sigma^2}{n-m-1} \chi^2(n-m-1), \tag{2.8}$$

and since s^2 is a sum of the independent squared normally distributed random variables $\hat{\epsilon_i}$, it is χ^2-distributed. The distribution relationship can be written as

$$\frac{\hat{\boldsymbol{\epsilon}}^T \hat{\boldsymbol{\epsilon}}}{\sigma^2} \sim \chi^2(n-m-1). \tag{2.9}$$

The estimated standard error of the regression is used to estimate the standard error for each model coefficient β_j. The formula is given by

$$\hat{\sigma}_{\beta_j} = \sqrt{s^2 \left(\mathbf{X}^T\mathbf{X}\right)^{-1}_{jj}}, \tag{2.10}$$

where $(\mathbf{X}^T\mathbf{X})^{-1}_{jj}$ refers to the j:th element on the diagonal of the covariance matrix, $(\mathbf{X}^T\mathbf{X})^{-1}$.

If a coefficient $\beta_j = 0$, the fraction between the estimated coefficient and the coefficient standard error, also called the coefficient t-value, is T-distributed with $n-m-1$ degrees of freedom. This can be seen by rearranging the terms as

$$t_{\beta_j} = \frac{\hat{\beta}_j}{\hat{\sigma}_{\beta_j}} = \frac{\left[\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{Y}\right]_j}{\sqrt{\frac{\hat{\boldsymbol{\epsilon}}^T\hat{\boldsymbol{\epsilon}}}{n-m-1}\left(\mathbf{X}^T\mathbf{X}\right)^{-1}_{jj}}} \sim \frac{\mathbf{N}(0,1)}{\sqrt{\frac{\chi^2(n-m-1)}{(n-m-1)}}} = T\left(n-m-1\right). \tag{2.11}$$

The result is a fraction between a normal distribution and the square root of a χ^2-distribution divided by its degrees of freedom. This is the definition of a T-distribution [Blom et al., 2005, p. 293]. Generally, the T-distribution origins from the normal distribution and as the degrees of freedom grows towards infinity, the T-distribution approaches the $N(0,1)$-distribution as illustrated in Figure 2.1.

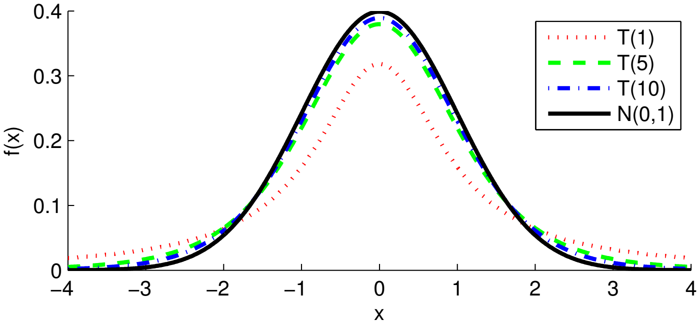


Figure 2.1: *Probability density function for T-distributions with various degrees of freedom compared to the $N(0,1)$-distribution.*

The T-distribution is useful to determine whether a regression coefficient $\beta_j = 0$, in other words, whether the explanatory variable x_{j-1} affects the response variable at all [Enqvist, 2007, pp. 27–32]. The coefficient p-value

$$p_{\beta_j} = P\left(|t| \geq |t_{\beta_j}| \,\middle|\, \beta_j = 0\right), \tag{2.12}$$

is a measure of how far out in the T-distribution the coefficient t-value lies.

For instance, if $p_{\beta_j} = 0.049$, it is possible to state that $\beta_j \neq 0$ at a confidence level of 95 %. Figure 2.2 shows the 95th percentile for the $T(5)$-distribution. If the t-value is above ≈ 2, the p-value is lower than 0.1 (since the distribution is symmetric) and it is possible to state that the coefficient is non-zero at a confidence level of 90 %.

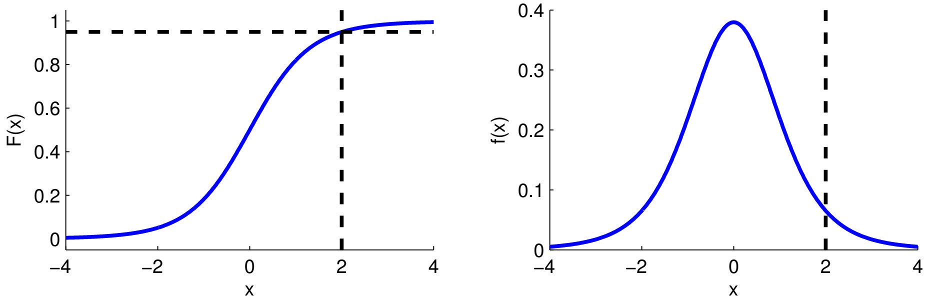


Figure 2.2: Cumulative distribution function and probability density function for a T-distribution with 5 degrees of freedom. The 95th percentile is dashed in both plots.

It is important to remember that these conclusions are only valid under the assumption that the residuals are normally distributed. Otherwise, the t- and p-values gives no information about the coefficients β_j since the coefficient standard errors will not be T-distributed.

However, if the residuals are normally distributed, a T-test can be used to reduce the number of explanatory variables by removing the variable most likely to have a coefficient equal to zero. This can be done by removing the variable with the largest p-value and perform the least squares regression with the remaining variables as proposed by Lee and Filipi [2011].

2.1.2 Measure of regression fit

The R^2-statistic is a measure of how well the estimated regression equation fits the observed data. The value represents how much of the variations in the response variable y, that can be explained by the regression model [Renaud and Victoria-Feser, 2010].

The formula is given by

$$R^2 = \frac{Q_{regr}}{Q_{tot}} = \frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} = 1 - \frac{Q_{res}}{Q_{tot}}, \tag{2.13}$$

where $\bar{y}$ is the mean value of the observed response variable and $\hat{y}_i$ is the response variable derived from the estimated model. Q_{tot} is the total amount of variations in the observed response variable, Q_{regr} is the variations accounted for by the regression model, and Q_{res} describes the variations that the model is unable to capture.

If R^2 is large ($\gtrsim 0.9$), the regression model with the estimated coefficients β_j explains most of the variations in the response variable and the equation shows a good fit to the observed data.

R^2 is useful when a stepwise regression is performed. A limit can be set and the removal of explanatory variables can be stopped when the model no longer shows a large enough fit (when R^2 becomes smaller than a predefined limit).

A property of R^2 is that it always grows if more explanatory variables are added to the model. This fact in combination with a small sample size can cause overfitting of the data, and more variables than necessary can be included in the model. This can however be compensated for by using

$$R^2_{adj} = 1 - (1 - R^2)\frac{n-1}{n-m-1}, \tag{2.14}$$

where n is the sample size and m is the number of explanatory variables in the model equation (not counting the constant term) [Harrell, 2001, p. 91].

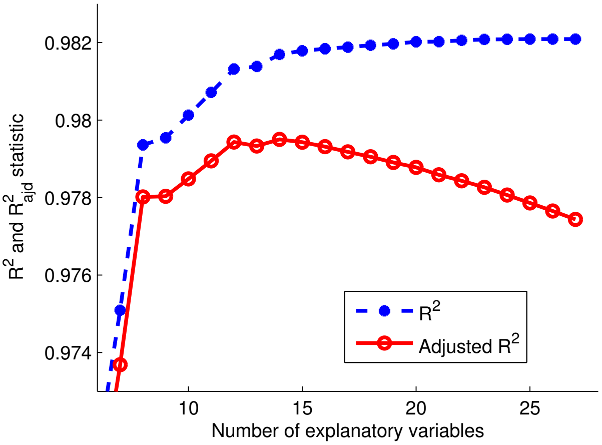


Figure 2.3: *Regression analysis statistics for different number of regressors.*

The R^2_{adj}-statistic compensates for the number of explanatory variables in the equation and unlike the R^2-statistic, it can decrease if too many variables are included in the model. Figure 2.3 shows both the statistics from a regression analysis containing $n = 132$ samples and different number of explanatory variables.

2.1.3 LASSO regression

In order to obtain a regression model with fewer explanatory variables than the ordinary least squares method described above, it is possible to add an extra constraint to the minimization problem. The objective with the least absolute shrinkage and selection operator method (LASSO) is to reduce the number of explanatory variables while at the same time obtain a model that can predict the response variable. These specific properties are obtained by penalizing the non-zero model coefficients β_j by using a regularization parameter λ and the L^1-norm of the model coefficients [Tibshirani, 1996]

$$\hat{\beta} = \operatorname{argmin}\left\{\sum_{i=1}^{n}(y_i - \beta_1 - \beta_2 x_{1,i} - \ldots - \beta_{m+1}x_{m,i})^2 + \lambda\sum_{j=2}^{m+1}|\beta_j|\right\}. \tag{2.15}$$

Solving (2.15) leads to more coefficients, β_j, being zero than in the ordinary least squares case. The larger the regularization parameter λ is set, the more coefficients will be equal to zero in the final model.

Since the LASSO regression already has the property of not including unnecessary variables, the regression fit can be measured using the ordinary R^2-statistic instead of the adjusted one mentioned above.

2.2 Hierarchical clustering of variables

In order to reduce the number of variables that describes a set of data, a hierarchical clustering method can be used to group closely related variables together. The concept of hierarchical clustering is well described by Everitt et al. [2011] and is illustrated in Figure 2.4.

There are many different methods to determine how closely related two variables are, e.g. correlation or euclidean distance. The distance between two clusters can also be defined in many ways, i.e. the average distance between the variables in the two clusters or simply the closest distance from a variable in the first cluster to a variable in the second cluster [Everitt et al., 2011]. In this thesis, the distance between two clusters (or variables) i and j is defined as

$$d_{i,j} = 1 - PC_1, \tag{2.16}$$

where PC_1 is the amount of within cluster variations accounted for by the first

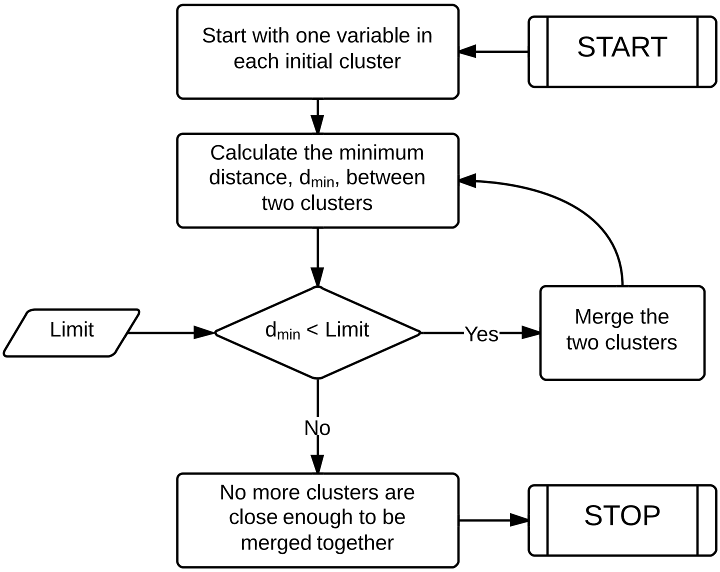

Start with one variable in each initial cluster
START
Calculate the minimum distance, d_{min}, between two clusters
Limit
d_{min} < Limit
Yes
Merge the two clusters
No
No more clusters are close enough to be merged together
STOP

Figure 2.4: *Hierarchical agglomerative clustering procedure.*

principal component (FPC). The FPC is obtained from a principal component analysis (PCA) on the variables in the combined cluster (see Section 2.2.1).

The clustering method used in this thesis is an agglomerative clustering method, meaning that all variables are assigned to an initial cluster. The clusters are grouped together as long as the distance between them falls below a predefined limit.

2.2.1 Principal component analysis

Principal component analysis (PCA) is a method to determine how orthogonal a set of variables are. By changing the base from the original variables to an orthogonal base consisting of principal components, it is possible to see in how many dimensions the variables actually varies, and especially, how one-dimensional the variations are. For further information about the concept and a complete theory, see Jolliffe [2002].

Assume that m variables have been observed n times. The variables then forms a matrix $\mathbf{X}$ where each row corresponds to a variable, where mean values of each variable is removed and each variable is scaled with its standard deviation. By performing a singular value decomposition of $\mathbf{X}$ as described by Jolliffe [2002, pp. 44–46], three new matrices are obtained. In other words, $\mathbf{X}$ is factorized as

$$\mathbf{X}_{mxn} = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_m \end{bmatrix} = \mathbf{U}_{mxm}\boldsymbol{\Sigma}_{mxn}\mathbf{V}^T_{nxn}, \tag{2.17}$$

where **U** is a unitary matrix with columns forming an orthonormal basis for **X**. The amount of variance explained by the principal components PC_i can be derived from the singular values σ_i in the diagonal matrix Σ using

$$PC_i = \frac{\sigma_i^2}{\sum_{j=1}^m \sigma_j^2}. \tag{2.18}$$

Specially, PC_1 is the amount of variance explained by the FPC and is a measure of the linearity in the set.

Figure 2.5 illustrates the procedure with two variables. The left picture shows *mean positive acceleration* and *acceleration standard deviation* derived from 447 driving cycles. The variables are correlated and by performing a PCA, it is possible to see that the FPC explains 96 % of the total variations in the original variables. The figure to the right shows the variables in the principal component base.

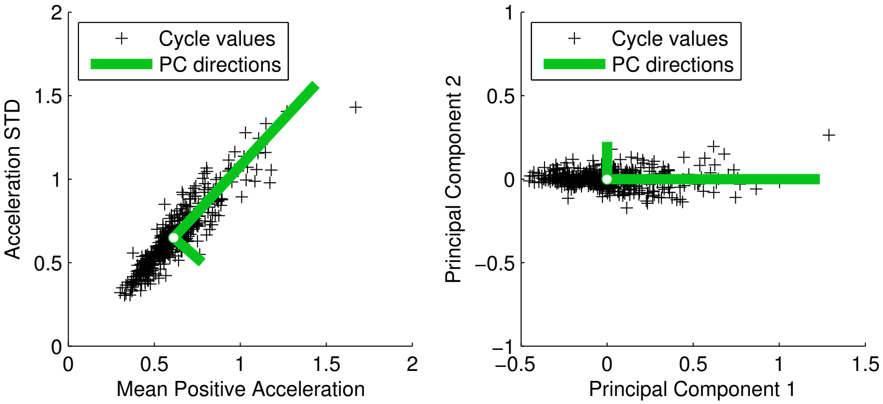


Figure 2.5: Two-dimensional principal component base change. (The length of the direction lines is not proportional to the amount of variance explained by the principal components.)

2.3 Mean tractive force

A measure of how a driving cycle affects the vehicle is the *mean tractive force* (MTF). The use of the MTF as representative response was proposed by Lee and Filipi [2011] (also called *specific energy at wheels*), and the definition given here can be found in Guzzella and Sciarretta [2007].

The MTF is defined as the mean positive force at the wheels necessary for a vehicle to follow the driving cycle. This means that only time instances when the powertrain provides power to the vehicle ($i \in trac$) are taken into account. The definition of MTF is given by

$$\overline{F}_{trac} = \frac{1}{x_{tot}} \int_{i \in trac} F(t) \cdot v(t) dt, \tag{2.19}$$

where $F(t)$ is the sum of all forces acting at the wheels, $v(t)$ is the velocity and x_{tot} is the driving cycle distance. The contributions to $F(t)$ are modeled and (2.19) is rewritten as

$$\overline{F}_{trac} = \overline{F}_{trac,a} + \overline{F}_{trac,r} + \overline{F}_{trac,m} \tag{2.20}$$

where $\overline{F}_{trac,a}$, $\overline{F}_{trac,r}$ and $\overline{F}_{trac,m}$ are the MTF values of aerodynamic, rolling resistance and acceleration resistance forces acting at the wheels. Forces on the wheels caused by road gradient are neglected when the power demand is calculated. They are each modeled as

$$\overline{F}_{trac,a} = \frac{1}{x_{tot}} \cdot \frac{1}{2} \cdot \rho_a \cdot A_f \cdot C_d \cdot \sum_{i \in trac} \overline{v}_i^3 \cdot T_s \tag{2.21}$$

$$\overline{F}_{trac,r} = \frac{1}{x_{tot}} \cdot m_v \cdot A_f \cdot g \cdot C_r \cdot \sum_{i \in trac} \overline{v}_i \cdot T_s \tag{2.22}$$

$$\overline{F}_{trac,m} = \frac{1}{x_{tot}} \cdot m_v \cdot \sum_{i \in trac} \overline{a}_i \cdot \overline{v}_i \cdot T_s, \tag{2.23}$$

where A_f is the vehicle frontal area, ρ_a is the air density and C_d is the drag coefficient. Furthermore, m_v is the vehicle mass, g is the gravitational constant, C_r is the rolling resistance coefficient, and T_s is the time between velocity samples.

Only samples where the vehicle operates in traction mode ($F(t) > 0$) are considered when the MTF is calculated. Another way to determine if the vehicle is in traction mode is to calculate the coasting velocity

$$v_c(t) = \frac{\beta}{\alpha} \cdot \tan\left\{\arctan\left(\frac{\alpha}{\beta} \cdot v_c(0)\right) - \alpha \cdot \beta \cdot t\right\} \tag{2.24}$$

where α and β are defined as

$$\alpha = \sqrt{\frac{1}{2 \cdot m_v} \cdot \rho_a \cdot A_f \cdot c_d} \tag{2.25}$$

$$\beta = \sqrt{g \cdot c_r}. \tag{2.26}$$

[Guzzella and Sciarretta, 2007]. If a velocity sample v_i in the driving cycle is higher than the coasting velocity $v_c(T_s)$, determined by using $v_c(0) = v_{i-1}$ and $t = T_s$ in (2.24), the vehicle is operating in traction mode in the interval between the samples $i - 1$ and i.

Figure 2.6 illustrates which intervals that are considered in the calculation of the MTF. The white areas indicates that the vehicle operates in braking mode, and therefore does not provide any traction force.

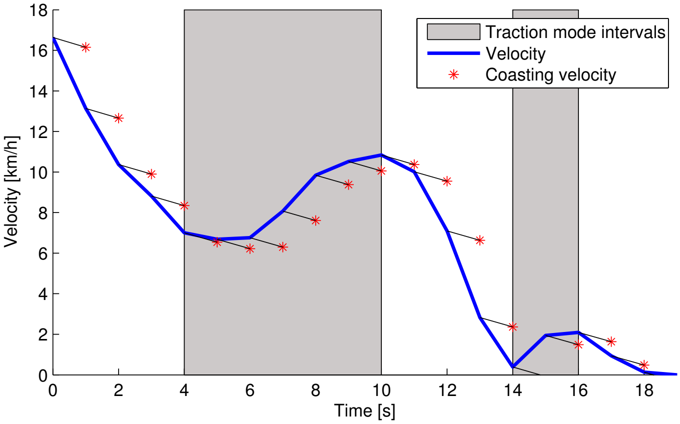


Figure 2.6: *Coasting velocities and traction mode illustration.*

2.4 Markov chain

Markov chain is a mathematical theory used to model a random process. The process is based on the Markov property that the next state, X_{n+1}, depends entirely on the current state, X_n, and not any preceding or following states [Gubner, 2006],

$$P\left(X_{n+1} = x \mid X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n\right) = P\left(X_{n+1} = x \mid X_n = x_n\right). \tag{2.27}$$

The probabilities of reaching a specific state at the next time instance varies depending on the current state. The states, x_i does not necessarily have to be one-dimensional. In this thesis, each state is defined by a two-dimensional vector $[v, a]$, and each combination of the discrete variables v and a corresponds to a specific state, x_i.

The Markov chain used in this thesis is considered stationary since all probabilities are time homogeneous [Gubner, 2006, p. 480]. It is possible to write the one-step transition probability from state x_i to state x_j as

$$p_{ij} = P\left(X_{n+1} = x_j \mid X_n = x_i\right). \tag{2.28}$$

All one-step state probabilities can be arranged in a matrix, called the transition probability matrix (TPM), where each element contains the probabilities for every other state to be the next in the chain. One important note is that all probabilities for leaving a state (including the probability of staying in the same state) must sum up to one. This is mathematically described as

$$\sum_j p_{ij} = \sum_j P\left(X_{n+1} = x_j \mid X_n = x_i\right) = 1, \quad \forall i. \tag{2.29}$$

3

Data Analysis

The following chapter describes how real-world data is processed and analyzed to later be used in the generation of transition probability matrices (TPMs).

3.1 Preprocessing

All driving data used to generate new stochastic driving cycles are provided by Volvo Cars in Gothenburg. A total of nine vehicles have logged speed and torque for several weeks during the summer of 2012.

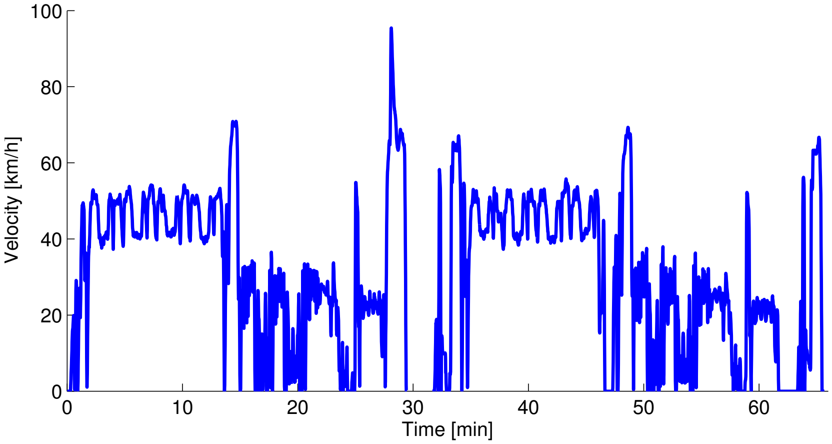
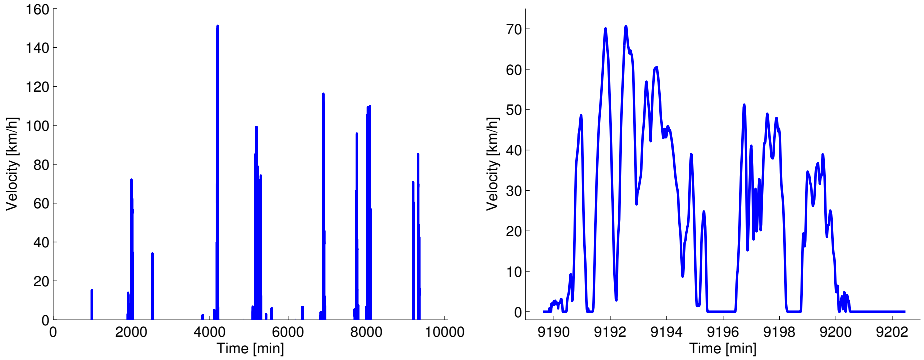


Figure 3.1: *Example of non-natural driving cycles.*

However, only three of the vehicles are assumed to have been driven in normal traffic conditions. The data from the remaining vehicles contains driving patterns with tendencies to be measured on a test track. Repetitive patterns were frequently occurring as can be seen in Figure 3.1.

Since the available data have been logged for entire weeks, as can be seen in Figure 3.2a, there is a need to split each week of data into multiple driving cycles. Each vehicle logged speed and torque while the engine was running. Figure 3.2b shows the velocity profile from one of the measured driving cycles.



(a) *Data from a week.* ***(b)*** *Zoomed in view of a driving cycle.*

Figure 3.2: *Examples of given real-world data.*

In Figure 3.2b, it is also possible to see the idle periods at the beginning and end of each driving cycle. These extra measurements do not describe the driving cycle when the vehicle is active and are therefore removed.

There are also some driving cycles that have unusually long idle periods. This was initially considered to be stops due to traffic lights. But when the duration of the idle periods were studied further, it was clear that a few of the stops could not have come from such scenarios. Figure 3.3 shows a driving cycle that has an idle time for approximately eight minutes between two non-zero velocity intervals. Such a scenario is considered to occur when the vehicle is left running while the driver is away, doing something else. All such events are therefore divided into two separate driving cycles if the stoppage time is longer than 3 minutes.

Some of the available driving cycles did not start and end with a zero velocity measurement. This is considered to be some kind of fault in the data logging process. However, most of these driving cycles have otherwise good measurements so instead of discarding multiple driving cycles, they are trimmed until they start and end with a zero velocity sample.

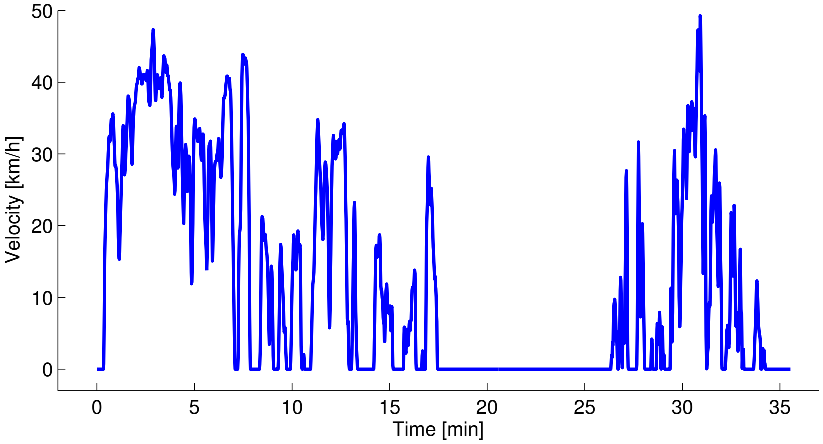


Figure 3.3: *Approximately eight minutes pause in the middle of a driving cycle.*

3.2 Data input specification

A specification for how all input data must be constructed is defined. Each driving cycle has to be a Matlab structure with fields according to Table 3.1. Furthermore, the structures has to be chained together in an array.

Table 3.1: *Data input specification.*

Field	Type	Explanation	Unit
`velocity`	Vector	Sampled velocity	[km/h]
`Ts`	Scalar	Sample time	[s]
`carCharacteristics`	Structure	(optional) See Table B.2.	-

The field `carCharacteristics` in Table 3.1 is an optional structure, that is mainly used when calculating the response variable in the regression analysis. Default values are used when the field does not exist in the input data. The specifications for `carCharacteristics` are given in Table 3.2, as well as default values for each parameter. Furthermore, all input driving cycles must have identical sample times.

It is recommended to use a sample time of 1 sample per second or faster. If a longer sample time is used, there is a risk of losing information about the changes in the driving cycles. If a shorter sample time is used, it will increase the complexity of the driving cycles and will not, in most cases, give any additional information. It will also result in a slower generation process since more samples has to be generated to achieve the desired driving cycle duration.

Table 3.2: *`carCharacteristics` input specification.*

Field	Type	Explanation	Default value	Unit
`mv`	Scalar	Vehicle mass	1600	[kg]
`Cd`	Scalar	Aerodynamic drag coefficient	0.4	[-]
`Cr`	Scalar	Rolling resistance coefficient	0.013	[-]
`Af`	Scalar	Vehicle frontal area	2.15	[m^2]

3.3 Data processing

All incoming data go through a processing stage according to

1. Calculate acceleration.
2. Averaging velocity.
3. Discretize data.
4. Extract statistical variables.

The following sections will explain each step further. Step 1 and 2 are calculated as in [Guzzella and Sciarretta, 2007, pp.23–24]. Step 3 and 4 are done as in [Lee and Filipi, 2011].

3.3.1 Accleration

The acceleration is approximated by calculating the velocity change in each interval

$$a(t) = \bar{a}_i = \frac{v_i - v_{i-1}}{3.6 \cdot T_s} \qquad \forall t \in [t_{i-1}, t_i). \tag{3.1}$$

3.3.2 Velocity

The average velocity between measurements is calculated as

$$v(t) = \bar{v}_i = \frac{v_i + v_{i-1}}{2} \qquad \forall t \in [t_{i-1}, t_i). \tag{3.2}$$

The velocity and acceleration measurements are defined in the same time intervals, which is important for upcoming calculations.

3.3.3 Discretization

To be able to generate a useful TPM, described in Section 4.1, there is a need to discretize all measurements. Averaged velocities and accelerations are therefore rounded to the closest neighboring discretization step as

$$\bar{v}_i^d \in \{0,\ v_{res},\ 2v_{res},\ \ldots\} \tag{3.3}$$

$$\bar{a}_i^d \in \{\ldots,\ -a_{res},\ 0,\ a_{res},\ 2a_{res},\ \ldots\}, \tag{3.4}$$

where the default values for the discretization resolution is shown in Table 3.3.

Table 3.3: *Default resolution steps for discretization.*

Type	**Variable**	**Stepsize**
Velocity	v_{res}	1.0 km/h
Accleration	a_{res}	0.2 m/s^2

3.3.4 Statistical analysis

One of the most important steps in the initial processing is the statistical analysis. The values extracted here are later used for data filtering (Section 3.4), representative variable analysis (Section 3.6), and validation (Section 4.3) among others. The variables extracted are presented in Table 3.6 and in Appendix A.

3.4 Data filtering

All real-world driving cycles are by this point processed and they have statistical properties available for further study. A couple of filtering criteria are defined to remove unwanted driving cycles. Data is filtered based on the following aspects

- Mean positive velocity.
- Driving time with positive velocity.

All driving cycles with a mean positive velocity below 10 km/h are removed since they are not considered natural. An example of such a driving cycle can be seen in Figure 3.4.

Driving cycles that have a non-zero velocity for shorter than 60 seconds are also removed. As can be seen in Figure 3.5, the driving time for the entire driving cycle is close to two minutes but the amount of time where the vehicle is driving with a positive velocity is below the limit, and the cycle is therefore removed.

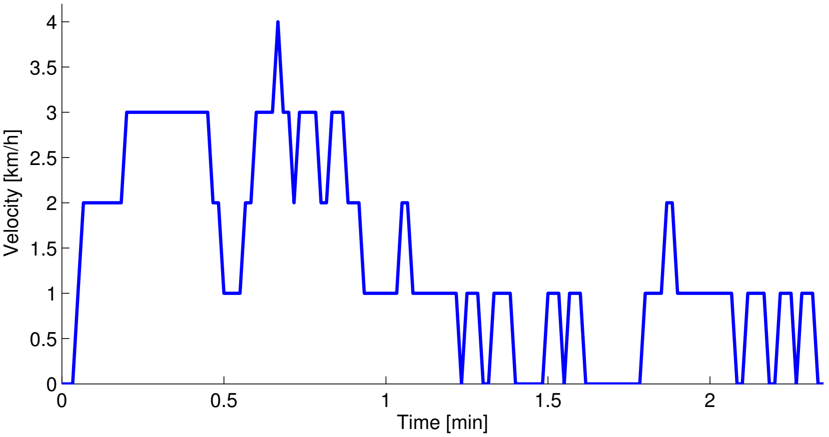


Figure 3.4: *Driving cycle with a mean positive velocity below 10 km/h.*

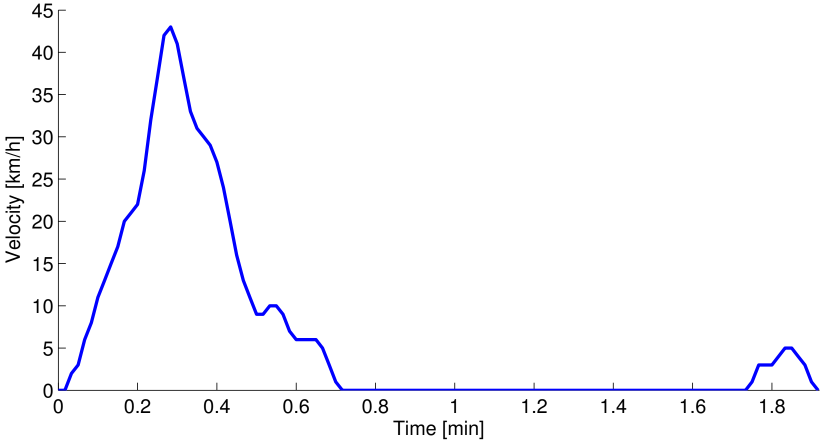


Figure 3.5: *Driving cycle with short timespan at positive velocity.*

3.5 Data categorization

A driving cycle can be categorized into different types, e.g. by distinguishing between driving cycles that are measured while driving in the city and driving cycles measured on the freeway. Since given data have a wide spread of driving types, it is possible to split the set of driving cycles into more specific categories. Categories used in this thesis are based on those defined by Lee and Filipi [2011] and can be seen in Table 3.4 and Table 3.5.

As can be seen in the third column in the tables (number of cycles), there are only three categories that have a substantial amount of data. Most effort is there-

Table 3.4: *Categories based on mean positive velocity.*

Category	Limits [km/h]	# Cycles
Urban	$0 < \bar{v}_{pos} \leq 40$	328
Mixed	$40 < \bar{v}_{pos} \leq 72$	133
Freeway	$72 < \bar{v}_{pos} < \infty$	5

Table 3.5: *Categories based on driving distance.*

Category	Limits [km]	# Cycles
Short	$0 < d \leq 14$	409
Medium	$14 < d \leq 32$	42
Long	$32 < d < \infty$	15

fore focused on these categories since the other ones do not have enough driving cycles to perform a proper statistical analysis.

3.6 Representative variables

Four different methods are implemented that determines a set of representative variables for a set of driving cycles, i.e. driving cycles from a specific category. Each one of the methods is tested on the driving cycles that are categorized as *short*, *urban* and *mixed*. Each method generates a subset of the statistical variables listed in Table 3.6, that may be considered sufficient to describe the characteristics of a driving cycle from the given category. The variables selected are later used to evaluate the representativeness of generated driving cycles.

3.6.1 Iterative regression analysis

The first implemented method is the iterative regression analysis proposed by Lee and Filipi [2011]. The objective is to single out the variables among the 27 proposed ones that explains the response variable, *mean tractive force* (MTF), described in Section 2.3. Unlike the method used by Lee and Filipi [2011], the implementation in this thesis is completely automated.

At first, the mutual correlation between the 27 explanatory variable candidates are examined. This leads to the removal of several variables that shows a strong correlation with another variable/variables. Each of the candidate explanatory variables are compared to the other ones in terms of linear correlation. The linear correlation coefficient between two explanatory variables, $\mathbf{X}_i = [x_{i,1}, ..., x_{i,n}]$ and $\mathbf{X}_j = [x_{j,1}, ..., x_{j,n}]$ observed together for n driving cycles is defined as

$$r_{i,j} = \frac{\operatorname{cov}(\mathbf{X}_i, \mathbf{X}_j)}{\sigma_i \cdot \sigma_j} = \frac{\sum_{k=1}^{N}(x_{i,k} - \overline{x}_i)(x_{j,k} - \overline{x}_j)}{\sqrt{\sum_{k=1}^{N}(x_{i,k} - \overline{x}_i)^2}\sqrt{\sum_{k=1}^{N}(x_{j,k} - \overline{x}_j)^2}}, \tag{3.5}$$

where $\overline{x}_i$ and $\overline{x}_j$ are the observed variable means [Blom et al., 2005].

If two variables show a strong linear correlation, $|r_{i,j}| > 0.75$, one of them is removed. The variable with the largest individual correlation with the response variable, MTF, is kept for the regression analysis as an explanatory variable.

The limit, $|r_{i,j}| > 0.75$, is selected based on visual examinations of the relationships. Figure 3.6 shows two examples of the correlation between candidate explanatory variables. In both cases, the mutual correlation exceeds the limit and one of the variables is removed.

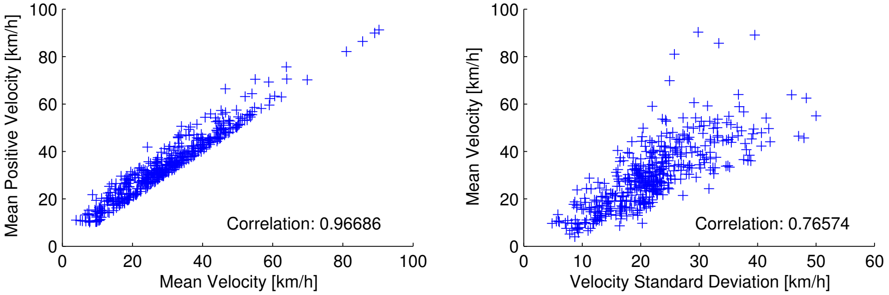


Figure 3.6: *Mutual correlation between explanatory variable candidates.*

A test where exponential correlations were taken into account was also performed. The test gave almost identical results as the linear correlation tests. A decision was therefore made to only use the linear correlations when determining the initial explanatory variables.

When the mutual correlation between the variables has been examined, a stepwise regression analysis is performed in order to determine the smallest set of variables that can be used to explain the driving cycles MTF.

An initial model is estimated using all the remaining variables. In order to further reduce the number of variables in the model, a T-test for each model coefficient is performed. The variable corresponding to the model coefficient with the largest p-value, p_{β_j}, is removed from the set of explanatory variables. The procedure is repeated and an explanatory variable is removed in each step until the model no longer satisfies the adjusted R-square limit, $R^2_{adj} > 0.9$.

The variable removed in the last step is returned to the model when the regression fit falls below the limit. The remaining variables are selected as representative for the driving cycles used in the analysis.

Table 3.6: *Driving cycle characteristics.*

Category	#	Explanatory variable	Unit
Velocity	1	Mean positive velocity	[km/h]
	2	Mean velocity	[km/h]
	3	Maximum velocity	[km/h]
	4	95th percentile maximum velocity	[km/h]
	5	Standard deviation of velocity	[km/h]
Acceleration	6	Mean positive acceleration	[m/s^2]
	7	Mean negative acceleration	[m/s^2]
	8	Positive acceleration time	[s]
	9	Negative acceleration time	[s]
	10	95th percentile maximum acceleration	[m/s^2]
	11	95th percentile minimum acceleration	[m/s^2]
	12	Maximum acceleration	[m/s^2]
	13	Minimum acceleration	[m/s^2]
	14	Standard deviation of acceleration	[m/s^2]
	15	Percentage of driving time under positive acceleration	[%]
	16	Percentage of driving time under negative acceleration	[%]
Distance and time	17	Driving distance	[km]
	18	Driving time	[s]
Driving characteristics	19	Idle time	[s]
	20	Percentage of idle time	[%]
	21	Cruise time	[s]
	22	Percentage of cruise time	[%]
	23	Number of stops	[-]
	24	Number of stops per km	[/km]
	25	Mean specific power	[W/km]
	26	Maximum specific power	[W/km]
	27	Minimum specific power	[W/km]

3.6.2 LASSO regression

To avoid unnecessary number of explanatory variables, another method based on regularized least-squares regression is implemented, namely the LASSO method described in Section 2.1.3. The minimization problem solved to estimate the model coefficients for different λ is given by

$$\hat{\beta} = \arg\min \left\{ \sum_{i=1}^{n} (\mathbf{Y} - \mathbf{X}\beta)^2 + \lambda \sum_{j=2}^{m+1} |\beta_j| \right\}. \tag{3.6}$$

The minimization problem is essentially the same as in the linear stepwise regression. The only difference is that the L^1-norm of the coefficient vector is included with the regularization coefficient λ.

Since a large λ-value results in many model coefficients β_j being zero, the coefficient value is lowered until the limit $R^2 > 0.9$ is fulfilled. In order to avoid an unnecessary high number of representative variables, the lowering of λ also stops if the number of non-zero β_j becomes larger than ten.

3.6.3 Hierarchical clustering of variables

A variable clustering method is implemented to determine a minimal subset of representative variables from the 27 variables listed in Table 3.6. The theory of clustering variables can be found in Section 2.2.

Unlike the iterative regression method, MTF is not used as a representative response. Instead, the implemented clustering method intends to explain the variations in all statistical variables.

Mean values are removed from each variable since it is the variation in the variables that is to be investigated. They are also scaled with their standard deviations to avoid that high-valued variables affect the result more than low-valued ones. For example, *maximum velocity* is normally much larger than *number of stops*.

The clustering procedure starts with each statistical variable in a separate cluster. At each iteration, the clusters closest to each other are combined as long as the distance between them is small enough.

The implemented distance measure between clusters makes use of the principal component analysis (PCA), described in Section 2.2.1. The distance between two clusters i and j, $d_{i,j}$ is obtained from a PCA on the variables in the combined cluster.

An upper triangular distance matrix D is calculated at each iteration before combining the closest clusters. D have the form

$$D = \begin{pmatrix} 0 & d_{1,2} & \cdots & d_{1,n-1} & d_{1,n} \\ 0 & 0 & \cdots & d_{2,n-1} & d_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & d_{n-1,n} \\ 0 & 0 & \cdots & 0 & 0 \end{pmatrix}, \tag{3.7}$$

where n is the number of clusters at the current iteration. The two clusters corresponding to the smallest non-zero value in D are combined in the subsequent step.

When the smallest non-zero value in the distance matrix no longer falls below the limit

$$d_{i,j} = 1 - PC_{1,ij} < 0.25, \tag{3.8}$$

the grouping stops and the final clusters are determined by the set of clusters at that point. The limit used is determined through several tests and visual examination of the clusters obtained in different categories.

Figure 3.7 illustrates the procedure of clustering variables for the driving cycles in the category *short*. The statistical variables are listed on the x-axis and the distance between the combined clusters are shown on the y-axis. The dotted line corresponds to the limit after which no more clusters are combined. (Due to the fact that the combined cluster variables no longer shows the one-dimensional behavior that is required in order to group them together.)

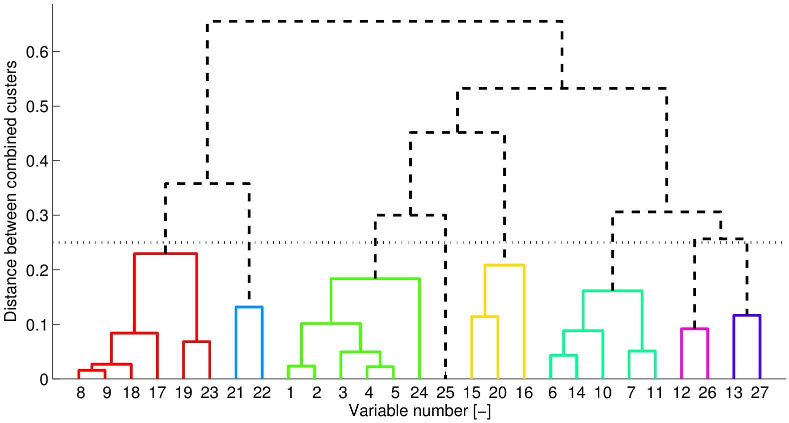


Figure 3.7: *Resulting dendrogram from the clustering of the variables in the category short.*

When the final clusters have been determined, one variable from each cluster is selected as the cluster representative and as a final representative variable in the validation of the generated driving cycles. The chosen variable from each cluster is the one with the largest distance to its closest neighboring cluster. The procedure to select a cluster representative is described in Figure 3.8 and explained further below.

The decrease in variations explained by the FPC when a variable, v, is added to a cluster, c, is calculated as

$$\Delta PC_1^{v,c} = PC_1^c - PC_1^{v,c}, \tag{3.9}$$

where PC_1^c is the variance explained by the FPC in cluster c, and $PC_1^{v,c}$ is the variance explained by the FPC when the variable v is added to the cluster.

Assume that there are k final clusters with various number of variables in them. Each variable $i \in [1, 2, \ldots, m_j]$ in cluster $j \in [1, 2, \ldots, k]$ is assigned a value s_i, that is the smallest FPC-decrease when the variable is added to another cluster

$$s_i = \min\left\{ \Delta PC_1^{i,c}\colon c \in [1, 2, \ldots, k],\ c \neq j \right\}. \tag{3.10}$$

Every variable in the cluster is compared to all other clusters and the variable selected as representative for cluster j, is the one with the largest s_i, determined by

$$i = \arg\max_i \left\{ s_i : i \in [1, 2, \ldots, m_j] \right\}. \tag{3.11}$$

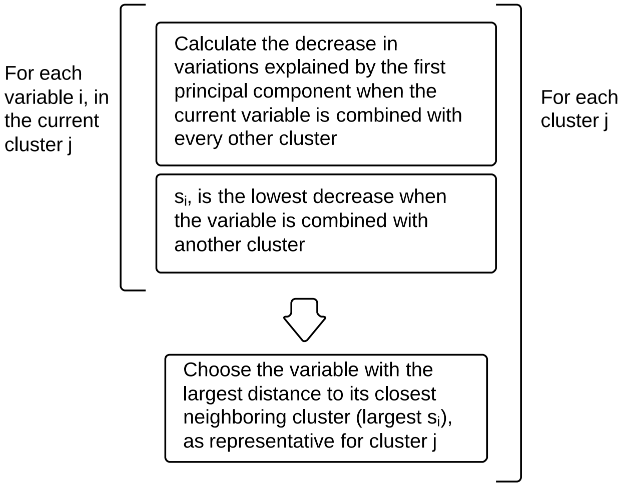

For each variable i, in the current cluster j
Calculate the decrease in variations explained by the first principal component when the current variable is combined with every other cluster
s_i, is the lowest decrease when the variable is combined with another cluster
Choose the variable with the largest distance to its closest neighboring cluster (largest s_i), as representative for cluster j
For each cluster j

Figure 3.8: *Procedure to choose a cluster representative in each final cluster.*

3.6.4 Combined regression and clustering

A fourth method is implemented to avoid the use of the response variable MTF when determining the initial variables for the stepwise regression analysis. Unlike the iterative regression analysis method, the procedure intends to remove correlated variables by using cluster analysis and then determine the final representative variables by using the method described in Section 3.6.1.

Instead of using the limit on PC_1 from Section 3.6.3, a lower one is used, namely that PC_1 in each cluster must exceed 0.9.

The resulting clusters obtained from the analysis in the category *short* can be seen in Figure 3.9. A total of 16 clusters, for which one representative is chosen using the same method as in Section 3.6.3, are nominated as explanatory variables for the regression analysis.

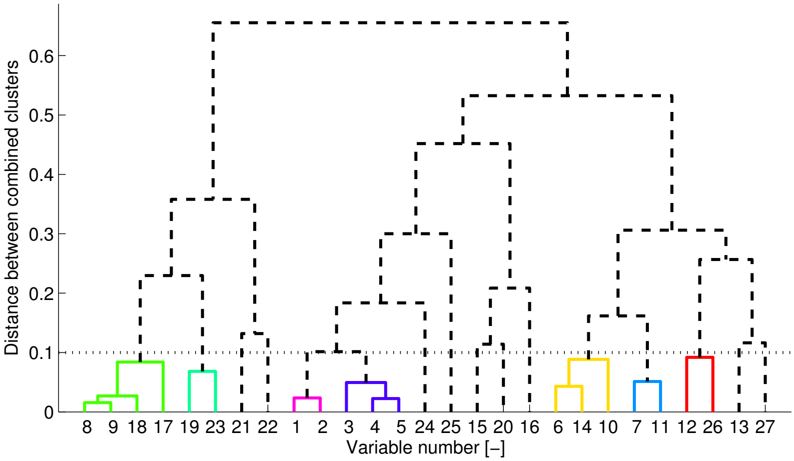


Figure 3.9: *Clustering dendrogram from the combined regression and clustering analysis.*

4 Driving Cycle Generation

Generation of driving cycles includes the process of generating both transition probability matrices (TPMs), described in Section 4.1, as well as driving cycles, described in Section 4.2. Section 4.3 goes into details on how the driving cycle validation works. The chapter also contains specifications on how data is specified within Matlab.

4.1 TPM construction

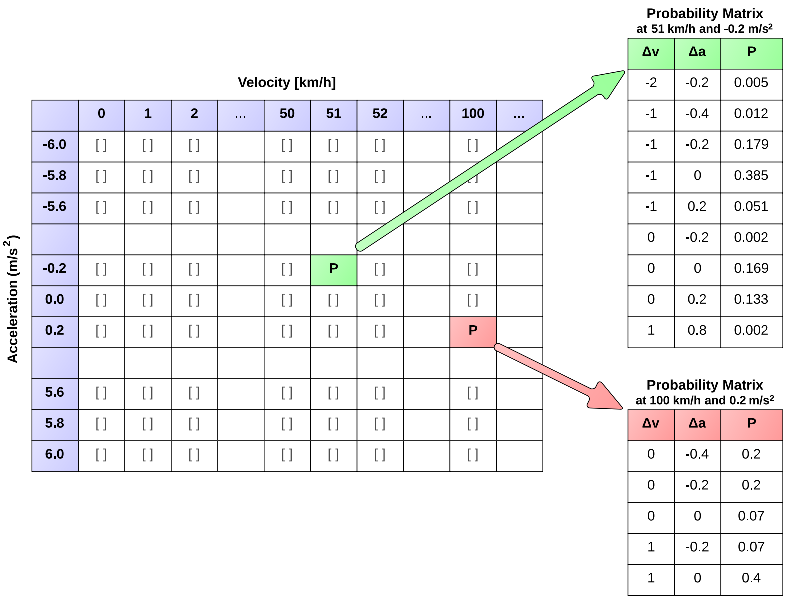
As described in Section 2.4, the TPM matrix contains probabilities to transition from one state to another state. Each state is defined by the state variables, velocity and acceleration. To increase the readability, the TPM is constructed as a large matrix containing smaller sub-matrices, as can be seen in Figure 4.1. Each state corresponds to a specific element in the TPM, that contains a smaller matrix with the transition probabilities.

The size of the large matrix is determined by the maximum velocity and the absolute maximum acceleration combined with the resolutions for velocity and acceleration. The number of rows, n_r, and columns, n_c, are calculated as

$$n_r = 2 \cdot \frac{|a|_{max}}{a_{res}} + 1 \tag{4.1}$$

$$n_c = \frac{v_{max}}{v_{res}} + 1. \tag{4.2}$$

For example, if the maximum velocity is 180 km/h, and the resolution is 1 km/h,



	0	1	2	...	50	51	52	...	100	...
-6.0	[]	[]	[]		[]	[]	[]		[]	
-5.8	[]	[]	[]		[]	[]	[]		[]	
-5.6	[]	[]	[]		[]	[]	[]		[]	
-0.2	[]	[]	[]		[]	P	[]		[]	
0.0	[]	[]	[]		[]	[]	[]		[]	
0.2	[]	[]	[]		[]	[]	[]		P	
5.6	[]	[]	[]		[]	[]	[]		[]	
5.8	[]	[]	[]		[]	[]	[]		[]	
6.0	[]	[]	[]		[]	[]	[]		[]	

Probability Matrix
at 51 km/h and -0.2 m/s^2

Δv	Δa	P
-2	-0.2	0.005
-1	-0.4	0.012
-1	-0.2	0.179
-1	0	0.385
-1	0.2	0.051
0	-0.2	0.002
0	0	0.169
0	0.2	0.133
1	0.8	0.002

Probability Matrix
at 100 km/h and 0.2 m/s^2

Δv	Δa	P
0	-0.4	0.2
0	-0.2	0.2
0	0	0.07
1	-0.2	0.07
1	0	0.4

Figure 4.1: *Example of a TPM.*

there will be 181 columns. If the absolute maximum acceleration is 8.2 m/s^2, and the resolution is 0.2 m/s^2, there will be 83 rows in the TPM. The first column in the matrix corresponds to zero velocity and the middle row to zero acceleration.

When the size of the large matrix is defined, it is possible to generate the sub-matrices. This is done by stepping through each input driving cycle and saving each state transition in the correct sub-matrix. A new row is added to the sub-matrix for each time a state is visited, changing the size of the sub-matrix.

When all driving cycles have been sorted into the TPM, there is a need to sort and summarize the sub-matrices. A value of how many times a unique transition has occurred is calculated and the transition probabilities are derived. Example of the final representation of the TPM can be seen in Figure 4.1.

4.1.1 TPM specification

The TPM is constructed as a Matlab structure, since there is a need to store different kinds of data within it. Instead of sending several individual variables between functions, it is possible to just send one structure with all information that is needed. It will also make it easier to store several different settings for the driving cycle generation which will make it possible to reuse the same generated TPM in the future. The specification on how a TPM is configured can be seen in Table 4.1.

Table 4.1: *TPM specification.*

Field	Explanation
`matrix`	The generated probability matrix
`velRes`	Velocity resolution
`accRes`	Acceleration resolution
`Ts`	Sample time
`nrOfCycles`	Number of cycles the TPM is based on
`variableIntervals`	Validation intervals
`statMatrix`	Cycle statistics matrix
`repVariables`	Structure with representative variables
`analysisInfo`	Information from the data analysis

4.2 Driving cycle construction

When a TPM has been created, it is possible to start generating driving cycles. The process starts by calculating the desired driving cycle duration. This is done by calculating the median for all driving cycles that the TPM is based on. This is the driving duration that the process aims for, but it is not the definite duration of the finished driving cycle.

The process, described in Figure 4.2, starts in the idle state (zero velocity and acceleration). The first transition is leaving the idle state and the driving cycle is then built up through random state transitions in the TPM, based on the transition probabilities. Each sub-matrix contains all state transitions available with corresponding probabilities. Two examples of how the sub-matrices are built can be seen in Figure 4.1. The iterative process continues until the desired duration is exceeded at the same time as the end state has a velocity equal to zero.

There is also a desire to have only one zero velocity state at the end of each driving cycle,

$$v(t_{end}) = 0 \tag{4.3}$$

$$v(t_{end} - 1) \neq 0, \tag{4.4}$$

which is obtained by removing all but one zero velocity state from the end. However, this trimming is very rare since it only occurs when the velocity is zero in an interval before and up to the desired duration.

Finally, the driving cycle goes through the validation process described in the next section. If the driving cycle is deemed valid, it is presented to the user. If it

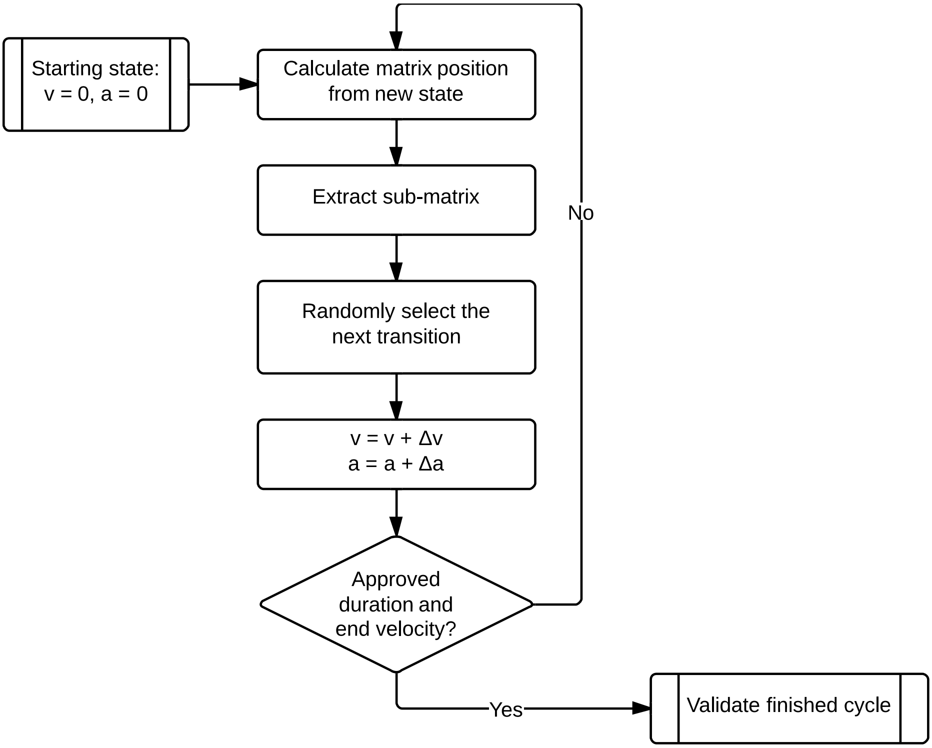


Figure 4.2: *The driving cycle generation process.*

is considered invalid, it is discarded, and a new driving cycle is generated. This continues until a valid driving cycle is found.

4.2.1 Driving cycle specification

The final generated driving cycle is a Matlab structure configured as in Table 4.2. The fields `velocity` and `acceleration` corresponds to the velocity and acceleration profiles obtained from the Markov process. The driving time can be found in the field `duration` and `Ts` is the sample time.

The field `characteristics` is a structure that contains values for all statistical variables, described in Appendix A. The last field, `TPMname`, contains a string with the name of the TPM used to create the driving cycle.

4.3 Validation

Since the generated driving cycles are created from a Markov process, there is no guarantee that they will be good representatives for the chosen data set. It is therefore necessary to validate each generated driving cycle. The validation is per-

Table 4.2: *Driving cycle specification.*

Field	Explanation	Unit
`velocity`	Velocity vector	[km/h]
`acceleration`	Acceleration vector	[m/s^2]
`duration`	Cycle duration	[s]
`Ts`	Sample time	[s]
`characteristics`	Cycle statistics	-
`TPMname`	Name of TPM used	-

formed using the representative variables obtained from the analysis described in Section 3.6.

Initially, the validation method used the average values for all statistical variables derived from the driving cycles in the TPM. These values were compared to the same variables for the generated driving cycles, and the deviation was calculated in percent. However, this method has several problems,

- Variables with a large value get a big validation range.
- Variables with a low value get a small validation range. This could result in validation limits, for which it is impossible to generate an approved driving cycle.
- The variables natural deviations was not taken into consideration.

The percentage validation method was for these reasons replaced with a new type of validation, based on percentiles.

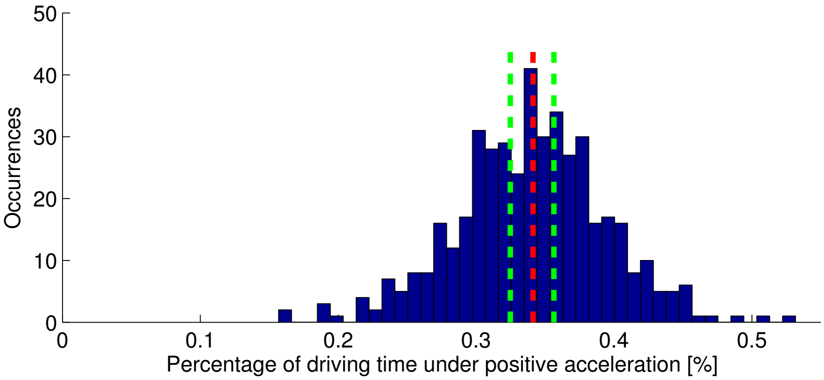


Figure 4.3: *Histogram for a statistical variable with median and 25 % limit presented.*

As an example, the 10th percentile is the value for which 10 % of all the observations falls below. The median value is by that logic found at the 50th percentile. A range is then constructed using this knowledge. If a validation should be done with a 20 % limit, then this is converted to a range from the 40th percentile to the 60th percentile. Another example can be seen in Figure 4.3, where a generated driving cycle is approved if it obtains a value between the validation limits.

Using percentiles solved the problems that occurred with percentage validation. All variables are allowed to be within an interval for which it is possible to approve the generated driving cycles. Variables that have a large variance in the measured driving cycles are also allowed a larger variance in the generated driving cycles. The opposite is true for the variables with narrow distributions.

5 Results

The main results from the process of generating stochastic driving cycles by using the described methods can be summarized in two groups.

First, some of the generated driving cycles are presented and their speed-acceleration frequency distribution (SAFD) is compared to the SAFD of the real-world driving cycles, in order to ensure their representativeness.

Second, results from the four proposed methods to determine representative variables to a set of real-world driving cycles (see Section 3.6) are presented in Section 5.2. The results from the validation of the generated driving cycles are presented in Section 5.3.

5.1 Generated driving cycles

The software, described in Appendix B, can output a valid driving cycle. An example can be seen in Figure 5.1, where the driving cycle has been constructed from the TPM produced from the driving cycles with a driving distance shorter than 14 km, as described in Section 3.5.

It is possible to see some similarities when generating several driving cycles from the same category. They have roughly the same duration and many of the statistical variables are in the same range, even those that the driving cycle was not validated against. This is because some of the statistical variables are related, described further in Section 5.2.

As previously mentioned in Section 3.5, only some of the categories have enough measured driving cycles to generate a TPM that does not have traces of separate driving cycles. When generating a driving cycle with those TPMs, it is often pos-

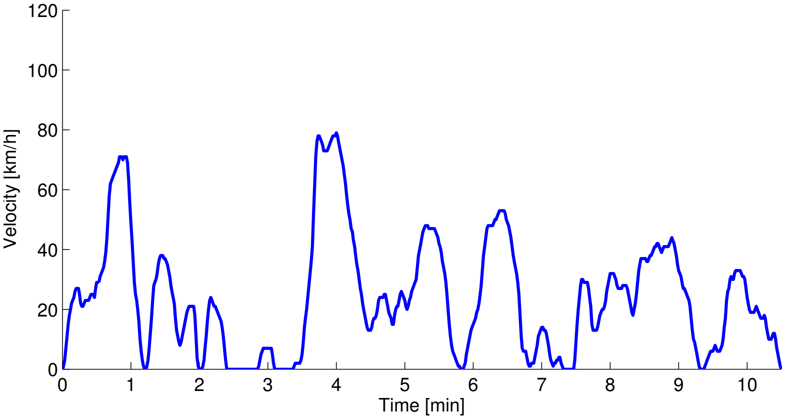


Figure 5.1: *Generated driving cycle from the category short.*

sible to see identical snippets compared to the measured driving cycles. This is due to the fact that some states in the TPM have only one transition available, and that the Markov chain will continue on the same path until the process arrives at a state that has multiple state transitions.

The generated TPMs in the categories *short*, *urban* and *mixed* contains a large number of real-world driving cycles. Examples of generated driving cycles in those categories can be seen in Figures 5.1 - 5.3.

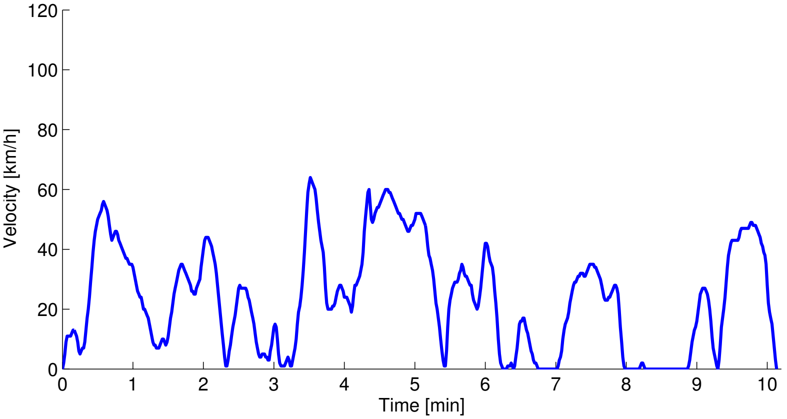


Figure 5.2: *Generated driving cycle from the category urban.*

It is possible to generate driving cycles from the other categories (*median*, *long* and *freeway*), but since the analysis is affected by the small data sets, there is no guarantee that the generated driving cycles are representative for their respective category.

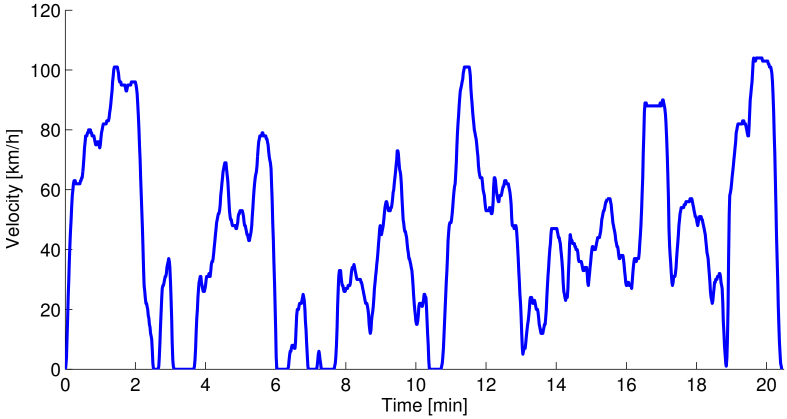


Figure 5.3: *Generated driving cycle from the category mixed.*

5.1.1 Distribution of generated driving cycles

When generating driving cycles, there is a desire that the output should have the same speed-acceleration frequency distribution (SAFD) as the input data. A test is performed, where one million driving cycles from the category *short* is generated and the SAFD is compared to the SAFD of the used TPM. The deviation from the TPM is calculated as

$$\text{Deviation} = 100 \cdot \frac{\text{SAFD Generated} - \text{SAFD TPM}}{\text{SAFD TPM}},$$

and the generation process is valid if the deviation is close to zero for all states.

The result of the SAFD deviation test is presented in Figure 5.4. The negative peak at the idle state (zero velocity and acceleration) origins from the restriction on the first transition when a new driving cycle is generated. The first transition has to leave the idle state. This causes the probability of the idle state to decrease in comparison with the SAFD of the TPM.

Because the deviation for the idle state has such a large negative value, it will increase the deviation for all other states a couple of percent. The result in Figure 5.4 can be compared to the test when no edge trimming of the generated driving cycles is performed, seen in Figure 5.5. When running the 'no-trimming' test, there is no large deviation for the idle state, and all other state deviations are close to zero.

The second thing to notice is that the deviation is very small, where velocities and accelerations are low. At the same time, the deviation is larger at the edges of the figure. In the TPM for the category *short*, there is a high frequency of low velocity and acceleration states, while samples with high velocities and accelerations are less common.

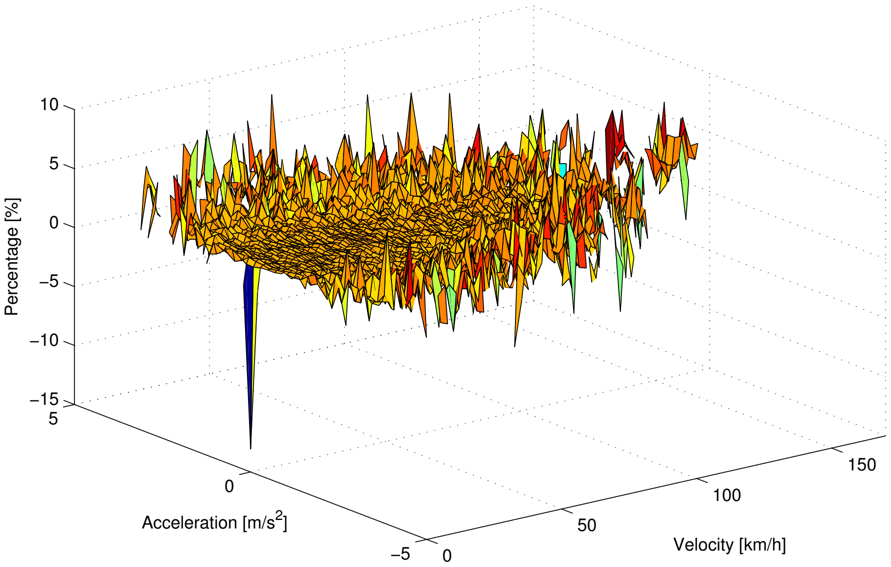


Figure 5.4: *SAFD deviations from the TPM distribution for 1 million generated driving cycles in the category short.*

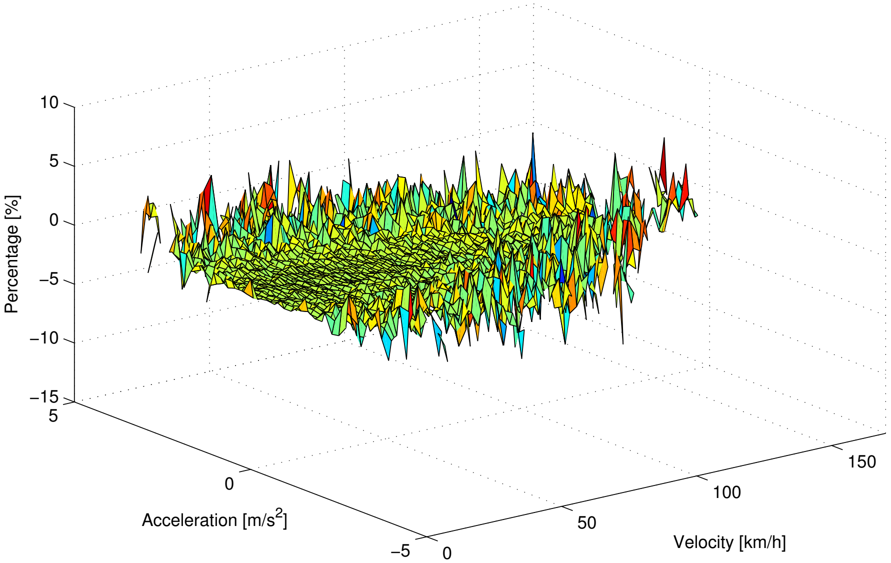


Figure 5.5: *SAFD deviations from the TPM distribution for 1 million generated driving cycles in the category short without trimming zero velocity measurement from the edges.*

States in the middle of the SAFD are more frequent when multiple driving cycles are generated. The deviations will therefore converge towards zero faster than the deviations for the states with high values of either acceleration or velocity. If more driving cycles were generated, around one billion, there would be close to zero deviation at the edges as well.

The fact that the SAFD of the generated driving cycles differ slightly from the TPM distribution might cause the generated driving cycles to differ from the expected. This is however handled in the validation process where the non-representative driving cycles are rejected.

5.2 Selected validation variables

The subsets of variables selected by the four proposed methods, applied to the driving cycles categorized as *short*, *urban* and *mixed*, can be seen in Table 5.1. Since some of the variables are highly correlated, the results can be misleading. One of two correlated variables can be selected by one of the methods, whereas the other variable can be selected by another method. Since the two variables are correlated, it can be seen as that the same feature has been selected rather than two different variables.

By grouping variables that are linearly correlated to each other with an absolute Pearson correlation coefficient,

$$|r_{i,j}| = \left|\frac{\operatorname{cov}(\mathbf{X_i}, \mathbf{X_j})}{\sigma_i \sigma_j}\right| = \left|\frac{\sum_{k=1}^{N}(x_{i,k} - \overline{x}_i)(x_{j,k} - \overline{x}_j)}{\sqrt{\sum_{k=1}^{N}(x_{i,k} - \overline{x}_i)^2}\sqrt{\sum_{k=1}^{N}(x_{j,k} - \overline{x}_j)^2}}\right|, \tag{5.1}$$

larger than 0.9, the table can be reduced in size. The variables correlated above $|r_{i,j}| = 0.9$ are listed in Table 5.2. When three variables are listed in one group, all within group correlations $|r_{i,j}|$ exceeds 0.9. The revised table, Table 5.3, shows the number of selected variables in each group of correlated variables. (An X in the table indicates that only one variable in the group is selected.)

At least one variable from Group 2 is selected by all methods in all categories, except by the cluster analysis in the categories *short* and *urban*. The cluster analysis selects the variable *number of stops per kilometer* instead of a variable from Group 2. *Number of stops per kilometer* is selected because it is clustered together with all the five velocity related variables and is used as cluster representative.

The *percentage of time in cruise mode* and the variables related to the variations in the acceleration are also frequently selected as representative in various categories. This indicates that the *acceleration standard deviation* and the mean accelerations captures some important property of driving cycles. They are all related to the aggressiveness of the driving cycle.

Table 5.1: *Representative variables selected by four different methods.*

	Short				**Urban**				**Mixed**				
	Regression	Clustering	Combined	LASSO	Regression	Clustering	Combined	LASSO	Regression	Clustering	Combined	LASSO	Var. group
Mean pos. vel.							X	X					1
Mean vel.					X								1
Max. vel.			X	X	X		X	X	X	X		X	2
95 % max. vel.											X	X	2
Vel. STD	X											X	2
Mean pos. acc.				X			X	X				X	3
Mean neg. acc.	X	X	X	X	X	X		X	X		X	X	4
Pos. acc. time								X					5
Neg. acc. time								X					5
95 % pos. acc.	X		X		X			X		X			-
95 % neg. acc.													4
Max. acc.		X			X	X				X			-
Min. acc.		X			X	X				X			-
Acc. STD				X				X				X	3
% time pos. acc.							X			X			-
% time neg. acc.		X			X	X	X			X			-
Driving dist.													6
Driving time													5
Idle time		X			X	X				X			-
% idle time					X								-
Cruise time		X			X								6
% cruise time	X		X	X	X	X	X	X					-
Nr. of stops								X					-
Nr. of stops /km		X			X	X	X	X		X			-
Mean s.p.		X			X	X		X		X			-
Maximum s.p.													-
Minimum s.p.				X	X		X	X				X	-

Table 5.2: Variables grouped together due to strong correlation.

Variable group	Variable 1	Variable 2	Variable 3
1	Mean pos. vel.	Mean vel.	-
2	Max. vel.	95 % max. vel.	Vel. STD
3	Mean pos. acc.	Acc. STD	-
4	Mean neg. acc.	95 % neg. acc.	-
5	Pos. acc. time	Neg. acc. time	Driving time
6	Driving dist.	Cruise time	-

Table 5.3: Representative variables selected by four different methods. Correlated variables have been grouped together.

	Short				Urban				Mixed				
	Regression	Clustering	Combined	LASSO	Regression	Clustering	Combined	LASSO	Regression	Clustering	Combined	LASSO	Σ
Group 1 (2)					X		X	X					3
Group 2 (3)	X		X	X	X		X	X	X	X	X	3	12
Group 3 (2)				2			X	2				2	7
Group 4 (2)	X	X	X	X	X	X		X	X		X	X	10
Group 5 (3)								2					2
95 % pos. acc.	X		X		X			X		X			5
Max. acc.		X			X	X				X			4
Min. acc.		X			X	X				X			4
% time pos. acc.							X			X			2
% time neg. acc.		X			X	X	X			X			5
Group 6 (2)		X			X								2
Idle time		X			X	X				X			4
% idle time					X								1
% cruise time	X		X	X	X	X	X	X					7
# stops								X					1
# stops per km.		X			X	X	X	X		X			6
Mean s.p.		X			X	X		X		X			5
Maximum s.p.													0
Minimum s.p.				X	X		X	X				X	5
Σ	4	8	4	6	14	8	8	13	2	9	2	7	

The number of representative variables selected by different methods applied in different categories varies widely. It is only the number of variables selected by the cluster analysis that remains stable between the categories. There are nine variables selected in the *mixed* category and eight in the *short* and *urban* categories. The additional variable in the *mixed* category can be interpreted as a result of the restrictions on the velocity and distance in the *urban* and *short* categories.

5.2.1 Regression analysis results

The estimated regression model shows a large fit to the data in all categories. Figure 5.6 shows the calculated MTF compared to the predicted MTF calculated with the estimated model in the category *short*. Only four representative variables are selected, and the model fit exceeds the limit set on the R^2_{adj}-statistic.

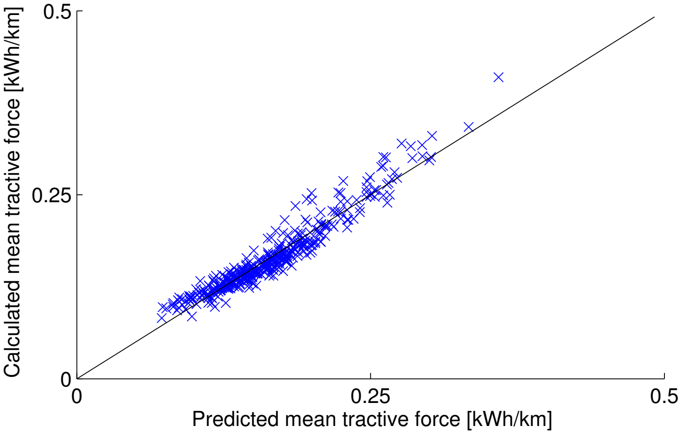


Figure 5.6: *Predicted MTF plotted against calculated MTF for the driving cycles categorized as short.*

The estimated models in the other categories shows similar fits, but the number of variables included in the final models differs. The *urban* category requires 14 variables to meet the requirements, whereas the *mixed* category only requires two. This is further discussed in the next chapter.

5.2.2 Cluster analysis results

Table 5.3 shows the resulting variables selected from the cluster analysis. The process of clustering the statistical variables in the three categories *urban*, *mixed* and *short* are illustrated in Figures 5.7-5.9. Each figure has the variables listed on the x-axis and the distance between the clusters grouped together on the y-axis.

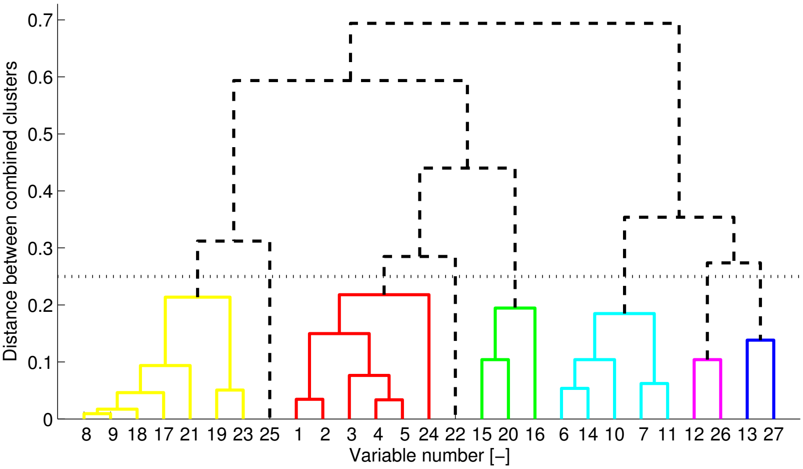


Figure 5.7: *Resulting dendrogram from the cluster analysis in the category containing urban driving cycles.*

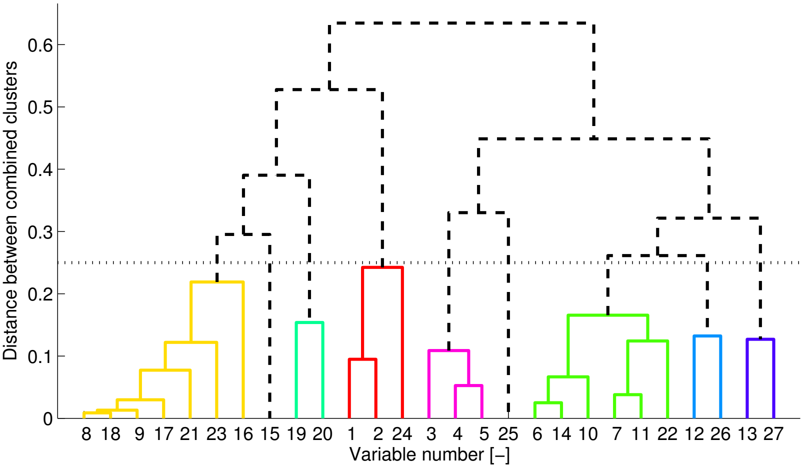


Figure 5.8: *Resulting dendrogram from the cluster analysis in the category containing mixed driving cycles.*

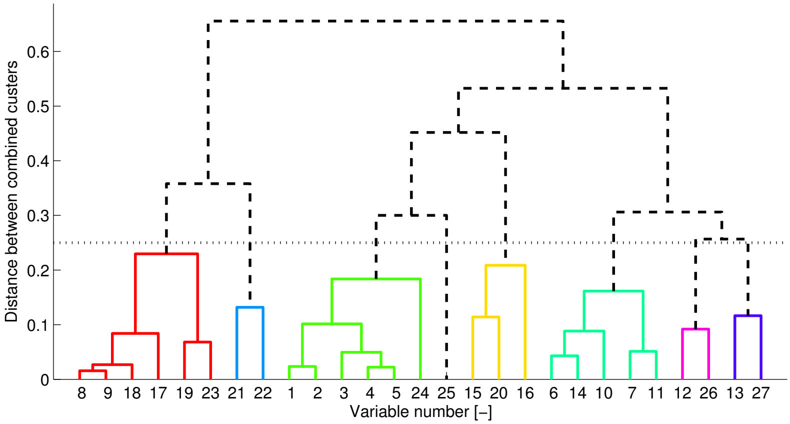


Figure 5.9: *Resulting dendrogram from the cluster analysis in the category containing short driving cycles.*

The final clusters obtained from the analysis in the *short* driving cycle category can be seen in Figure 5.9. A total of eight clusters are determined and the variables in each cluster represents a specific feature in the set of real-world driving cycles. The clusters are composed as follows (from left to right in Figure 5.9):

1. Time-related variables such as *idle time* and *positive acceleration time*.
2. Cruise time related variables (21 and 22).
3. All variables related to driving cycle velocity.
4. The variable *mean specific power* (25).
5. Variables related to the amount of time spent in various driving modes, namely idle, acceleration and deceleration.
6. Variables related to the aggressiveness of the driving cycle, i.e. *mean positive acceleration* and *acceleration standard deviation*.
7. Variables related to maximum acceleration. (*Maximum acceleration* and *maximum specific power*.)
8. Variables related to maximum retardation. (*Minimum acceleration* and *minimum specific power*.)

The only difference between the final clusters in the *short* and *urban* categories is that the variable *cruise time* moves from the second cluster to the first. The variable *cruise time* is highly related to the *percentage of cruise time* in the *short* category since the length of the driving cycles are restricted. The length of the driving cycles varies more in the category *urban*, causing *cruise time* to move to the time-related variables cluster.

5.2.3 Combined regression and cluster analysis results

The variables selected by the combined cluster and regression analysis are similar to the variables selected by the regression analysis method. The categories containing *short* and *urban* driving cycles obtains fewer representative variables from the combined analysis than from the regression analysis, indicating that the removal of variables due to mutual correlation do not have the expected effect. This is discussed further in Chapter 6.

5.2.4 LASSO results

It can be seen in Table 5.3 that the LASSO method tends to select variables that are highly correlated to each other. For example, all three variables in Group 2 (*maximum velocity*, *95 % maximum velocity* and *velocity standard deviation*) are selected in the category *mixed*. This can be explained by the fact that no variables are removed from the set of possible explanatory variables due to correlation before the regression model is estimated. An LASSO method where some of the 27 initial variables are removed in advance might solve the problem.

5.3 Validation

The validation data for the driving cycle in Figure 5.1 can be seen in Figure 5.10. The figure shows the deviations from the TPM medians for all the statistical variables. The horizontal lines represent the limits set, and the stems with a bigger marker represent the variables that the driving cycle is validated against. A driving cycle is approved if all the validation variables obtains values within the bounds.

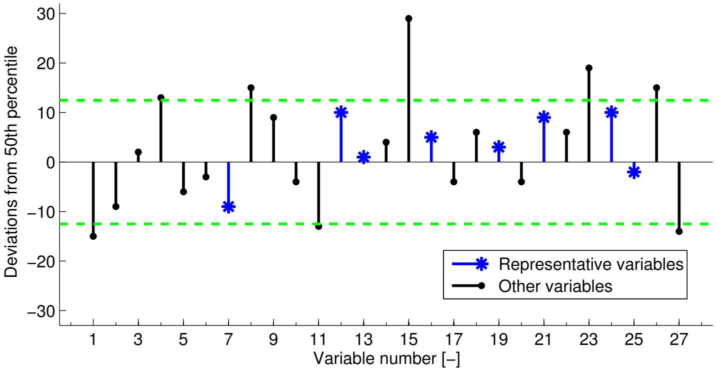


Figure 5.10: *Deviations from the category median values for the driving cycle in Figure 5.1.*

The number of iterations needed to generate a valid driving cycle varies and mainly depends on the number of representative variables used in the validation, but also on which variables that are used. The average number of iterations for each method of determining representative variables in multiple categories are shown in Table 5.4.

Table 5.4: *Number of iterations needed to approve a generated driving cycle, based on an average over 10 generations.*

		Iterations	Number of representative variables
Short	Regression	25	4
	Clustering	3800	8
	Combined	25	4
	LASSO	100	6
Urban	Regression	27000	14
	Clustering	4000	8
	Combined	300	8
	LASSO	4100	13
Mixed	Regression	10	2
	Clustering	8000	9
	Combined	5	2
	LASSO	75	7

The reason for the amount of iterations necessary to get a valid driving cycle can be seen in Figure 5.11. A driving cycle is valid when it has a statistical value within the dotted lines, and since some of the variables have a large deviation from their median values, they will not be approved easily.

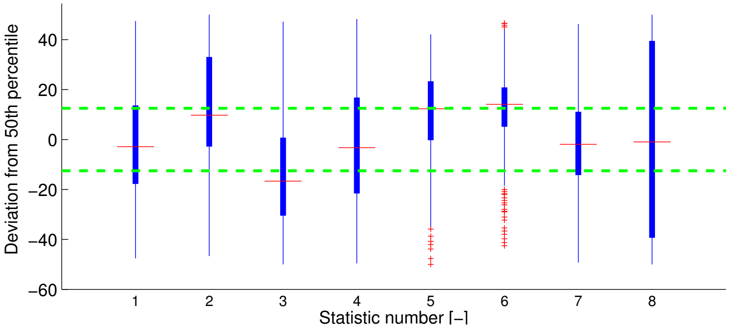


Figure 5.11: *Statistical deviations during a generation with 20 000 iterations.*

6 Discussion

The objective with this thesis was to generate stochastic driving cycles from a Markov process. The desired result was to generate driving cycles that resembles real-world driving in terms of statistical criteria.

The statistical variables calculated for the driving cycles are to some extent affected by the discretization. However, since they are derived from the discretized real-world driving cycles, they are still valid for comparison with the generated driving cycles. One way to motivate the discretization could be to argue that the vehicles which the driving cycles are applied to will erase the effects, and that they will instead resemble the original real-world driving cycles.

The discretization also affects the generation of the TPMs. If for instance a velocity $v = 0.4$ km/h, is measured together with the acceleration, $a = -0.15$ m/s^2, the resulting discrete state is $[v, a] = [0, -0.2]$, if the default discretization steps are used. It might seen odd that the vehicle can stand still while having a negative acceleration when the vehicle is assumed to never have a negative velocity. However, the resulting velocity profile in the generated driving cycles are not affected. It is however clear that further studies needs to be performed in order to identify the impact of the discretization.

The speed-acceleration frequency distribution shows that the generated cycles have almost to the same distribution as the SAFD from the TPM. The main differences are visible in the idle state. The large deviation is due to the removal of superfluous zero velocity states at the beginning and end of the driving cycles. Even though this removal of states changes the SAFD for the generated driving cycles, it is still reasonable since they do not add any relevant information to the driving cycle.

Lin and Niemeier [2002] also performed a SAFD test on their generated driving cycles. However, they compared the differences while this thesis concentrates on the deviations, defined in Section 5.1.1. Analyzing the deviations will give a better representation over the entire distribution since high values on speed and acceleration have a low frequency and the difference will be close to zero in comparison even though the values differ. Low values for velocity and acceleration have a higher frequency and the difference can be visible even though the deviation is relatively small. For these reasons, it is better to analyze the deviations.

One strength with the method of determining representative variables presented in this thesis compared to Lee and Filipi [2011] is that it uses an automatic process and finds a set of representative variables for each category. In [Lee and Filipi, 2011], they analyzed all available driving cycles and used the result regardless of the type of driving cycles. Different categories usually have different kinds of driving cycles and needs separate sets of representative variables to be described properly.

The implemented software performs an automated stepwise regression analysis, and the number of variables selected as representative differs a lot between categories as stated in the previous chapter. The reason can be seen in Figure 6.1. The models estimated from the categories *short*, *urban* and *mixed* shows similar development of the R^2_{adj}-statistic when the number of explanatory variables decreases. The hard limit forces the number of variables to 14 in the category *urban*, even though the R^2_{adj}-statistic is very close to the limit when only four explanatory variables are used.

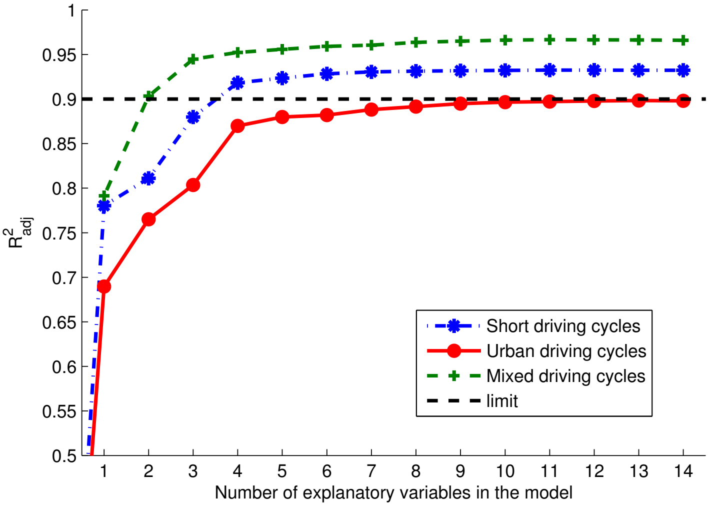


Figure 6.1: *Adjusted R^2-statistic for estimated regression models with various number of regressors.*

The reason for why the regression model estimated in the category *mixed* shows a larger fit than the models in the other categories might depend on which variables that are removed in advance due to mutual correlation. It might also depend on some of the difficulties listed below. Further studying of the phenomena is needed in order to determine the cause of the results.

Some other difficulties when automating the regression analysis process are:

- No coefficients are added back into the regression equation once they have been removed. This can be a problem since the t-value for a coefficient depends on the regression model and can vary between iterations.
- The amount of observations needed to ensure that no over-fitting is made is approximately 10 to 20 times the number of explanatory variables used in the regression equation. This means that the number of driving cycles needed to perform a regression analysis is at least $n = 100$, assuming that at least 10 explanatory variables are selected at the first iteration step.
- Two explanatory variables might not show a linear correlation, but some of them might be related in other ways. It can be exponential relationships, or relations where one variable can be derived from several others. These scenarios will not result in a situation where variables are removed, and that might lead to a situation where the assumption of independent explanatory variables do not hold.
- The fact that the explanatory variables are ranked according to their individual correlation with the response variable might lead to the selection of the wrong set of explanatory variables. A variable that together with another one explains a lot of the response can be omitted because it can not explain the response good enough on its own.

The use of MTF as a representative response in the automated regression analysis may result in some difficulties explaining certain features of the driving cycles. The contributions to the MTF are only calculated from traction mode samples, which means that information from the braking and idle parts of a driving cycles are not accounted for. These modes are increasingly important when designing electrical vehicles. For example, electrical vehicles generates energy from the braking power which is not accounted for while calculating the MTF, but highly affects the needed power. However, the application implemented in this thesis considers general driving cycles and do not study the differences between vehicles operating in them.

Lee and Filipi [2011] used a regression analysis method to determine representative variables for driving cycles in general, which became the starting point for this thesis as well. However, it has been shown that regression analysis does not always work as expected. When comparing the different methods, it is clear that cluster analysis provides a more easily interpreted set of representative variables for a specific set of driving cycles. It gave a similar amount of representative variables for each category, and the variables that got clustered seemed reasonable.

Cluster analysis also avoids the problems that occur when the MTF is used as a representative response, since the clustering explains the variations in a specific set of driving cycles rather than the MTF.

6.1 Future work

Some of the improvements and extensions to the software that could be of interest are listed below.

1. *First principal component* - The selection of a cluster representative can be performed in many ways. A PCA method is used here, which selects *one* variable from each cluster. Another, perhaps better way, would be to use the FPC to define a statistic (linear combination of all the variables in the cluster) in each cluster that captures the most of the within cluster variations.

2. *User defined car parameters* - When developing cars, there is a desire to calculate or simulate how much emissions the vehicle will emit. Make it possible for the user to enter car specific parameters such as

   - Vehicle mass.
   - Frontal area.
   - Aerodynamic coefficients.

   When car parameters are set and a model for emissions has been implemented into the software, it will be possible to calculate the emissions over several driving cycles of the same type. Since the cycles are stochastically generated they will differ enough to avoid cycle beating when optimizing parameters.

3. *Connection to Simulink model* - A common way to simulate vehicles is by using a Simulink model. By connecting the cycle generation software to a vehicle model, it will be possible to analyze the performance of the modeled vehicle.

4. *Generating cycles based on speed limits* - Another way to categorize data is based on speed limits. For Sweden, the driving will be categorized into bins of 30 km/h, 40 km/h, 50 km/h, ... , and 120 km/h. When generating a driving cycle, it should be possible to either

   - Set a complete route: 50 km/h for 8.3 km followed by 70 km/h for 1.2 km and so on.
   - Set a route ratio: 25% of the route is in 50 km/h, 30% is in 70 km/h, and drive for a total of 40 km.

   This includes collecting new data where the driving location is known, extracting data about speed limits from a database and categorize all measurements depending on speed limits. Example of such database is NVDB [Trafikverket, 2012].

If the second method of generating driving cycles is used, there is also a need to calculate the probabilities to switch between different speed limits.

5. *A validation if the driving cycle is realistic* - The implemented validation process checks if the statistical values of the driving cycle is valid, and approves it if everything checks out. But there is no check if the generated driving cycles are realistic.

    - Can a vehicle go from a cold start to this velocity in that time.
    - Is it reasonable for a cycle to have that many stops in such a short of a timespan.

7 Conclusion

An application has been developed in Matlab that can be used to generate stochastic driving cycles based on a given set of real-word driving cycles. The generated driving cycles resembles real-world driving cycles in terms of SAFD and selected statistical properties.

Markov chain theory is used to randomly select state transitions in the velocity profile to ensure the randomness of the generated driving cycles, and minimizing the risk of cycle beating.

The representativeness of the generated driving cycles can be investigated using either regression analysis or hierarchical cluster analysis. A set of statistical variables that have to coincide with the generated driving cycle values are determined. The former method, suggested by Lee and Filipi [2011], proved to be difficult to automate and the assumption that the same statistical variables can be used to represent all types of driving cycles proved to be wrong. The variables describing a set of driving cycles are highly dependent on the driving conditions in the driving cycle, i.e. amount of traffic or the type of road.

Overall, the most important conclusions can be stated as

- A Markov process can be used to ensure the randomness of the generated driving cycles.
- The characteristics of a driving cycle varies between types of driving and the validation must therefore be specific for each driving category.
- The proposed hierarchical cluster analysis can be used to determine a set of variables sufficient to represent a specific set of driving cycles.

Bibliography

M. André. Driving cycles development: Characterization of the methods. *SAE Technical Paper Series, vol. 961112SAE (Society of Automotive Engineers)*, 1996.

G. Blom, J. Enger, G. Englund, J. Grandell, and L. Holst. *Sannolikhetsteori och statistikteori med tillämpningar.* Studentlitteratur, 2005.

E. Enqvist. *Grundläggande regressionsanalys.* BOKAB Linköping, June 2007.

B. S. Everitt, S. Sabine, M. Leese, and D. Stahl. *Cluster analysis.* Wiley, first edition, 2011.

M. Fellah, A. Rousseau, S. Pagerit, E. Nam, and G. Hoffman. Impact of real-world drive cycles on PHEV battery requirements. *SAE Technical Paper*, pages 01–1383, 2009.

J. A. Gubner. *Probability and random processes for electrical and computer engineers.* Cambridge University Press, 2006. 476–488.

L. Guzzella and A. Sciarretta. *Vehicle propulsion systems.* Springer-Verlag Berlin Heidelberg, 2007.

F. E. Harrell. *Regression modeling strategies.* Springer-Verlag New York, Inc., 2001.

I. T. Jolliffe. *Principal component analysis.* Springer, second edition, 2002.

P. Kågeson. Cycle-beating and the EU test cycle for cars. *European Federation for Transport and Environment. T&E*, 98(3), 1998.

T-K. Lee and Z. S Filipi. Synthesis of real-world driving cycles using stochastic process and statistical methodology. *International Journal of Vehicle Design*, 57(1):17–36, 2011.

J. Lin and D. A. Niemeier. An exploratory analysis comparing a stochastic driving cycle to California's regulatory cycle. *Atmospheric Environment*, 36(38):5759–5770, 2002.

O. Renaud and M-P. Victoria-Feser. A robust coefficient of determination for regression. *Journal of Statistical Planning and Inference*, 140(7):1852–1862, 2010.

V. Schwarzer, R. Ghorbani, and R. Rocheleau. Drive cycle generation for stochastic optimization of energy management controller for hybrid vehicles. In *proceedings of the 2010 IEEE International Conference on Control Applications (CCA)*, pages 536–540, sept. 2010.

S. Shahidinejad, E. Bibeau, and S. Filizadeh. Statistical development of a duty cycle for plug-in vehicles in a north american urban setting using fleet information. *IEEE Transactions on Vehicular Technology*, 59(8):3710–3719, 2010.

R. Tibshirani. Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society*, 58(1):267–288, 1996.

Trafikverket. Nationell vägdatabas (NVDB), 2012. URL `https://nvdb2012.trafikverket.se/`. Accessed: 2013-05-02.

A
Driving Cycle Characteristics

The following appendix describes the 27 statistical variables that were proposed by Lee and Filipi [2011] as possible explanatory variables in a regression model. How the variables are defined and calculated are described in detail. The variables have been categorized as velocity, acceleration, distance and time related variables as well as variables depending on driving characteristics. The equation numbers correspond to the numbers mentioned in Table 3.6, which also lists the variable units.

Each variable is calculated using the averaged and discretized driving cycles velocity, v_i, and acceleration, a_i, defined in the time intervals between the original velocity samples ($i \in [1, 2, \ldots, N]$ when the number of samples in the measured velocity equals $N + 1$). The velocity unit is km/h and the acceleration unit is m/s^2. Furthermore, the sample time, T_s, is assumed to be constant through the entire driving cycle.

A.1 Velocity

A total of five variables related to the driving cycle velocity are defined. First, there are two mean velocity statistics. The first one, **mean positive velocity** is defined as

$$\bar{v}_{pos} = \frac{1}{N_{v_{pos}}} \sum_{i:v_i>0} v_i, \tag{A.1}$$

where $N_{v_{pos}}$ is the number of samples with a positive velocity ($v_i > 0$) in the driving cycle.

The second one, **mean velocity**, which also includes zero velocity samples, is calculated as

$$\overline{v} = \frac{1}{N} \sum_{i=1}^{N} v_i, \tag{A.2}$$

where N is the total number of samples in the cycle.

Two statistics depends on the driving cycles high velocity samples, namely **maximum velocity** and **95th percentile maximum velocity**. The former is defined as the maximum sample velocity, $v_{max} = \max\{v_1, v_2, ..., v_n\}$. The latter is the value for which 95 % of the sampled velocities are lower.

The last velocity related statistic is the **standard deviation of velocity**, defined as

$$\sigma_v = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (v_i - \overline{v})^2}, \tag{A.5}$$

where $\overline{v}$ is the cycle mean velocity. (The standard deviation is defined using the $N-1$ denominator in order to obtain a mean real estimation.)

A.2 Acceleration

Eleven variables related to the driving cycle acceleration are defined. **Mean positive acceleration** and **mean negative acceleration** are defined as

$$\overline{a}_{pos} = \frac{1}{N_{a_{pos}}} \sum_{i:a_i>0} a_i \tag{A.6}$$

$$\overline{a}_{neg} = \frac{1}{N_{a_{neg}}} \sum_{i:a_i<0} a_i, \tag{A.7}$$

where $N_{a_{pos}}$ and $N_{a_{neg}}$ are the number of positive and negative acceleration samples, respectively. The acceleration periods, **positive acceleration time** and **negative acceleration time** can also be derived using $N_{a_{pos}}$, $N_{a_{neg}}$ and T_s as

$$t_{a_{pos}} = N_{a_{pos}} T_s \tag{A.8}$$

$$t_{a_{neg}} = N_{a_{neg}} T_s. \tag{A.9}$$

There are four cycle statistics related to the extremes of the acceleration. The first pair, **95th percentile maximum acceleration** and **95th percentile minimum acceleration** are the 95th and 5th percentiles in the acceleration samples distribution.

The second pair is **maximum acceleration** and **minimum acceleration** and they are defined as $a_{max} = \max\{a_1, a_2, ..., a_N\}$ and $a_{min} = \min\{a_1, a_2, ..., a_N\}$, respectively. The **standard deviation of acceleration** is calculated for all accelerations (including both positive and negative values) and is defined as

$$\sigma_a = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(a_i - \overline{a})^2}, \tag{A.14}$$

where $\overline{a}$ is the mean cycle acceleration. In order to obtain a mean real estimation of the standard deviation, it is defined using the denominator $N-1$.

The last two acceleration related variables are **percentage of driving time under positive acceleration** and **percentage of driving time under negative acceleration** and they are calculated as

$$pct_{a_{pos}} = \frac{N_{a_{pos}}}{N} \tag{A.15}$$

$$pct_{a_{neg}} = \frac{N_{a_{neg}}}{N}, \tag{A.16}$$

where $N_{a_{pos}}$ and $N_{a_{neg}}$ are the same as in (A.6) and (A.7).

A.3 Driving distance and time

Two variables depend on the driving cycle distance and duration. The first one is the total distance driven in the cycle, denoted **driving distance** and the second one is the cycle duration, denoted **driving time**. The variables are calculated as

$$d = \frac{T_s}{3600} \cdot \sum_{i=1}^{N} v_i \tag{A.17}$$

$$t_{drive} = N \cdot T_s. \tag{A.18}$$

A.4 Driving characteristics

The vehicle is assumed to operate in idle mode when the cycle velocity $v_i = 0$, and the first two variables associated with driving characteristics are **idle time** and **percentage of idle time**, defined as

$$t_{idle} = N_0 \cdot T_s \tag{A.19}$$

$$pct_{t_{idle}} = \frac{N_0}{N}, \tag{A.20}$$

where N_0 is the number of samples with a velocity $v_i = 0$. An alternative definition could be to include the condition that also the acceleration $a_i = 0$, but that would only increase the complexity and serves no purpose.

The second pair consists of **cruise time** and **percentage of cruise time**. According to [Shahidinejad et al., 2010, pp.3712] a sample i is defined as cruise if the velocity $v_i > 5$ m/s and the acceleration $|a_i| < 0.1$ m/s^2. The definition used in this thesis is the same and the variables are derived in a similar way as the variables associated with the time spent in idle mode. The variables are defined as

$$t_{cruise} = N_c \cdot T_s \tag{A.21}$$

$$pct_{t_{cruise}} = \frac{N_c}{N}, \tag{A.22}$$

where N_c is the number of samples with an acceleration $|a_i| < 0.1$ m/s^2 and a velocity $v_i > 5$ m/s.

Two variables are related to the frequency of idle periods in the driving cycles, **number of stops** and **number of stops per kilometer**. The former one is the total number of idle periods in a driving cycle, calculated as

$$N_{stops} = \sum_{i=2}^{N} c_i, \qquad c_i = \begin{cases} 1, & \text{if } v_{i-1} \neq 0,\ v_i = 0 \\ 0, & \text{otherwise.} \end{cases} \tag{A.23}$$

The latter one is defined as the total number of stops divided by the total cycle distance, namely

$$N_{stops/km} = \frac{N_{stops}}{d}. \tag{A.24}$$

The last three statistical variables are all derived from the specific power, defined as $\mathrm{SP}_i = 2\frac{v_i}{3.6}a_i$ W/kg. The **mean specific power**,

$$\overline{\mathrm{SP}} = \frac{1}{N}\sum_{i=1}^{N}\mathrm{SP}_i, \tag{A.25}$$

is the average specific power over the entire cycle. **Maximum specific power** and **minimum specific power**,

$$\mathrm{SP}_{max} = \max\{\mathrm{SP}_1, \mathrm{SP}_2, ..., \mathrm{SP}_N\} \tag{A.26}$$

$$\mathrm{SP}_{min} = \min\{\mathrm{SP}_1, \mathrm{SP}_2, ..., \mathrm{SP}_N\}, \tag{A.27}$$

are the individual sample maximum and minimum.

B

User Manual

How to use the application, *Driving Cycle Generation v. 1.0*, is described here in detail. The software can generate stochastic driving cycles based on a provided set of real-world data. The data provided to the program must be configured as described in Section B.1.

The software is completely controlled from within a graphical user interface (GUI) described in Section B.2. How the data is converted to a transition probability matrix (TPM) and used to generate driving cycles are described in Section B.3.

There is also a short troubleshooting guide in Section B.4 in case any errors occur. The software was created and tested in Matlab R2011b and above and require the statistics toolbox to function.

B.1 Data input specifications

A correctly formatted data file will be a **.mat* file containing an array of structures configured as in Table B.1. Each structure has to contain a single driving cycle.

Table B.1: *Data input specification*

Field	Type	Explanation	Unit
`velocity`	Vector	Sampled velocity	[km/h]
`Ts`	Scalar	Sample time	[s]
`carCharacteristics`	Structure	(optional) See Table B.2.	-

The field carCharacteristics is an optional structure configured as in Table B.2. There is no requirement to attach this field since default values exist, although the result of the regression analysis will be improved if this is correctly defined. See Section B.3.3 for more information about analysis methods. Example B.1 shows an example of a correctly formatted set of input driving cycles.

Table B.2: *carCharacteristics input specification*

Field	Type	Explanation	Default value	Unit
mv	Scalar	Vehicle mass	1600	[kg]
Cd	Scalar	Aerodynamic drag coefficient	0.4	[-]
Cr	Scalar	Rolling resistance coefficient	0.013	[-]
Af	Scalar	Vehicle frontal area	2.15	[m^2]

B.1 Example

The input data should be combined in a structure array where each element is a driving cycle. The input here consists of 123 driving cycles.

```
>> inputData

inputData =

1x123 struct array with fields:
    velocity
    Ts
    carCharacteristics
```

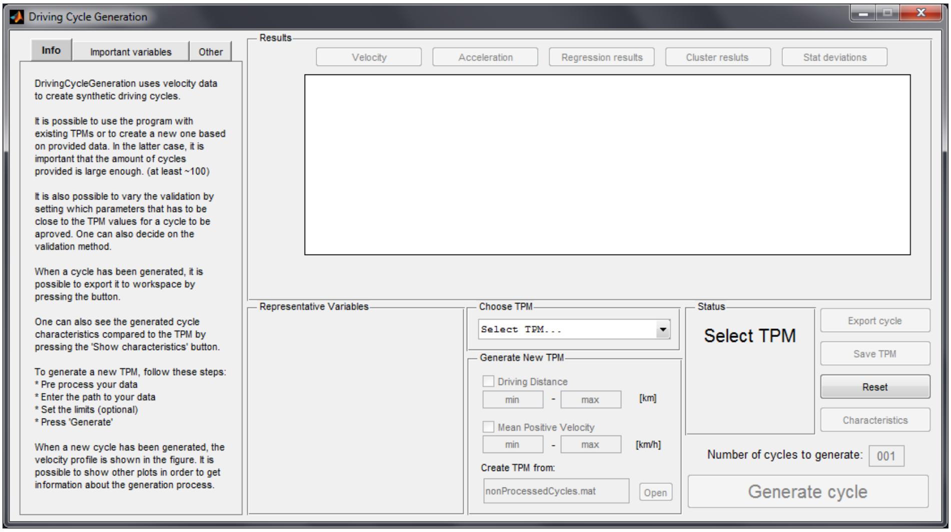
The last field is optional but if defined, it should be formatted as follows.

```
>> inputData(3).carCharacteristics

ans =

    mv: 1600
    Cd: 0.4000
    Cr: 0.0130
    Af: 2.1500
```

B.2 Graphical user interface

The developed GUI can be seen in Figure B.1. By using the interface, it is possible to change settings and analyze the result in a more convenient way than using the available Matlab commands.

It is also a practical way to visually examine the generated driving cycles and its characteristics before exporting them for further use. The review of the generated driving cycles is easily done by pressing a couple of buttons in the GUI.



DrivingCycleGeneration uses velocity data to create synthetic driving cycles.

It is possible to use the program with existing TPMs or to create a new one based on provided data. In the latter case, it is important that the amount of cycles provided is large enough. (at least ~100)

It is also possible to vary the validation by setting which parameters that has to be close to the TPM values for a cycle to be aproved. One can also decide on the validation method.

When a cycle has been generated, it is possible to export it to workspace by pressing the button.

One can also see the generated cycle characteristics compared to the TPM by pressing the 'Show characteristics' button.

To generate a new TPM, follow these steps:
* Pre process your data
* Enter the path to your data
* Set the limits (optional)
* Press 'Generate'

When a new cycle has been generated, the velocity profile is shown in the figure. It is possible to show other plots in order to get information about the generation process.

Choose TPM: Select TPM...

Number of cycles to generate: 001

Create TPM from: nonProcessedCycles.mat

Figure B.1: Graphical user interface (GUI).

The information panel to the left gives a quick overview of the software and which steps to take. It is however recommended to read this manual before starting to generate driving cycles.

B.3 How to use the software

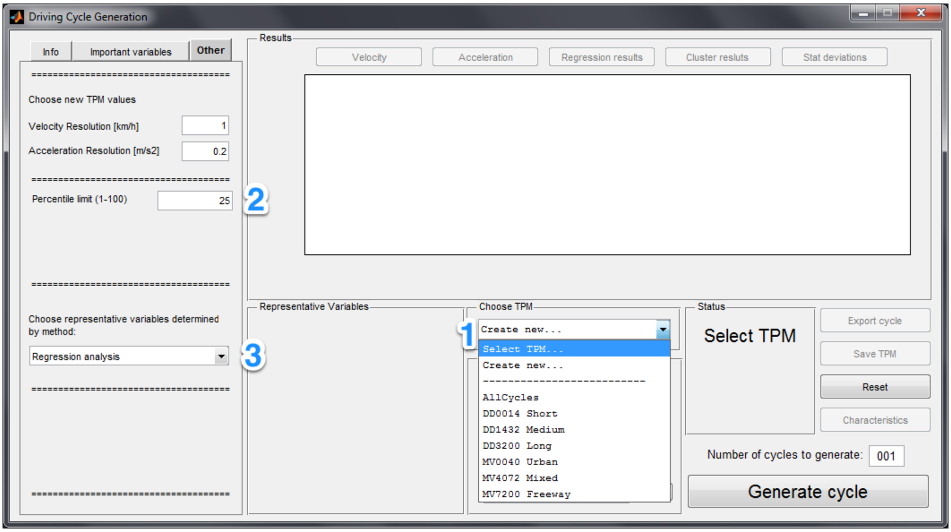
The GUI functions are described here, and the process of generating driving cycles is illustrated.

B.3.1 Use an existing TPM

By pressing the drop-down menu (**1**), pointed out in Figure B.2, a list of already existing TPMs is presented. When a new TPM is saved, it will show up here the next time an existing TPM is to be chosen.

Even though the TPMs are already generated and can instantly be used in the generation of a new driving cycle, there are still some settings available.

As shown in Figure B.2, there is a setting for the percentile limit (**2**) that affect the validation of the generated driving cycles. There is also a setting that lets the user define which set of representative variables to use when the generated cycles are validated (**3**). Both these settings can be found in the *Other*-tab.



Choose new TPM values

Velocity Resolution [km/h] 1

Acceleration Resolution [m/s2] 0.2

Percentile limit (1-100) 25

Choose representative variables determined by method: Regression analysis

Number of cycles to generate: 001

Figure B.2: *Use existing TPM to generate driving cycles.*

B.3.2 Create a new TPM

When creating a new TPM, there are several fields that can be changed to customize the resulting driving cycles, pointed out in Figure B.3. To generate a new TPM, select the option *Create new...*, in the drop-down menu described above.

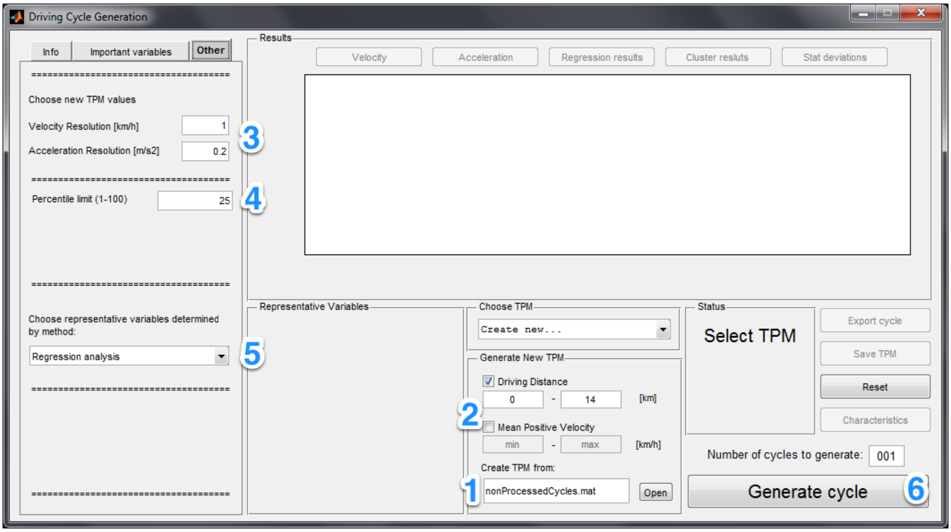
The most important step is to give the software some data to work with. By pressing the *Open* button (**1**), a window will be presented where you need to find a data file formatted as described in Section B.1.

When input data has been defined, it is possible to set some categorization limits. This is done by checking the box for *Driving Distance* and/or *Mean Positive Velocity* (2) and enter the variable range in the fields below. All input driving cycles will be used if no categorization limits are set. For example, if a categorization limit on the driving distance is entered as in Figure B.3, only the provided driving cycles with a distance lower than 14 km will be used to create the TPM.

Everything is now set to generate a new TPM and driving cycle, but if there is a need to change the resolution for the data discretization, it is possible in the *Other*-tab as seen in Figure B.3 (**3**).

There are also settings for changing the validation limits (**4**) and for which validation method to use (**5**), described in Section B.3.3. However, all methods and all limits will be calculated so that it is possible to reuse the same TPM with several different settings in the future.

When the prefered settings have been entered, it is possible to enter how many driving cycles to generate and pressing the *Generate cycle* button (**6**). The TPM will be created as a part of the process.



Choose new TPM values

Velocity Resolution [km/h]: 1

Acceleration Resolution [m/s2]: 0.2

Percentile limit (1-100): 25

Choose representative variables determined by method: Regression analysis

Choose TPM: Create new...

☑ Driving Distance: 0 - 14 [km]

□ Mean Positive Velocity: min - max [km/h]

Create TPM from: nonProcessedCycles.mat

Status: Select TPM

Number of cycles to generate: 001

Figure B.3: *Create a new transition probability matrix (TPM).*

B.3.3 Choose method of determining representative variables

There are four methods for determining representative variables

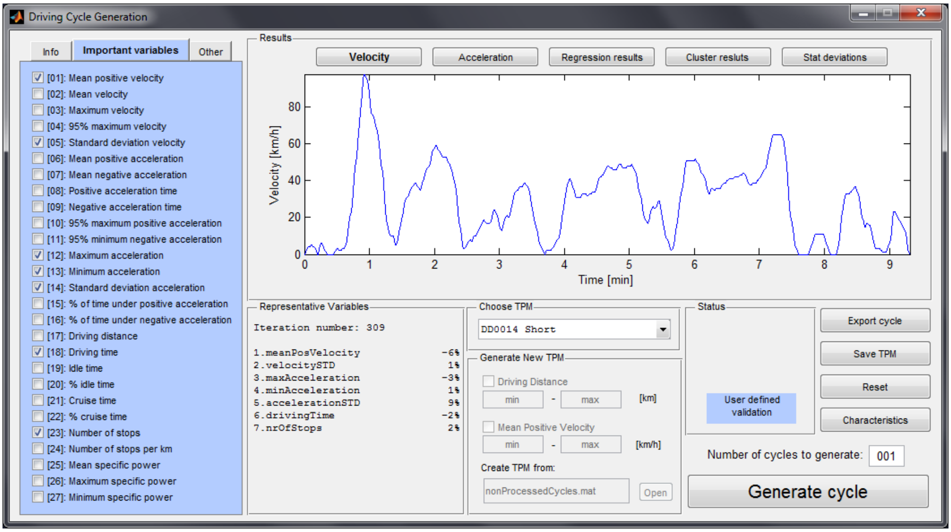
- Regression analysis
- Cluster analysis
- Combo analysis (Cluster + Regression)
- LASSO analysis

Different methods will give different representative variables and will affect both number of iterations and the distinguishing features of the generated driving cycle.

The user can also define their own variables using the *Important variables* tab, see Figure B.4. The variables selected will be used in the validation and it does not matter which method for determining representative variables is selected.

B.3.4 Analyze generated driving cycles

When a driving cycle has been generated, it is possible to look at different aspects of the generation process, as shown in Figure B.5.



☑ [01]: Mean positive velocity
☐ [02]: Mean velocity
☐ [03]: Maximum velocity
☐ [04]: 95% maximum velocity
☑ [05]: Standard deviation velocity
☐ [06]: Mean positive acceleration
☐ [07]: Mean negative acceleration
☐ [08]: Positive acceleration time
☐ [09]: Negative acceleration time
☐ [10]: 95% maximum positive acceleration
☐ [11]: 95% minimum negative acceleration
☑ [12]: Maximum acceleration
☑ [13]: Minimum acceleration
☑ [14]: Standard deviation acceleration
☐ [15]: % of time under positive acceleration
☐ [16]: % of time under negative acceleration
☐ [17]: Driving distance
☑ [18]: Driving time
☐ [19]: Idle time
☐ [20]: % idle time
☐ [21]: Cruise time
☐ [22]: % cruise time
☑ [23]: Number of stops
☐ [24]: Number of stops per km
☐ [25]: Mean specific power
☐ [26]: Maximum specific power
☐ [27]: Minimum specific power

```
Iteration number: 309

1.meanPosVelocity          -6%
2.velocitySTD               1%
3.maxAcceleration          -3%
4.minAcceleration           1%
5.accelerationSTD           9%
6.drivingTime              -2%
7.nrOfStops                 2%
```

Choose TPM: DD0014 Short

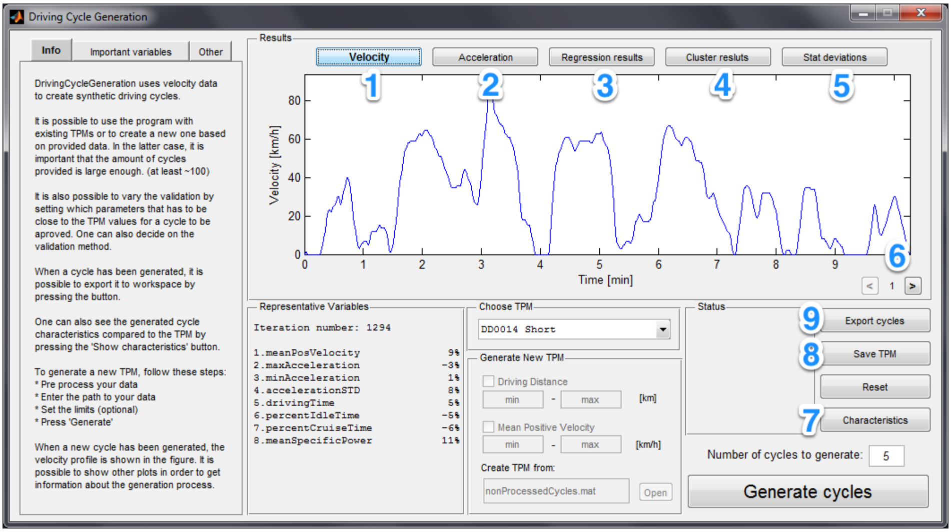
☐ Driving Distance min - max [km]

☐ Mean Positive Velocity min - max [km/h]

Create TPM from: nonProcessedCycles.mat

Number of cycles to generate: 001

Figure B.4: *User defined validation activated.*



DrivingCycleGeneration uses velocity data to create synthetic driving cycles.

It is possible to use the program with existing TPMs or to create a new one based on provided data. In the latter case, it is important that the amount of cycles provided is large enough. (at least ~100)

It is also possible to vary the validation by setting which parameters that has to be close to the TPM values for a cycle to be aproved. One can also decide on the validation method.

When a cycle has been generated, it is possible to export it to workspace by pressing the button.

One can also see the generated cycle characteristics compared to the TPM by pressing the 'Show characteristics' button.

To generate a new TPM, follow these steps:
* Pre process your data
* Enter the path to your data
* Set the limits (optional)
* Press 'Generate'

When a new cycle has been generated, the velocity profile is shown in the figure. It is possible to show other plots in order to get information about the generation process.

```
Iteration number: 1294

1.meanPosVelocity           9%
2.maxAcceleration          -3%
3.minAcceleration           1%
4.accelerationSTD           8%
5.drivingTime               5%
6.percentIdleTime          -5%
7.percentCruiseTime        -6%
8.meanSpecificPower        11%
```

Choose TPM: DD0014 Short

☐ Driving Distance min - max [km]

☐ Mean Positive Velocity min - max [km/h]

Create TPM from: nonProcessedCycles.mat

Number of cycles to generate: 5

Figure B.5: *GUI view after the generation of five driving cycles.*

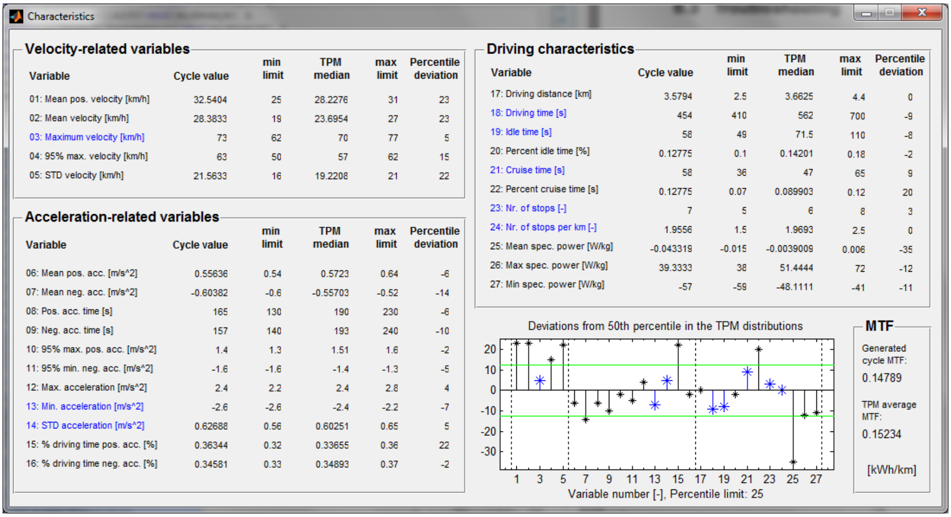
There are five buttons at the top of the GUI

- Velocity (**1**) - Driving cycle velocity profile (default view).
- Acceleration (**2**) - Acceleration values from the Markov chain.
- Regression results (**3**) - Results from the regression analysis.
- Clustering results (**4**) - Results from the clustering process
- Statistical deviations (**5**) - Deviations from the 50th percentile over all iterations.

The graph will update according to which mode is selected.

And if multiple driving cycles are generated, there are buttons to look at the other driving cycles (**6**). Regression results and clustering results are the same for all generated driving cycles since they are related to the TPM and not the driving cycles.

The button *Characteristics* (**7**), opens a new window that shows deviations for all the 27 variables and their validation limits, see Figure B.6.



Velocity-related variables

Variable	Cycle value	min limit	TPM median	max limit	Percentile deviation
01: Mean pos. velocity [km/h]	32.5404	25	28.2276	31	23
02: Mean velocity [km/h]	28.3833	19	23.6954	27	23
03: Maximum velocity [km/h]	73	62	70	77	5
04: 95% max. velocity [km/h]	63	50	57	62	15
05: STD velocity [km/h]	21.5633	16	19.2208	21	22

Acceleration-related variables

Variable	Cycle value	min limit	TPM median	max limit	Percentile deviation
06: Mean pos. acc. [m/s^2]	0.55636	0.54	0.5723	0.64	-6
07: Mean neg. acc. [m/s^2]	-0.60382	-0.6	-0.55703	-0.52	-14
08: Pos. acc. time [s]	165	130	190	230	-6
09: Neg. acc. time [s]	157	140	193	240	-10
10: 95% max. pos. acc. [m/s^2]	1.4	1.3	1.51	1.6	-2
11: 95% min. neg. acc. [m/s^2]	-1.6	-1.6	-1.4	-1.3	-5
12: Max. acceleration [m/s^2]	2.4	2.2	2.4	2.8	4
13: Min. acceleration [m/s^2]	-2.6	-2.6	-2.4	-2.2	-7
14: STD acceleration [m/s^2]	0.62688	0.56	0.60251	0.65	5
15: % driving time pos. acc. [%]	0.36344	0.32	0.33655	0.36	22
16: % driving time neg. acc. [%]	0.34581	0.33	0.34893	0.37	-2

Driving characteristics

Variable	Cycle value	min limit	TPM median	max limit	Percentile deviation
17: Driving distance [km]	3.5794	2.5	3.6625	4.4	0
18: Driving time [s]	454	410	562	700	-9
19: Idle time [s]	58	49	71.5	110	-8
20: Percent idle time [%]	0.12775	0.1	0.14201	0.18	-2
21: Cruise time [s]	58	36	47	65	9
22: Percent cruise time [s]	0.12775	0.07	0.089903	0.12	20
23: Nr. of stops [-]	7	5	6	8	3
24: Nr. of stops per km [-]	1.9556	1.5	1.9693	2.5	0
25: Mean spec. power [W/kg]	-0.043319	-0.015	-0.0039009	0.006	-35
26: Max spec. power [W/kg]	39.3333	38	51.4444	72	-12
27: Min spec. power [W/kg]	-57	-59	-48.1111	-41	-11

Deviations from 50th percentile in the TPM distributions
20
10
0
-10
-20
-30
1
3
5
7
9
11
13
15
17
19
21
23
25
27
Variable number [-], Percentile limit: 25

MTF

Generated cycle MTF: 0.14789

TPM average MTF: 0.15234

[kWh/km]

Figure B.6: Characteristics for the generated driving cycle.

B.3.5 Save a generated TPM

When the process of generating a new TPM is finished, it will be possible to save it for later use by pressing the *Save TPM* button (**8**), seen in Figure B.5. When asked to, enter a name for the generated TPM and press *OK*.

B.3.6 Export generated driving cycles

When all desired driving cycles are generated, it is possible to export to the current workspace by pressing the *Export cycles* button (**9**), seen in figure Figure B.5. The exported driving cycles can then be accessed as in example B.2.

B.2 Example

The output in Matlabs command window after a generation of five driving cycles

```
>> ExportedCycles

ExportedCycles =

1x5 struct array with fields:
    velocity
    acceleration
    duration
    Ts
    characteristics
    TPMname
```

B.4 Troubleshooting

Here are some common errors listed together with possible solutions.

Q: It seems to generate forever

Sometimes there will be a combination of representative variables that are extremely hard or even close to impossible to finish with the current selected validation limit. The only option is to open the Matlab window and press *Ctrl+C* followed by a restart of the GUI. Try again with different settings when the GUI has reloaded.

Q: I get multiple warnings during the regression analysis

This is because there are very few driving cycles in use. It will still be possible to generate a TPM and driving cycles with these settings but it is still strongly recommended to change your categorization limits or add more data since the representative variables may not be accurate.

Upphovsrätt

Copyright