Synthesis of realistic driving cycles with high accuracy and computational speed, including slope information

Citation for published version (APA):

Document license:
TAVERNE

DOI:
10.1109/TVT.2016.2546338

Document status and date:
Published: 01/06/2016

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:
• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher’s website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:
www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:
openaccess@tue.nl
providing details and we will investigate your claim.
Synthesis of Realistic Driving Cycles With High Accuracy and Computational Speed, Including Slope Information

Emilia Silvas, Kobus Hereijgers, Huei Peng, Theo Hofman, and Maarten Steinbuch, Senior Member, IEEE

Abstract—This paper describes a new method to synthesize driving cycles, where not only the velocity is considered but the road slope information of the real-world measured driving cycle as well. Driven by strict emission regulations and tight fuel targets, hybrid or electric vehicle manufacturers aim to develop new and more energy- and cost-efficient powertrains. To enable and facilitate this development, short, yet realistic, driving cycles need to be synthesized. The developed driving cycle should give a good representation of measured driving cycles in terms of velocity, slope, acceleration, and so on. Current methods use only velocity and acceleration and assume a zero road slope. The heavier the vehicle is, the more important the road slope becomes in powertrain prototyping (as with component sizing or control design); hence, neglecting it leads to unrealistic or limited designs. To include the slope, we extend existing methods and propose an approach based on multidimensional Markov chains. The validation of the synthesized driving cycle is based on a statistical analysis (as the average acceleration or maximum velocity) and a frequency analysis. This new method demonstrates the ability of capturing the slope, acceleration, and so on. Furthermore, results show that the proposed method outperforms current methods in terms of accuracy and speed.

Index Terms—Driving cycle compression, hybrid electric vehicle (HEV) design, multidimensional Markov chains, slope and velocity synthesis.

I. INTRODUCTION

DESIGNING efficient hybrid electric vehicles (HEVs) with low ownership costs requires detailed feasibility studies before building prototypes. In the design process of a HEV, a drive cycle is necessary to evaluate its fuel consumption, emissions, and performance. Furthermore, once this driving cycle is known, one can determine the mission of the vehicle, the optimal component sizes, and the costs of hybridization [1], [2]. These characteristics make the driving cycle selection an important step in the design of new HEVs [3].

From country to country, or built by various parties, standardized driving cycles are numerous and exist in two forms [4]: transient cycles, such as the VAIL2NREL driving cycle, and modal cycles such as the New European Driving Cycle (NEDC), both shown in Fig. 1. The primary difference is that transient cycles involve many velocity variations, typical of on-road driving conditions, whereas modal cycles are a compilation of constant acceleration, deceleration, and possibly constant velocity segments. For example, the NEDC consists of a succession of acceleration and deceleration segments that represent city driving (velocity up to 15 m/s) and an acceleration to highway velocity (up to 35 m/s). Modal cycles can easily highlight per-segment vehicle performance criteria (e.g., acceleration or deceleration) and are often used for testing powertrain components (e.g., emission tests). Yet, since they are not very realistic, in powertrains design, transient cycles are preferred.

Since driving cycles are considerably different, a HEV optimally designed using one cycle will not be optimally designed with respect to another cycle. Therefore, having a realistic cycle that reflects real-world driving scenarios is crucial for design. To avoid this drawback (of designs based on a single cycle), in [5]–[8], several cycles are used, and the sensitivity of the design is investigated. Although this is an improved approach for the cycle selection, using two or a very limited amount of cycles will not reach a HEV design robust to all real-life driving scenarios. Moreover, when sizing and control studies are done with optimization algorithms, such as dynamic programming, long cycles will increase the simulation times significantly.

To decrease the length of several cycles, in [9], microtrips from several cycles are combined to create a typical driving
cycle, i.e., one that has certain statistical characteristics similar to the original cycles (percentage of city, highway and suburban driving, average speed, and idling time).

In recent years, to improve the quality of the synthesized cycle speed, other methods have been proposed [10]–[12]. In addition to matching the driving cycle statistics, these include the random nature of a driving cycle as well. These methods are mostly based on Markov Chains and show better results than previous methods. In [13], a 2-D Markov chain is introduced that also considers information of acceleration. However, no method so far considers information about road altitude. This is an important characteristic of the driving cycle, and neglecting it leads to unrealistic HEV designs.

In this paper, we introduce and compare two cycle synthesis methods, which also consider altitude information. Building upon existing methods, we discuss the usability of 2-D or 3-D Markov chains for cycle generation. Furthermore, we use both statistical and spectral analysis to identify good cycle candidates. As proven by results, the proposed method can create a new cycle, representing measured cycles well.

The remainder of this paper is organized as follows. In Section II, existing methods for driving cycle synthesis are explained. Then, in Section III, a detailed description is presented for the used methods, based on single- and multidimensional stochastic Markov chains. Next, in Section III two new methods are introduced that also contain road slope information. In Section IV, results of extensive simulations of the chosen method are discussed. Finally, in Section V, conclusions on the synthesis of driving cycles with road slope information are drawn.

II. EXISTING DRIVING CYCLE SYNTHESIS METHODS

In recent years, several methods were proposed for driving cycle synthesis. Starting from concatenating measured cycles [6]–[8], to cut-and-clip type of methods [9] and, later on, to Markov Chain based-methods [13], [14], all methodologies have in general the following structure:

Step 1: preprocessing of input (measured) cycles, i.e., segmentation into micro trips [11], [12];
Step 2: synthesis of a new driving cycle by using rule-based [9] or using Markov Chains [10]–[13], [15];
Step 3: validation of the results, i.e., each generated cycle, by using a statistical, spectral, or time-domain analysis. Often, a combination of multiple criteria is used.

Most often and when needed, this procedure is an iterative one as it will be exemplified also later on. Every existing method to synthesize driving cycles that will be discussed here, uses measured real-world driving data as input.

Note that, given one measured cycle $\Lambda_m$

$$\Lambda_m = \left[ \begin{array}{c} v(t) \\ \theta(t) \end{array} \right] \quad \forall t \in [0, t_f]$$

with $v$ being the velocity, $\theta$ being the slope of the driving cycle, and $t_f$ being the final time of the driving cycle; the current methods assume

$$\theta(t) = 0 \quad \forall t \in [0, t_f].$$

Moreover, as mentioned earlier, some methods use also the acceleration $a(t)$, either measured or found by $a = dv/dt$. The first and third steps are not included in all existing driving cycle procedures, whereas the synthesis procedure is always present.

1) Data Preprocessing: Before using the real-world driving data for driving cycle synthesis, in several works, the data are preprocessed. A first way to do this is by segmenting the velocity values into different classes [10], [13]. For instance, all velocities can be segmented into discrete velocity classes, referred to also bins, with a constant bin width. For example, for a 0.2-m/s (0.72 km/h) bins width, all $v(t) \in [0, 0.2]$ m/s are in the first velocity class $C^1$, all velocities $v(t) \in [0.2, 0.4]$ m/s are in the second class $C^2$, and so forth. Thus, prior to the cycle synthesis method, the velocity vector is defined as

$$v(t) = \left[ \begin{array}{c} 0, \ldots, 0.2 \\ 0.2, \ldots, 0.4 \\ \vdots \\ v_{\text{max}} - 0.2, \ldots, v_{\text{max}} \end{array} \right]_{C^N}$$

A second method to preprocess the measured driving data is to divide the original velocity profile into microtrips. These form different modal events, such as cruising, idle time, acceleration, and deceleration, based on velocity and acceleration characteristics [11], [12]. Thus, the velocity vector becomes (4), shown at the bottom of the page. Every modal event collection contains microtrips of the original driving cycle, having the same velocity and acceleration characteristics, but can be different in duration (i.e., the bin width is variable). The process of segmenting the original driving cycle into microtrips is usually rule based (engineering knowledge).

A more elaborated driving condition recognition tool is introduced in [12]. In this method, several steps are performed

$$v(t) = \left[ \begin{array}{c} v_{1_{\text{min}}}, \ldots, v_{1_{\text{max}}} \\ v_{2_{\text{min}}}, \ldots, v_{2_{\text{max}}} \\ \vdots \\ v_{m_{\text{min}}}, \ldots, v_{m_{\text{max}}} \end{array} \right]$$

1The superscript 1 does not refer to “C to the power 1” but indicates the first velocity class.
prior to the driving cycle synthesis step. Smoothing of the instantaneous acceleration, computed directly from the measured velocity data, was done using the Epanechnikov density kernel smoothing algorithm. This was done to eliminate numerical differentiation errors. Next, using statistical values and a neural network, data segmentation and classification are applied. The microtrips are glued together, such that parts of the original driving cycle are retained in the synthesized driving cycle.

2) Synthesis Procedure: The different synthesis and compression processes use rule-based, statistical, or stochastic methods to synthesize a driving cycle that resembles the original driving cycle. Rule-based methods rely on expert opinion and target at matching a limited number of characteristics from the measured driving cycles [9], [16]. Such a criterion can be represented by the percentages of city, suburb, and highway speeds.

Statistical methods aim at matching statistical parameters on velocity and acceleration, such as percentage time in positive acceleration, percentage of standstill time, and average velocity. The synthesis procedure of a driving cycle is based on an improvement in matching those parameters, by connecting segments of the original driving cycle. This is done to maintain the same characteristics as in real-world driving [17], [18].

To enhance the probabilities of the driving cycle synthesis procedure (in terms of randomness and choosing an arbitrary length for the synthesized driving cycle), a combination of statistic and stochastic methods for combining different segments of a real-world driving cycle is explained in [11], [19]. Stochastic methods use Markov chain theory to capture a model of the input driving cycles and to generate a representative driving cycle. This machine learning method has also been used in other fields for prediction, e.g., in estimating the energy consumption in buildings [20] or for estimating wind speeds [21]. Using Markov chains, to improve the results, in [10] and [13], the acceleration is added as an extra dimension for the transition probability matrix (TPM). This TPM is computed from velocity (and acceleration) classes with fixed bin width, as described in (3), whereas in [11] and [12], the TPM is computed from the probabilities of segments from the original driving cycle, as described in (4). In the latter two works, instead of computing the probabilities of transitions between velocities, the probabilities of transitions between different modal events [11] or microsegments [12] are computed. Modal events are, for example, acceleration, deceleration, cruising, and so on, whereas microsegments are segments of real-world driving, separated by consecutive stops. At every time step, instead of using velocity values and combining them to a driving cycle, this method works with segments of the original driving cycle. Therefore, this method generates a driving cycle that contains parts of the original driving cycle. Yet, it has the big disadvantage that it does not capture the characteristics of the complete measured cycles, as done in [10] and [13].

Thus far, the stochastic method that uses velocity and acceleration classes and generates a purely synthetic driving cycle, from [13] and [14], has shown better results then other existing methods. Likewise, using stochastically obtained Markov chain driving cycles, in [15], it is shown that one can reach improved vehicle designs at reduced computational costs. These results motivate the incorporation of stochastic driving cycles in HEV optimization studies [15].

3) Post-Processing and Validation: Several different ways to validate the synthesized driving cycle exist. These are based on the comparison of the generated cycle with the input data (measured cycles). One such method is used in [10]–[12], where the distributions of velocity and, optionally, acceleration, within the cycle are analyzed.

A second method is to compare statistical parameters of both the measured cycles and the synthesized cycle [11], [13]. For instance, these parameters are average velocity, maximal velocity, standard deviation of acceleration, and so on. In [13], an analysis is made on which of these are more crucial for obtaining a good synthesized cycle for HEV design. Ultimately, as done in [10], one can analyze the power spectra resemblance of the velocity (and optionally acceleration), between the original and synthesized driving cycles.

Motivated by these existing methods and the desire to include altitude information in the synthesized driving cycle, in this paper, we investigate the applicability of multidimensional (multistates) Markov chains for driving cycle synthesis. In addition, to eliminate the rule-based approach, we apply a post-processing method based on fixed width classes. For validation, we combine all of the given described methods (e.g., statistical analysis and spectral analysis) to select a candidate cycle, validate the proposed method, and analyze the quality of the result.

III. DRIVING CYCLE SYNTHESIS INCLUDING SLOPE INFORMATION

The desired outcome of a newly developed synthesis procedure is a driving cycle with a velocity $v(t)$, acceleration $a(t)$, and road slope profile $\theta(t)$. These characteristics are all important in control and design studies of HEVs, with slope being particularly important for heavier vehicles. To introduce such a methodology, one should be aware of the inherit dependence between these parameters and capture this in the newly developed cycle.

A. Correlation Between Velocity, Acceleration, and Slope

The longitudinal dynamics of a vehicle are given by $m_v(dv/dt)v(t) = F_t - (F_r + F_a + F_g)$, where $F_r$ represents the rolling resistance force, $F_a$ represents the aerodynamic friction force, $F_g$ represents force loss caused by gravity in nonhorizontal driving, and $m_v(dv/dt)v(t)$ represents the kinetic force need to achieve desired acceleration $(d/dt)v(t)$. From this, it can be deducted that a change in the road angle $\theta$ produces a change in $F_g$, which influences the vehicle speed [22].

Considering existing methods [10]–[13] for driving cycle synthesis, two parallel synthesis processes can be performed, one to synthesize a velocity profile $\dot{v}(t)$ and another to synthesize a road slope profile $\dot{\theta}(t)$. An example of the resulting velocity and road slope profile is shown in Fig. 2. This is an undesired unrealistic output since in this case, there is no dependence between $\dot{v}(t)$ and $\dot{\theta}(t)$. When the road slope is positive (driving uphill), the acceleration will be slower than for a road
slope that is zero or even negative (downhill driving). Furthermore, it is very likely that the maximum velocity is reached at \( \theta \leq 0 \), not for \( \theta > 0 \). Therefore, the velocity profile and road slope profile should be synthesized together, such that their dependence is retained in the synthesized driving cycle.

### B. Driving Cycle Models Based on Discrete Markov Chains

A Markov chain is a random process on a discrete state space for which the conditional probability property holds. This implies that the probability of an event to occur, given that another event has occurred, at any future time depends only on its value at the current time. In other words, the future and the past are conditionally independent given the present. The description of the present state at time fully captures all the information that could influence the future step.

1) One-Dimensional Markov Chain: As shown in [13] and [14], a driving cycle \( \Lambda_m \) defined in (1) can be characterized through a discrete-time Markov chain. Thus, from here onwards, we restrict ourselves to the discrete-time case [23]. For this, the conditional probability property is defined as follows.

**Definition 1:** Let \( \{ X_k \} \) be a discrete-time stochastic process that takes its values in a space \( S = \{ s_1, s_2, \ldots, s_r \} \). Let \( s_j \subset S \). If

\[
P \{ (X_{k+1} = s_j) | X_0, X_1, \ldots, X_k \} = P \{ (X_{k+1} = s_j) | X_k \}
\]

then \( \{ X_k \} \) is said to be a discrete-time Markov process.

**Definition 2:** The probability \( P \) of taking a step from current state \( s_i \subset S \) to a next state \( s_j \subset S \), which is denoted \( P_{ij} \), is

\[
P_{ij} = P(X_{k+1} = s_j \mid X_k = s_i).
\]

All transition probabilities \( P_{ij}, \forall s_i, s_j \in S \) are captured in a TPM denoted by \( F \). This matrix \( F \) has the property that all its entries \( P_{ij} \geq 0 \) and all its row sum (i.e., all probabilities of leaving a state) are unity, i.e.,

\[
\sum_j P_{ij} = \sum_j P(X_{k+1} = s_j \mid X_k = s_i) = 1.
\]

The speed vector of a driving cycle, which is postprocessed as in (3) or (4), will become

\[
v \in C^1 \quad \cdots \quad v \in C^N
\]

\[
F = \begin{pmatrix}
    v(t) \\
\end{pmatrix}
\]

We denote this as a 1-D Markov chain that models the speed of one or more given cycles. The incorporation of acceleration or slope increases the dimensionality of \( F \) and will be discussed in the following. Moreover, the construction of a new cycle, using one or multiple-dimensional Markov models, will be addressed in Section III-D.

### C. Two-Dimensional Markov Chain

Building a driving cycle synthesis method that also considers slope or acceleration in addition to velocity requires a 2-D Markov chain to be considered. In the 2-D case, the states in \( F \) are defined by two parameters, for instance, \( v(t) \) and \( \theta(t) \). Thus, every state \( s \in S \) is a combination of the velocity class \( C^v \) and the road slope class \( C^\theta \). Now, \( F \) contains the probabilities to go to \( C^v \) and \( C^\theta \) at the next time step.

Such a framework is applied in [13] and [14] to synthesize a driving cycle considering velocity and acceleration. In many real-world driving cycles, the acceleration is not measured directly but obtained from differentiating the measured velocity. Furthermore, real-world velocity signals are often measured only every second. This large sampling time leads to an imprecise acceleration signal where specific acceleration information (e.g., the aggressiveness of the driver while driving) is lost. Thus, generating an acceleration profile is not necessary in the driving cycle synthesis method. Motivated by these, in this paper, we will use a 2-D Markov-chain-based method to capture the velocity and the slope information, rather than the velocity and the acceleration.

Analog to (8), to construct \( F, v(t) \) and \( \theta(t) \) from (1) are segmented into classes. As motivated in Section II we assume for these fixed class width, with a predefined number of classes \( N \) for slope and \( M \) for velocity, which are defined by \( \Delta v = (v_{\max} - v_{\min})/M \) and \( \Delta \theta = (\theta_{\max} - \theta_{\min})/N \). The subscripts \( (\min) \) and \( (\max) \) represent the minimum and maximum values of these parameters in \( \Lambda_m \). The driving cycle now has \( M \) velocity classes and \( N \) road slope classes. Every road slope value and every velocity value falls in classes \( C^v \) and \( C^\theta \), i.e.,

\[
v(t) \in C^v = \{ C^1, C^2, \ldots, C^M \}
\]

\[
\theta(t) \in C^\theta = \{ C^1, C^2, \ldots, C^N \}
\]

The TPM is a 2-D matrix \( F \in \mathbb{R}^{M \times N} \), with \( M \) rows for the velocity classes and \( N \) columns for the road slope classes. Every element of \( F \) consists of a \( (M \times N) \) matrix containing the probabilities of going from the current state \( s_i \) at \( t_k \) to the next state \( s_j \) at time \( t_{k+1} \).
D. Selecting Synthesized Driving Cycle Samples

To synthesize a new driving cycle \( \Lambda_s = [\hat{v}(t) \ \hat{\theta}(t)]^T \), an initial state \( s_i \) and \( F \) are used to compute future states \( s_j \). To this purpose, a Monte Carlo sampling method [24] based on a Poisson distribution of the probabilities is used. This Markov chain–Monte Carlo (MCMC) technique has been successfully used in earlier driving cycle synthesis methods [10] and in weather forecasting (e.g., wind speed estimation models [21]). Furthermore, the Poisson distribution used in the synthesis process is built from the measured (input) driving cycle.

To apply MCMC, the matrix \( F \) is transformed into a new matrix \( T \), using two steps. First, all the probabilities within each element of the matrix \( F \) (i.e., \( F_{i,v}(j_v,j_\theta) \)^2 \( \forall i,j \)) are transformed in a vector \( \tilde{T}_{i,v,i_\theta} \in \mathbb{R}^{1,M,N} \) as

\[
\tilde{T}_{i,v,i_\theta} = \left[ [F_{i,v,i_\theta}^1, \ldots, F_{i,v,i_\theta}^M]^T, \ldots, [F_{i,v,i_\theta}^N]^T \right]^T
\]

(10)

with \( \| \) representing the concatenation of two vectors. This results into a matrix \( \tilde{T} \), in which each element is a vector. Then, for each element of \( \tilde{T} \), the cumulative sum is determined, and the vector is augmented with 0 as follows:

\[
T_{i,v,i_\theta}(q) = \left[ 0, \sum_{p=1}^{q} \tilde{T}_{i,v,i_\theta}(p) \right] \quad \forall q \in [1,2,\ldots,M \cdot N] = \left[ 0, \tilde{T}_{i,v,i_\theta}(1), \tilde{T}_{i,v,i_\theta}(1) + \tilde{T}_{i,v,i_\theta}(2), \ldots \right].
\]

(11)

By doing this two-step transformation of the matrix \( F \), every element in the newly obtained matrix \( T \) is represented by a vector with values starting at 0 and ending at 1. Given the construction from (11), it holds that the difference between two consecutive elements of \( T_{i,v,i_\theta} \) is equal to the probability of going to this state \( s_i \) in \( F \). For instance

\[
F_{i,v,i_\theta}(6,1) = T_{i,v,i_\theta}(6) - T_{i,v,i_\theta}(5).
\]

(12)

Therefore, the probability that a transition to \( F_{i,v,i_\theta}(6,1) \) occurs is equal to the probability that a randomly generated number \( \mu \in [0,1] \) falls in the interval between \( T_{i,v,i_\theta}(5) \) and \( T_{i,v,i_\theta}(6) \). In Monte Carlo sampling, a repeated random sampling \( \mu \in [0,1] \) is selected to generate new samples. When selecting \( \mu \in [0,1] \), the first index in the corresponding row of \( T_{i,v,i_\theta} \) for which

\[
\mu \leq T_{i,v,i_\theta}(z)
\]

holds is the index that should be selected, i.e., the upper limit of the interval in which \( \mu \) falls. In (13), \( i_v, i_\theta \) stand for the classes \( C_{i_v} \) and \( C_{i_\theta} \) at the current time \( t_k \), and the parameter \( z \) indicates the index of the sample that will be selected. Thereafter,

the corresponding classes of \( j_v \) and \( j_\theta \) in the next time step \( t_j \) can be derived from this index \( z \). For example, given a randomly generated \( \mu \cong 0.25 \), assume that the corresponding indexes of cell \( T_{i,v,37} \) are \( T_{i,v,37}(37) \) and \( T_{i,v,37}(38) \). Therefore, at \( t_{k+1} \), \( C_{j_v}^w \) and \( C_{j_\theta}^\theta \) that correspond to this index \( z = 38 \) will be chosen. This process of selecting a sample for driving cycle synthesis is graphically shown in Fig. 3. Converting the index of \( T_{i,v,i_\theta}(z) \) into, respectively, the velocity class \( C_{i_v}^w \) and the road slope class \( C_{i_\theta}^\theta \) of the new driving cycle sample is done by

\[
C_{j_v}^w = T_{i,v,i_\theta}(z) \cdot M
\]

\[
C_{j_\theta}^\theta = \left[ \frac{T_{i,v,i_\theta}(z)}{N} \right].
\]

(14)

Note that in (14), the operators \( [\ ] \) and \( \| \) stand for floor and ceiling functions, i.e., \( [T_{i,v,i_\theta}(z)/M] = \{ \alpha \in \mathbb{Z} \mid \alpha \leq T_{i,v,i_\theta}(z)/M \} \) and \( [T_{i,v,i_\theta}(z)/M] = \{ \beta \in \mathbb{Z} \mid \beta \geq (T_{i,v,i_\theta}(z)/M) \} \).

When \( C_{j_v}^w \) and \( C_{j_\theta}^\theta \) are known, the new velocity value \( \hat{v} \) and road slope value \( \hat{\theta} \) are obtained from taking the average value of the lower and upper limits of the class, assuming a normal distribution within each class, i.e.,

\[
\hat{v} = v_{avg}(\eta) \quad \forall \eta \in C_{j_v}^w,
\]

\[
\hat{\theta} = \theta_{avg}(\eta) \quad \forall \eta \in C_{j_\theta}^\theta.
\]

(15)

This procedure, to select driving cycle samples for drive cycle synthesis, is a step in the cycle synthesis method. This complete method is shown in Fig. 4 for a 2-D case and holds for single- or multidimensional Markov chains.

In addition to the method presented here for sample selection, updating the probability based on the selected samples improves the resulting synthesized driving cycle. By doing this, for sufficiently long synthesized driving cycles, the distribution of \( \hat{v}(t) \) and \( \hat{\theta}(t) \) will approach asymptotically the original distribution. This update is made by subtracting the transitions already selected from the TPM. Once the update is applied, the matrix \( T \) from (11) is recalculated, and the process is reiterated, as shown in Fig. 4.
E. Cycle Evaluation and Validation

The initial state of the velocity and the road slope \( s_i \) are selected randomly from the TPM. In this way, the starting point of the newly synthesized driving cycle is always different. The remaining instances are selected as described in Section III-D and as shown in Fig. 4. Due to the stochastic characteristic of the process, the newly created cycle will not always end at a velocity \( v = 0 \) m/s. This can pose a problem for hardware testing, as for example on a dynamometer, but should not pose a problem for simulations. Furthermore, the process should be iterated until the synthesized driving cycle is validated, i.e., matching the original driving cycle(s) according to predefined requirements on similarity.

The validation of a new cycle is often done based on matching statistical parameters between the original and the synthesized driving cycle in time domain [11]–[13]. Examples include parameters based on velocity (e.g., average velocity, maximal velocity, and so on) and acceleration. The second criterion is to compare the newly developed cycle to the input data in the frequency domain. Using the power spectral density, one can make a conclusion as to whether the cycles match in the interest frequency band (≤ 0.2 Hz) [10].

In this paper, to validate a newly developed cycle and to keep or reject a solution, we use all the aforementioned methods. The parameters considered are shown in the following:

- **Mean velocity**
  \[ v_a = \frac{1}{n} \sum_{i=1}^{n} v_i \]

- **Standard deviation velocity**
  \[ s_v = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (v_i - v_a)^2} \]

- **Maximal velocity**
  \[ v_{\text{max}} = \max_{i=1}^{n} (v_i) \]

- **Mean acceleration**
  \[ a_a = \frac{1}{n} \sum_{i=1}^{n} a_i \]

- **Standard deviation positive acc.**
  \[ s_a = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (a_i - a_a)^2}, \forall a_i > 0 \]

- **Maximal acceleration**
  \[ a_{\text{max}} = \max_{i=1}^{n} (a_i) \]

- **Maximal deceleration**
  \[ a_{\text{max}} = \max_{i=1}^{n} (a_i), \forall a_i < 0 \]

We restrict these parameters of each candidate cycle to match the ones of the measured driving cycle within maximum 10% difference, as also used in the work of Brady et al. [12]. This, in order to guarantee that a particular synthesized cycle satisfactorily represents the input real-world driving cycle. In the works of [11] and [14], these parameters are used to analyze the synthesized cycle but not in the generation process. Additionally, as this method considers road slope information, we introduced statistical parameters for road slope in the following and constrain them to match within a 15% bound.

- **Mean slope**
  \[ \theta_a = \frac{1}{n} \sum_{i=1}^{n} \theta_i \]

- **Standard deviation slope**
  \[ s_{\theta} = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (\theta_i - \theta_a)^2} \]

- **Maximal slope**
  \[ \theta_{\text{max}} = \max_{i=1}^{n} (\theta_i) \]

- **Minimal slope**
  \[ \theta_{\text{max}} = \min_{i=1}^{n} (\theta_i) \]

In addition to these parameters, to compare the 2-D method to the 3-D method (described in Section III-F), we also analyze computational time. Parameters such as the percentage of idle time and number of stops are excluded from our analysis since they are not present in the measured cycle.

F. Three-Dimensional Markov Chain

When available, in addition to the velocity and the slope, one can use the acceleration as an input to the Markov chain. This requires a 3-D Markov model, which can be an extended and more complex form of the two-parameter case. To construct the new matrix \( F \), the acceleration, \( a(t) \), \( v(t) \), and \( \theta(t) \) information are segmented into classes \( C^v \), \( C^\theta \), and \( C^a \). By choosing the number of classes \( M \), \( N \), and \( O \), the intervals are defined similar to the 2-D case, by \( \Delta v = (v_{\text{max}} - v_{\text{min}})/M \), \( \Delta \theta = (\theta_{\text{max}} - \theta_{\text{min}})/N \) and \( \Delta a = (a_{\text{max}} - a_{\text{min}})/O \). The driving cycle now has \( M \) velocity classes, \( N \) road slope classes, and \( O \) acceleration classes. Every velocity, every road slope, and every acceleration falls in a class \( C^v \), \( C^\theta \), \( C^a \), and a TPM \( F \in \mathbb{R}^{M \times N \times O} \) can be built. Moreover, every element of this matrix is composed of a \((M \times N \times O)\) matrix again. Each of these \((M \times N \times O)\) matrices contain the probabilities of going from the current state \( s_i \) at \( t_k \) to a next state \( s_j \) at \( t_{k+1} \). An example
of such a matrix is shown in Fig. 5, where the probabilities for leaving a state with classes \( C_t^v = 2 \), \( C_t^\theta = 5 \), and \( C_t^a = 1 \) are shown. This matrix is denoted by \( F_{2,5,1} \), and from it, one can observe that the probability of staying in the same velocity, road slope, and acceleration class is the highest, namely \( F_{2,5,1}(2, 5, 1) = 0.1538 \). The probability of going to \( C_j^v = 3 \), \( C_j^\theta = 4 \), and \( C_j^a = 2 \) is \( F_{2,5,1}(3, 4, 2) = 0.0308 \), and the probability of a transition to a state with \( C_j^v = 1 \), \( C_j^\theta = 6 \), and \( C_j^a = 3 \) is \( F_{2,5,1}(1, 6, 3) = 0 \).

To synthesize a new driving cycle, new samples are selected using the procedure explained in Section III-D. Note that the dimensionality of the matrices \( F \) and \( T \) have increased, with each element of \( T \in \mathbb{R}^{1, M \times N \times O} \) being defined as

\[
T_{v_i, \theta_i, a_i} = \left[ F_{v_i, \theta_i, a_i}(1, 1, 1), \ldots, F_{v_i, \theta_i, a_i}(M, 1, 1) \right]^T \parallel \left[ F_{v_i, \theta_i, a_i}(1, 2, 1), \ldots, F_{v_i, \theta_i, a_i}(M, 2, 1) \right]^T \parallel \ldots \parallel \left[ F_{v_i, \theta_i, a_i}(1, N, O), \ldots, F_{v_i, \theta_i, a_i}(M, N, O) \right]^T.
\]

(16)

Sequentially, the cumulative sum is determined over all these probabilities, i.e.,

\[
T_{v_i, \theta_i, a_i}(q) = \sum_{p=1}^{q} T_{v_i, \theta_i, a_i}(p) \quad \forall q \in [1, 2, \ldots, M \cdot N \cdot O].
\]

(17)

By following the process described in Fig. 4, the index \( z \) is found with the use of a randomly generated number \( \mu \), by \( \mu \leq T_{v_i, \theta_i, a_i}(z) \). The corresponding classes, with index \( z \), of velocity and road slope are found as in (14) and for acceleration by

\[
C_j^a = \frac{T_{v_j, \theta_j, a_j}(z)}{M \cdot N}.
\]

(18)

The procedure is then completed by the steps explained in Section III-D and Fig. 4.

IV. RESULTS

Given measured cycles, to determine which method is most suitable for the construction of a new driving cycle, here, we compare the 2-D Markov chain-based method and the 3-D Markov chain-based method.

To validate the two proposed methods, one measured driving cycle has been used, which is representative for heavy-duty long-haul routes across Europe (see Fig. 6). The length of this cycle is 29,282 data points that corresponds to 8.3 h of driving (with a sampling time of 1 s). In this cycle, both the road slope and the velocity are measured.

For this analysis, \( v(t) \), \( \theta(t) \), and \( a(t) \) are segmented into 50 classes; hence, \( M = N = O = 50 \). For the particular input cycle, this results, respectively, in class widths of

\[
\Delta v = 0.50 \text{ [m/s]} \quad \Delta \theta = 0.28 \text{ [\degree]} \quad \Delta a = 0.072 \text{ [m/s}^2].
\]

(19)

The discretization resolution (\( M, N, \) and \( O \)) should be small enough to be representative of the vehicle dynamics on the complete velocity, slope, and acceleration ranges, as also motivated in [10] and [13]. Moreover, this selection is a limitation for the accuracy of the results and a computational challenge. Next, the desired driving cycle length is chosen to be 4500 data points.

A. Two-Dimensional Method Compared With the Three-Dimensional Method

Here, the methods described in Section III-C and F are compared, as schematically shown in Fig. 7. This is done to evaluate the results accuracy and speed of using one or another method. As explained in Section III, increasing the driving cycle synthesis method, from two to three parameters, also increases the transition probability matrices from being 2-D to 3-D. This greatly effects both the complexity of calculations and the computation time needed to synthesize a complete new driving cycle. By averaging the results on the computation time and resemblance between the original and the synthesized driving cycle gives a good indication of the performance of both methods. For each of the two methods, the resemblance of the two cycles can be improved when performing more iterations. However, since the process is purely stochastic, there is no guarantee that if a particular cycle generate with one method (after \( q \) iterations) is a good candidate, the other method will
Fig. 7. Schematic representation of the compared methods.

Table I: Comparison of the 2-D and 3-D Methods Using Statistical Results for Velocity \( v(t) \), Slope \( \theta(t) \), and Acceleration \( a(t) \) with Respect to the Error Percentage

<table>
<thead>
<tr>
<th>Variable Name (Criteria)</th>
<th>2D Markov chain using ( v(t) ) and ( \theta(t) )</th>
<th>3D Markov chain using ( v(t) ), ( \theta(t) ), and ( a(t) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean velocity</td>
<td>12.66%</td>
<td>13.04%</td>
</tr>
<tr>
<td>Standard deviation velocity</td>
<td>60.71%</td>
<td>56.18%</td>
</tr>
<tr>
<td>Maximal velocity</td>
<td>1.87%</td>
<td>2.16%</td>
</tr>
<tr>
<td>Mean slope</td>
<td>1.58%</td>
<td>1.50%</td>
</tr>
<tr>
<td>Standard deviation slope</td>
<td>12.64%</td>
<td>16.07%</td>
</tr>
<tr>
<td>Maximal slope</td>
<td>5.15%</td>
<td>7.30%</td>
</tr>
<tr>
<td>Minimal slope</td>
<td>16.69%</td>
<td>17.32%</td>
</tr>
<tr>
<td>Mean acceleration</td>
<td>0.015%</td>
<td>0.08%</td>
</tr>
<tr>
<td>Standard deviation positive</td>
<td>24.36%</td>
<td>44.16%</td>
</tr>
<tr>
<td>Maximal acceleration</td>
<td>18.10%</td>
<td>21.71%</td>
</tr>
<tr>
<td>Maximal deceleration</td>
<td>24.18%</td>
<td>32.57%</td>
</tr>
<tr>
<td>Computation time</td>
<td>0.52 seconds</td>
<td>27.43 seconds</td>
</tr>
</tbody>
</table>

(5) increases significantly and poses computational challenges (for instance MATLAB memory problems). Yet, increasing the number of classes could lead to improved accuracy and decreased computation time, as will be discussed in the following. The presence of a measured acceleration signal, with a small enough sampling time, could possibly improve the results of the 3-D method.

B. Enhanced Performance Analysis for the 2-D Method

To evaluate further the 2-D proposed method and to find a valid synthetic cycle, the number of road slope classes \( M \) and velocity classes \( N \) are set to 200 classes. The desired length of the synthesized driving cycle \( n \) is chosen again at \( n = 4500 \). Given a measured driving cycle of 29 282 datapoints (29 282 seconds) shown in Fig. 6, this implies that the original driving cycle is reduced in length by approximately a factor of six. In 5433 iterations, a valid cycle is found as shown in Fig. 8. In this cycle, the correlation (interdependence) between the velocity and slope is evidenced. This is particularly important, since neglecting it leads to a generated cycle with a significantly higher power demand for the powertrain. Table II shows the error percentage between the measured and the synthesized cycle, where all the velocity and acceleration criteria \( \leq 10\% \) and all slope-related criteria \( \leq 15\% \). To compare further the matching of the velocity, the road slope, and the acceleration of the generated and the measured cycle, in Figs. 9–11, the distribution of these is shown. Furthermore, in Fig. 12, the power spectra for both the road slope and the velocity are depicted. The results...
Fig. 9. Histogram of measured and synthesized velocity using the 2-D method and iterative evaluation of candidate cycles.

Fig. 10. Histogram of measured and synthesized road slope using the 2-D method and iterative evaluation of candidate cycles.

Fig. 11. Histogram of the acceleration (measured and synthesized) using the 2-D method and iterative evaluation of candidate cycles.

demonstrate the resemblance of the real-life driving cycle with high accuracy.

This resemblance is shown also in Figs. 13 and 14, where the current velocity $v(k)$ and current slope $\theta(k)$ are depicted and compared with $v(k-10)$ and $\theta(k-10)$ (10 s in the past). With increasing time delay, these figures shift from a straight line to a scatter plot between zero and the maximum velocity. The shorter the generated cycle is required to be, the more difficult it is to obey the $\leq 15\%$ slope related criteria. Nonetheless, the length of the desired cycle is not the purpose of this paper and will not be considered here.

Increasing the number of classes $M$ and $N$ leads to better accuracy of the results, but at the same time, it increases the computational time. The same tradeoff holds for choosing the synthesized driving cycle length $n$ closer to the original driving cycle length. Whereas, in this example, only one driving cycle is used as input, multiple driving cycles can be used as input as well, such that the synthesized driving cycle contains characteristics of every input cycle. If the number of classes is too small, then the validation criteria from Table II might...
be impossible to achieve. When this is the case, the validation criteria should be relaxed or the number of classes redefined. For instance, one could reason that the slope information should not be sampled as often or one could segment the velocity, acceleration, and slope in different number of classes for the TPM.

These results demonstrate that the proposed 2-D method to generate driving cycles is able to synthesize a good new driving cycle. The outcome of the process is a synthetic driving cycle that represents its original in terms of time- and frequency-domain performance criteria. Moreover, the use of acceleration, as a third parameter in the Markov process, does not bring a significant improvement in the resemblance, but it increases the computational burden significantly (for a cycle with 1-s sampling time). If actual measurements of acceleration would be available with a lower sampling time, this hypothesis should be revalidated. Yet, as long the acceleration is derived from the measured velocity (see Fig. 7), the high accuracy obtained with the 2-D method on the acceleration performance criteria motivates the usage of the 2-D method over the 3-D method (i.e., with lower sampling time).

Each candidate cycle can be used in assessing the vehicle’s performance, for which a vehicle model is necessary, as used in [25]. It can occur that the power demanded by a synthesized cycle is larger then what the vehicle can deliver. If that is the case, this would require a postprocessing of the cycle and is not addressed here. To further improve the results, the optimal number of velocity and road slope classes $M$ and $N$ can be found. Moreover, depending on the designer, the choice of the synthesized driving cycle length can be changed.

V. CONCLUSION

The possibility of creating a synthetic driving cycle, which is short and representative of a measured driving cycle, is very important in hybrid or electric vehicle design. By using a synthetic driving cycle, design simulation time can be significantly decreased. Moreover, the characteristics of the driving cycle used immediately influence the size and control parameters of the vehicle. Using a short synthetic cycle, which is representative of more real driving cycles, eliminates the need to iteratively simulate and analyze power train designs for several driving cycles, such as driving in the city or on the highway. Hence, the use of one compact driving cycle that reflects all real-world driving, is preferred above the use of multiple driving cycles or one very long driving cycle.

This paper has introduced two novel methods to synthesize a driving cycle which are required as input in powertrain design studies. In addition to the attributes considered so far in literature (i.e., velocity and acceleration), this methods consider also road the slope information from the measured driving cycle. Using Markov chain theory and building upon existing methods (Monte Carlo sampling and the cumulative Poisson distribution function), two methods were introduced and compared using measured data. Both time-domain and frequency-domain analyses were used to validate the results. In this regard, we show that the proposed 2-D method is able to synthesize or compress driving cycles fast and with high accuracy. Moreover, the method is independent of the input cycle length or number, and it is therefore widely applicable. Further work will address the influence of this driving cycle and its specific parameters on the optimal design and control of HEVs. More insight in the effect of, for example, average velocity or standard deviation of velocity on the component sizes of a HEV, could improve the robustness of HEV design and reduce the effect of cycle beating (where a vehicle is designed based on a driving cycle that does not reflect real-world driving).

REFERENCES


Emilia Silvas received the B.Sc. degree in automatic control and computer science from Politehnica University of Bucharest, Bucharest, Romania, in 2009 and the M.Sc. degree in systems and control and the Ph.D. degree from Eindhoven University of Technology, Eindhoven, The Netherlands, in 2011 and 2015, respectively.

Since 2016, she has been a Research Scientist with the Netherlands Organization for Applied Scientific Research (TNO) Technical Sciences, Helmond, The Netherlands, working in the area of cooperative vehicle systems and mobile robots. Her research interests include advanced control, system identification and modeling, machine learning techniques, and optimal system design.

Kobus Hereijgers received the B.Sc. and M.Sc. degrees in mechanical engineering from Eindhoven University of Technology, Eindhoven, The Netherlands, in 2013 and 2015, respectively. His M.Sc. thesis, in collaboration with DAF Trucks N.V., focused on driving cycle synthesis and its effects on hybrid-electric-vehicle fuel consumption. His main research interests include optimal design, modeling, and control of motion and transportation systems. In 2014, for three months, he was a Visiting Research Scholar with the Department of Mechanical Engineering, the University of Michigan, Ann Arbor, MI, USA.

Huei Peng received the Ph.D. degree in mechanical engineering from the University of California, Berkeley, CA, USA, in 1992.

He is currently a Professor with the Department of Mechanical Engineering, University of Michigan, MI, USA. He is also a Changjiang Scholar with the State Key Laboratory of Automotive Safety and Energy, Department of Automotive Engineering, School of Mechanical Engineering, Tsinghua University, Beijing, China. His research interests include the design and control of electrified vehicles, connected/automated vehicles, and adaptive control and optimal control, with emphasis on their applications to vehicular and transportation systems.

Dr. Peng is a Fellow of Society of Automotive Engineers and American Society of Mechanical Engineers.

Theo Hofman was born on September 5, 1976. He received the M.Sc. (with honors) and Ph.D. degrees in mechanical engineering from Eindhoven University of Technology, Eindhoven, The Netherlands, in 1999 and 2007, respectively.

From 1999 to 2003, he was a Researcher and a Project Manager with the R&D Department, Thales Cryogenics B.V., Eindhoven. From 2003 to 2007, he was a Scientific Researcher with Drivetrain Innovations B.V. (Punch Powertrain 2013+), Eindhoven. From 2007 to 2009, he was a Postdoctoral Fellow with the Control Systems Technology (CST) Group, Department of Mechanical Engineering, Eindhoven University of Technology. Since 2010, he has been an Assistant Professor with the CST Group. His research interests include the development of model-based system design methods for dynamical systems (including discrete topology design) with applications to powertrain systems.

Maarten Steinbuch (S’83–M’89–SM’02) received the Ph.D. degree from Delft University of Technology, Delft, The Netherlands, in 1989. Since 1999, he has been a Full Professor with the Control Systems Technology Group, Department of Mechanical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands. His research interests include the modeling, design, and control of motion systems and automotive powertrains.

Dr. Steinbuch serves as the Scientific Director of the Centre of Competence High Tech Systems of the Federation of Dutch Technical Universities. He also serves as the Editor-in-Chief of IFAC Mechatronics.