


Article

A Sticky Sampling and Markov State Transition Matrix Based Driving Cycle Construction Method for EV

Li Zhao ^{1,2}, Kun Li ^{1,*} , Wu Zhao ³, Han-Chen Ke ¹ and Zhen Wang ¹

¹ School of Mechanical and electrical Engineering, Beijing Information Science and Technology University, Beijing 100192, China; zhaoli@bistu.edu.cn (L.Z.); kehanchen2020020106@bistu.edu.cn (H.-C.K.); 2021020023@bistu.edu.cn (Z.W.)

² Collaborative Innovation Center of Electric Vehicles in Beijing, Beijing 100192, China

³ Jiangsu Vocational Institute of Architectural Technology, Xuzhou 221116, China; zhaowu@jsjz.edu.cn

* Correspondence: likun1995@bistu.edu.cn

Abstract: Driving cycle (DC) plays an important role in designing and evaluating EVs, and many Markov chain-based DC construction methods describe driving profiles of unfixed-line vehicles with Markov state transition probability. However, for fixed-line electric vehicles, the time-sequence of microtrips brings huge influences on their brake, drive, and battery management systems. Simply describing topography, traffic, location, driving features, and environment in a stochastic manner cannot reflect the continuity characteristics hidden in a fixed route. Thus, in this paper, we propose a sticky sampling and Markov state transition matrix based DC construction algorithm to describe both randomness and continuity hidden in a fixed route, in which a data structure named “driving pulse chain” was constructed to describe the sequence of the driving scenarios and several Markov state transition matrices were constructed to describe the random distribution of velocity and acceleration in same driving scenarios. Simulation and experimental analysis show that with sliding window and driving pulse chain, the proposed algorithm can describe and reflect the continuity characteristics of topography, traffic, and location. At the same time, the stochastic nature of the driving cycle can be preserved.

Keywords: driving cycle (DC); electric vehicle (EV); sticky sampling algorithm (SSA); Markov state transition probability matrix; driving pulse chain



Citation: Zhao, L.; Li, K.; Zhao, W.; Ke, H.-C.; Wang, Z. A Sticky Sampling and Markov State Transition Matrix Based Driving Cycle Construction Method for EV. *Energies* **2022**, *15*, 1057. <https://doi.org/10.3390/en15031057>

Academic Editors: Thierry A. Meynard and Jaime W. Zapata

Received: 22 December 2021

Accepted: 29 January 2022

Published: 31 January 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Considering fossil fuel depletion, greenhouse gas emissions, air pollution, and increasingly stringent fuel efficiency standards, the electric vehicle (EV) is becoming an alternative to internal combustion engines [1,2]. As a core technology in the automobile industry, the driving cycle (DC) plays an important role in designing and evaluating EVs. Most of the current DC construction methods were designed for traffic management, vehicular pollution measurement, and fuel consumption estimation on internal combustion engine vehicles (ICEVs) [3,4]. Torque and power characteristics, transmission efficiency, powertrain, driving behavior, and the energy recovery braking system of EV have significant differences with ICEV's. Simply designing and evaluating EVs with a traditional DC generation tool will generate a large error, especially in the process of energy management, battery state of charge (SOC) estimation, and driving range calculation. DC is a velocity-time profile that describes the driving characteristics of vehicles under real-world driving conditions [5,6]. To design an efficient powertrain and energy storage management system for EVs, a deep understanding of real-world DC is essential. At present, many researchers focus on DCs constructed for specific regions and vehicle types. “Microtrips” or the Markov chain are two common methods. The “microtrips” method defines driving activities between adjacent stops as “microtrips” and generates DCs by selecting them from observed data with some given criteria [7,8]. The correlation coefficient of characteristic

parameters and the similarity of velocity-acceleration joint probability distribution matrix are two commonly used comparison criteria. The “microtrips” method is not robust and reflective of the stochastic nature of the driving data [9,10].

To address the limitation of the “microtrips” method, Lin [11] proposed a mode-based cycle construction method, in which real-world driving is viewed as a random process and is categorized into four modes: acceleration, deceleration, cruise, and idle. Jakov Topic [12] presented a Markov chain-based method to synthesize multidimensional driving cycles. With a corresponding 8D transition probability matrix, the algorithm can generate a large set of synthetic driving cycles and can be implemented in a sparse form to improve computational efficiency and reduce memory requirements. Numerous synthetic driving cycles were generated by sampling from the state transition probability matrix (STPM), and the values of the selected statistical features were calculated for each of them. Based on the Markov and Monte Carlo methods, Zhao [13] constructed urban driving conditions for electric vehicles in Xi’an and conducted a comparison test of driving distance in simulation and real environments. It indicated that the urban driving conditions of electric vehicles developed by the proposed method can truly reflect the driving status of Xi’an. To really describe the operating characteristics of hybrid buses, Peng [14] constructed Zhengzhou urban driving cycle (ZZUDC) operating conditions, and statistically analyzed the operating characteristic with the Markov theory-based method. Correlation analysis shows that the proposed construction method can provide more help for the design of low-speed hybrid electric vehicles (HEVs). Having compared the driving patterns obtained in real-world with the Urban Dynamometer Driving Schedule (UDDS), new European driving cycle (NEDC), and Japan-10, Brady [15] pointed out that the difference is significant and real-world driving cycles are essential for EV powertrain design, battery management system construction and running range estimation. With speed-acceleration frequency distribution and Markov chain theory, Nguyen [16] constructed typical driving cycles for buses in Hanoi and provided a powerful tool for studying energy consumption and improving the air quality of buses. In order to improve the performance of energy management systems in a plug-in hybrid electric vehicle, Nguyen [17] proposes a global driving cycle construction method based on the real-time traffic information and Markov transfer matrix, with which the reconstructed driving cycle can further reflect the real-time road condition and the fuel efficiency of the HEV was improved by 19.83%. To discuss the big gap between standard driving cycle and actual driving cycle, caused by uneven data collection, inaccurate driving cycle construction method, and geographical factors, Ashtari [18] collected a large amount of real driving data and partitioned them with a k-means clustering method. With Markov models, the Winnipeg driving cycle (WPG01) was constructed by extracting the segments of speed and time from the constructed database. Compared with the matching characteristics of 14 driving parameters from the existing eight standard driving cycles, the constructed driving cycle can reflect the local driving characteristics more effectively. With the statistic and the Markov chain method, Gong [19] developed several Beijing driving cycles to accurately evaluate the performance of electric vehicles and accomplished the comparison with NEDC, the federal test procedure (FTP-75), and Japan-10–15.

In fact, driving profiles are dependent on a large set of properties, such as topography, traffic, location, driving characteristics, and environment [20]. Furthermore, simply modeling these features in a stochastic manner cannot reflect the driving nature of a vehicle. Especially for fixed-line HEVs, the order of the driving scenarios that appeared is not random [21,22]. The sequence of the “microtrips” plays an important role in braking energy recovery and fuel economy improvement. Thus, in this paper, we propose a sticky sampling and Markov state transition matrix based DC construction algorithm to describe both randomness and continuity, in which a data structure named “driving scenario chain” was constructed to describe the inheritance of the driving scenarios and a Markov state transition matrix was constructed to describe the randomness of velocity and acceleration that appeared in “microtrips”.

The proposed algorithm distinguishes itself from others by maintaining the sequence of the driving scenario (DS) steadily and generating driving pulse (DP) indeterminately. The contributions of this paper lie in: (1) The sticky sampling algorithm was used to calculate the velocity frequent item statistic vector, with which the computational time and storage space were greatly reduced; (2) DS label sequences were built and used to construct DCs, with which the order of DSs and the correlation between driving pulses appeared in a fixed route were clearly reflected. The rest of this paper is organized as follows: Firstly, the related work on the Markov state transition matrix and sticky sampling algorithm was reviewed. Secondly, the proposed algorithm was introduced and some theorem analyses were represented. Finally, several experiments were implemented to examine the parameters and the efficiency of the algorithm.

2. Related Work

2.1. Markov State Transition Probability Matrix

Markov chain is a random state sequence with Markov property, in which each state value depends only on the preceding finite number of states [23]. As a typical stochastic process, the Markov process is very suitable to deal with the random property of the driving cycle, in which the future states depend only on the current states and are irrelevant with past states.

Let $X(X_1, X_2, \dots, X_n)$ be a state space and every state X_i can be described by the combination of velocity v_i and acceleration a_i ; P_{ij} is the means transition probability from state X_i to state X_j , N_{ij} is the number of system switches from state X_i to X_j , and the Markov state transition probability P_{ij} can be calculated like this:

$$P_{ij} = \frac{N_{ij}}{\sum_{j=1}^n N_{ij}} \quad i = 1, 2, \dots, n; \quad j = 1, 2, \dots, n. \quad (1)$$

The state transition probability Matrix P on state space X can be expressed with Formula (2),

$$P = \begin{pmatrix} P_{11} & \dots & P_{1n} \\ \vdots & \ddots & \vdots \\ P_{n1} & \dots & P_{nn} \end{pmatrix} \quad (2)$$

In which, the transition probability P_{ij} must satisfy the following two conditions:

$$0 \leq P_{ij} \leq 1 \quad (3)$$

$$\sum_{j=1}^n P_{ij} = 1 \quad (4)$$

2.2. Sticky Sampling Algorithm

A sticky sampling algorithm is a sampling-based algorithm for computing an ϵ deficient synopsis over a data stream with singleton items [24]. When the algorithm is executed, three parameters (support s , error ϵ , and probability of failure δ) and a data structure DS (with the form (e, f) , f is the frequency of element e), and should be specified previously. The algorithm can compute an ϵ -deficient synopsis with a probability of at least $1 - \delta$ using at most $2 * \log(s^{-1}\delta^{-1}) / \epsilon$ expected number of entries.

The procedure for computing an ϵ deficient synopsis was presented below:

1. Empty the data structure DS and initialize the parameters s, ϵ, δ ;
2. For the i th incoming element e , if an entry for e already exists in DS, increment the corresponding frequency f ; otherwise, sample the element with rate r . If the element is selected by sampling, add an entry $(e, 1)$ to DS; otherwise, ignore e and move on to the next element in the stream;

3. Change the sampling rate r over the lifetime of the data stream like this: Let $t = \log(s^{-1}\delta^{-1})/\varepsilon$, sample the first $2t$ elements at rated $r = 1$, sample the next $2t$ elements at rated $r = 2$, sample the next $4t$ elements at rated $r = 4$, and so on;
4. Whenever the sampling rate changes, update entries in DS like this: For each entry (e, f) , repeatedly toss an unbiased coin until the coin toss is successful, diminishing f by 1 for every unsuccessful outcome; If $f = 0$ during this process, delete the entry from DS;
5. Output those entries in S where $f \geq (s - \varepsilon)N$;
6. $i = i + 1$, go (2).

3. The Sticky Sampling and Markov State Transition Probability Matrix Based DC Construction Algorithm

3.1. The Randomness and Inheritance of Microtrips

In general, driving profiles are dependent on a large set of properties, such as topography, traffic, location, driving characteristics of the operator, and environment [25]. These dependence features generate significant complexities, which make it virtually impossible to model each feature of a driving profile in a stochastic manner. For unfixed-line vehicles, the order of appeared driving scenarios is randomness. For fixed-line vehicles (such as a bus, sweeper, and sanitation and logistics truck), the order of driving scenarios is sequential. For EVs/HEVs/PHEVs with a complex energy management strategy, the different time-sequence of microtrips carry huge influences on the brake system, drive system, and battery management system.

In Figure 1(left) and Figure 2, the driving cycles of two fixed-line buses with the same route were shown. The driving scenario types were represented by different colored lines and the start and end positions were shown in Figure 1. From this, we can find that with the same topography/traffic conditions and different drivers, the appearance orders of driving scenarios on two contiguous buses are very similar. At the micro-level, the acceleration, cruising, deceleration, and idle stages of driving pulses of the two vehicles' DCs are very different. In Figure 1(right) and Figure 3, four kinds of driving scenarios of an unfixed-line private car (rural, highway, urban, and congestion) were represented by different colored lines. We can find that with the same driver but different topography/traffic conditions, the orders of driving scenarios on the same car are very different.

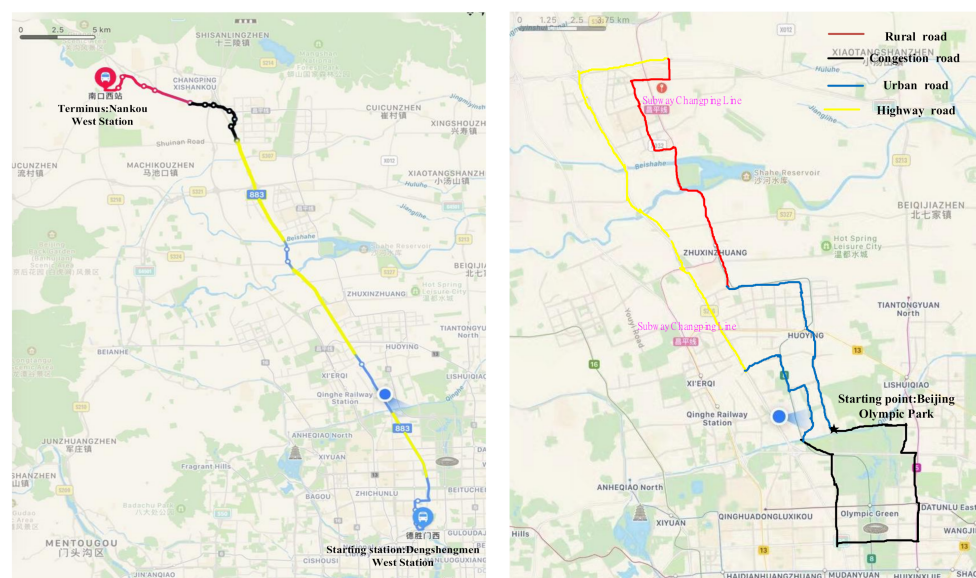


Figure 1. Routes of Bus 883 and a private car in Beijing.

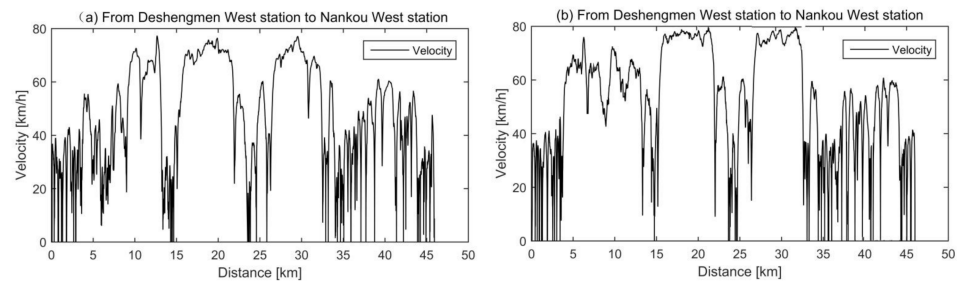


Figure 2. Driving cycles of two contiguous buses in Beijing.

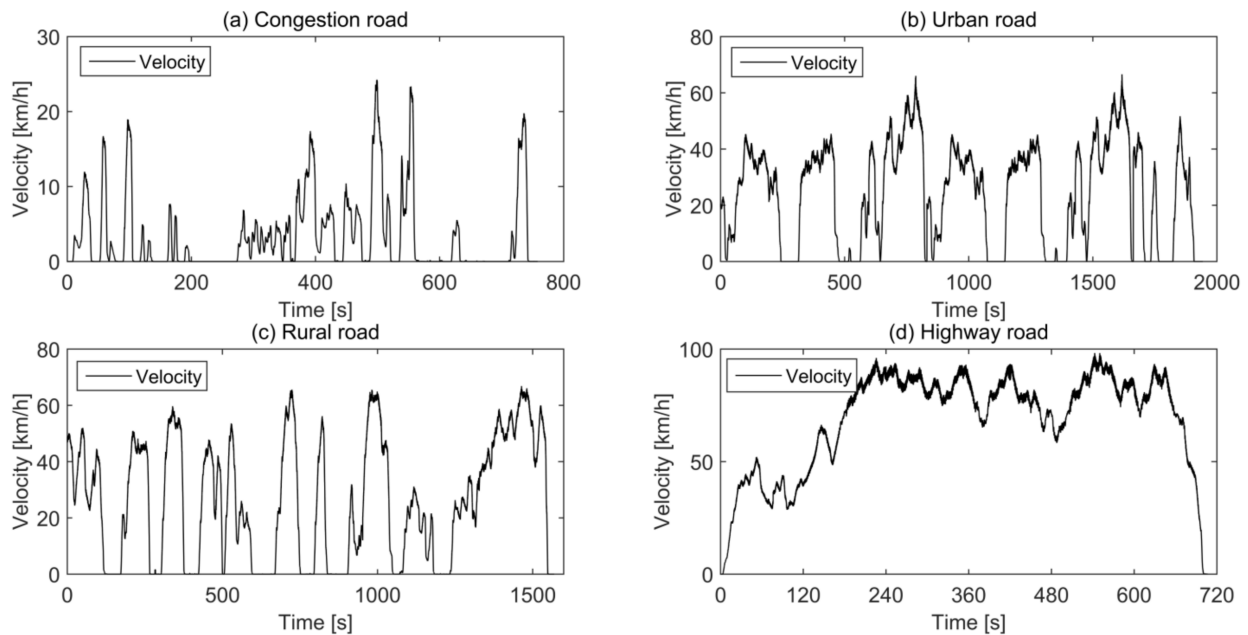


Figure 3. Four kinds of driving cycles of a private car in Beijing.

To further compare the differences between the six driving cycles, the driving cycles were converted from a time domain to a frequency domain. Generally speaking, the low-frequency component of the amplitude spectrum is largely affected by road topology, while the high-frequency component is largely affected by driving habits, road conditions, and the environment. It can be seen in Figure 4 that the low-frequency coefficients of the amplitude spectrum of the two fixed-line buses are very similar. On the contrary, the four amplitude spectrums of the unfixed-line car are very different. When we develop a driving cycle generation system for fixed-line vehicles, how to describe the randomness of microtrips under the premise of ensuring the sequence of scenarios?

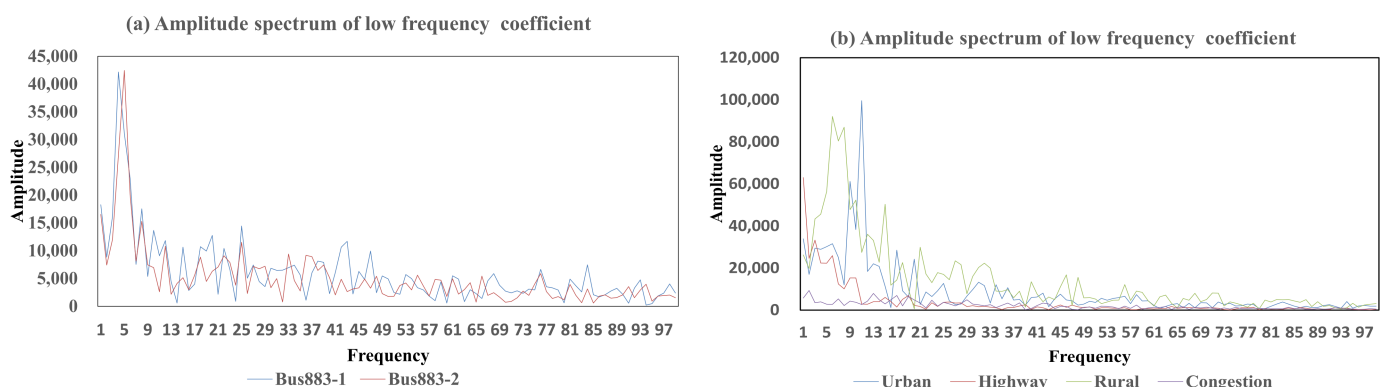


Figure 4. Amplitude spectrums of six driving cycles.

3.2. Characterize the Consecutiveness of Driving Scenarios with Sliding Window and Sticky Sampling Algorithm

For fixed-line electric vehicles, the sequence of driving scenarios is largely decided by the topography, traffic, and location. It is virtually impossible to model all the features for a driving cycle. Hence, we have to make some simple assumptions and classify the driving scenarios into four categories: urban (expressway, trunk road, secondary trunk road, branch road), rural (fewer pedestrians, fewer signal lights, higher speed), highway (no pedestrians, no signal lights, high speed) and congestion (more pedestrians, poor road conditions, slower speed). The detailed statistical characteristics of speed and acceleration of the four DSs will be discussed and shown in Section 4.1. To characterize the time-sequence of driving scenarios, we defined a data structure, “driving scenario chain”, $C_1 \cup C_2 \cup \dots \cup C_n$, and describe the sequence of driving scenarios with it.

Let $\{(W_n, n \in N)\}$ be a window sequence, C_n be the corresponding driving scenario category of window W_n , state space $C = \{C_{rural}, C_{highway}, C_{urban}, C_{congestion}\}$, $C_1 \cup C_2 \cup \dots \cup C_n$ be the categorization label sequence for the driving cycle, V_n be the velocity frequency item distribution vector for window W_n , V_{rural} , $V_{highway}$, V_{urban} , and $V_{congestion}$ be the velocity frequency item distribution vector for four standard driving scenarios: rural, highway, urban and congestion; the similarity between V_n and four standard vector V_x ($x = urban, rural, highway$ and $congestion$) can be calculated like this:

$$S(V_n, V_x) = \sqrt{\frac{\sum_{i=1}^M |f_{i,V_n} - f_{i,V_x}|^2}{M}} \quad (5)$$

where f_{i,v_n}, f_{i,v_x} be the frequencies of the i^{th} velocity interval (the whole velocity distribution is divided into M continuous velocity intervals).

To obtain the categorization label sequence C_n on the driving cycle, firstly, we should calculate the velocity frequent term vector V_n for the arriving window n with a sticky sampling algorithm. The calculation process can be seen in Section 2.2. Then, compare the similarities between V_n and four standard velocity frequent term vectors, V_x ($x = urban, rural, highway, and congestion$). After doing that, the velocity frequent vector V_x should be updated immediately. Lastly, the sliding window n moves on and the categorization label sequence C_n continues to grow. In fact, the size of the sliding window has a large effect on the categorization label sequence, $C_1 \cup C_2 \cup \dots \cup C_n$, and should be optimized to easily calculate the difference between V_n and V_x . This can be seen in Figure 5.

3.3. Characterize the Randomness of Microtrips with Markov State Transition Probability Matrix

For fixed-line vehicles, although the consecutiveness of driving scenarios is relatively fixed, the microtrips in the same driving scenario randomly appeared. The road condition, driving characteristics of the operator, and environment bring great uncertainty to driving pulses. The DC generation tool should produce driving pulses stochastically instead of producing them with recorded data or data snippets. Thus, in this section, we generate DPs in a driving scenario with Markov state transition probability matrix P , the element P_{ij} appearing in the matrix expresses the transition probability from state X_i to state X_j .

For each DP, its driving profile can be divided into four parts: initial acceleration phase, cruise phase, deceleration phase, and idle phase, as shown in Figure 6. The distribution range of velocity and acceleration can be used for defining the DP phase. For example, in the acceleration phase, the vehicle acceleration should be greater than $\delta \text{ m/s}^2$ and the vehicle velocity should be positive [19]. A typical DP in an urban environment could be a trip between two traffic lights. In order to better depict the road, traffic, and environmental conditions, we constructed a unique state transition probability matrix for each phase. Each element in the matrix is a non-negative value and the sum of them is equal to 1. The whole range of acceleration distribution is divided into k intervals and each of them represents a state. The value of variable k needs to be optimized—too large will reduce the calculation

velocity and too small will reduce the quality of DCs generated. This will be discussed in Section 4.2. With the acceleration state transition probability matrix built for each kind of scenario/phase, the characteristics of micro-trip can be described, respectively.

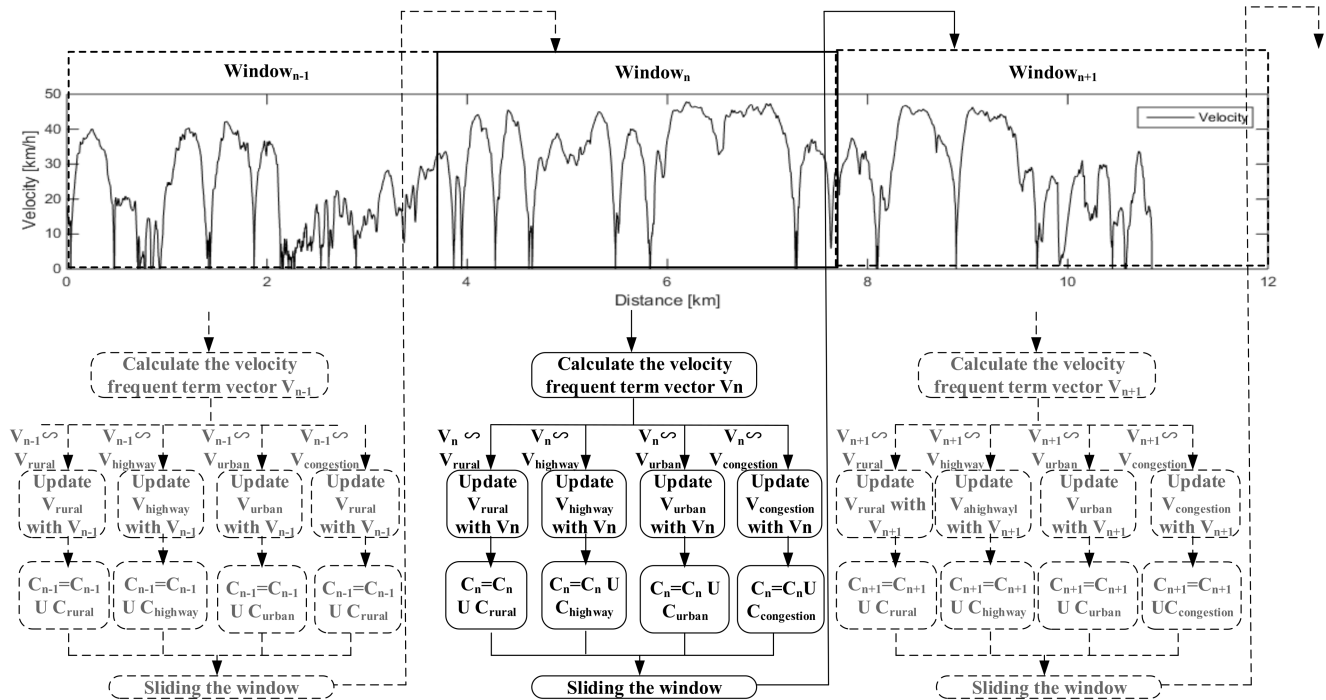


Figure 5. Constructing driving scenario chain with sliding window.

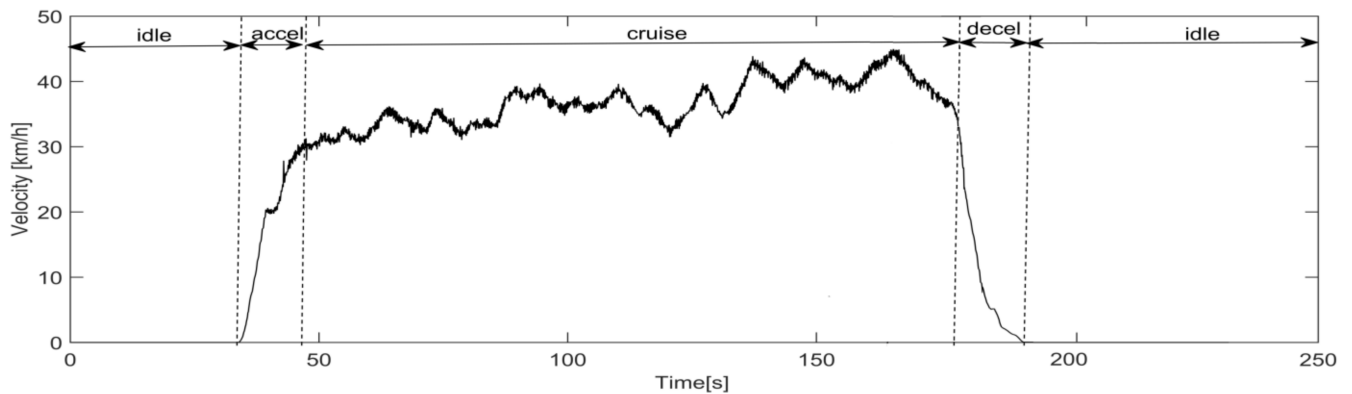


Figure 6. Four DP phases (acceleration, cruise, deceleration, and idle phase).

Why do we divide a driving pulse into four parts and construct a state transition probability matrix for each of them? The reason is that the state transition probability distribution of the four phases is very different. It can be seen in Figure 7. If we combine four state transition probability matrices into one, the characteristics of the topography, traffic, location, driver, vehicle, and environment of the four phases in the synthetic DS will disappear.

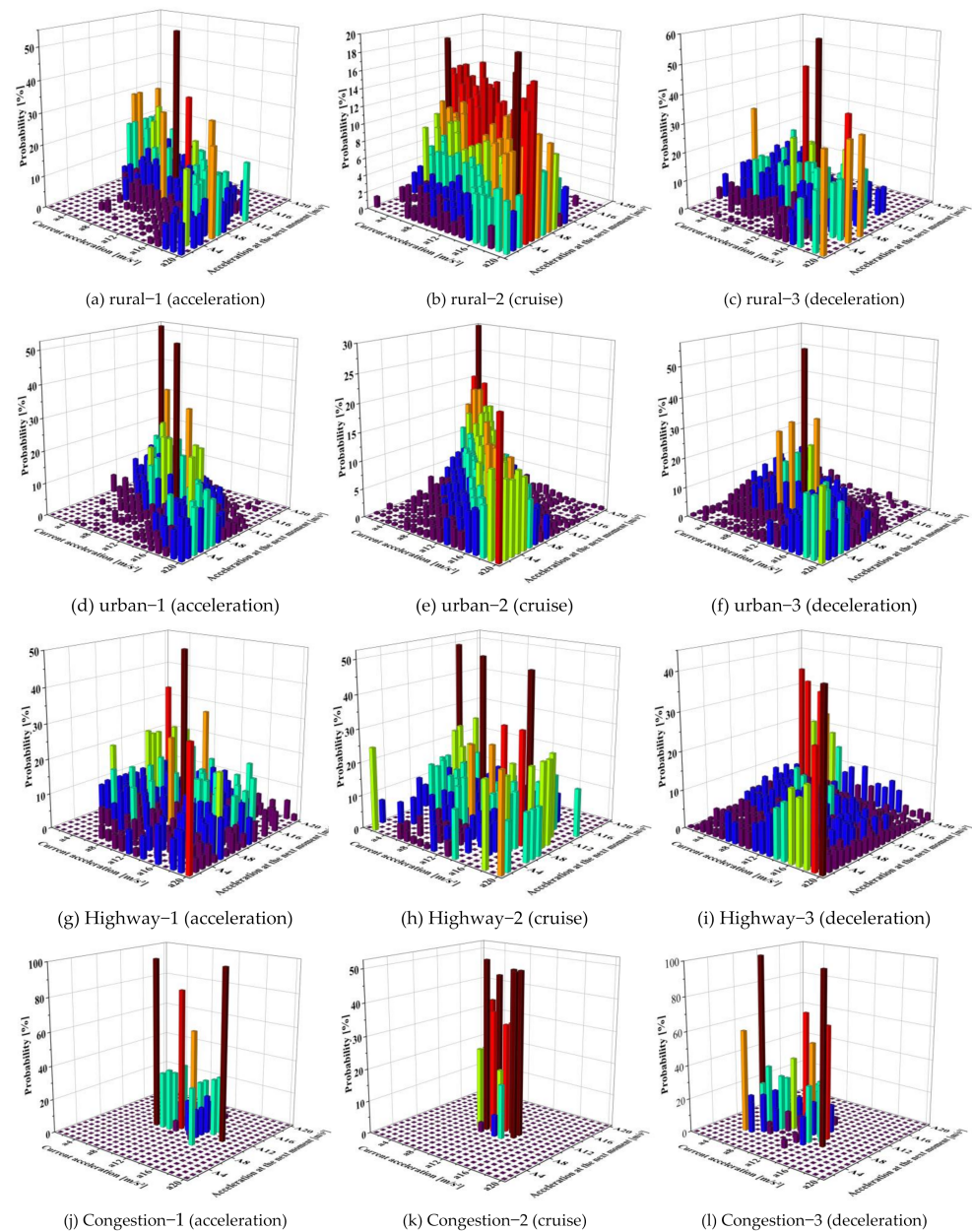


Figure 7. State transition probability matrix of acceleration, deceleration, and cruise.

3.4. The Sticky Sampling and Markov State Transition Probability Matrix Based DC Construction Algorithm

The sticky sampling and state transition probability matrix based DC construction algorithm (SS-STPM) proposed in this section distinguish itself from other methods by maintaining the sequence of DSs and generating DPs in a non-deterministic way. It does not use data snippets to generate cycles and all key parameters in a DS are stochastically described. In fact, in the generated DC, the sequence of DS is deterministic and the length of them is arbitrarily designated by the user. It constructs a DS sequence to reflect road topology and calculates the accelerated velocity state transition probability matrix to describe the randomness of traffic and environment. The algorithm is divided into two phases. In Phase_1, a DS categorization label sequence is produced; it can be used to record the succession of DS. In total, 16 accelerated/velocity state transition probability matrixes are constructed for four types of DS and DP.

3.4.1. Phase 1: Calculate DS Categorization Label Sequence and State Transition Probability Matrix

In this section, four steps are described, with which we produce the DS label sequence and acceleration state transition probability matrices. We can see this in Figure 8 and the following steps:

1. Real-world vehicle trajectory collection: Record the vehicle velocity at a fixed interval, produce a velocity sequence, $V = \{V_{(1)}, V_{(2)}, \dots, V_{(T)}\}$;
2. Velocity frequent item statistics calculation: Calculate the frequent item statistics for each velocity interval in the n^{th} window with sliding window function, $SW(V, n)$ and sticky sampling function, $SSA(SW(V, n))$, generate the velocity frequent item distribution vector V_n ;
3. State transition probability updating and DS sequence generating: Calculate the similarity between velocity frequent item distribution vector V_n and V_x ($x = urban, rural, highway, and congestion$). Select the most similar scenario DS and update its frequent item distribution vector V_x . Decide the DS type for current window n , and update the acceleration state transition probability matrix P for the selected DS;
4. DS label sequence production: Produce the current DS label sequence C_n by set operator 'U'.

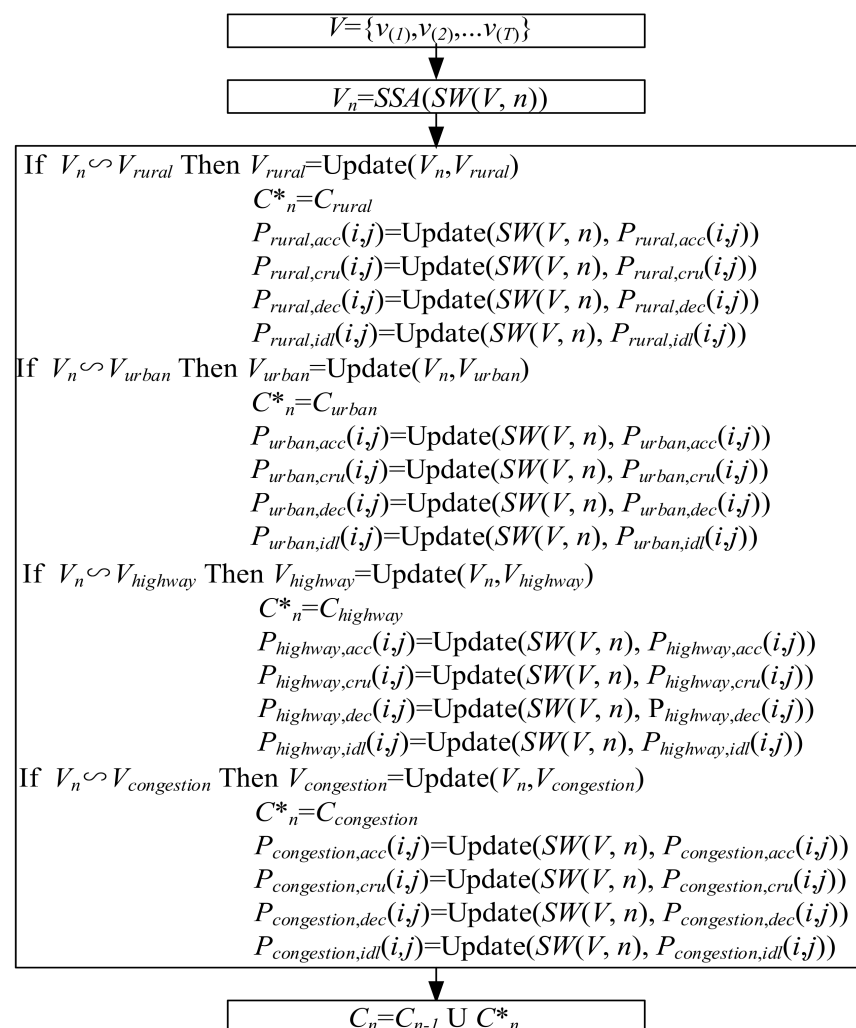


Figure 8. Calculate DS label sequence and state transition probability matrix.

3.4.2. Phase 2: Generate DC with DS Label Sequence and State Transition Probability Matrix

The proposed sticky sampling and state transition probability matrix based DC construction algorithm (SS-STPM) is a modular approach based on nesting. It can be seen in Figure 9. The outermost module (module-1) is the DC itself. The DC duration t_{DC} is decided by the user and the DS sequence that appeared in it is decided by DS label sequence C_n . Overall, DC consists of three modules: driving scenario, driving pulse, and pulse duration. In this paper, driving scenarios are classified into four categories: urban, rural, highway, and congestion. The classification is based on the characteristic statistics of velocity and acceleration distribution and it can be seen in Section 4.1. In fact, in Phase-1, the state transition probability P_{ij} and the frequent item distribution vector V_n for each of these driving scenarios are dynamically updated. The duration of scenarios t_{DS} is proportional to the number of the contiguous scenario label C_n^* in the same type.

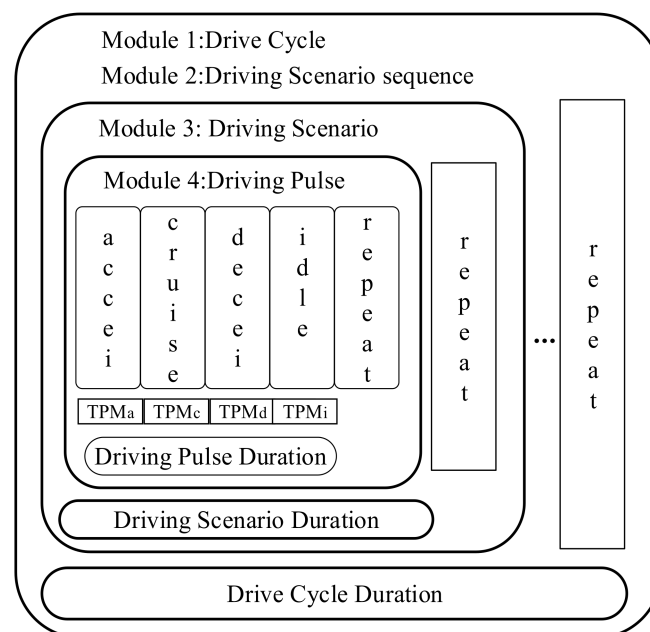


Figure 9. Graphical illustration of the DC production algorithm.

A DP consists of an initial acceleration phase, a cruise phase, a deceleration phase, and an idle phase. The DP duration t_{DP} is determined by the following equation:

$$t_{DP} = t_{acc}(DP) + t_{cru}(DP) + t_{dcc}(DP) + t_{idl}(DP) \quad (6)$$

The duration of acceleration phase t_{acc} , cruising phase t_{cru} , deceleration phase t_{dcc} , and idle phase t_{idl} are decided by the duration probability P_{acc} , P_{cru} , P_{dcc} , and P_{idl} . The acceleration interval is divided into n equal parts, and each interval represents a state A_i . In the acceleration state spaces $A(A_1, A_2, \dots, A_n)$, the acceleration state transition probability from state A_i to state A_j is P_{ij} . In the same DP phase, let N_{ij} be the number of systems switched from state A_i to A_j , the acceleration state transition probability P_{ij} can be calculated with Formula (1).

Each time we generate a velocity sequence, a random number $\delta \in [0,1]$, the current velocity v_t , acceleration state A_i , and state transition probability matrix P for the current DP phase are needed. The velocity v_{t+1} at the next moment can be expressed as:

$$v_{t+1} = v_t + a_t \times \Delta t \quad (7)$$

where acceleration a_t was decided by current acceleration state A_i , state transition probability matrix P_{ij} , and random number δ .

4. Performance Evaluation

One year of operating condition data of ten electric buses and ten hours of driving data of an electric test car was acquired in Beijing. Large amounts of fixed-line data make state transition probabilities more representative. The collected driving data includes four scenarios (22% Rural, 26% Highway, 27% Urban, and 25% congestion). A pure electric private car was used to collect the operating data of non-fixed lines. The parameters of it were shown in Table 1. A VBOX 3i SL GPS device manufactured by Racelogic was used for acquiring the test car data. This can be seen in Figure 10. The operating condition data of buses were obtained from the National Big Data Alliance of New Energy Vehicles.

Table 1. Parameters of BJ-EV 150.

Category	Parameter	Value	Unit
Vehicle Parameters	Curb weight	1370	kg
	Length \times Width \times Height	3398 \times 1720 \times 1503	mm
	Maximum speed	120	km/h
	Maximum grade	25	%
	Drive type	Front Predecessor	
	Drive range	150	km
	Wheelbase	2500	mm
	Minimum ground clearance	150	mm
Motor parameters	Motor peak power	45	kw
	Motor rate power	20	kw
	Motor peak torque	144	Nm
	Motor Maximum efficiency	92	%
Battery parameters	Battery type	Lithium Iron Phosphate	
	Rated capacity	91.5	Ah
	Rated voltage	320	V
	Rated power	30	kw

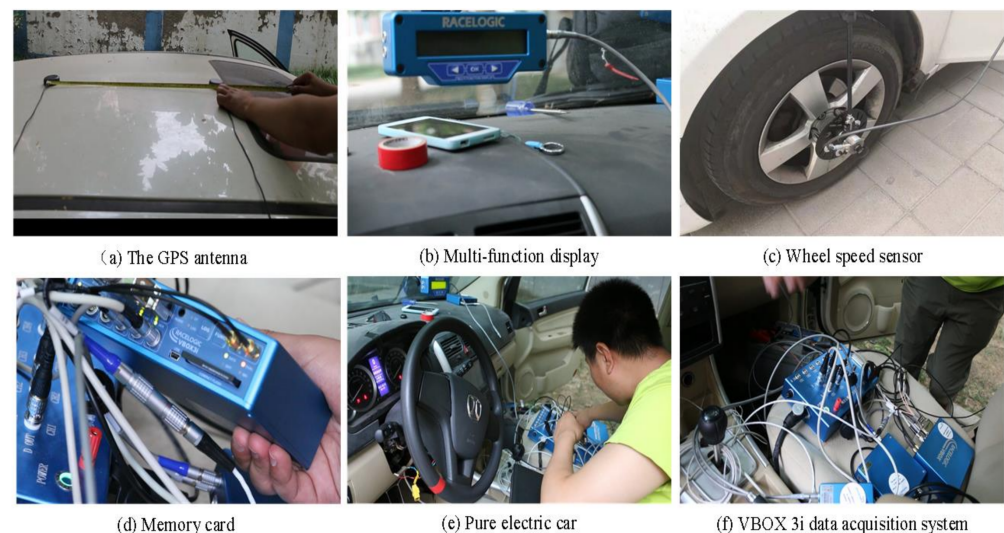


Figure 10. Test equipment.

4.1. Calculating Window DS Label with Velocity Frequency Item Distribution Vector or Acceleration Frequency Item Distribution Vector

For fixed-line electric vehicles, the time-sequence of microtrips has great influences on braking energy recovery systems, the drive system, and the battery management system [26]. To characterize the consecutiveness of driving scenarios, a sliding window and SSA algorithm were used to produce a driving pulse chain. A circular test between Beijing's fifth and sixth ring roads was carried out. The test vehicle was a private car, EV150,

and four driving scenarios were marked in advance. This can be seen in Figures 11–13. After comparing the velocity frequency item distribution and acceleration frequency item distribution, we can find that the difference of velocity frequency item distributions within four scenarios is larger than that of acceleration frequency item distributions. Thus, we select velocity frequency item distribution vectors as classification standards.

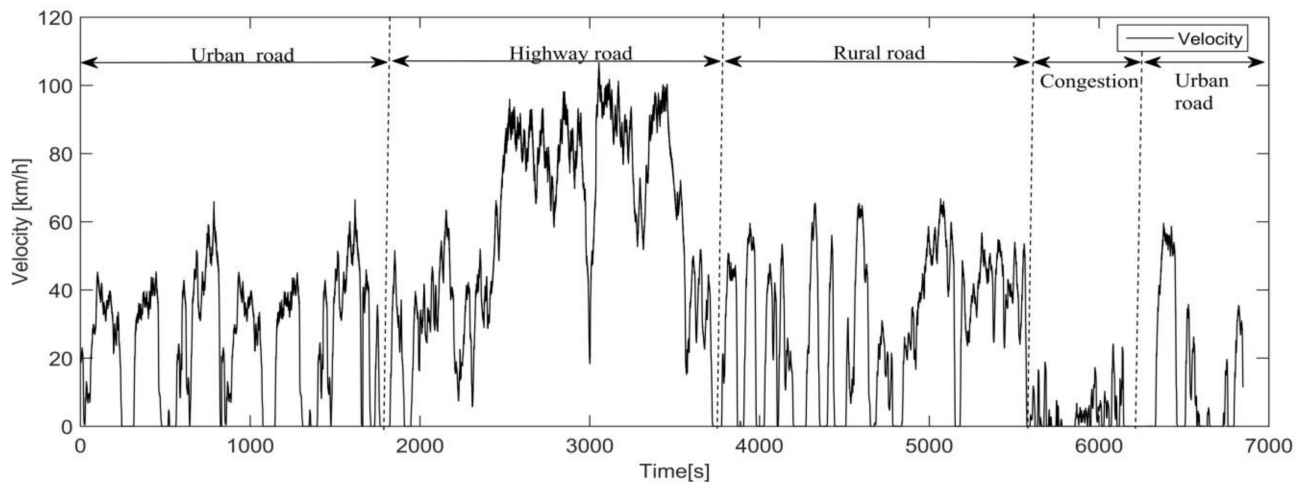


Figure 11. The driving circle of a private car in Beijing.

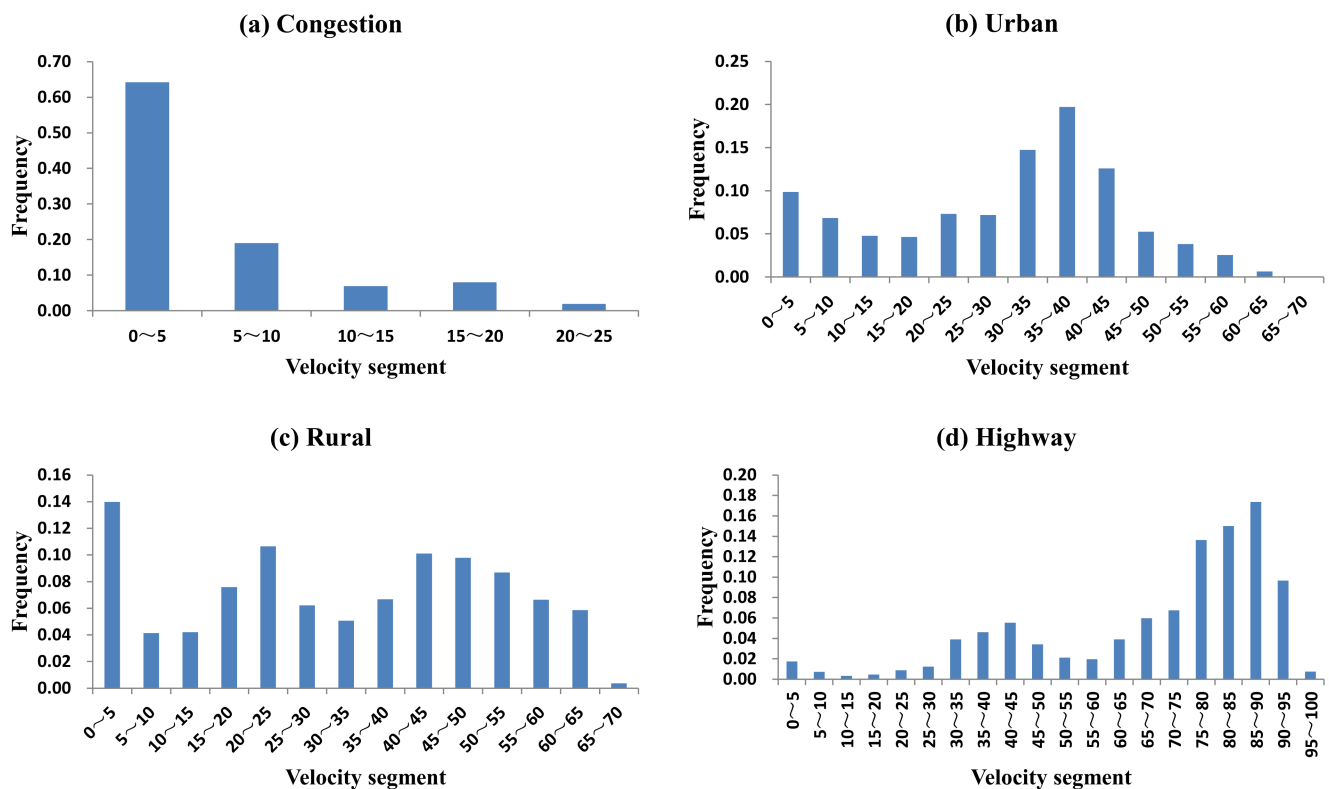


Figure 12. The distribution of speed frequent terms in congestion, urban, rural, and highway scenarios.

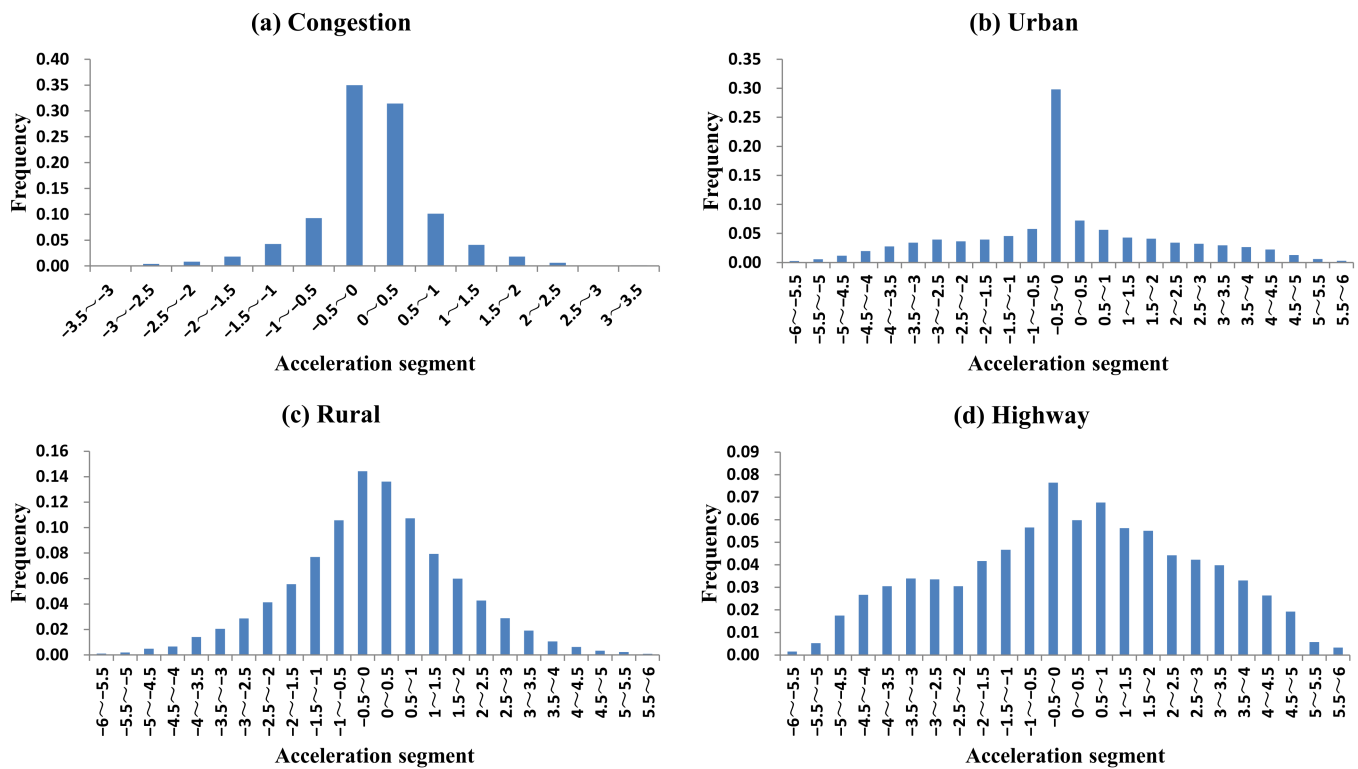


Figure 13. The distribution of acceleration frequent terms in congestion, urban, rural, and high-way scenarios.

Furthermore, for vehicles that are running, the current driving scenario type is decided by calculating the difference between the velocity frequent term distribution vectors obtained in the four standard operating conditions and the vector obtained within the current sliding window. In this section, the driving scenario that appeared in each of the sliding windows was classified and labeled with its velocity frequency item distribution. Each driving pulse that appeared in the window was classified as four driving scenarios, urban, rural, highway, and congestion. The size of the sliding window is 600s and the classification process can also be seen in Figure 14. Within the first 2400 s, the Euclidean distance between the sliding window and standard urban classification vector is less than others when we calculated it with the velocity frequency item distribution vector. However, if we calculated it with an acceleration frequency item distribution vector, the current sliding window should belong to congestion. In window (2400–3000 s), if we classified DPs with the Euclidean distance of velocity frequency item distribution vector, the window should belong to the highway category. However, if we classified them with the Euclidean distance of acceleration frequency item distribution vector, the window belongs to rural. Comparing this with Figures 11 and 14, we can find that classifying windows with the velocity frequency item distribution vector is more accurate than the acceleration frequency item distribution vector ($Stvfd/Stafd$ is the difference between the window velocity/acceleration frequency item distribution vector and standard velocity/acceleration frequency item distribution vector). Thus, in the generation process of the DS label sequence, we classify windows by calculating the similarity between the velocity frequency item distribution vector and four standard scenario classification vectors.

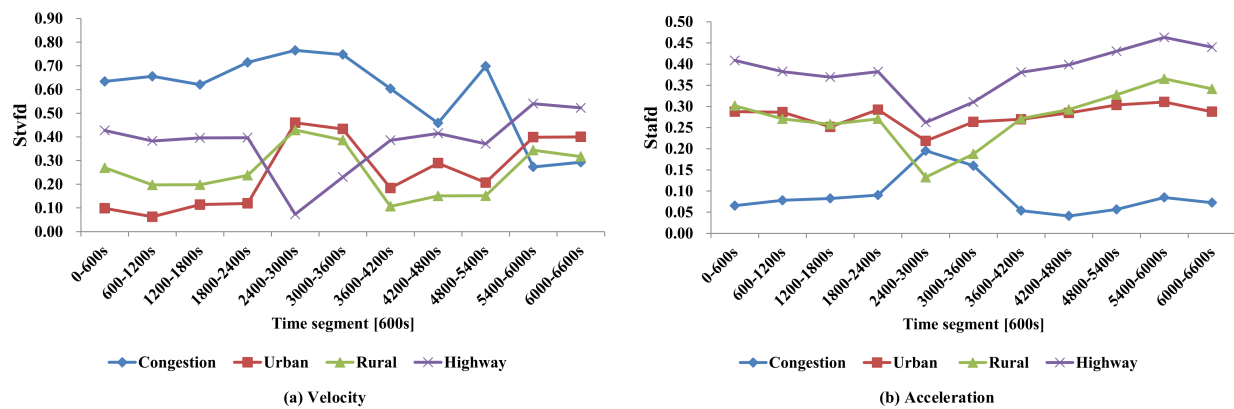


Figure 14. Classifying windows with velocity vector and acceleration vector.

4.2. The Granularity of Markov Acceleration State Transition Probability Matrix

To characterize the randomness of microtrips appearing in driving pulses, the Markov state transition probability matrix was constructed for each kind of phase [27–29]. The range of acceleration was divided into n consecutive intervals and each interval was defined as acceleration state. Within different kinds of driving scenarios, the state transition probability for acceleration phase, cruising phase, deceleration phase, and idle period are different. With the SSA algorithm, we can rapidly calculate the state transition probability P_{ij} with Formula (1), in which the state space X is determined solely by the state of acceleration.

In Figure 15, we divided the range of acceleration into 12, 18, 20, and 42 equal intervals and calculate the state transition probability for each of them. Then, construct the Markov acceleration state transition probability matrix P with them, and generate DCs with these matrices. Finally, we evaluated these generated DPs with real recorded DPs in the same scenarios. This can be seen in Figure 16, that with the decrease of the state transition probability matrix's size, the gap between the generated DPs and the real recorded DPs became larger and larger.

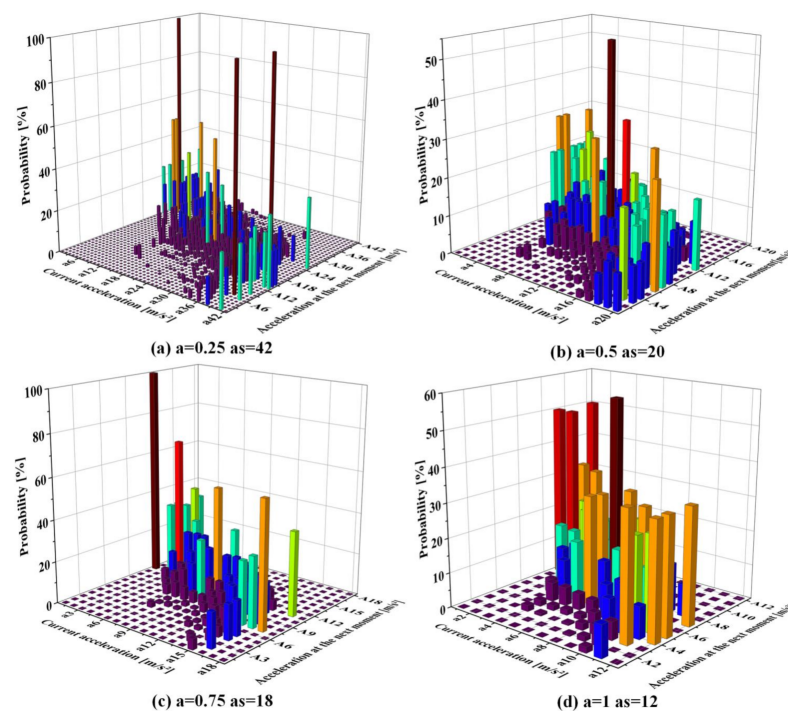


Figure 15. State transition probability matrix generated in different granularity.

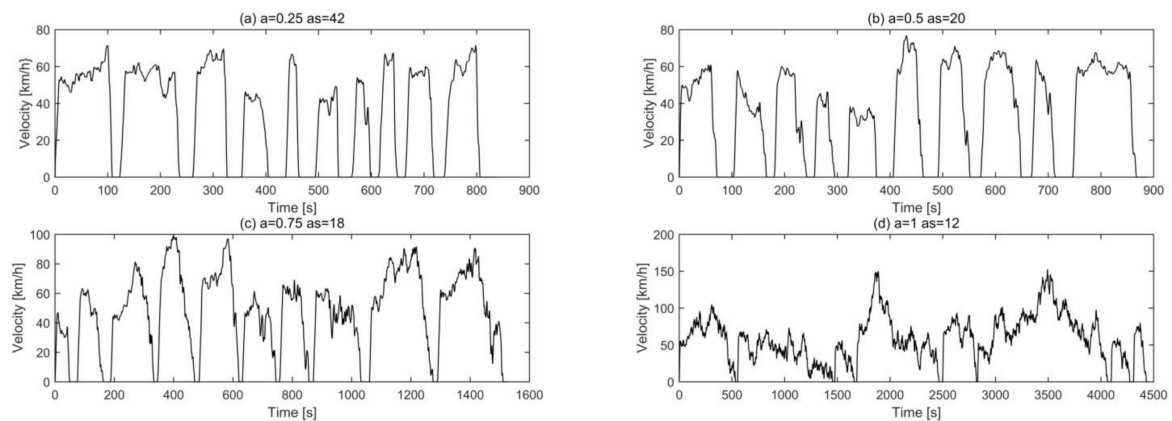


Figure 16. Generating DCs with different acceleration granularity.

4.3. Comparison

For many traditional Markov state transition probability-based driving cycle developing methodologies, the stochastic nature of the driving data is perfectly reflected. However, the correlation of driving scenarios hidden in fixed routes is not mined deeply and the correlation will bring large influences on designing and evaluating brake systems, drive systems, and battery management systems of electric vehicles. Thus, to clearly show the features of the proposed DC developing method, we compare three kinds of DCs generated separately by the worldwide-harmonized light vehicles test cycle (WLTC Class 3), the proposed method, and a Markov chain-based method (the state transition progress appeared in acceleration, cruise, deceleration and idle period are determined by Markov acceleration state transition probability matrix and the DSs appeared probabilistically). This can be seen in Figure 17, in which three velocity-time profiles were given. In real WLTC, no periodic acceleration, and deceleration were given and different traffic jams can be reflected. The test period was 1800 s and the test average speed was 46 km/h. In Figure 17b,c, the Markov acceleration state transition probability was calculated with real WLTC. From Figure 17a–c we can find that, compared with the Markov chain-based method, the sequence of DS generated by the proposed method is closer to that of real WLTC. In fact, for testing fixed-line vehicles, the overall distance of real WLTC is short and the number of DP is small.

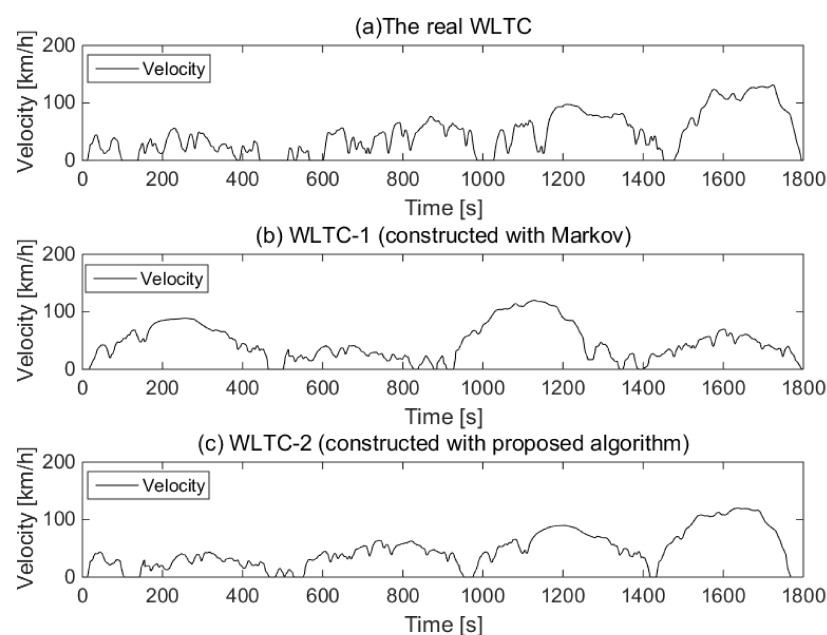


Figure 17. Comparisons of three kinds of WLTCs.

To clearly compare the order of driving scenarios, three velocity-distance profiles of Bus 883 were given in Figure 18. From Figure 18a,c we can find that, although the speed of each fixed position is different, scenario types of them are approximately the same. It is mainly due to the consistency of the fixed-line topology. On the contrary, for DC-1 generated by a traditional algorithm, the sequence of DS is random and the scenario types for each fixed position cannot keep up with that that appeared in real DC.

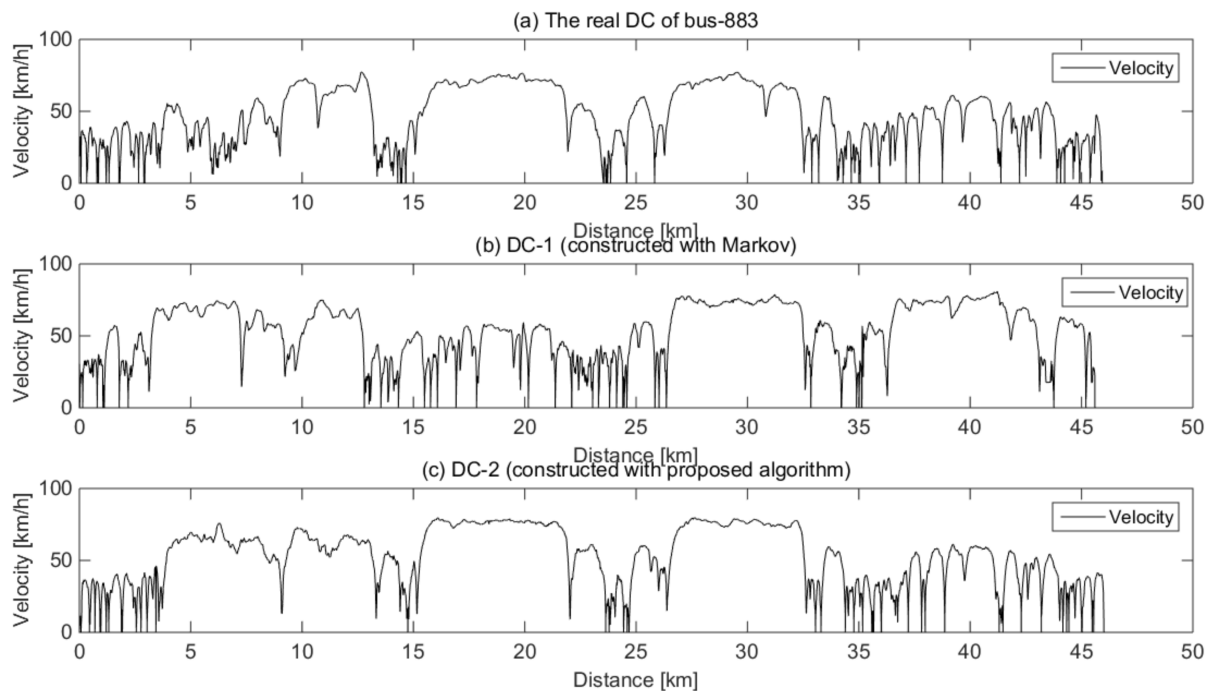


Figure 18. Comparison of three kinds of DC.

To further illustrate the similarities and differences between the three DCs, we listed 14 statistical indicators, such as velocity, distance, and average acceleration. From Table 2, we can find that the stochastic characteristics that appeared in the three DCs are very similar. Although the time span of the three DCs is slightly different, most indicators, such as total distance, average acceleration, average deceleration, and others, are very similar. This is due to the fact that in the DP generation module, the same state transition probability matrix was adopted in the two DC generating methods.

Table 2. Characteristic parameters in real DC, DC-1, and DC-2.

Statistical Items	Real Bus 883	DC-1(Markov)	DC-2
Time (s)	5646	5248	5484
Mileage (km)	45.9	45.6	45.9
Average acc. (m/s^2)	0.364	0.388	0.372
Average dec. (m/s^2)	−0.379	−0.401	−0.395
Idling proportion (%)	14.04%	22.43%	15.55%
Cruising proportion (%)	59.2%	57.13%	57.58%
Acc.proportion (%)	16.62%	11.32%	14.73%
Dec.proportion (%)	10.14%	9.12%	12.14%
Maximum velocity (km/h)	77.36	80.66	78.52
Average velocity (km/h)	30.11	33.54	31.28
Velocity standard deviation	12.468	13.346	12.748
Maximum acc (m/s^2)	3.689	3.597	3.731
Minimum dec (m/s^2)	−3.792	−3.935	−3.874
Acceleration standard deviation	0.603	0.692	0.657

Additionally, we compared the DS label sequences and DS transition probability for real DC, DC-1, and DC-2. It can be seen in Figure 19 that DC-2 has a more consistent DS label with real DC than DC-1. In the range of 15–24 km and 37–44 km, DC-1 has a large gap with real DC and DC-2. Statistically speaking, the DS label sequence of real DC is 87.5%, similar to DC-2, and 54.6% similar to DC-1. To further illustrate the differences within three DCs, DPs appeared in each DC and were divided into four categories, DS-1, DS-2, DS-3, DS-4, depending on the scenario. We calculated scenario transition probability matrices within DS-1, DS-2, DS-3, DS-4 and compared the differences between DCs with them. This can be seen in Figure 20, that the scenario transition probability “DS4→DS1”, “DS1→DS4”, and “DS2→DS2” have a larger difference between “Real-world” and “DC-1” than between “Real-world” and “DC-2”. From this, we can find the correlation of driving scenarios hidden in the fixed route is perfectly mined by the proposed algorithm and is not found by the traditional Markov strain-based algorithm.

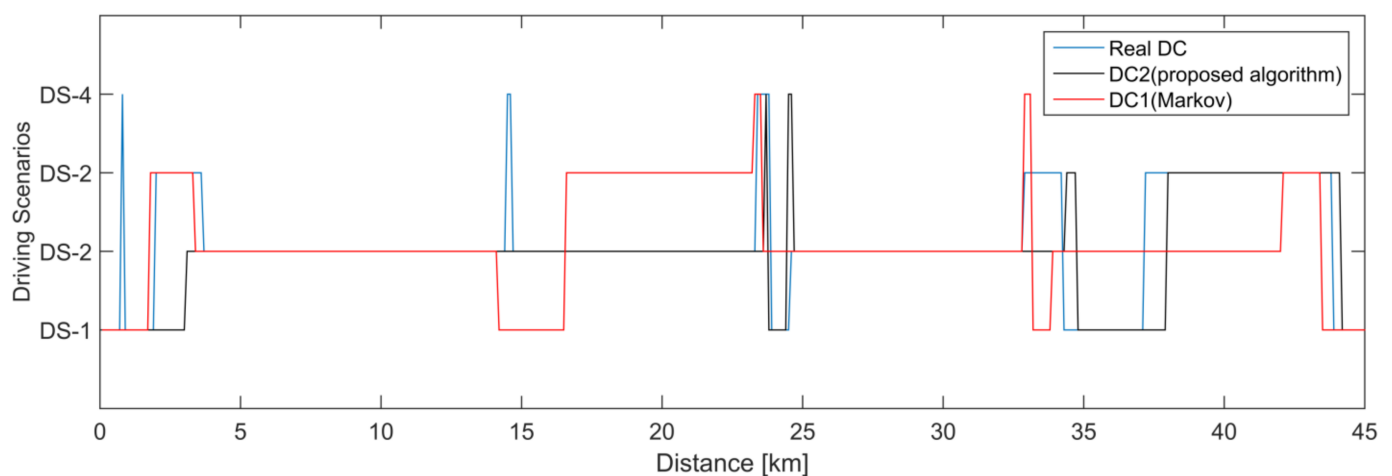


Figure 19. Driving scenario labels (real DC, DC-1, and DC-2).

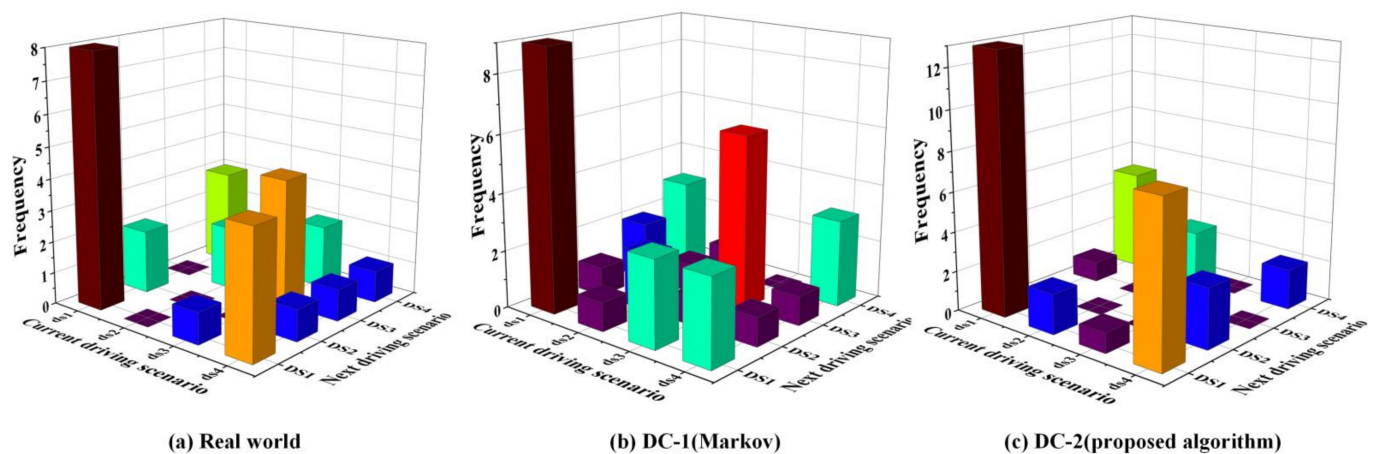


Figure 20. Driving scenario transition probability (real DC, DC-1, and DC-2).

5. Conclusions

In this article, a novel DC expressing and constructing method was presented. It was shown that the proposed method can not only describe the stochastic nature of driving data but also reflect the correlation of driving scenarios. The major advantage of it is that: (1) it constructs a steady DS sequence to reflect the road topology for fixed-line EVs, which is very suitable to evaluate and compare the performance of the braking system, drive system, and energy recovery system in hybrid electric vehicles or electric vehicles; (2) it respectively constructs an acceleration state transition matrix for acceleration, cruise, deceleration,

and idle periods, and the characteristics of the traffic, location, driving behavior and environment are included in these state transition matrices. It is very important for fixed-line EVs/HEVs with the braking system, drive system, and energy recovery system, hence, the performance of these vehicles has large correlations with the sequence of microtrips. In future work, we will discuss the application of this approach in designing and developing electric vehicles, such as electric buses, sweepers, sanitation, and logistic trucks.

Author Contributions: Conceptualization, L.Z. and K.L.; methodology, L.Z.; software, K.L.; validation, W.Z.; formal analysis, H.-C.K. and Z.W.; investigation, K.L.; resources, L.Z.; data curation, W.Z.; writing—original draft preparation, K.L.; writing—review and editing, L.Z., W.Z., H.-C.K. and Z.W.; visualization, K.L.; supervision, L.Z. and W.Z.; project administration, Z.W.; funding acquisition, L.Z. and W.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (Grant No.52077007); Science and Technology Plan of Beijing Municipal Education Commission (Grant No.KM201811232003); the Postgraduate Science and Technology Innovation Project of Beijing Information Science and Technology University (Grant No.5112110835); the Beijing Information Technology University education reform project (Grant No.2021Jgyb02); and the Xuzhou Science and Technology project (Grant No. 90250886701313).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

DC	Driving cycle
Ev	Electric vehicle
SSA	Sticky Sampling algorithm
DP	Driving Pulse chain
DS	Driving Scenarios
ICEVs	Internal Combustion Engine Vehicles
SOC	State of Charge
STPM	State Transition Probability Matrix
UDDS	Urban Dynamometer Driving Schedule
NEDC	New European driving cycle
FTP-75	Federal Test Procedure
ZZUDC	Zheng Zhou Urban Driving Cycle
SS-STPM	Sticky Sampling and State Transition Probability Matrix based DC construction algorithm

References

1. Yuan, X.; Zhang, C.; Hong, G.; Huang, X.; Li, L. Method for evaluating the real-world driving energy consumptions of electric vehicles. *Energy* **2017**, *141*, 1955–1968. [\[CrossRef\]](#)
2. Huertas, J.I.; Giraldo, M.; Quirama, L.F.; Díaz, J. Driving Cycles Based on Fuel Consumption. *Energies* **2018**, *11*, 3064. [\[CrossRef\]](#)
3. Tong, H. Development of a driving cycle for a supercapacitor electric bus route in Hong Kong. *Sustain. Cities Soc.* **2019**, *48*, 101588. [\[CrossRef\]](#)
4. Xiong, R.; Cao, J.; Yu, Q.; He, H.; Sun, F. Critical Review on the Battery State of Charge Estimation Methods for Electric Vehicles. *IEEE Access* **2017**, *6*, 1832–1843. [\[CrossRef\]](#)
5. Zhang, H.; Zhao, L.; Chen, Y. A Lossy Counting-Based State of Charge Estimation Method and Its Application to Electric Vehicles. *Energies* **2015**, *8*, 13811–13828. [\[CrossRef\]](#)
6. Berzi, L.; Delogu, M.; Pierini, M. Development of driving cycles for electric vehicles in the context of the city of Florence. *Transp. Res. Part D Transp. Environ.* **2016**, *47*, 299–322. [\[CrossRef\]](#)
7. Cui, Y.; Xu, H.; Zou, F.; Chen, Z.; Gong, K. Optimization based method to develop representative driving cycle for real-world fuel consumption estimation. *Energy* **2021**, *235*, 121434. [\[CrossRef\]](#)

8. Huertas, J.I.; Quirama, L.F.; Giraldo, M.; Díaz, J. Comparison of Three Methods for Constructing Real Driving Cycles. *Energies* **2019**, *12*, 665. [\[CrossRef\]](#)
9. Schwarzer, V.; Ghorbani, R. Drive Cycle Generation for Design Optimization of Electric Vehicles. *IEEE Trans. Veh. Technol.* **2012**, *62*, 89–97. [\[CrossRef\]](#)
10. Jing, Z.; Wang, G.; Zhang, S.; Qiu, C. Building Tianjin driving cycle based on linear discriminant analysis. *Transp. Res. Part D Transp. Environ.* **2017**, *53*, 78–87. [\[CrossRef\]](#)
11. Lin, J.; Niemeier, D.A. An exploratory analysis comparing a stochastic driving cycle to California’s regulatory cycle. *Atmos. Environ.* **2002**, *36*, 5759–5770. [\[CrossRef\]](#)
12. Topić, J.; Škugor, B.; Deur, J. Synthesis and Feature Selection-Supported Validation of Multidimensional Driving Cycles. *Sustainability* **2021**, *13*, 4704. [\[CrossRef\]](#)
13. Zhao, X.; Zhao, X.; Yu, Q.; Ye, Y.; Yu, M. Development of a representative urban driving cycle construction methodology for electric vehicles: A case study in Xi’an. *Transp. Res. Part D Transp. Environ.* **2020**, *81*, 102279. [\[CrossRef\]](#)
14. Peng, J.; Jiang, J.; Ding, F.; Tan, H. Development of Driving Cycle Construction for Hybrid Electric Bus: A Case Study in Zhengzhou, China. *Sustainability* **2020**, *12*, 7188. [\[CrossRef\]](#)
15. Brady, J.; O’Mahony, M. Development of a driving cycle to evaluate the energy economy of electric vehicles in urban areas. *Appl. Energy* **2016**, *177*, 165–178. [\[CrossRef\]](#)
16. Nguyen, Y.-L.T.; Nghiem, T.-D.; Le, A.T.; Bui, N.-D. Development of the typical driving cycle for buses in Hanoi, Vietnam. *J. Air Waste Manag. Assoc.* **2018**, *69*, 423–437. [\[CrossRef\]](#) [\[PubMed\]](#)
17. Hongwen, H.; Jinquan, G.; Jiankun, P.; Huachun, T.; Chao, S. Real-time global driving cycle construction and the application to economy driving pro system in plug-in hybrid electric vehicles. *Energy* **2018**, *152*, 95–107. [\[CrossRef\]](#)
18. Ashtari, A.; Bibeau, E.; Shahidinejad, S. Using Large Driving Record Samples and a Stochastic Approach for Real-World Driving Cycle Construction: Winnipeg Driving Cycle. *Transp. Sci.* **2014**, *48*, 170–183. [\[CrossRef\]](#)
19. Gong, H.; Zou, Y.; Yang, Q.; Fan, J.; Sun, F.; Goehlich, D. Generation of a driving cycle for battery electric vehicles: A case study of Beijing. *Energy* **2018**, *150*, 901–912. [\[CrossRef\]](#)
20. Wang, H.; Zhang, X.; Ouyang, M. Energy consumption of electric vehicles based on real-world driving patterns: A case study of Beijing. *Appl. Energy* **2015**, *157*, 710–719. [\[CrossRef\]](#)
21. Shen, P.; Zhao, Z.; Li, J.; Zhan, X. Development of a typical driving cycle for an intra-city hybrid electric bus with a fixed route. *Transp. Res. Part D Transp. Environ.* **2018**, *59*, 346–360. [\[CrossRef\]](#)
22. Ho, S.-H.; Wong, Y.-D.; Chang, V.W.-C. Developing Singapore Driving Cycle for passenger cars to estimate fuel consumption and vehicular emissions. *Atmos. Environ.* **2014**, *97*, 353–362. [\[CrossRef\]](#)
23. Shi, S.; Lin, N.; Zhang, Y.; Cheng, J.; Huang, C.; Liu, L.; Lu, B. Research on Markov property analysis of driving cycles and its application. *Transp. Res. Part D Transp. Environ.* **2016**, *47*, 171–181. [\[CrossRef\]](#)
24. Manku, G.S.; Motwani, R. Approximate frequency counts over data streams. *Proc. VLDB Endow.* **2012**, *5*, 1699. [\[CrossRef\]](#)
25. Wang, L.; Ma, J.; Zhao, X.; Li, X. Development of a Typical Urban Driving Cycle for Battery Electric Vehicles Based on Kernel Principal Component Analysis and Random Forest. *IEEE Access* **2021**, *9*, 15053–15065. [\[CrossRef\]](#)
26. Qin, S.; Qiu, D.; Lin, H.; Bing, W.; Li, Y. Support vector machine-based driving cycle recognition for dynamic equivalent fuel consumption minimization strategy with hybrid electric vehicle. *Adv. Mech. Eng.* **2018**, *10*. [\[CrossRef\]](#)
27. Wang, Z.; Zhang, J.; Liu, P.; Qu, C.; Li, X. Driving Cycle Construction for Electric Vehicles Based on Markov Chain and Monte Carlo Method: A Case Study in Beijing. *Energy Procedia* **2019**, *158*, 2494–2499. [\[CrossRef\]](#)
28. Zhang, Q.; Deng, W. An Adaptive Energy Management System for Electric Vehicles Based on Driving Cycle Identification and Wavelet Transform. *Energies* **2016**, *9*, 341. [\[CrossRef\]](#)
29. Lin, C.; Zhao, L.; Cheng, X.; Wang, W. A DCT-Based Driving Cycle Generation Method and Its Application for Electric Vehicles. *Math. Probl. Eng.* **2015**, *2015*, 178902. [\[CrossRef\]](#)