energies

MDPI

Concept Paper

A Novel Adaptive Equivalence Fuel Consumption Minimisation Strategy for a Hybrid Electric Two-Wheeler

Naga Kavitha Kommuri *, Andrew McGordon, Antony Allen and Dinh Quang Truong

Warwick Manufacturing Group (WMG), University of Warwick, Coventry CV4 7AL, UK; a.mcgordon@warwick.ac.uk (A.M.); a.allen@warwick.ac.uk (A.A.); t.dinh@warwick.ac.uk (D.Q.T.)
* Correspondence: k.kommuri@warwick.ac.uk; Tel.: +91-9945668025

Abstract: One of the major challenges in implementing the equivalent fuel consumption minimisation strategy in hybrid electric vehicles is the adaptation of the equivalence factor to real-world driving. In this paper, a novel adaptive equivalent fuel consumption minimisation strategy (A-ECMS) has been developed for a hybrid two-wheeler to further improve fuel savings by predicting the drive cycles and thereby estimating and adapting the equivalence factor online for the ECMS energy management control. A learning vector quantitative neural network (LVQNN)-based classifier was first proposed to recognise the real-world driving cycle based on a fixed time window of past driving information. Along with standardised drive cycles, real-world driving data were used in the learning process to increase the robustness of the learning. The A-ECMS is then capable of regulating its equivalence factors online based on the LVQNN controller output. Numerical simulation results indicated that there was considerable improvement in fuel economy of the vehicle with the proposed methodology, up to 10.7%, compared to the use of traditional ECMS which was manually optimised for a single drive cycle. The average improvement in fuel economy over the ten drive cycles considered for testing is 3.93%.

Keywords: optimal real-time control; ECMS; hybrid two-wheeler; equivalence factor adaptation; neural network; drive cycle recognition

check for updates

Citation: Kommuri, N.K.; McGordon, A.; Allen, A.; Truong, D.Q. A Novel Adaptive Equivalence Fuel Consumption Minimisation Strategy for a Hybrid Electric Two-Wheeler. *Energies* **2022**, *15*, 3192. https://doi.org/10.3390/en15093192

Academic Editor: Mario Marchesoni

Received: 29 December 2021
Accepted: 21 April 2022
Published: 27 April 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

Concerns over climate change, constraints on energy resources, stringent regulations on emissions and poor energy efficiency are all pushing the transportation industry to focus more on alternative technologies, such as hybrid electric vehicles (HEVs) [1]. Plugin HEVs (PHEVs) have become the prime focus in recent times. The key goal for using the full potential of hybridisation is by developing an intelligent real-time implementable energy management strategy (EMS).

Studies have suggested that energy management controllers for HEVs can be divided into rule-based controllers and optimisation-based controllers [2–4]. Rule-based controllers are based on predefined rules based on experience, calibration and tuning of controllers for the desired output without prior knowledge of the trip [5,6]. A hybrid electric vehicle is more sensitive to a driving cycle than an internal combustion vehicle when fuel consumption and tailpipe emissions are concerned [7]. Studies show that real-world drive cycles and driving trends have a large negative impact on fuel consumption and exhaust emissions with a traditional rule-based EMS used in hybrid vehicles [8]. Optimisation controllers overcome the inherent rigidity of rule-based controllers by using an optimal control strategy that minimises a cost function [9,10].

One of the most popular solutions for the real-time optimisation of the energy management controller is the equivalent fuel-consumption minimisation strategy (ECMS) and it is generally accepted to be a promising real-time controller because of its feasibility and optimality [4,11,12]. The equivalent consumption minimisation strategy is an instantaneous approach derived from Pontryagin's Minimum Principle [11]. Unlike many optimisation strategies, the ECMS does not require an a priori driving profile to exist before optimisation. Feasibility along with fast computation makes ECMS a potentially real-time implementable strategy. However, one of the most important challenges in implementing an ECMS in the production of hybrid electric vehicles is the estimation and adaptation of equivalence factor for real-world driving because the optimal equivalence factor is not readily available without the trip information. Using an optimally tuned equivalence factor for a specific drive cycle does not guarantee the optimality of the ECMS controller for other driving data and by doing so, the actual purpose of using optimal control is completely lost. Adapting the equivalence factor for on-road driving conditions accurately is essential for the optimal working of ECMS. The optimal equivalence factor significantly affects the fuel economy, and the size of this effect varies with the drive cycle [12]. Therefore, precise estimation of the equivalence factor is crucial for the performance of the ECMS and a challenge for successful implementation on the production vehicle.

Unlike a four-wheeler, with a two-wheeler, the cost for hybridisation causes a larger change of marginal cost in the production of the vehicle [13]. This drives a need for a cost-effective, adaptive ECMS controller, which has a low offline and online computational load. The real-world drive data would not necessarily be like any of the standardised drive cycles or previously collected on-road driving data sets. Storage of thousands of potential drive cycles on board a two-wheeler is not feasible. Thus, the hybrid two-wheeler platform requires a cost-effective solution, and this manuscript reports a methodology based on drive cycle recognition to adapt the equivalence factor with precision and efficient computation for the hybrid two-wheeler.

There is limited literature found for online adaptation of equivalence factors based on driving cycle or pattern recognition for ECMS implementation. In the paper [14], a similarity weight is assigned to each reference driving cycle using a fuzzy clustering method. Fuzzy weights represent the similarity of an unknown driving cycle to each of the reference driving cycles. Thus, drive cycle recognition is used for estimating the drive cycle and thereby estimating the equivalence factor online for ECMS. Musardo et al. implemented an adaptive ECMS method by updating the equivalence factor online by predicting the future driving cycles using a neural network. This method provided an online estimation of the equivalence factor [11]. This method showed results very close to those obtained using a global optimal solution; dynamic programming (DP). However, the method required additional computation load to implement the predictor, which would not be a suitable solution for a low-cost energy management application. Jeon et al. first generated five representative driving patterns (RDP) by rules. A rule-based control algorithm was extracted from the result of the optimal solution provided by dynamic programming (DP) on each RDP. Finally, a multi-mode driving control was realised by switching the control parameters in each RDP [15]. However, this method requires large computational capability. In 2009, Huang et al. used four features of drive cycle characteristics for distinguishing between the driving cycle types for the equivalence factor adaptation for ECMS implementation. However, the fuel benefit of this method was not evaluated [5]. The study considered only two standardised driving cycle types with two non-fuzzy rule-based control strategies for each of the driving cycles. Most of the research in the literature focuses on the fuzzy logic algorithm to predict the driving cycle, the driving intention, or the driving patterns for adaptive equivalence factors for online ECMS implementation. Most of the considered literature used statistical and stochastic velocity forecast approaches for energy management optimisation. However, a study done by Chao et al. [16] demonstrates that a data-driven neural network (NN) exhibits better performance when compared to statistical and stochastic prediction approaches concerning the prediction precision and computational cost.

Neural networks (NNs) have been used successfully for many applications such as pattern classification, decision-making, predicting and adaptive control [17]. The data-driven type classifiers exhibit the best performance as they can learn the short-term driving behaviour of a vehicle and capture its nonlinearity with low computation and high precision [18]. A well-designed NN can fit into a look-up table and can adapt itself by training to update the table data. It is a powerful computational method, which learns and generalises from training data. Therefore, it would be very beneficial to use NN for the drive cycle recognition and estimation of equivalence factors for the successful implementation of ECMS online. Langari and Won proposed an intelligent energy management agent (IEMA) which is based on a fuzzy rule-based energy management strategy for parallel hybrid vehicles, which contained a learning vector quantisation (LVQ) network roadway type identifier [9]. During their research, 47 parameters were selected for LVQ classification and most of the drive cycle segments could be correctly classified. Lei et al. analysed the impacts on identification results caused by the dissimilarity measures used in driving pattern recognition (DPR). A micro-trip extraction method is used to optimise the training of the LVQ identifier [19]. Research results show that realizing DPR through calculating the Euclidean distance is more adaptable. The LVQ neural network recognition algorithms are exactly based on the calculation of Euclidean distances. In the paper [20], an automated feature extraction scheme based on convolution neural networks (CNNs) and Kernel PCA (KPCA) for real-time driving pattern recognition (RTDPR) is proposed to achieve the consistent performance of the energy management. Simulation and experimental results show that the proposed automated feature extraction strategy outperforms the conventional driving pattern recognition algorithms based on manual feature extraction. In the research study [21], the authors propose an improved adaptive equivalent consumption minimisation strategy (A-ECMS) based on long-term target driving cycle recognition and short-term vehicle speed prediction, and adapt it to personalised travel characteristics. In the offline part, typical driving cycles of a specific driver is constructed by analysing personalised travel characteristics in the historical driving data, and optimal SOC consumption under each typical driving cycle is optimised by DP. In the online part, the SOC reference trajectory is obtained by recognizing the target driving cycle from intelligent traffic system, and short-term vehicle speed is predicted by nonlinear auto-regressive (NAR) neural network which both adjust EF together. Simulation results show that compared with CD-CS, the fuel consumption of A-ECMS proposed in the paper is reduced by 8.7%. However, this method needs additional information from the Intelligent traffic system which adds cost to the vehicle.

There is no literature found which uses the data-driven LVQNN for drive cycle recognition using limited past drive cycle information for online equivalence factor estimation. In this study, a data-driven LVQNN-based drive cycle recognition based on past data of the driving cycle for online estimation of equivalence factor for ECMS implementation is proposed in this paper. In this study, an advanced adaptive equivalent fuel consumption minimisation strategy, (AECMS) is proposed and developed for the hybrid two-wheeler considered, based on the previously reported ECMS strategy [22]. An NN-based controller is used for drive cycle recognition and thereby the online estimation of equivalence factor for the adaptive ECMS implementation.

This paper reports the development of a novel control approach to estimate the equivalence factor by classifying the present drive cycle against a range of standardised and real-world drive cycles using the learning vector quantitative neural network (LVQNN) algorithm. This classification is based on eight parameters chosen to characterise the drive cycle. A lookup table is used to estimate the equivalence factor for the modified ECMS optimal controller based on the output from the supervisory controller.

This paper is organised as follows: In Section 2, the hybrid powertrain configuration is briefly introduced, and a mathematical model is developed for further investigation. In Section 3, the modified ECMS implementation is explained. In Section 4, the LVQNN-based control strategy is constructed and optimised using trained data. In Section 5 the simulation test results are presented and discussed, and the concluding remarks are provided in Section 6.

2. Vehicle Configuration and Modelling

The vehicle considered for this study is a full parallel plugin hybrid concept two-wheeler and the powertrain architecture for the vehicle is as shown in Figure 1. The system is composed of an engine, a centrifugal clutch, an electric machine, mechanical transmission and an energy storage device (high voltage battery). Figure 1 shows the mechanical and electrical power flow between the powertrain components. The electric machine is capable of power assist and charging from the engine along with pure electric drive depending on the wheel power requirement and the battery SOC (state of charge). Hard constraints corresponding to the physical limits of the powertrain components are applied to the control input.

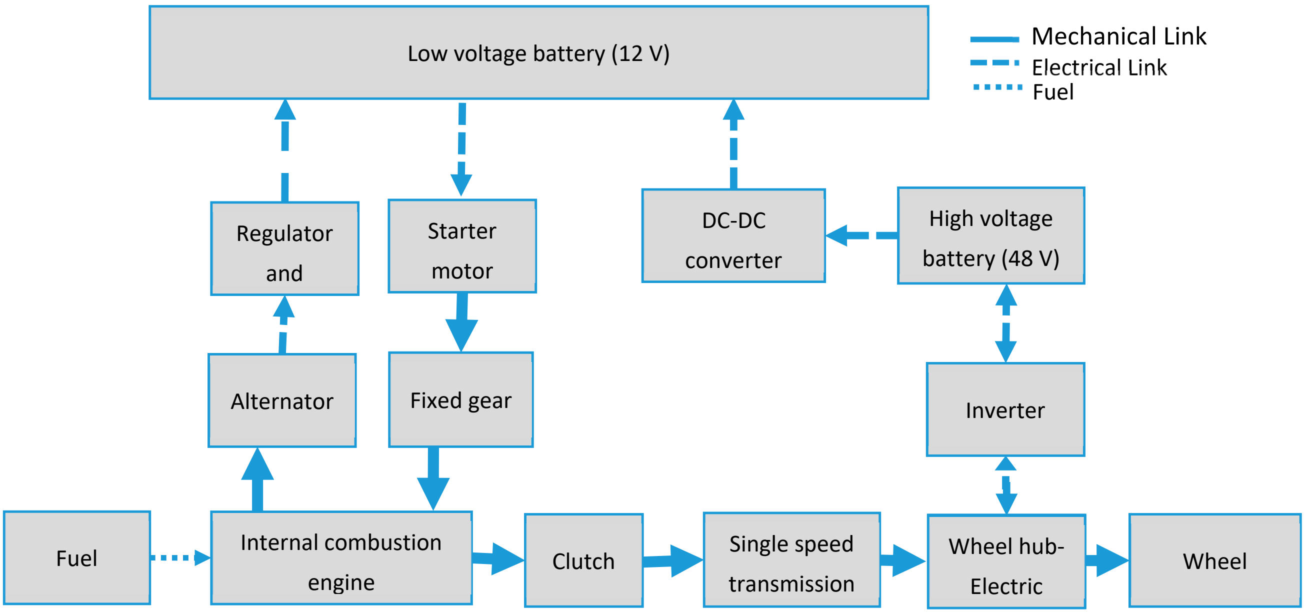


Figure 1. Powertrain architecture of the hybrid two-wheeler.

Two different approaches to HEV modelling can be adopted: backward or forward-facing modelling concerning the physical causality principles [23,24]. The powertrain data for the two-wheeler hybrid available drives this study to use a simplified appropriate backward-facing model for this purpose [22]. A backward model developed by sourcing the detailed technical specifications and experimental data of engine, electric machine, and battery from the hybrid two-wheeler considered for this study is explained in detail in the previous work by the author published in Energies journal *'Evaluation of a Modified Equivalent Fuel-Consumption Minimization Strategy Considering Engine Start Frequency and Battery Parameters for a Plugin Hybrid Two-Wheeler*—Section 2*: Vehicle model and system configuration'* [22]. The power sources, transmission and control system were developed using MATLAB/Simulink/State flow environment. Since the vehicle is a concept two-wheeler and not production-ready, the actual technical details of the powertrain components are not disclosed. In turn, the normalised values are displayed wherever necessary.

Figure 2 shows the engine characteristics. It shows the engine's BSFC map, engine power lines, engine max torque and engine optimum operating line. The Figure 3 shows the efficiency map of the traction machine along with max motor and generator torque.

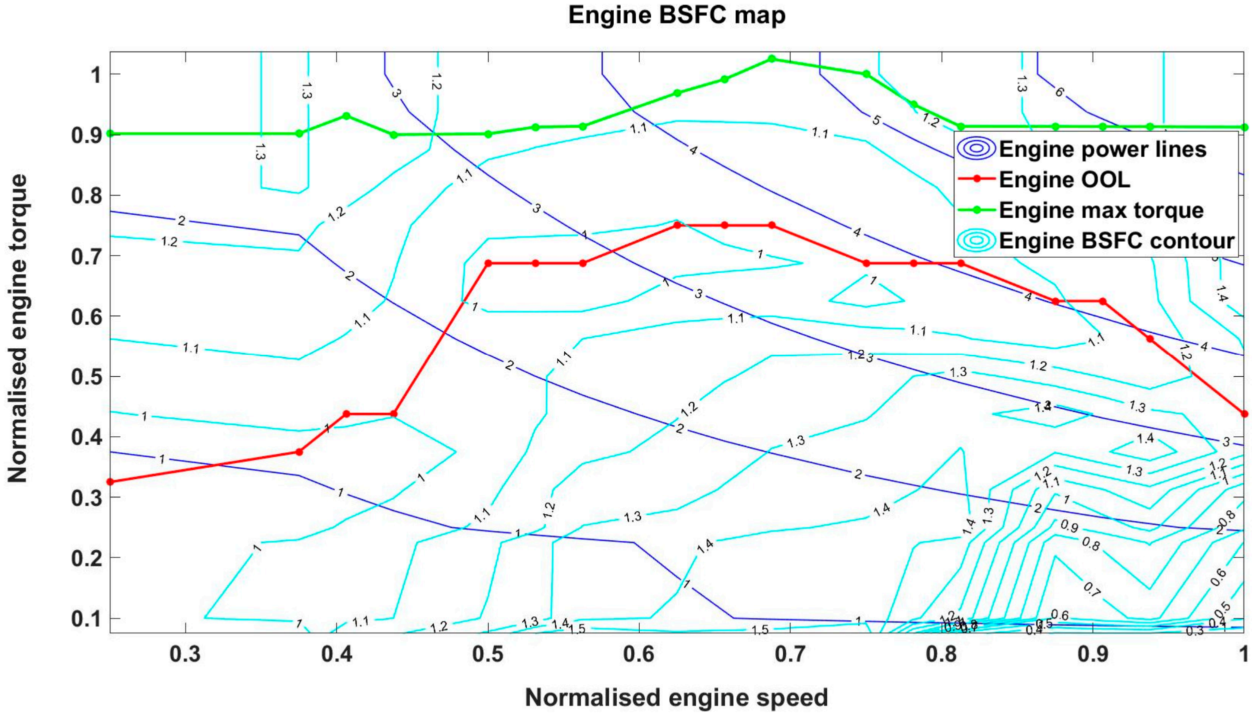


Figure 2. Normalised engine BSFC map.

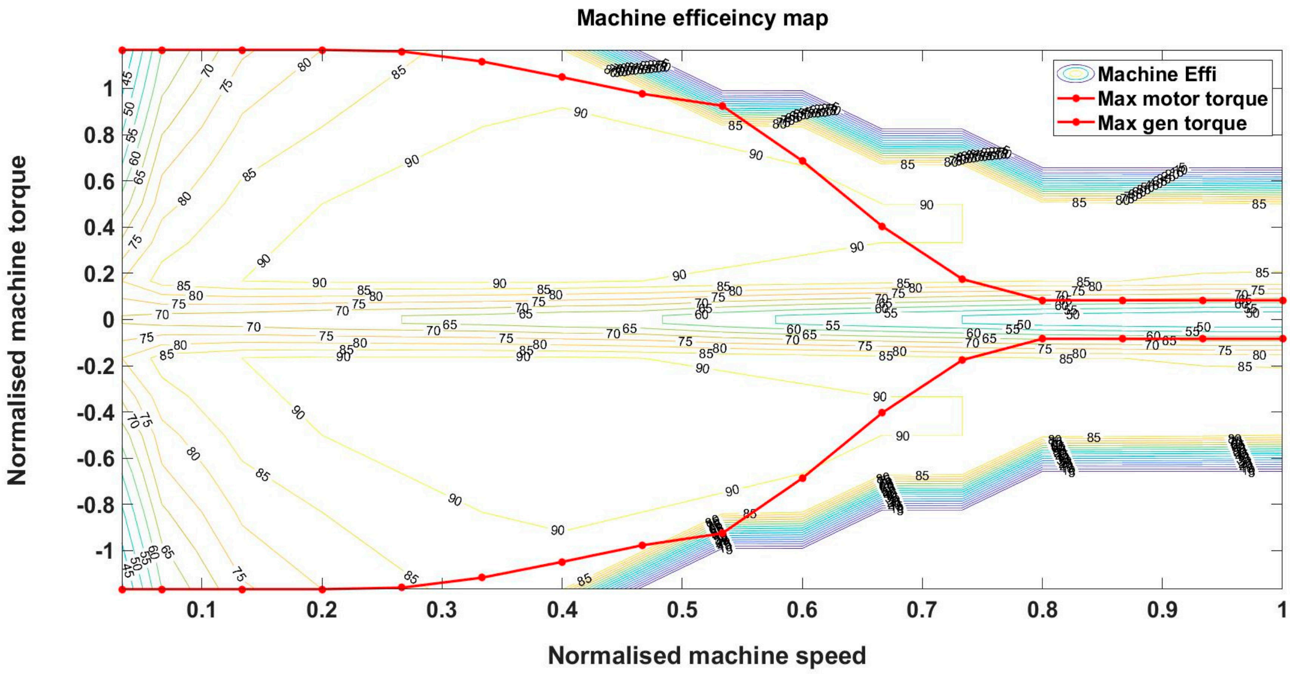


Figure 3. Efficiency map of the electric machine.

3. Novel Adaptive Equivalent Fuel Consumption Minimisation Strategy (ECMS_LL)

Previously, a modified novel equivalent fuel consumption minimisation strategy referred to as ECMS_LL has been designed and developed, which is published in [22]. The proposed modified ECMS attempts its maximum possibility to bring the engine operating points toward the engine optimum operating region by deriving a penalty function and implementing it into the cost function evaluation. The penalty function depends on engine operating points and their deviation from the engine optimum operating line at every time instant. The addition of this penalty function enhances the engine operating towards the engine optimum operating line, thereby improving the mean engine efficiency for the drive cycle considered.

The cost function for the conventional ECMS is represented as follows [22]:

$$J(x_t, u_t) = \dot{m}_{eqv}^{fuel}(t) = \left(\dot{m}_{ICE}^{fuel}(t) + \dot{m}_{BAT}^{fuel}(t)\right) = \left(\dot{m}_{ICE}^{fuel}(t) + \frac{s}{Q_{lhv}} P_{BAT}(t)\right) min \quad (1)$$

where $\dot{m}_{eqv}$ is the instantaneous equivalent fuel consumption, $\dot{m}_{ICE}$ is the instantaneous fuel consumption from the engine, $\dot{m}_{BAT}$ is the instantaneous equivalent fuel consumption from the battery power (both in charging and discharging), s is the equivalence factor, which represents the conversion of electric power into fuel consumption and P_{BAT} is the battery power, Q_{lhv} is the low heating value of the fuel. All the parameters in the equations are in SI units.

The novel modified ECMS, ECMS_LL has been developed where the cost function is modified with a new penalty factor based on the deviation of engine operating points from engine optimum operating line (OOL) [22].

The cost function for the modified ECMS_LL is represented as follows [22]:

$$J(x_t, u_t) = \dot{m}_{eqv}^{fuel}(t) = \left(\dot{m}_{ICE}^{fuel}(t) + \frac{\gamma S}{Q_{lhv}} P_{BAT}(t) + \beta\left(\frac{ICE_{OP}^{fueleff}}{ICE_{OOP}^{fueleff}}(t)\right)\right) \text{min} \quad (2)$$

where $\beta\left(\frac{ICE_{OP}^{fueleff}}{ICE_{OOP}^{fueleff}}(t)\right)$ is the new penalty factor in the cost function that is based on deviation engine operating point from optimum operating point. $ICE_{OP}^{fueleff}$ is the fuel efficiency of the engine operating point at a particular time instant, and $ICE_{OOP}^{fueleff}$ is the fuel efficiency of the engine's optimum operating point at that time instant. The addition of $\beta\left(\frac{ICE_{OP}^{fueleff}}{ICE_{OOP}^{fueleff}}(t)\right)$ affects the equivalence factor of conventional ECMS for the charge-sustained requirement. Therefore, while implementing this modified ECMS_LL in real-time, along with the challenge of estimating the equivalence factor 'γS' accurately, there is added task of estimating the penalty weight 'β'.

The hybrid electric vehicle is more sensitive to a driving cycle than an internal combustion vehicle when it comes to fuel consumption and tailpipe emissions [7]. Studies show that real-time drive cycles and driving trends have a large impact on fuel consumption and exhaust emissions of a vehicle [8]. The optimal equivalence factor significantly influences the fuel economy, and its influence varies with the drive cycle [12]. This makes it very important to estimate the equivalence factor accurately so that it could have a benefit on the fuel economy. One of the most important challenges in implementing an ECMS in the production of Hybrid electric vehicles is the estimation and adaptation of equivalence factors for real-world driving. Thus, the development of an adaptive ECMS control strategy, which is adaptable, robust and intelligent concerning real-time driving conditions, is required for real-world applications.

4. Proposed A-ECMS Methodology

In the proposed adaptive equivalence factor estimation strategy, a novel low computation-based, real-time implementable equivalence factor estimation algorithm has been designed and implemented for the considered hybrid two-wheeler. A learning vector quantitative neural network (LVQNN) technique-based classifier is proposed, designed and implemented which is trained using standardised drive cycles and real-world driving data. Figure 4 shows the control flowchart of the adaptive ECMS implemented.

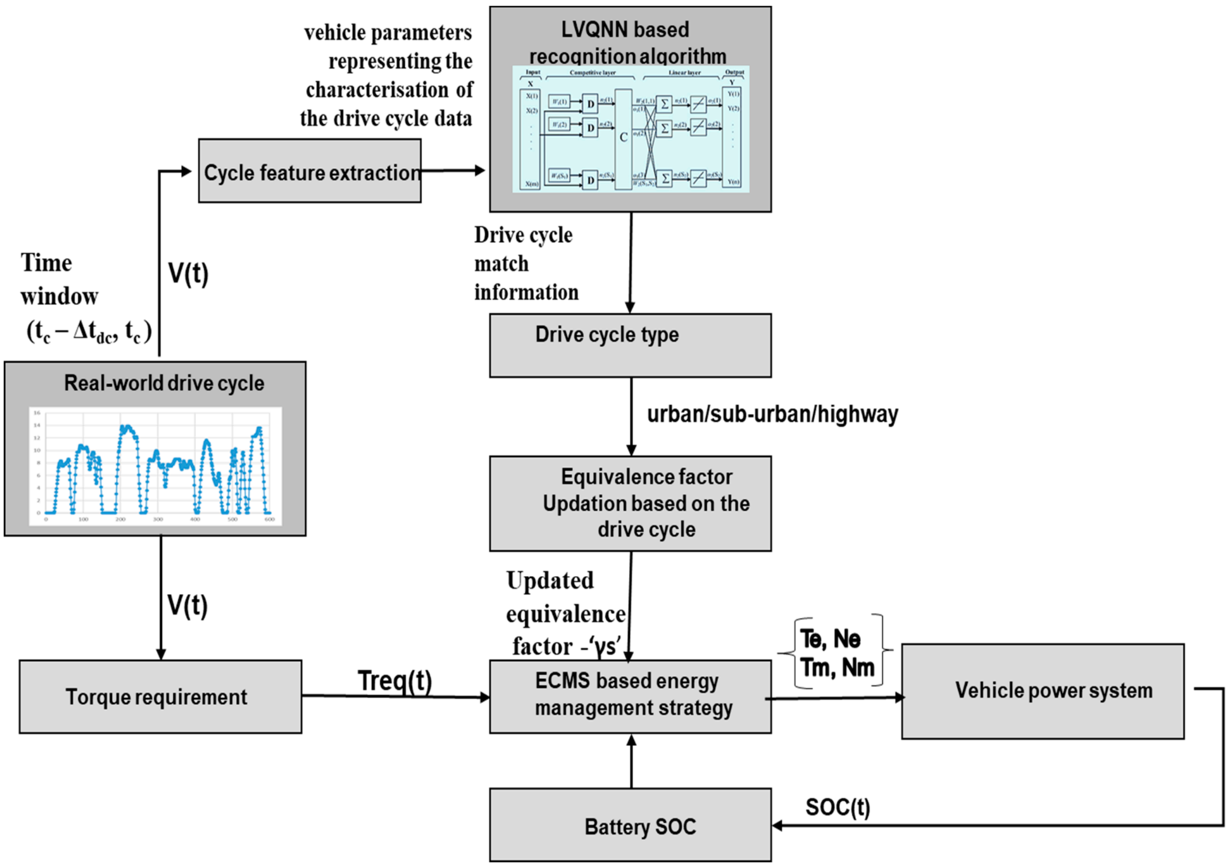


Figure 4. The control flowchart of adaptive ECMS (A-ECMS).

There are several steps involved in training and implementing the LVQNN-based equivalence factor estimation method. The first step involves offline training of the LVQNN neural network controller for a set of standardised and real road driving data sets. To keep the offline computation and online computation load low, instead of using all the drive cycles for training, three drive cycle sets, each corresponding to a different driving pattern (urban, suburban and highway category) were chosen for learning. These chosen drive cycles were characterised using identified driving parameters, which were generated from the speed profile over a pre-defined time window t_c. Figure 2 shows the block diagram of the online implementation of the LVQNN-based equivalence factor estimation. During the online computation, the real-world driving data are characterised and matched to one of the driving patterns used for the training (urban/suburban/highway) using the LVQNN recognition algorithm. The equivalence factor of the closest matched drive cycle is used from a stored value in a map.

The drive cycles considered for this study, the characterisation of the corresponding drive cycles and the training process of LVQNN methodology are explained in the later sections.

4.1. LVQNN Learning Methodology

Neural networks (NNs) are computing systems inspired by the biological neural networks that constitute animal brains. An NN is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain. They can learn from a set of data and construct weight matrices to represent the learning patterns [25]. NNs are usually classified as supervised or unsupervised learning based on their training processes. If the training is based on the desired responses to given stimuli, then it is termed supervised learning and if the training is based on clustering of stimuli without specified responses, then it is called unsupervised learning [26]. Here we use LVQNN, which is a hybrid network that uses advanced behaviours of both competitive learnings and thereby it applies a well-known Kohonen feature map for classification.

Figure 5 shows the structure of an LVQNN framework. The LVQNN structure contains the following four layers [17]:

First layer: Input layer with m nodes;
Second layer: Competitive layer with S_1 nodes;
Third layer: Linear layer with S_2 nodes;
Fourth layer: Output layer with n nodes.

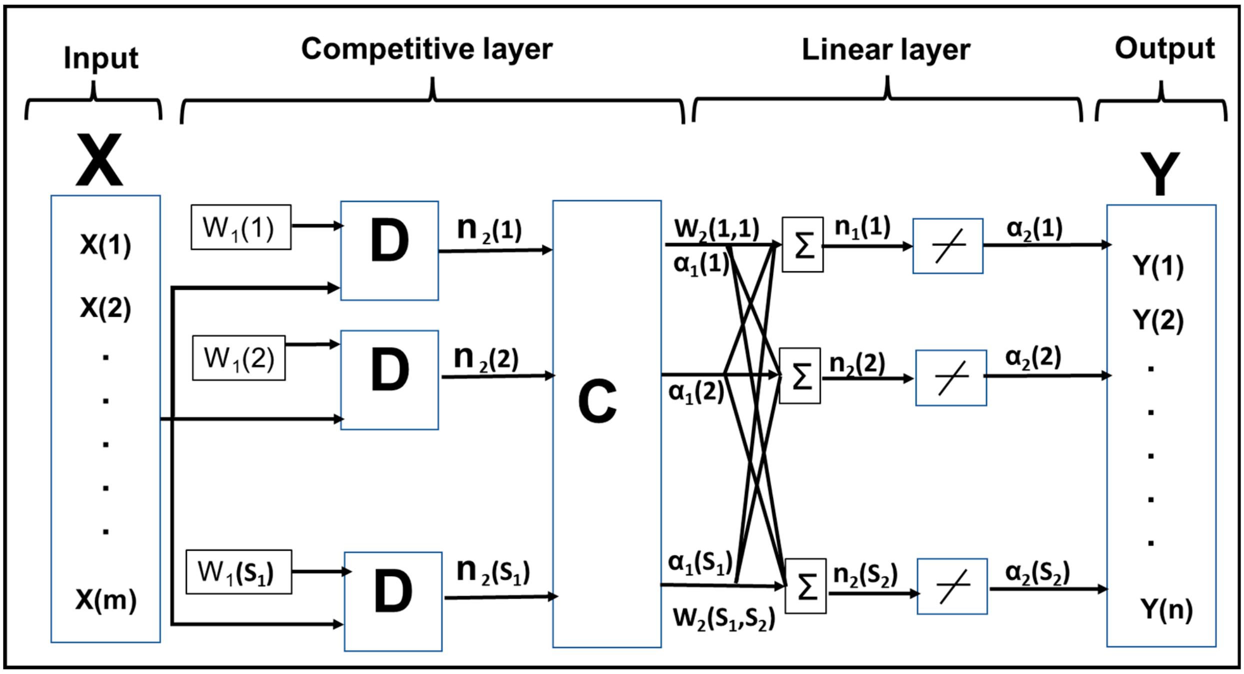


Figure 5. Structure of the LVQNN [19].

During the learning operation, the input layer with m modes is initialised in the first layer. In the second layer, the competitive layer is initialised, and then it maps the input vectors into the clusters through training. In the competitive layer, these clusters are merged into classes based on the input target data. Here the number of clusters to be merged is dependent on the hidden neurons considered. The bigger the hidden layer, the better is the learning and thereby mapping of input to target classes can be performed. With the proper selection of structure and training of the weighting factors, the LVQNN can classify the information of any system. The LVQNN is based on the nearest-neighbour method. For real-valued input variables, the most popular distance measure is Euclidean distance. Euclidean distance is calculated as the square root of the sum of the squared differences between a new point (X) and an existing point (Xi) for each attribute j [17]:

$$n_j = \mathrm{D}(\mathrm{X}, W_1(j)) = \sqrt{\sum_{i=1}^{m} (X(i) - W_1(j,i))^2}, \quad j = 1, \ldots S_1 \tag{3}$$

where X is the input vector and $W_1(j, i)$ is the weight of the jth node in the competitive layer corresponding to the ith element of the input vector. Thereby the Euclidean distances are passed onto the competitive transfer function which returns an output vector O_1. This vector is given as input to the linear layer and the output is derived for each element that relates to a node of the output layer and is computed as follows [17]:

$$Y(k) = O_2(k) = k_W(k)n_2(k) = k_W(k)\sum_{j=1}^{S_1} W_2(k,j)O_1(j), \quad k = 1, \ldots n, (n = S_2) \tag{4}$$

where $W_2(k,j)$ is the weight of node k in the linear layer corresponding to element j of the competitive output vector; and $k_W(k)$ is the linearised gain of node k in the linear layer. In the learning process, the weights of LVQNN are updated by the well-known Kohonen rule by the following equation [17]:

$$\begin{cases} W_1^{t+1}(j) = W_1^{t+1}(j) + \mu\left(X - W_1^t(j)\right) \\ W_1^{t+1}(j) = W_1^{t+1}(j) - \mu\left(X - W_1^t(j)\right) \end{cases}, \; j = 1 \ldots S_1 \tag{5}$$

where μ is the learning ratio which is positive, and decreased concerning the number of training iterations ($n_{iteration}$), $\mu = \frac{1}{n_{iteration}}$.

4.2. Drive Cycles Considered for the Study

For this study, several standardised drive cycles and some real-time on-road driving data sets are considered for training and testing. Table 1 shows the ten different drive cycles considered for this study. The first six drive cycles considered in the table are standardised drive cycles used for emission and fuel economy tests. The last four drive cycles in the table are on-road hybrid two-wheeler driving data. Each of these drive cycles can be categorised into one of these; urban, suburban, or highway.

Table 1. Drive cycles considered for the study (standardised drive cycle and on-road driving data (*)).

Drive Cycle	Maximum Velocity (m/s)	Average Velocity (m/s)	Max Acceleration (m/s^2)	Max Deceleration (m/s^2)	Drive Cycle Length (s)
WMTC	13.89	6.41	1.66	–1.94	600
IDC	11.67	6.03	0.67	–0.63	120
ECE15	13.70	4.98	1.03	–0.89	210
10–15_Japanese	19.44	6.41	0.93	–1.11	660
NYCC	12.33	3.16	2.66	–2.62	598
Manhattan	11.24	3.03	2.04	–2.48	1089
IND_SU *	17.27	6.38	1.72	–2.09	2689
IND_H *	20.98	13.18	2.10	–1.90	881
IND_U *	18.20	7.90	2.28	–2.61	867
IND_CD *	13.20	4.47	2.13	–2.71	5434

Hybrid electric vehicle performance may change dramatically between urban driving and highway driving [12]. For example, highway cycles usually have higher average vehicle speed while urban cycles usually have larger vehicle idling time and comparatively lower average vehicle speeds [12]. The characteristics of the drive cycles can be analysed and clustered into urban, suburban and highway. This segregation would make it easier for the training and adaptation process.

For the training LVQNN, three drive cycles are chosen: one from urban, one from sub-urban and the third one from highway driving pattern. The three drive cycles chosen for the training data were WMTC (urban), IND_SU drive cycle (suburban) and IND_H (highway) as shown in Figure 6. However, for the testing phase, all the ten-drive cycles were extended to 3000 s (16 km).

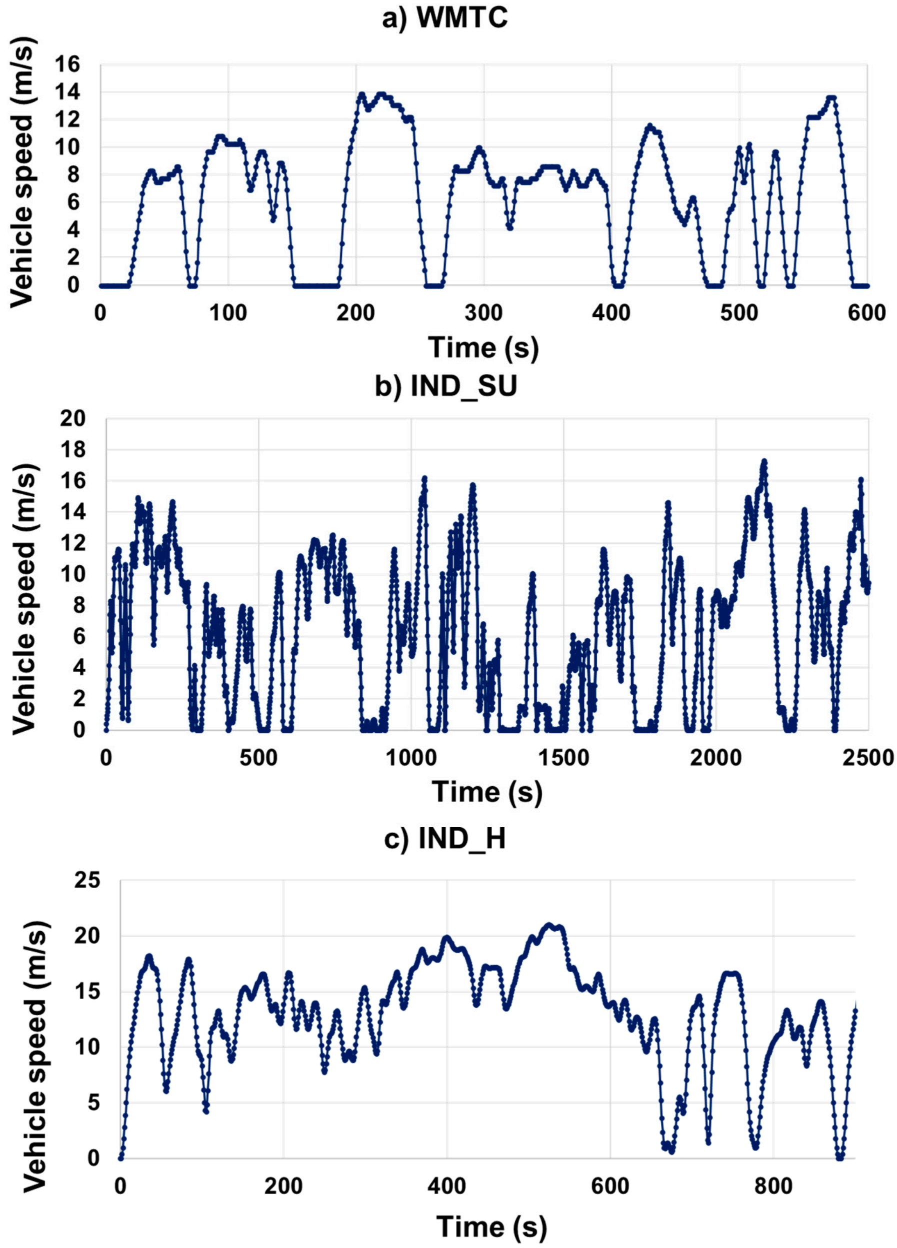


Figure 6. Drive cycles considered for training (**a**) WMTC, (**b**) IND_SU and (**c**) IND_H.

4.3. Characterisation of Drive Cycles

After the consideration of the drive cycles, the drive cycle characteristic parameters need to be analysed further. Dembski et al. [27] introduced a systematic method of analysing driving cycles and clustering. Twenty-one statistical metrics were used to represent the characteristics of a driving cycle. In [28], Beta et al. discussed the basic drive cycle characteristic parameters, which directly relate to fuel consumption and emissions. Twenty-four parameters including the grade information of the road are chosen to characterise driving patterns. Each of these twenty-four parameters was weighted based on its contribution to driving pattern recognition. In this paper, around ten drive cycle parameters that have a significant influence on fuel consumption and emissions are identified and considered [28]. The selection of representative features has a great impact on the effect of drive cycle recognition [29]. In [15] the study used 24 features of the driving pattern recognition. In [30], around 17 driving pattern parameters were considered for characterisation. In [31], around 15 drive cycle parameters were considered in the study for driving pattern recognition. In [29], 62 different driving parameters that might have influenced fuel economy and emissions of hybrid electric vehicles have been studied. The study concluded that out of sixty-two, nine drive cycle parameters had a significant effect on fuel economy and emissions of hybrid vehicles. A similar method is used in this study to classify the driving cycles. As shown in Table 2, for our study the eight most important drive cycle parameters responsible for the significant influence on fuel consumption are identified based on various literature studies [17,25,29]. All the velocity and acceleration related characteristic are taken into consideration.

Table 2. Drive cycle characteristics and their description.

Drive Cycle Characteristic DCC[i]	Characteristic Name	Description of Characteristic	Unit
DCC[1]	V_{avg}	Average velocity over the time window $(t_c - \Delta t_{dc}, t_c)$	m/s
DCC[2]	V_{max}	Maximum velocity over the time window $(t_c - \Delta t_{dc}, t_c)$	m/s
DCC[3]	ACC_{avg}	Average acceleration over the time window $(t_c - \Delta t_{dc}, t_c)$	m/s^2
DCC[4]	$DECC_{avg}$	Average deceleration over the time window $(t_c - \Delta t_{dc}, t_c)$	m/s^2
DCC[5]	V_{st}	Start velocity at $(t_c - \Delta t_{dc})$	m/s
DCC[6]	V_{end}	End velocity at t_c	m/s
DCC[7]	ACC_{max}	Maximum acceleration over the time window $(t_c - \Delta t_{dc}, t_c)$	m/s^2
DCC[8]	$DECC_{max}$	Maximum deceleration over the time window $(t_c - \Delta t_{dc}, t_c)$	m/s^2

A characteristic-selection algorithm was developed with the recorded speed profile as the input and a set of eight variables as the output, namely: average vehicle speed (V_{avg}), maximum vehicle speed (V_{max}), Average acceleration (ACC_{avg}), average deceleration ($DECC_{avg}$), vehicle speed start (V_{st}), vehicle speed end (V_{end}), maximum acceleration (ACC_{max}) and maximum deceleration ($DECC_{max}$) as shown in Table 1. All these parameters are calculated respectively for the time window defined as $(t_c - \Delta t_{dc}, t_c)$ where t_c is the present time of the drive cycle and Δt_{dc} is the predefined time window. All these parameters are used for the investigation and training of NN.

A neural network-based prediction of driving trends was developed using periodically updated eight drive cycle characteristic parameters over the predefined time window t_c as described above in the table. A time window range of 25 to 150 s has been used in previous similar studies for drive cycle or driving pattern recognition methods [14–16]. A higher time window range requires higher memory for data (drive cycle characteristic parameters) storage. Hence, for this study, the time window (Δt_{dc}) is fixed to the lower time window range of 25 s.

4.4. Application of LVQNN for Learning

In a network-training problem, the preceding task is to collect the system behaviour data to improve the performance of the training process [17]. To perform the investigation, three different categories of drive cycle are considered for the NN learning. The number of hidden neurons can influence the error on the nodes to which their output is connected. The stability of neural network is estimated by error. The minimal error reflects better stability, and higher error reflects worst stability. The excessive hidden neurons will cause over fitting; that is, the neural networks have over-estimated the complexity of the target problem. One of the problems that occur during neural network training is called overfitting. The error on the training set is driven to a very small value, but when new data are presented to the network the error is large. The network has memorised the training examples, but it has not learned to generalise to new situations. A lower hidden neurons cause underfitting and cause larger error during training as well as in new data. One of the major challenges in the design of neural network is the fixation of hidden neurons with minimal error and highest accuracy. A neuron range of 10–60 has been used in previous similar studies [17].

To investigate the performance of the LVQNN concerning different drive cycles, training was performed using the selected data set by varying the number of inputs from 2 to 8 (number of drive cycle characteristic parameters), and the number of hidden neurons from 20 to 60. After each training process, the correlation between the simulation output and target output was taken as indicative of the success of the training. The range of the number of neurons used typically in similar studies is 10–60 neurons [17]; the LVQNN training was therefore tested for a range of 10–60. The performance was evaluated by RMSE (root means square error). However, it was found that the test results of the range 20–60 neurons (nodes of the competitive layer) gave a good fit. Considering below 20 neurons gave a low success rate (accuracy levels were below the acceptable range and larger errors). Considering neurons above 60 gave a low success rate. The training results (goodness of fit (%)) of the LVQNN are shown in Figure 7 and Table 3. The results indicate that the most suitable LVQNN structure is comprised of six parameters in the input vector (average velocity, maximum velocity, average acceleration, average deceleration, start velocity and end velocity over the time window respectively) and 60 nodes in the competitive layer. The highest learning success rate was ~88.3% in this case, which is considered sufficient for the recognition of the drive cycle because usually anything above 75% is considered a good fit [17]. The corresponding weights matrix W1 and W2 from equations x and y are taken from the best-fit point (6, 60; representing the number of input parameters and the number of neurons respectively) and used for the online estimation and classification using LVQNN.

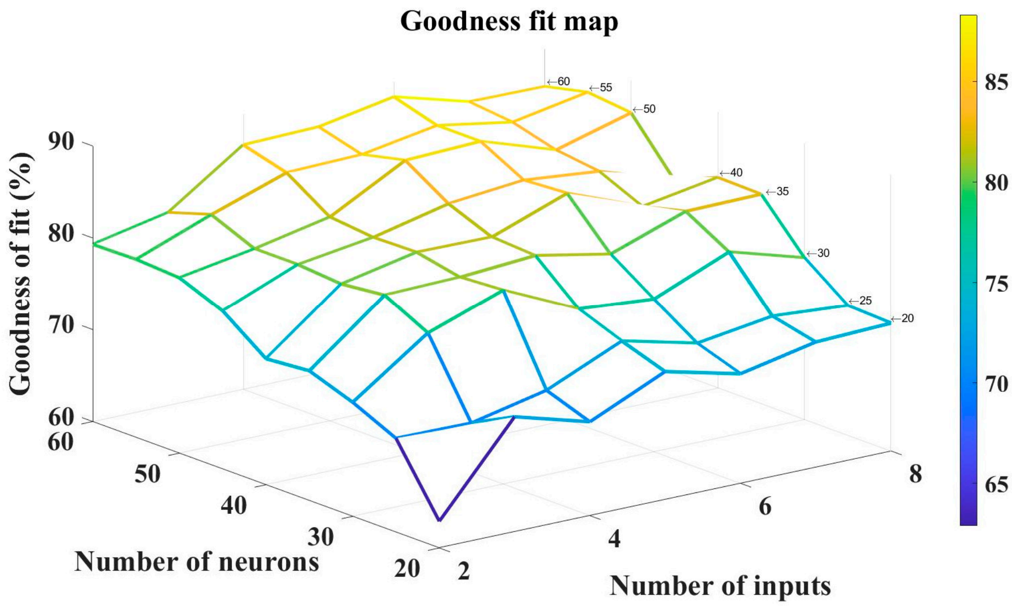


Figure 7. Goodness fit—3D map.

Table 3. The learning success rate of LVQNN (%).

Number of Inputs Considered	Number of Nodes in Hidden Layer								
	20	25	30	35	40	45	50	55	60
2	62.9	70.2	72.4	74.1	73.7	77.2	79.1	79.4	79.3
3	72.5	70.1	78.2	80.6	80.1	80.5	80.5	82.5	81
4	70.2	71.9	81.1	80.8	81.8	81.7	82.2	85.3	86.6
5	73.9	75.5	77.3	81.4	81.7	83.6	86.6	85.5	86.8
6	71.9	73.5	76.6	79.8	84.7	84.4	86.9	86.9	**88.3**
7	73.6	74.7	80	82.7	81.5	83.6	84.2	85.5	86
8	73.9	74.1	77.6	82.8	82.9	80.9	86.5	87	85.9
	Goodness of fit in %								

The red signifies the maximum learning success rate.

After the training process, the second step is the implementation and the integration of LVQNN-based drive cycle recognition with the ECMS controller. As shown in Figure 4, a predefined time window t_c past driving information with driving parameters characterizing the drive cycle is updated every one second. The LVQNN_DCR block is fed with these vehicle characteristic parameters. The block takes these inputs and, based on the previously trained information, identifies the present driving cycle and matches it to one of the nearest representative standardised drive cycles used for training. Later, based on the LVQNN_DCR output, the equivalence factor and penalty weights corresponding to the recognised drive cycle are used by the ECMS controller.

5. Evaluation of the Proposed Method for Different Drive Cycles

In this section, the capability of the proposed LVQNN_DCR controller for estimating the equivalence factor is evaluated. LVQNN_DCR is tested and evaluated for all ten-drive cycles shown in Table 1. From the results obtained from the LVQNN learning, the block LVQNN_DCR is constructed and used for the estimation of the equivalence factor in real-time. The LVQNN_DCR block recognises the candidate/new drive cycle and classifies the drive cycle to the nearest of the standardised drive cycles used for training (WMTC (urban), IND_SU drive cycle (suburban) and IND_H (highway)) as shown in Figure 8. In every time instant, the LVQNN_DCR block evaluates instantaneous equivalence factor based on the drive cycle parameters analysed for previous time windows of 25 s.

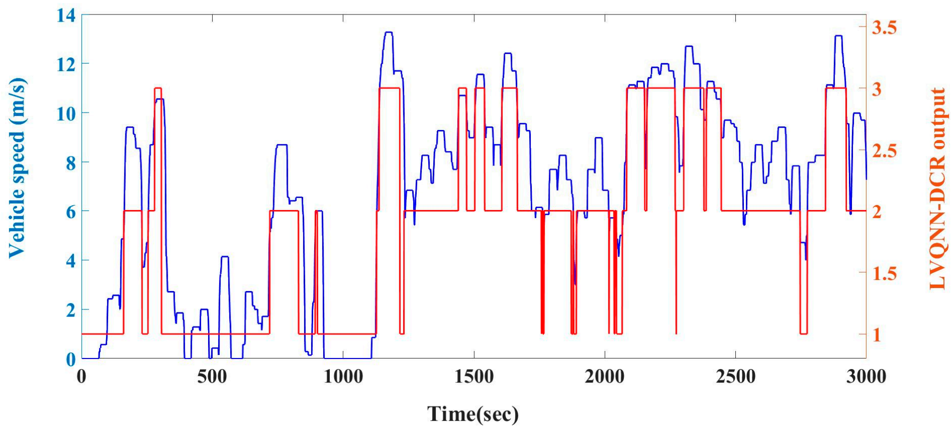


Figure 8. LVQNN_DCR output.

Figure 8 shows the LVQNN_DCR block output for the IND_CD input drive cycle; this is an 'unseen' real-time drive cycle. The output of LVQNN_DCR varies between 1, 2 and 3. Each number implies the match of the present drive cycle to the respective standardised drive cycles; 1, 2 and 3 correspond to the WMTC, IND_SU and IND_H drive cycles which were used during the LVQNN training respectively. Based on LVQNN_DCR output, the pre-optimised values of equivalence factor stored in a map are used correspondingly.

5.1. Performance Evaluation of LVQNN for Different Drive Cycles

The performance evaluation of the LVQNN is done using root mean square error (RMSE) and R-squared (R^2). The performance of LVQNN was tested for all the ten drives cycles considered for testing and evaluation. The RMSE and R^2 values for different drive cycles are as shown in Table 4.

Table 4. RMSE and R^2 values when LVQNN tested for various drive cycle.

Drive Cycles	WMTC	IDC	ECE 15	JP 10-15	NYCC	Manhattan	IND_SU	IND_H	IND_U	IND_CD
RMSE	0.013	0.009	0.014	0.014	0.018	0.22	0.014	0.012	0.01	0.008
R2	0.97	0.96	0.98	0.98	0.97	0.98	0.986	0.99	0.988	0.98

R^2 represents the proportion of variance explained by the model. R^2 is formulated as:

$$R^2 = 1 - (\mathrm{SSE}/\mathrm{SST})$$

SSE is the sum of squared errors, the sum of the squared differences between the actual values and predicted values. SST is the total sum of squares, the sum of the squared differences between the actual values and the mean of the actual values. A model that explains no variance would have an R^2 of 0. A model with an R^2 of 1 would explain all of the variances. Higher scores show better results. Most of the R-squared values in the Table 4 tested for various drive cycles show that the model covers nearly 98% variance on average.

The predicted drive cycles versus the actual drive cycle were plotted for all the drive cycles considered. Figure 9 shows the plot between the predicted and the actual drive cycle values for real-world test data.

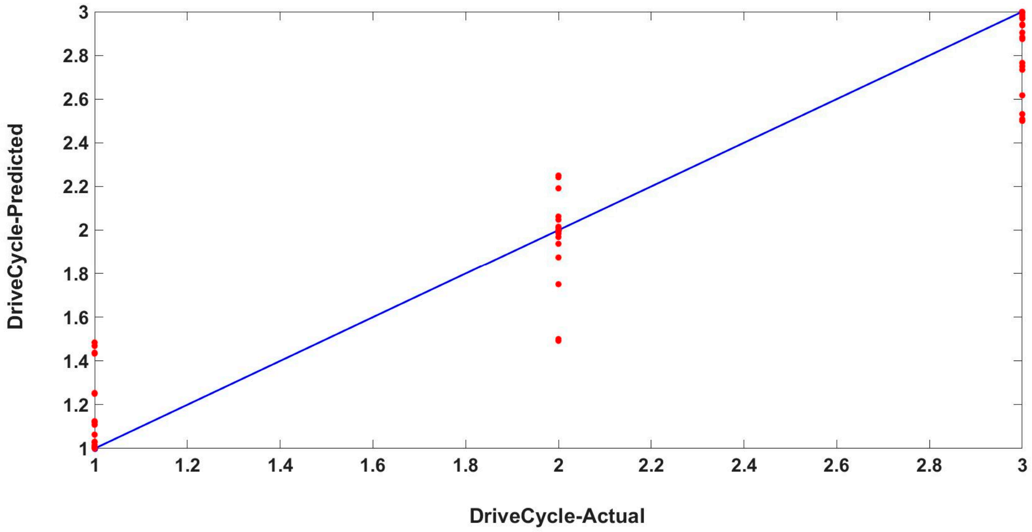


Figure 9. Predicted and actual drive cycle match for a real-world test data.

The actual drive cycle data matches one of the three drive cycles used for training and validation (1, 2 and 3 each referring to a particular drive cycle used for training). The predicted drive cycle data are as shown in the figure as red dots. Though there is some variation in the predicted drive cycle, it can predict most of the time within the limits of error.

5.2. FC Comparison for Optimal and Adaptive Equivalence Factor

The ten drive cycles considered for this study were tested with two controllers. First, the manually tuned equivalence factor (EQ_{WMTC}) in EMCS_LL was optimised for the WMTC drive cycle. Second, the equivalence factor (EQ_{LVQNN}) in AECMS was derived from the LVQNN-based estimation. The test cycles were run for a standard time of 3000 s, approximately equivalent to 16 km of driving. Table 5 shows the normalised fuel consumption of both the controllers ECMS_LL and AECMS for ten different drive cycles considered. As seen in Table 5, a noticeable benefit in fuel savings was achieved with AECMS when compared to ECMS_LL for almost all the drive cycles considered. However, in the case of the WMTC drive cycle, there is a reduction in the fuel benefit when tested with AECMS when compared to using ECMS_LL for obvious reasons.

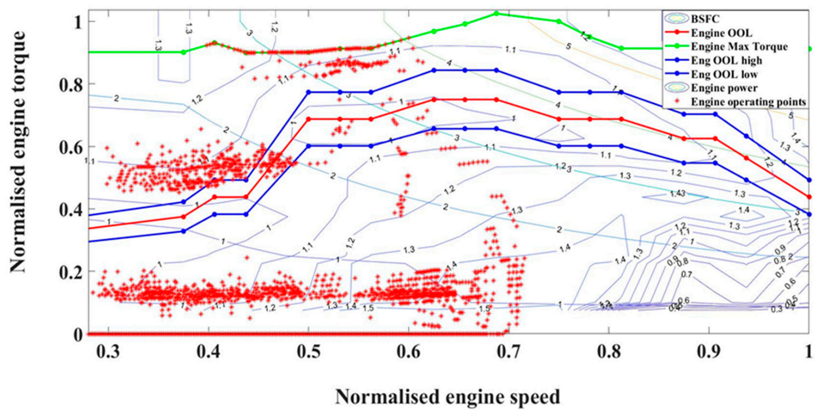
A maximum of 10.70% fuel benefit was seen by using the LVQNN-based estimation of equivalence factor in AECMS overusing an optimally tuned equivalence factor in ECMS_LL for the IND_CD drive cycle. The fuel economy benefit percentage varies with the drive cycle considered because the equivalence factor estimation is based on the nearest classification of the drive cycle to the standardised drive cycle. The average fuel economy benefits, when tested over the ten drive cycles considered for this study, was 3.93% demonstrating a consistent tangible benefit of this optimisation and adaptation strategy.

Table 5. Comparative study of fuel-saving for different drive cycles considered.

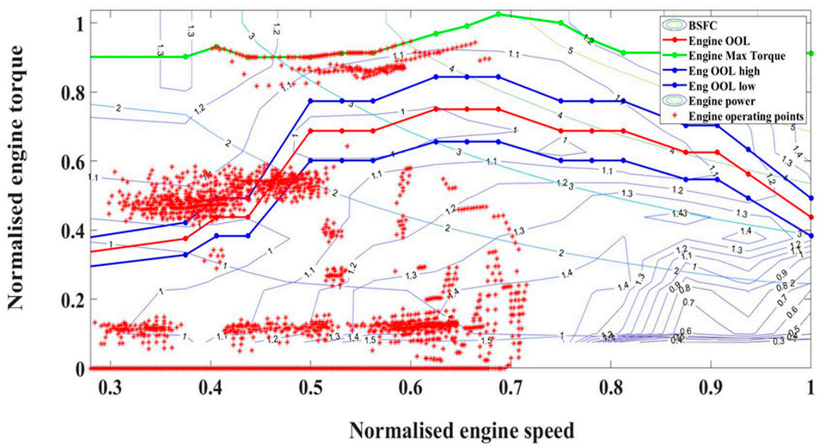
Drive Cycle	NFC with ECMS_LL	NFC with AECMS_LL	Fuel Savings (%)
WMTC	1.00	1.015	–1.47
IDC	1.00	0.938	6.21
ECE15	1.00	0.893	10.70
JP_10-15	1.00	0.991	0.90
NYCC	1.00	0.989	1.13
Manhattan	1.00	0.894	10.58
IND_SU *	1.00	0.974	2.64
IND_H *	1.00	0.994	0.61
IND_U *	1.00	0.973	2.71
IND_CD *	1.00	0.946	5.36

* Real world driving data.

Figure 10 shows the engine operating points for the ECMS_LL (a) and AECMS (b) controllers for the real-world driving data; IDC_CD. The AECMS controller uses the adaptive equivalence factor estimation using LVQNN, and ECMS_LL uses the equivalence factor optimised for the WMTC drive cycle. Thus, because of the better estimation of the equivalence factor with AECMS, the engine operating points are more optimally placed for the entire drive cycle thereby increasing the average engine efficiency.



(a) ECMS_LL with EQwmtc



(b) AECMS with EQLQNN

Figure 10. IDC_CD drive cycle with (**a**) ECMS_LL and (**b**) AECMS.

6. Discussion

The usual methodology of using a conventional ECMS is to manually tune the equivalence factor concerning the drive cycle considered. However, this does not work for a real-world application where the future drive cycles are unknown. Thus, an Adaptive Equivalent fuel Consumption Minimisation Strategy (A-ECMS) is essential to be developed and integrated to obtain a better estimation of the equivalence factor in real-time.

In this paper, a novel NN-based controller with better prediction precision and low computational load for cost-sensitive applications such as a hybrid two-wheeler has been developed. In this study, a learning vector quantitative neural network-based classifier was proposed, designed and implemented which was trained and tested on several standardised drive cycles and on-road driving data. The implementation of NN-based drive cycle recognition method involved a two-step process. The first step involved a learning process, where the system behaviour data, which was the drive cycle characterisation data, was used for the training process. To perform the investigation, three different categories of drive cycle were considered for the NN learning. The categorisation of driving patterns into urban, suburban and highway for training, eliminates the similar type of driving patterns to be used for training and thereby reduces the computational time for optimisation. The categories considered were urban, suburban and highway. Each of these drive cycles considered was characterised by eight drive cycle parameters. A detailed training process by varying the number of the inputs (drive cycle characterisation parameters) and the number of neurons (20–60) were run for all the combinations to get the best-fit matrix (weights W_1 and W_2). These weights are further used in the LVQNN-based drive cycle recognition controller LVQNN_DCR.

The LVQNN-based classifier dynamically identifies the present driving cycle and matches it to one of the nearest representative standardised drive cycles, based on the past driving information. During the operation, a predefined time window of 25 s past driving information of critical parameters defining the drive cycle was updated every second. A simple interpolated curve is used to estimate the equivalence factor and penalty weights based on the output of the LVQNN_DCR block.

Simulation results show that the real-time driving information can be matched to standardise drive cycles and provide better fuel economy compared to using conventional ECMS_LL while sustaining battery SOC within desired limits of target SOC. A maximum of 10.70% fuel benefit was seen by using the LVQNN-based estimation of equivalence factor in AECMS overusing a traditional ECMS_LL with equivalence factor optimised for WMTC. The average FE benefit over the ten drive cycles considered was 3.93%. However, the fuel benefit percentage depends on the driving data set considered. Along with this fuel benefit, the final SOC values also showed an acceptable limit of the SOC target. The simulation results showed that the final SOC values with adaptive ECMS_LL-based estimation were much closer to the target final SOC when compared to the ECMS_LL for a single drive cycle.

Unlike in the previous research [14], which uses the statistical and stochastic velocity forecast approaches for energy management optimisation, the proposed research is based on data-driven using Euclidean distance approach for drive cycle classification. Therefore, there is an advantage over prediction precision and computational load with the proposed method. In the previous study [11], an adaptive ECMS was implemented by updating the equivalence factor using prediction future cycles. Whereas in the proposed research, the past data over a fixed time window was used which prevents additional computational load, and adding a classifier would thereby make it a low computational solution in comparison. Similarly, In study [15], a rule-based control algorithm was extracted from the result of the optimal solution provided by dynamic programming (DP) on each RDP. Finally, a multi-mode driving control was realised by switching the control parameters in each RDP which required high computational capability making it non-suitable for cost-effective applications. Unlike the previous research, the results show that the proposed method based on LVQNN not only has a good prediction precision leading to significant fuel benefit but also has the advantage of low computational load. This makes the proposed method suitable for cost-sensitive applications such as the hybrid two-wheeler considered for this study.

The primary improvement of the proposed A-ECMS over other algorithms with similar objectives is that it does not require the knowledge of future driving cycles through external systems (GPS) or sensors or predictive models. The proposed adaptive strategy is designed and developed to achieve low computational burden on the controller. Results obtained in this research show that the driving conditions can be successfully recognised with better performance and can be achieved in various driving conditions while sustaining battery SOC within desired limits. The study uses real driving data set for training and testing along with the standardised drive cycles, unlike previous research which considers only standardised drive cycles. Thus, the proposed methodology has further enhanced the training and testing process making it much more suitable for real-world applications when compared to traditional methodologies.

This study emphasises that the potential implementation of ECMS in a production vehicle for a near to optimal solution is incomplete without estimating the equivalence factor accurately. The simulation results presented in this study provide insights that the equivalence factor is very sensitive and has a high influence when it comes to fuel economy and charge sustenance. The LVQNN-based estimation method developed and implemented shows improvement in fuel benefit and charge sustenance when compared to traditional ECMS_LL with equivalence factor optimised to a single drive cycle.

7. Conclusions and Future Work

In the previous paper [21] a novel equivalent ECMS strategy was developed, and which showed an improvement in the fuel consumption w.r.t to the traditional ECMS. However, to make the novel ECMS real-time capable, a method for online estimation of equivalence factor was necessary. In this research work, a novel LVQNN-based drive cycle recognition strategy has been developed and implemented for the online adaptation of the equivalence factor for ECMS. In this study, a learning vector quantitative neural network-based classifier was proposed, designed and implemented.

For this investigation, three different categories of drive cycles were considered for the NN learning. The driving cycle categories considered were one of each urban, sub-urban and highway. Unlike previous research, the proposed methodology has considered both standardised and real-world driving data for the training and testing of the drive cycle recognition, which makes this methodology more robust for the real world. Each of these drive cycles considered was characterised by eight drive cycle parameters. Keeping the prerequisite for production-ready solution, one of the requirements was to design a cost-effective EMS solution. Thus, the number and choice of the drive cycle used for training, the time window range, and the number of drive cycle characteristics considered for this adaptation methodology were consciously selected keeping the computation load in consideration. There was considerable improvement in the fuel economy with the proposed methodology when compared to the standard ECMS, optimised for a single drive cycle. The maximum FE benefit achieved with the novel adaptive strategy over the ECMS strategy optimised for a single drive cycle was around 10.7%. The average improvement in FE over the ten drive cycles considered for this study was found to be 3.93%.

The proposed method of adaptive equivalence factor is not only precise and efficient but also provides a low computational load suitable for cost-sensitive applications.

Author Contributions: Conceptulisation, methodology, data curation, validation and writing original draft: N.K.K.; Software: N.K.K. and D.Q.T.; Writing reviewing and editing: A.M., D.Q.T. and A.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

Nomenclature

HEV	Hybrid Electric Vehicle
PHEV	Plugin Hybrid Electric Vehicle
EMS	Energy Management Strategy
ECMS	Equivalent Fuel-Consumption Minimisation Strategy
DP	Dynamic programming
RDP	Representative Driving Pattern
NN	Neural Network
LVQNN	Learning Vector Quantitative Neural Network (LVQNN)
SOC	State of Charge
IEMA	Intelligent Energy Management Agent
DPR	Driving pattern Recognition
CNN	Convolution Neural Network
NAR	Non-Linear Auto Regressive
A-ECMS	Adaptive Equivalent Fuel-Consumption Minimisation Strategy
GPS	Global Positioning System
EF	Equivalence Factor

References

1. Chan, C.C. The State of the Art of Electric and Hybrid Vehicles. *Proc. IEEE* **2002**, *90*, 247–275. [CrossRef]
2. Wirasingha, S.G.; Emadi, A. Classification and Review of Control Strategies for Plug-In Hybrid Electric Vehicles. *IEEE Trans. Veh. Technol.* **2011**, *60*, 111–122. [CrossRef]
3. Salmasi, F.R. Control Strategies for Hybrid Electric Vehicles: Evolution, Classification, Comparison, and Future Trends. *IEEE Trans. Veh. Technol.* **2007**, *56*, 2393–2404. [CrossRef]
4. Zeng, Y.; Cai, Y.; Kou, G.; Gao, W.; Qin, D. Energy Management for Plug-In Hybrid Electric Vehicle Based on Adaptive Simplified-ECMS. *Sustainability* **2018**, *10*, 2060. [CrossRef]
5. Huang, Y.-J.; Yin, C.-L.; Zhang, J.-W. Design of an energy management strategy for parallel hybrid electric vehicles using a logic threshold and instantaneous optimization method. *Int. J. Automot. Technol.* **2009**, *10*, 513–521. [CrossRef]
6. Škugor, B.; Deur, J.; Cipek, M.; Pavković, D. Design of a power-split hybrid electric vehicle control system utilizing a rule-based controller and an equivalent consumption minimization strategy. *Proc. Inst. Mech. Eng. Part D J. Automob. Eng.* **2014**, *228*, 631–648. [CrossRef]
7. Lin, C.C.; Jeon, S.; Peng, H.; Lee, J.M. Control of a Hybrid Electric Truck Based on Driving Pattern recognition. In Proceedings of the Advanced Vehicle Control Conference, Hroshima, Japan, 9–13 September 2002.
8. Niu, L.; Yang, H.; Zhang, Y. Intelligent HEV fuzzy logic control strategy based on identification and prediction of drive cycle and driving trend. *World J. Eng. Technol.* **2015**, *3*, 215–226. [CrossRef]
9. Chen, B.-C.; Wu, Y.-Y.; Wu, Y.-L.; Lin, C.-C. Adaptive Power Split Control for a Hybrid Electric Scooter. *IEEE Trans. Veh. Technol.* **2011**, *60*, 1430–1437. [CrossRef]
10. Tang, L.; Rizzoni, G.; Onori, S. Energy Management Strategy for HEVs Including Battery Life Optimization. *IEEE Trans. Transp. Electrif.* **2015**, *1*, 211–222. [CrossRef]
11. Musardo, C.; Rizzoni, G.; Guezennec, Y.; Staccia, B. A-ECMS: An Adaptive Algorithm for Hybrid Electric Vehicle Energy Management. *Eur. J. Control* **2005**, *11*, 509–524. [CrossRef]
12. Gu, B.; Rizzoni, G. An adaptive algorithm for hybrid electric vehicle energy management based on driving pattern recognition. In Proceedings of the IMECE2006, ASME International Mechanical Engineering Congress and Exposition, Chicago, IL, USA, 5–10 November 2006; pp. 5–10.
13. Dhruv, T.V.S.; Sadashiv, K.A. Study ont he implementation of Hybrdi two-wheelrs in the Indian Market. *Int. Res. J. Eng. Technol. IRJET* **2018**, *5*, 417–422.
14. Langari, R.; Won, J.-S. Intelligent Energy Management Agent for a Parallel Hybrid Vehicle-Part I: System Architecture and Design of the Driving Situation Identification Process. *IEEE Trans. Veh. Technol.* **2005**, *54*, 925–934. [CrossRef]
15. Jeon, S.-I.; Jo, S.-T.; Park, Y.-I.; Lee, J.-M. Muti-mode driving control of a parallel hybrid electric vehicle using driving pattern recognition. *J. Dyn. Syst. Control* **2002**, *124*, 141–149. [CrossRef]
16. Chao, S.; Hu, X.; Moura, S.J.; Sun, F. Velocity Predictors for Predictive Energy Management in Hybrid Electric Vehicles. *IEEE Trans. Control. Syst. Technol.* **2015**, *23*, 1197–1204. [CrossRef]
17. Tria, N.M.; Truong, D.Q.; HoangThinh, D.; Binh, P.C.; Dung, D.T.; Lee, S.; Park, H.G.; Ahn, K.K. A novel control method to maximize the energy-harvesting capability of an adjustable slope angle wave energy converter. *Int. J. Renew. Energy* **2016**, *97*, 518–531. [CrossRef]
18. Sun, C.; He, H.; Sun, F. The Role of Velocity Forecasting in Adaptive-ECMS for Hybrid Electric Vehicles. *Energy Procedia* **2015**, *75*, 1907–1912. [CrossRef]

19. Feng, L.; Liu, W.; Chen, B. Driving Pattern Recognition for Adaptive Hybrid VEhicle Control. *SAE Int. J. Altern. Powertrains* **2012**, *1*, 169–179.
20. Xie, L.; Tao, J.; Zhang, Q.; Zhou, H. CNN and KPCA-Based Automated Feature Extraction for Real Time Driving Pattern Recognition. *IEEE Access* **2019**, *7*, 123765–123775. [CrossRef]
21. Yu, Y.; Jiang, J.; Wang, P.; Li, J. A-EMCS for PHEV based on real-time driving cycle prediction and personalized travel characteristics. *J. Math. Biociences Eng.* **2020**, *17*, 6310–6341.
22. Kommuri, N.K.; McGordon, A.; Allen, A.; Truong, D.Q. Truong Evaluation of a Modified Equivalent Fuel-Consumption Minimization Strategy Considering Engine Start Frequency and Battery Parameters for a Plugin Hybrid Two-wheeler. *Energies* **2020**, *13*, 3122. [CrossRef]
23. Delavaux, M.; Lhomme, W.; McGordon, A. Comparison between forward and backward approaches for the simulation of an electric vehicle. In Proceedings of the IEEE Vehicle Power and Propulsion Conference (VPPC2010), Lille, France, 3–5 September 2010.
24. Dixon, G.; Stobart, R.; Steffen, T. *Unified Backwards Facing and Forwards Facing Simulation of a Hybrid Electric Vehicle Using MATLAB Simscape*; SAE Technical Paper; SAE International: Detroit, MI, USA, 2015; Volume 1.
25. Sharif, M.S.; Abbod, M.; Amira, A.; Zaidi, H. Artificial neural network-statictical approach for PET volume analysis and classification. *Adv. Fuzzy Syst.* **2012**, *2012*, 327861.
26. Leng, F.; Tan, C.M.; Pecht, M. Effect of Temperature on the Aging rate of Li Ion Battery Operating above Room Temperature. *Sci. Rep.* **2015**, *5*, 12967. [CrossRef] [PubMed]
27. Dembski, N.; Guezennec, Y.; Soliman, A. *Analysis and Experimental Refinement of Real-World Driving Cycles*; SAE International Congress and Exposition Paper; SAE International: Detroit, MI, USA, 2002.
28. Bata, R.; Yacoub, Y.; Wang, W.; Lyons, D.; Gambino, M.; Rideout, G. *Heavy Duty Testing Cycles: SURVEY and Comparison*; SAE Paper; SAE International: Detroit, MI, USA, 1994; pp. 29–41.
29. Ericsson, E. Independant driving pattern factors and their influence on fuel-use and exhaust emission factors. *Transp. Res. Part D Transp. Environ.* **2001**, *6*, 325–345. [CrossRef]
30. Tian, Y.; Zhang, X.; Zhang, L.; Zhang, X. Intelligent Energy Management Based on Driving Cycle Identification Using Fuzzy Neural Network. In Proceedings of the 2nd International Symposium on Computational Intelligence and Design, Changsha, China, 12–14 December 2009; Volume 2, pp. 501–504.
31. Langari, R.; Won, J.-S. Integrated drive cycle analysis for fuzzy logic based energy management in hybrid vehicles. In Proceedings of the 12th IEEE International Conference on Fuzzy Systems, St. Louis, MO, USA, 25–28 May 2003; pp. 290–295.