

# Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering

<http://pid.sagepub.com/>

---

## Designing an intelligent control strategy for hybrid powertrains utilizing a fuzzy driving cycle identification agent

Ali Safaei, Mohammad Reza Ha'iri-Yazdi, Vahid Esfahanian, Mohsen Esfahanian, Masood Masih Tehrani and Hassan Nehzati

*Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* published online 5 November 2014

DOI: 10.1177/0954407014556116

The online version of this article can be found at:  
<http://pid.sagepub.com/content/early/2014/11/11/0954407014556116>

---

Published by:



<http://www.sagepublications.com>

On behalf of:



[Institution of Mechanical Engineers](#)

Additional services and information for *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* can be found at:

Email Alerts: <http://pid.sagepub.com/cgi/alerts>

Subscriptions: <http://pid.sagepub.com/subscriptions>

Reprints: <http://www.sagepub.com/journalsReprints.nav>

Permissions: <http://www.sagepub.com/journalsPermissions.nav>

Citations: <http://pid.sagepub.com/content/early/2014/11/11/0954407014556116.refs.html>

>> [OnlineFirst Version of Record - Nov 14, 2014](#)

[OnlineFirst Version of Record - Nov 5, 2014](#)

[What is This?](#)

# Designing an intelligent control strategy for hybrid powertrains utilizing a fuzzy driving cycle identification agent

**Ali Safaei<sup>1</sup>, Mohammad Reza Ha'iri-Yazdi<sup>1</sup>, Vahid Esfahanian<sup>1</sup>, Mohsen Esfahanian<sup>2</sup>, Masood Masih Tehrani<sup>3</sup> and Hassan Nehzati<sup>1</sup>**

## Abstract

In this paper, a new idea for designing the control strategy for the energy management of hybrid powertrains based on the driving cycle type is presented. Here, every instance of an unknown driving cycle is considered to be similar to the reference driving cycles using similarity weights. To determine the control output in the unknown driving cycle, the weights are applied to a linear combination of the optimal control decisions generated in each of the reference driving cycles. The weights which are between zero and one are determined using a fuzzy driving cycle identification agent based on the comparison of preselected driving features. The simulation studies in seven different driving cycles show that, while all driving patterns in every driving cycle are considered for the generation of energy management by online implementation of the proposed intelligent control strategy, some driving patterns would be eliminated by using a non-fuzzy identification agent. This leads to a significant reduction in the fuel consumption of the hybrid powertrain utilized with the fuzzy identification agent in some driving cycles in comparison with those without the use of non-fuzzy driving cycle identification. In addition, in some driving cycles, the intelligent control strategy has a performance close to that for the offline optimized control strategy.

## Keywords

Intelligent system, control strategy, hybrid powertrain, fuzzy driving cycle identification agent, dynamic programming

Date received: 28 April 2014; accepted: 17 September 2014

## Introduction

The design and implementation of a hybrid powertrain and, in particular, the distribution of the vehicle's torque demand between two power sources have presented some challenging problems. This torque distribution is performed through a control strategy, which is the core issue of this paper. In 2007, Salmasi<sup>1</sup> classified all control strategies for hybrid powertrains, including control strategies based on the driving cycle type, in two main groups. The first group includes rule-based control strategies which use some rules that are generated by experts. The rules could be in a non-fuzzy (or deterministic) scheme or a fuzzy scheme. The thermostat (on–off) control strategy and the power-follower control strategy are two popular categories within the group of deterministic control strategies. In the thermostat control strategy,<sup>2</sup> the state of charge (SOC) of the energy storage system is always maintained between its preset

top line and bottom line by turning the engine on and off. Despite its simplicity, this strategy cannot satisfy power demands by the vehicle in all operating conditions.<sup>1</sup> In the power-follower control strategy,<sup>2</sup> the engine is the primary source of power and the secondary power source is used to produce additional power when needed by the vehicle, while sustaining a charge in the energy storage system.<sup>1</sup> The control strategy used

<sup>1</sup>School of Mechanical Engineering, College of Engineering, University of Tehran, Tehran, Iran

<sup>2</sup>Department of Mechanical Engineering, Isfahan University of Technology, Isfahan, Iran

<sup>3</sup>School of Automotive Engineering, Iran University of Science and Technology, Tehran, Iran

## Corresponding author:

Ali Safaei, School of Mechanical Engineering, College of Engineering, University of Tehran, Tehran, Iran.

Email: ali.safaei@ut.ac.ir

for the Toyota Prius and the Honda Insight were developed on the basis of the power-follower approach.<sup>1</sup> Also, fuzzy logic seems to be the most suitable approach for specifying the control strategy rules of the hybrid powertrain by considering the powertrain as a multi-domain, non-linear and time-varying plant.<sup>1</sup> There are several investigations in designing a fuzzy control strategy for hybrid powertrains. Although most of these are designed for electric hybrid passenger vehicles,<sup>3–6</sup> however, some are for hydraulic hybrid powertrains.<sup>7</sup> Usually rule-based control strategies are far from the optimum strategy. The second group of strategies consists of those that are extracted by using an optimization algorithm. Dynamic programming (DP) is the most popular algorithm for performing an optimized control strategy in hybrid powertrains. Lin et al.<sup>8</sup> applied the DP technique to solve the optimal power management problem of a hybrid electric truck by minimizing a cost function over a driving cycle.<sup>1</sup> In addition, Wu et al.<sup>9</sup> performed a similar study for a hydraulic hybrid delivery truck. The problem of these control strategies is that they would be the optimum choices for only a specified driving cycle. Moreover, they are generated offline. A separate group can be defined for the control strategies of hybrid powertrains. Control strategies based on the driving cycle type which consider the environmental conditions of the vehicle are called intelligent control strategies. The types of driving cycle, the level of congestion on the road and the number of stops and starts of the vehicle in the driving cycle are examples of the environmental conditions of a vehicle, which affect the amount of fuel consumption. In 2002, Jeon et al.<sup>10</sup> and Lin et al.<sup>11</sup> incorporated a pattern recognition process in order to consider the driving environments in the energy management of a hybrid electric vehicle. In addition, in 2005, Langari and Won<sup>12</sup> and Won and Langari<sup>13</sup> presented an intelligent control strategy for an electric hybrid car. They generated a fuzzy rule-based torque distributor for each of nine reference driving cycles. A driving pattern recognition process was performed for an unknown driving cycle and a control strategy was chosen among the extracted strategies. This phase was performed according to the similarity of each portion of the driving cycle to one of the reference driving cycles. In 2007, Abdollahi et al.<sup>14</sup> extracted an optimal control strategy for each reference driving cycle using a static optimization algorithm. The rules of the fuzzy rule-based control strategy were generated using the results of the optimization algorithm. Furthermore, the number of features used for recognition of an unknown driving cycle was reduced to five features in contrast with the 16 features used in Won's work. In the above two investigations, a linear vector quantization network was used for driving pattern recognition. In 2009, Park et al.<sup>15</sup> developed a multi-layered multi-class neural network for this task. There, a special function was suggested for representing the fuel rate of an internal-combustion engine (ICE) in an electric hybrid

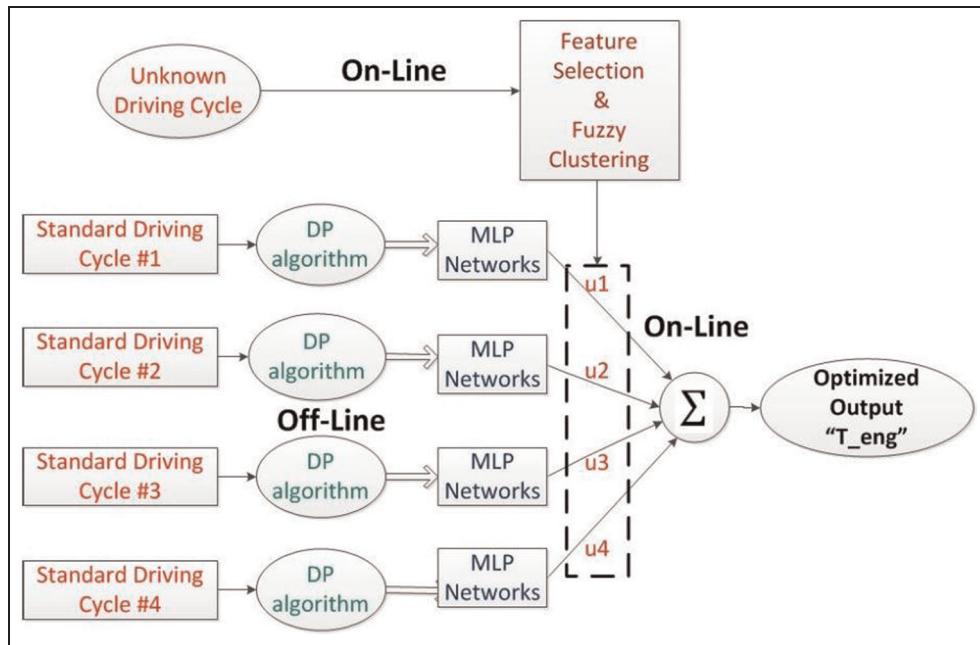
powertrain according to the type of the road. Finally, quadratic programming was used to determine the optimal torque distribution with respect to the recognized function for the fuel rate. Moreover, Chen et al.<sup>16</sup> and Murphrey et al.<sup>17</sup> continued these investigations by developing the proposed intelligent control strategy for hybrid electric passenger vehicles. In 2009, Huang et al.<sup>18</sup> used four features for distinguishing between the driving cycle types. They considered only two driving cycle types with two non-fuzzy rule-based control strategies for each of the driving cycles. Montazeri-Gh et al.<sup>19,20</sup> worked on the design of intelligent control strategies equipped with a pattern recognition process for an electric hybrid powertrain.

In all the above studies, each portion of a driving cycle has been assigned to only one reference driving cycle. Since the driving cycle used in a model is not completely similar to a reference driving cycle, a combination of the reference driving cycles can be used to identify the driving conditions more accurately. Here, 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. To implement the fuzzy clustering, a feature selection task is performed via a floating search method, which is flexible so that the features previously discarded and also the discarded features previously selected can be reconsidered.<sup>21</sup> Using a DP method, the optimal decisions for each case in four reference driving cycles were generated. Then, each of the four optimal control strategies are modelled using multi-layer perceptron (MLP) networks. A linear combination of optimal results for each reference driving cycle is used to generate the torque distribution in the hybrid powertrain using fuzzy similarity weights. Finally, in order to assess the capability of the presented intelligent control strategy, a parallel hydraulic hybrid bus model is simulated in seven distinct driving cycles with four different control strategies: a rule-based control strategy; an intelligent control strategy utilized with fuzzy identification; an intelligent control strategy utilized with non-fuzzy identification; an optimized control strategy.

## Design and implementation of the intelligent control strategy

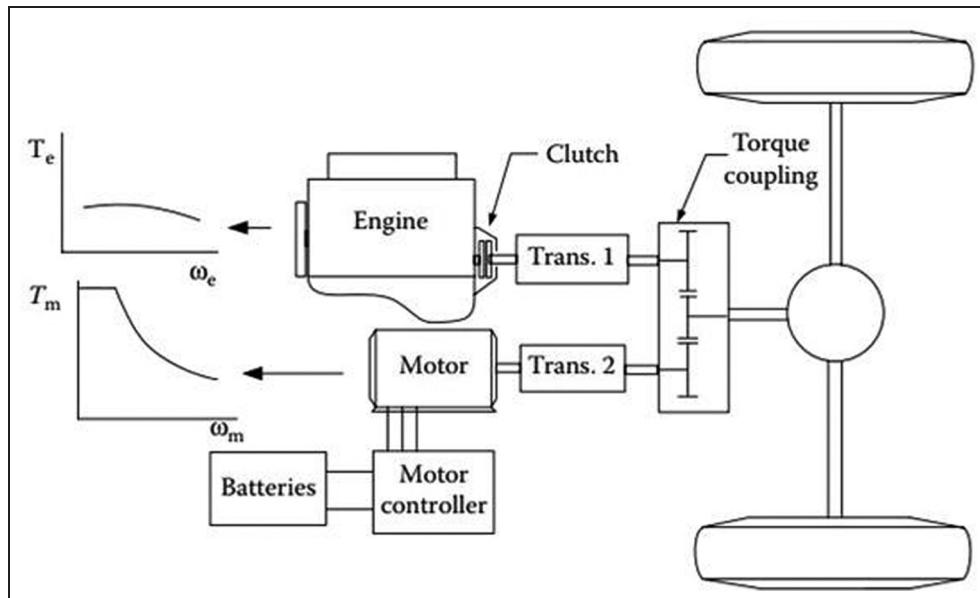
The intelligent control strategy determines the similarity weights of every instance of an unknown driving cycle to four reference driving cycles. These weights are involved in generating the reference torque signal for an ICE using a weighted linear combination of optimized control signals. The structure of the proposed intelligent control strategy is shown in Figure 1.

As can be seen in Figure 1, the intelligent control strategy consists of some online portions and also an offline portion. First, the optimized control strategies corresponding to the four reference driving cycles are



**Figure 1.** Structure of the proposed control strategy.

DP: dynamic programming; MLP: multi-layer perceptron.



**Figure 2.** A schematic diagram for the parallel hybrid powertrain electric hybrid.<sup>2</sup>

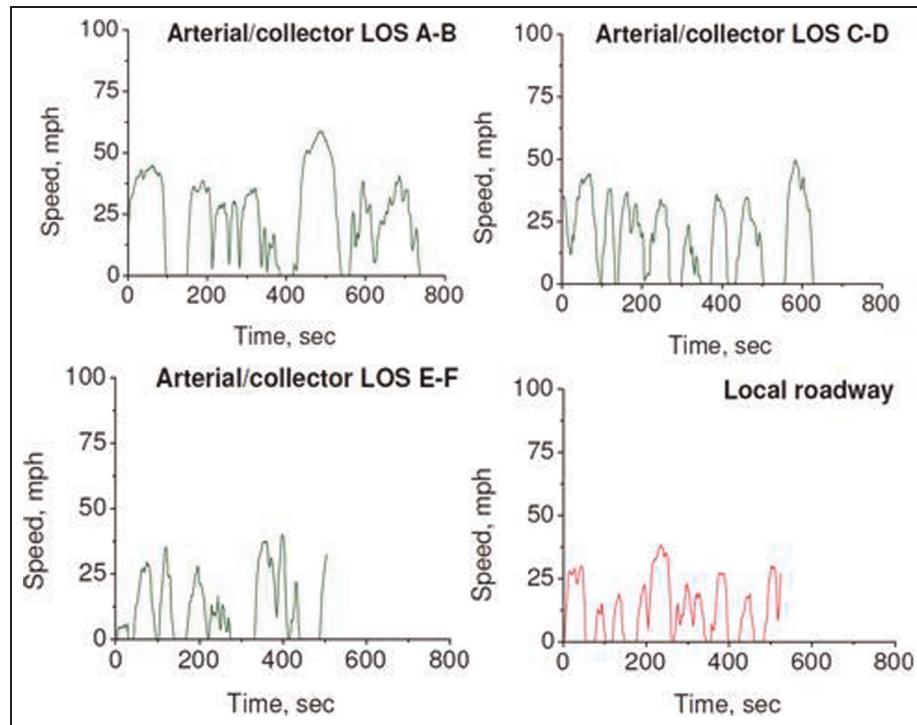
Trans.: transmission.

derived using the DP algorithm. Then, the MLP networks are trained using the results of the four optimized controllers. These two procedures are offline. A library containing the best set of driving features is produced using the feature selection algorithm. Using this library, the fuzzy  $c$ -means clustering is performed in an online manner to generate the similarity weights for every instance of the input driving cycle to the reference driving cycle. The unit which executes this is called the intelligent driving cycle identification agent. Finally, the output signal of the intelligent control strategy is determined by the linear combination of the outputs of

the MLP networks. The fuzzy similarity values are the weights of each output of the MLP networks in the linear combination

$$\text{Output} = \sum_{i=1}^4 u_i \text{MLP}_i \quad (1)$$

The desired value of the ICE torque is the output of the intelligent control strategy. The value for the torque of the secondary power source is determined by subtracting the optimized ICE torque from the value of the torque demand signal.



**Figure 3.** Four reference driving cycles.<sup>22</sup>

LOS: level of service; mph: mile/h.

Furthermore, in this paper, the design process for the intelligent control strategy is explained for implementation on a parallel hybrid powertrain. Each parallel hybrid powertrain has an ICE as the primary source of the required power, a secondary device for power generation (an electric motor in the electric hybrid powertrain and a hydraulic pump-motor in the mechanical hybrid powertrain) and also an energy storage system (a battery pack in the electric hybrid and hydraulic accumulators in the mechanical hybrid). A schematic diagram for the parallel hybrid powertrain is shown in Figure 2.

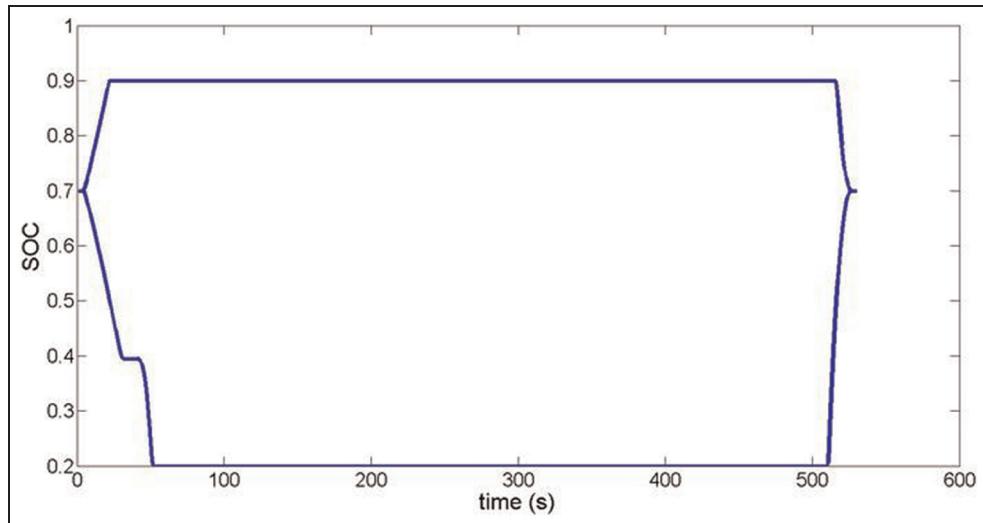
#### Optimized control strategy for the reference driving cycles using dynamic programming

The reference driving cycles considered in the intelligent control strategy (Figure 3) are adopted from those which have been developed by Sierra Research Inc.<sup>22</sup> In their work, nine reference driving cycles were developed for a vehicle, where five of them were for freeway driving conditions and the rest corresponded to urban driving conditions. Since the hybrid powertrains are mostly matched with the urban driving environments (containing many stop-and-go events), here only the driving cycles representing urban driving conditions are considered as the reference driving cycles. They are selected according to the popular driving cycle of an urban vehicle, i.e. not a freeway driving cycle.

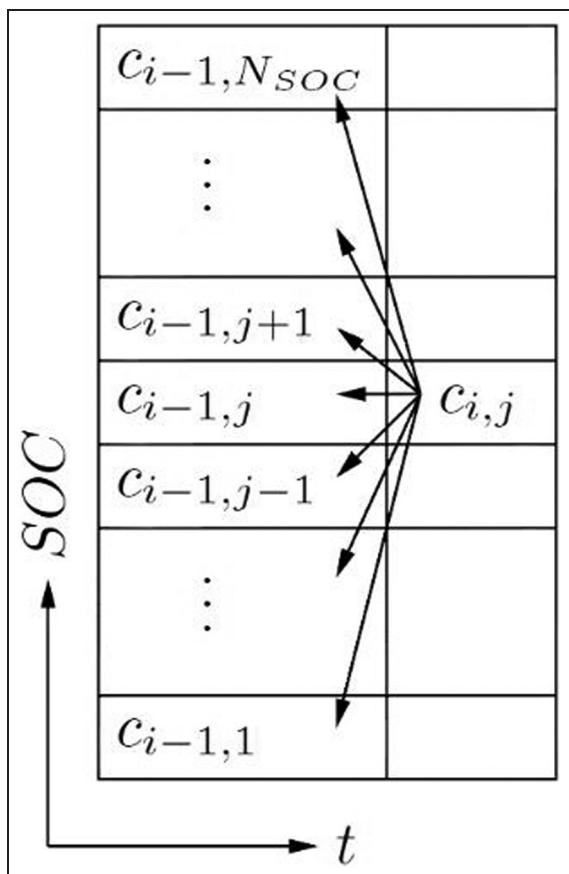
Here, the optimized control strategies for the reference driving cycles are generated using a discrete DP

algorithm<sup>23,24</sup> (see Appendix 2). The input variables for running the algorithm are the driver's torque demand, the vehicle speed and also the number of gears in the automatic gearbox. Since the gear shifting in the automatic gearbox is a function of the vehicle speed (which itself is an input parameter), it is decided to consider the number of gears as an input parameter rather than considering it as a state or control variable. The inputs are logged in every second. In addition, the SOC of the energy storage system is considered as the state variable in the optimal control problem. In order to perform the discrete DP algorithm, the solution space should be quantized. A two-dimensional space with time as the horizontal axis and the SOC as the vertical axis is called the solution space (Figure 5). The time axis is quantized for each second and the quantization step for the SOC axis is 0.001. The smaller the SOC quantization step, the smaller is the step for the torque of the secondary power source. Consequently, a more accurate solution can be achieved by choosing a smaller quantization step for the SOC. However, the value of the SOC step is limited by the specifications of the processing computer. The value of 0.001 is profitable regarding the two above considerations. Using the input variables, the presented DP algorithm proceeds as follows.

First, the upper and lower bands of the SOC values are determined considering the maximum torque and the minimum torque of the secondary power source. Here, the initial value and the final value of the SOC are constrained to be equal (i.e. 0.7 (Figure 4)). Then,



**Figure 4.** Upper band and lower band of the SOC for the collector–arterial E–F driving cycle.  
SOC: state of charge.



**Figure 5.** Forward direction of the DP algorithm in each time step.<sup>24</sup>  
SOC: state of charge.

the optimum state in the  $i$ th time step is determined for each SOC node in the  $(i + 1)$ th time step (Figure 5) in a forward direction. This procedure is continued through the whole time duration of the driving cycle. Finally, considering the equality of the initial SOC and

the final SOC, the optimum state at each time step is specified in a backward direction.

The output of the DP algorithm is the optimal path for the SOC of the energy storage system over the whole driving cycle in order to level the initial SOC and the final SOC. Moreover, the required output torques of the ICE in each time step are determined. In fact, it is the control variable in the present optimal control problem. The constraints on the problem are

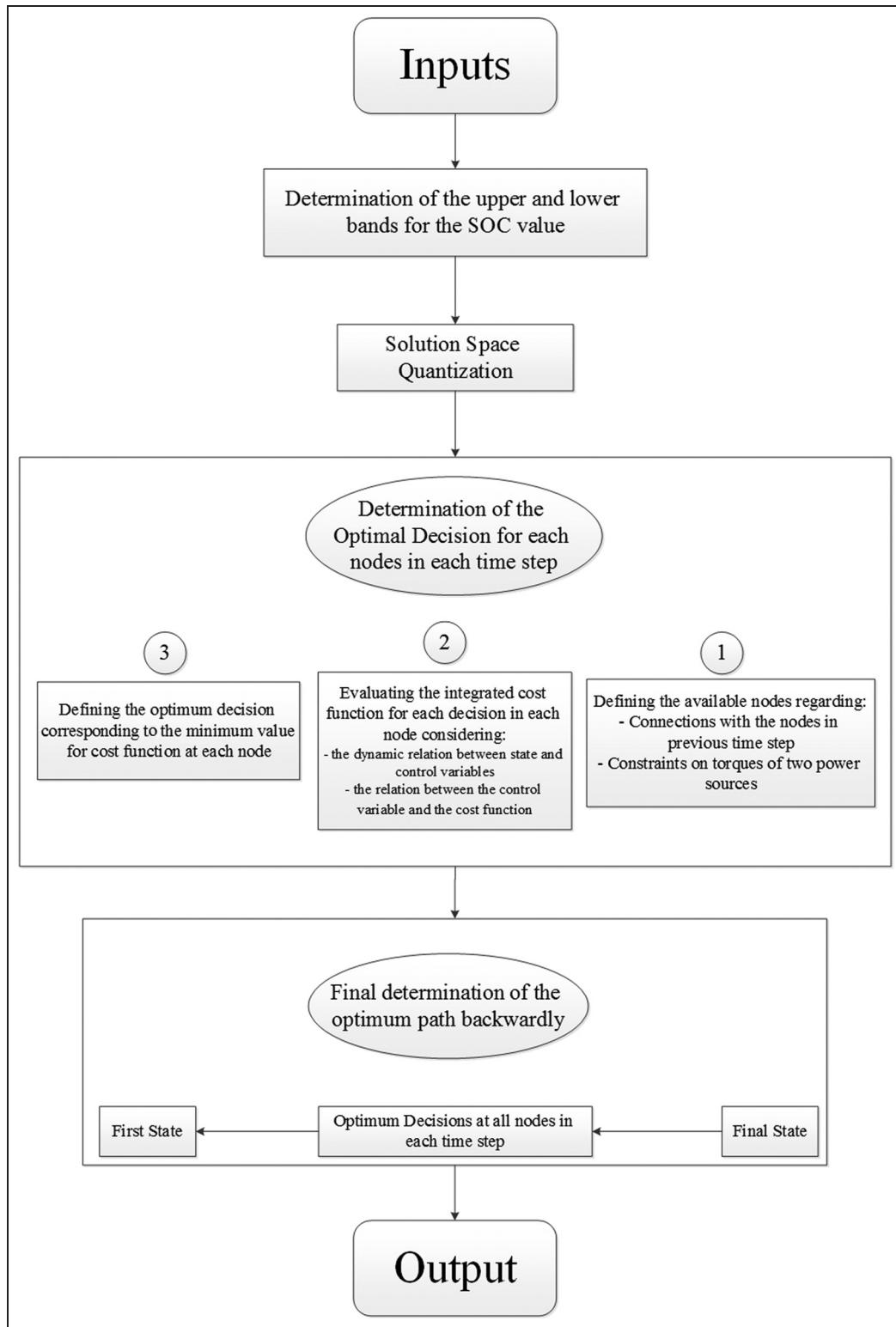
$$\begin{aligned} \text{SOC}_{\min} < \text{SOC} < \text{SOC}_{\max} \\ 0 < T_{\text{eng}} < T_{\text{eng\_max}} \\ T_{\text{sec\_min}} < T_{\text{sec}} < T_{\text{sec\_max}} \end{aligned} \quad (2)$$

The SOC boundaries are defined considering the safety limitations for the energy storage system. Also, the boundaries for the ICE torque and the secondary power source torque are gained from their corresponding performance diagrams. Ideally, the torque distribution has to be chosen to minimize the overall engine fuel consumption over a given driving cycle within the constraints listed above, according to

$$\min \left[ \sum \dot{m}_f(t) \right] \quad (3)$$

where  $\dot{m}_f(t)$  is the fuel rate of the ICE in each time step. As stated above, since the initial value and the final value of the SOC are assumed to be equal, there is no need to consider a special term in the cost function for balancing the SOC during the driving cycle. A flow chart illustrating the implemented DP algorithm is presented in Figure 6.

After extraction of the optimal values for the ICE torque at each time step of the four reference driving cycles, a model should be generated to use these values in an online manner. In other words, a model is required to choose the optimal decisions in further similar driving conditions according to each generated



**Figure 6.** Flow chart illustrating the proposed DP algorithm.  
SOC: state of charge.

optimal control strategy. This is required for implementation of the intelligent control strategy. To obtain this, some MLP networks are generated using the *nntool* toolbox in MATLAB. An MLP network is a strong tool utilized for modelling (especially interpolating) the non-linear systems. According to the very strong non-

linear behaviour of the DP algorithm results in the present problem, every portion of data is modelled by a specific network. This is because we have a set of models which are fitted as much as possible with the optimal control strategies generated using the DP algorithm. For each reference driving cycle, the DP

**Table 1.** Exact number of MLP networks for modelling the DP results in each of the reference driving cycles.

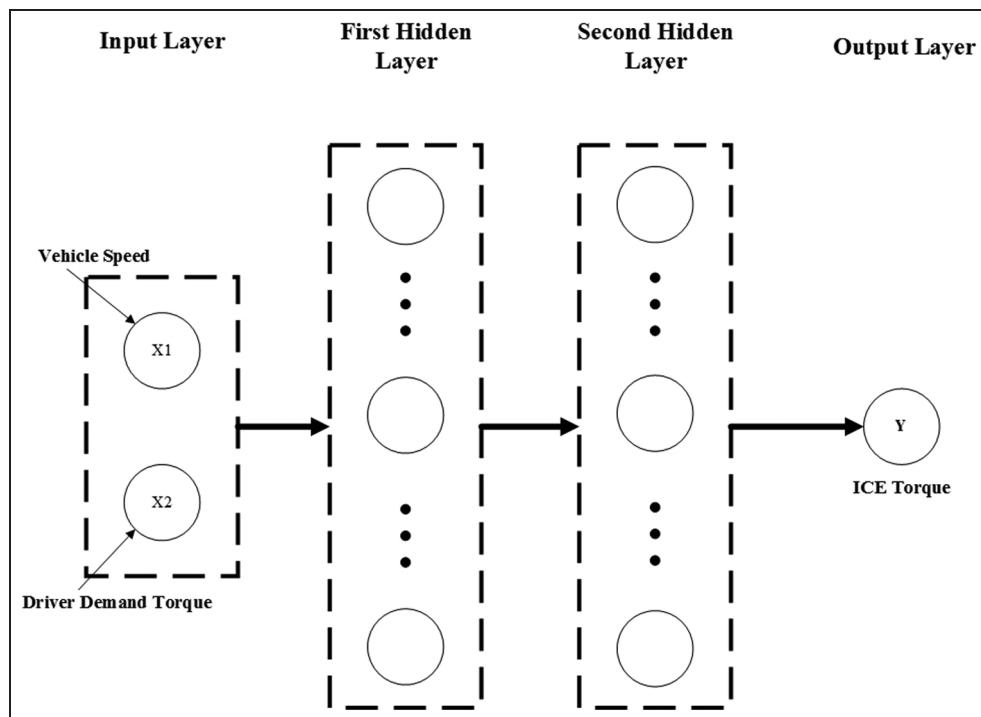
Driving cycle	SOC ranges	Number of SOC groups	Number of MLP networks
Local road	0.2–0.5, 0.5–0.6, 0.6–0.7, 0.7–0.8, 0.8–0.9	5	5
Collector–arterial E–F	0.2–0.4, 0.4–0.5	6	6
Collector–arterial C–D	0.5–0.6, 0.6–0.7	6	6
Collector–arterial A–B	0.7–0.8, 0.8–0.9	6	6

SOC: state of charge; MLP: multi-layer perceptron.

**Table 2.** The specifications of the MLP network for fitting with the optimized control strategy in the local-road driving cycle.

SOC range	Number of neurons in the first hidden layer	Number of neurons in the second hidden layer	Mean squared error of the output values for the validation data
0.2–0.5	10	15	0.005
0.5–0.6	30	60	0.031
0.6–0.7	40	60	0.025
0.7–0.8	30	60	0.011
0.8–0.9	30	40	0.01

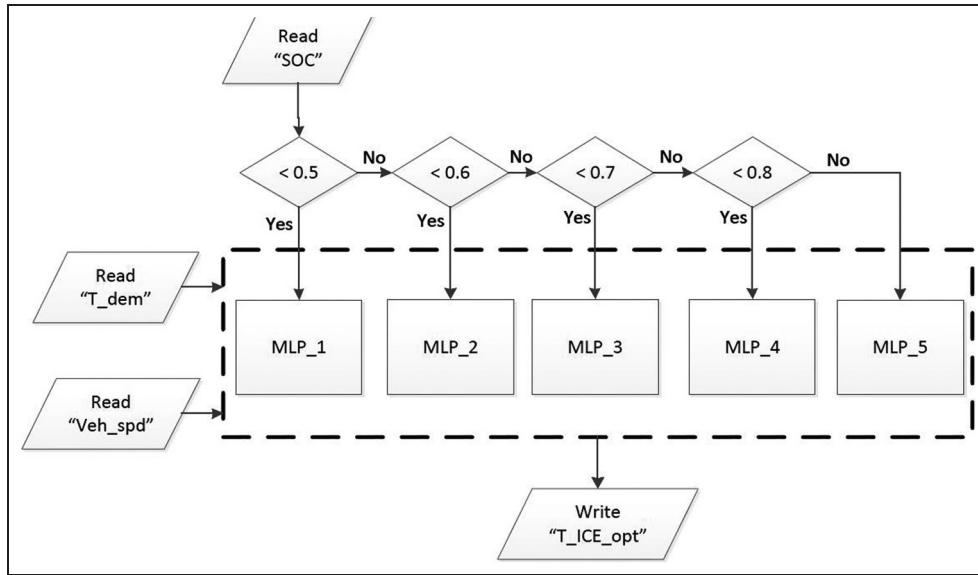
SOC: state of charge.

**Figure 7.** Structure of each MLP network.

ICE: internal-combustion engine.

results divided into five to six groups regarding the SOC values (Table 1). Then, an MLP with two layers and the specific number of neurons is modelled for each of these groups. For example, for the SOC values between 0.6 and 0.7 and also between 0.8 and 0.9, two different MLP networks are generated. The number of neurons is defined by checking all the possible choices to achieve the least value for the mean squared error of

the validation data. This is achieved by running several trial-and-error simulations. The general structure for each of the used MLP networks is depicted in Figure 7. Moreover, the information on the MLP network to fit the optimal control strategy generated for the local-road driving cycle is presented in Table 2. The optimal control strategy used for training this network is produced by the DP algorithm for the hybrid powertrain



**Figure 8.** Procedure for generating the outputs of the MLP networks for each of the DP algorithms.  
SOC: state of charge; MLP: multi-layer perceptron.

of the case study (see the third section). The results gathered from each second of the DP algorithm are rearranged to be in the required format, i.e. the input variables and the output variables. The data set is divided into an appropriate number of SOC groups. Then 80% of the data in each group are randomly chosen for training the MLP network, and the remaining data are used for validation.

The procedure for generating the outputs of the MLP networks is shown in Figure 8. In each of the MLP networks, the driver's torque demand and the vehicle speed are the inputs, and the output is the optimal ICE torque. All the input values and the output values are normalized. The outputs of the MLP networks generated here are used in implementation of the intelligent control strategy.

### Fuzzy driving cycle identification agent

Here, an identification unit is used to generate the similarity weights of an unknown driving cycle to four reference driving cycles. The system is called a fuzzy driving cycle identification agent (FDCIA), since it is performed using a fuzzy clustering method. Every driving cycle can be recognized through some driving features. Ericson<sup>25</sup> has defined 69 driving features for a given driving cycle. Won and Langari<sup>13</sup> and others who have worked on the design of intelligent control strategies have stated that all these features are not useful in driving pattern recognition. A feature selection process should be organized to select the useful features and to eliminate the others. Lists of all the features that are considered as the reference set in this study are presented in Table 3. These are chosen according to previous work.<sup>10–20</sup> Ericson presented the intervals for the speed, the acceleration and the deceleration regarding

the characteristics of a passenger car. In Table 3, the intervals are updated according to the driving characteristics in urban environment with many stop-and-go events. Also in this table, the positive kinetic energy (PKE) is derived from

$$\text{PKE} = \frac{\sum (v_f^2 - v_s^2)}{x} \quad \text{when } \frac{dv}{dt} > 0 \quad (4)$$

where  $v_f$  and  $v_s$  are the final velocity of the vehicle and the initial velocity of the vehicle respectively during the distance  $x$ . Also, the relative positive acceleration (RPA) is determined by

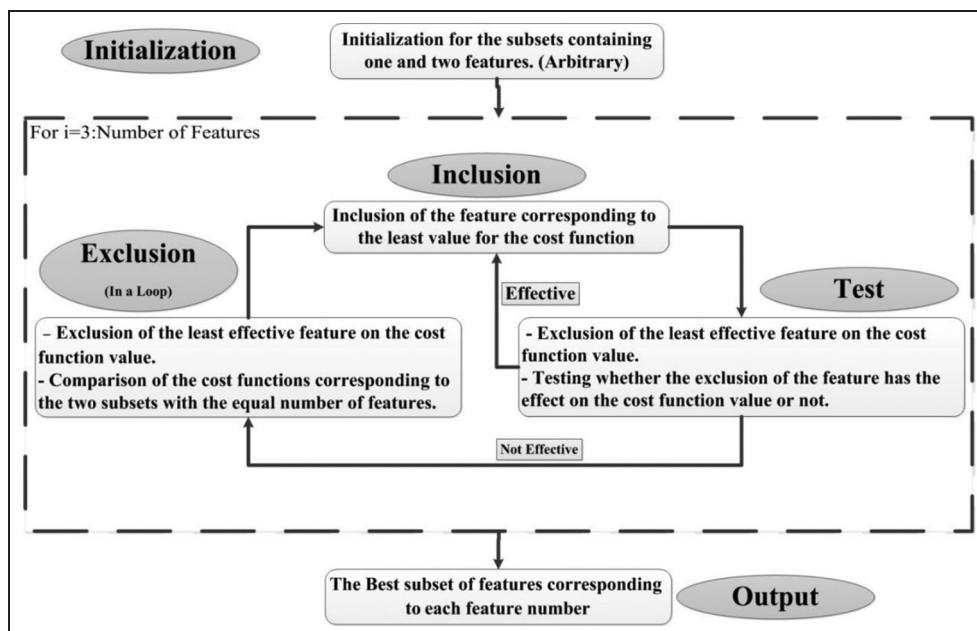
$$\text{RPA} = \frac{1}{x} \int^v a^+ dt \quad (5)$$

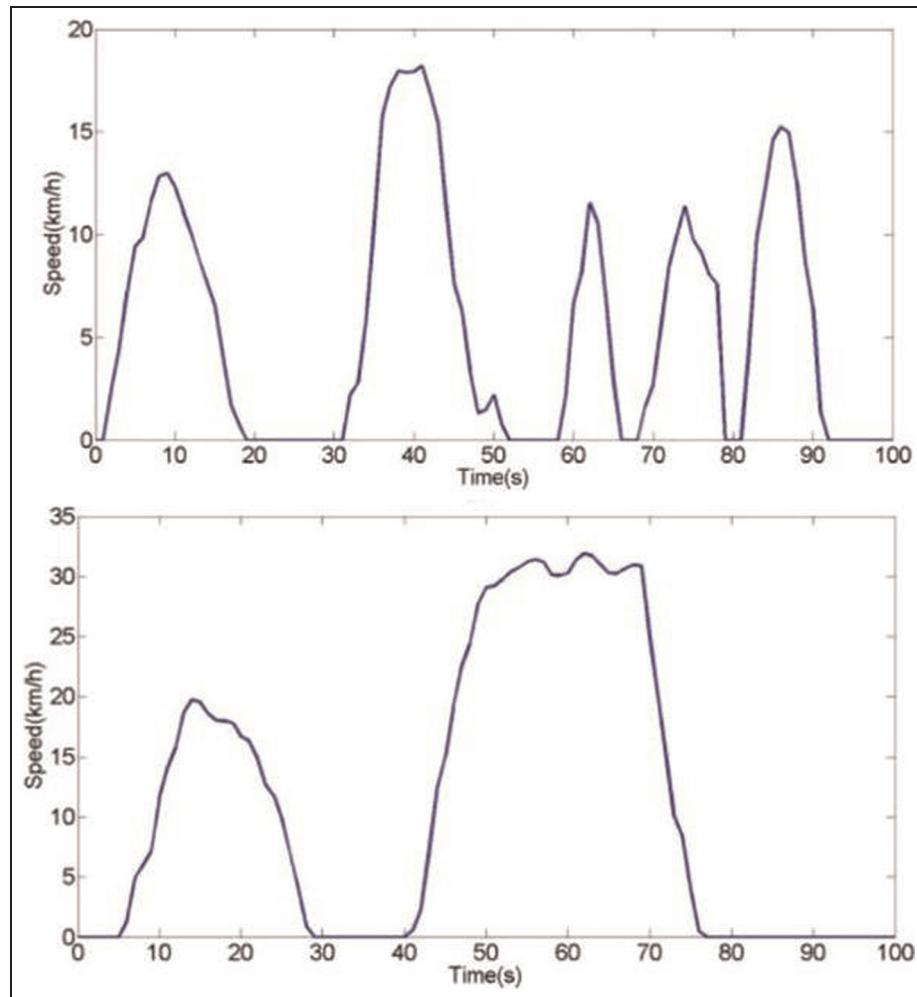
where  $x$  is distance of the cycle,  $v$  is the speed and  $a^+$  is the acceleration of the vehicle.

As the first step in designing the FDCIA, an algorithm known as the floating search method<sup>21</sup> is used to perform the feature selection task. The floating search method is not supervised but can obtain acceptable results.<sup>21</sup> In addition, the method has a much lower numerical complexity in comparison with other methods such as linear vector quantization. The floating search algorithm is presented in Figure 9. In this method the algorithm starts with one arbitrary feature and then continues by adding the best feature which achieves the minimum cost. Moreover, in every step the least sufficient feature is removed. Here, the cost in the feature selection process is determined by computing the clustering cost function (equation (6)), which defines the level of effectiveness for the clustering process. The process is continued until all features are considered. The final result of the floating search method is the best combination for every possible number of features.

**Table 3.** Reference set of driving features.

Number	Feature
1	Average speed (km/h)
2	Maximum speed (km/h)
3	Percentage of time that the speed is between 0 km/h and 10 km/h
4	Percentage of time that the speed is between 10 km/h and 20 km/h
5	Percentage of time that the speed is between 20 km/h and 30 km/h
6	Percentage of time that the speed is between 30 km/h and 40 km/h
7	Percentage of time that the speed is between 40 km/h and 50 km/h
8	Percentage of time that the speed is above 50 km/h
9	Average acceleration ( $m/s^2$ )
10	Maximum acceleration ( $m/s^2$ )
11	Percentage of time that the acceleration is between $0 m/s^2$ and $0.2 m/s^2$
12	Percentage of time that the acceleration is between $0.2 m/s^2$ and $0.4 m/s^2$
13	Percentage of time that the acceleration is between $0.4 m/s^2$ and $0.6 m/s^2$
14	Percentage of time that the acceleration is between $0.6 m/s^2$ and $0.8 m/s^2$
15	Percentage of time that the acceleration is between $0.8 m/s^2$ and $1 m/s^2$
16	Percentage of time that the acceleration is above $1 m/s^2$
17	Relative positive acceleration ( $m/s^2$ )
18	Average deceleration ( $m/s^2$ )
19	Maximum deceleration ( $m/s^2$ )
20	Percentage of time that the deceleration is between $0 m/s^2$ and $0.2 m/s^2$
21	Percentage of time that the deceleration is between $0.2 m/s^2$ and $0.4 m/s^2$
22	Percentage of time that the deceleration is between $0.4 m/s^2$ and $0.6 m/s^2$
23	Percentage of time that the deceleration is between $0.6 m/s^2$ and $0.8 m/s^2$
24	Percentage of time that the deceleration is between $0.8 m/s^2$ and $1 m/s^2$
25	Percentage of time that the deceleration is under $1 m/s^2$
26	Positive kinetic energy ( $m/s^2$ )
27	Number of stops per kilometre
28	Number of stops
29	Percentage of time that the product of the speed and the acceleration is between $0 m^2/s^3$ and $5 m^2/s^3$
30	Percentage of time that the product of the speed and the acceleration is between $5 m^2/s^3$ and $10 m^2/s^3$
31	Percentage of time that the product of the speed and the acceleration is above $10 m^2/s^3$
32	Percentage of time that the product of the speed and the deceleration is between $0 m^2/s^3$ and $-5 m^2/s^3$
33	Percentage of time that the product of the speed and the deceleration is between $-5 m^2/s^3$ and $-10 m^2/s^3$
34	Percentage of time that the product of the speed and the deceleration is below $-10 m^2/s^3$

**Figure 9.** The algorithm of the floating search method as a feature selection procedure.



**Figure 10.** Two of the sample driving cycles used for performing the floating search algorithm.

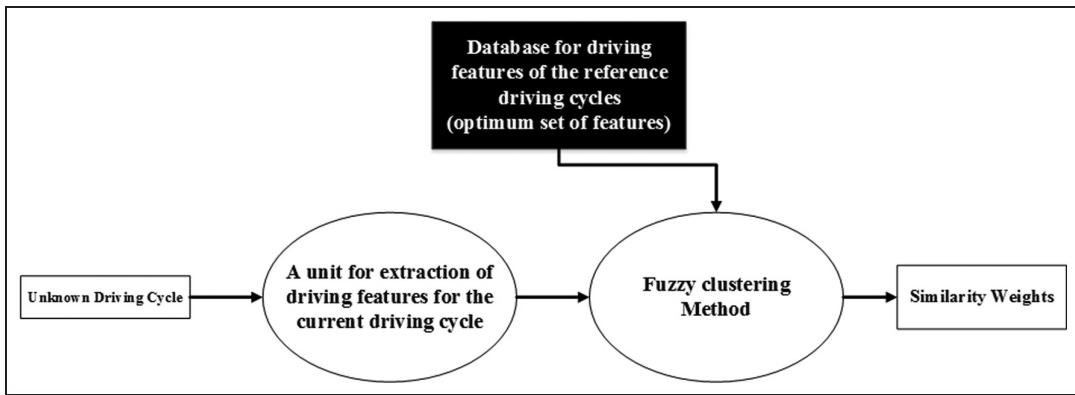
Here, the proposed floating search algorithm is run offline to determine the appropriate driving features. This means that the implementation of the algorithm is performed separately from the hybrid powertrain simulations. It is necessary to have some sample driving cycles as the data set to handle the algorithm. Several sample driving cycles extracted from the simulated driving cycles (those used in simulations of the hybrid powertrain; see Table 6 later) with the length of 100 s each are used to provide the required data set. Two of these sample driving cycles are depicted in Figure 10. As can be seen, the initial speed of each driving cycle is zero and they end in the stationary situation. It should be noted that the optimum set of driving features is determined so as to reach the optimum clustering performance using the floating search method. This is implemented on the data set, which contains all the sample driving cycles. Finally, the optimum set of features is used in the implementation of the FDCIA.

As mentioned before, here a fuzzy clustering method is utilized to represent the similarity of every unknown driving cycle to each reference driving cycle. The fuzzy  $c$ -means method<sup>21</sup> is the implemented clustering method. In a conventional fuzzy  $c$ -means method, there

are several reference clusters whose centre positions are variable. In each algorithm iteration, the positions of the centres of all clusters are updated. In a fuzzy clustering method, the membership degrees of each experimental datum is defined with regard to the distance between the data position and the position of each cluster centre. By updating the position of each cluster centre, the membership degrees are also updated. The algorithm is stopped if the clustering cost function decreases to the desired value.<sup>21</sup>

In this paper, the implemented fuzzy  $c$ -means method is a little different from the conventional method. Here, the focus is on the membership degrees or the similarity weights rather than on the position of each cluster centre. The features of each reference driving cycle are used to define the centre for each of four clusters (corresponding to the four reference driving cycles). In fact, the following two applications are considered for the fuzzy  $c$ -means algorithm based on the intended functionality of the paper.

In the implementation of the floating search method, the fuzzy  $c$ -means algorithm is used to determine the cost function. The positions of the cluster centres are fixed and only the features chosen to define each of the



**Figure 11.** A schematic diagram for the FDCIA.

cluster centres are changed. For adding the most appropriate feature and eliminating the least effective feature, the value of the clustering cost function is computed in each step. The cost function is specified by

$$J_q = \sum_{i=1}^N \sum_{j=1}^m u_{ij}^q d(x_i, \theta_j) \quad (6)$$

where  $u_{ij}$  is the fuzzy weight for the similarity of  $x_i$  driving cycles to the  $\theta_j$  cluster and  $d(x_i, \theta_j)$  is the dissimilarity between them. The parameter  $q$  is the clustering factor and is a real number greater than one. Here  $q = 2$ . Also the dissimilarity  $d(x_i, \theta_j)$  is computed from

$$d(x_i, \theta_j) = (x_i - \theta_j)^T (x_i - \theta_j) \quad (7)$$

The values of the fuzzy weights for each of the clusters are computed according to the fuzzy  $c$ -means algorithm. These weights are updated in every step of the feature selection process by<sup>21</sup>

$$u_j = \frac{1}{\sum_{k=1}^m [d(x, \theta_j) / d(x, \theta_k)]^{1/(q-1)}} \quad (8)$$

in which  $m$  is the number of clusters (i.e. four) and  $j$  is the reference driving cycle (i.e. the cluster).

In the online implementation of the FDCIA, a unit is required to produce the similarity weights for the input driving cycle. The unit operates according to the fuzzy  $c$ -means algorithm. In fact, the weights are computed in a single iteration by use of equation (8), where both the position of the reference driving cycles and the position of the input driving cycle are fixed. This is because the fixed set of driving features (i.e. the optimum set defined through the floating search algorithm), which constitutes the elements of  $x$  and  $\theta$  in equation (8), is considered.

Finally, a library of optimum features for the reference driving cycles together with an online system for generation of the similarity weights of the driving cycle is implemented in MATLAB/Simulink (Figure 11). The model for the FDCIA is used in implementing the intelligent control strategy. It should be noted that this model generates the similarity weights online and is

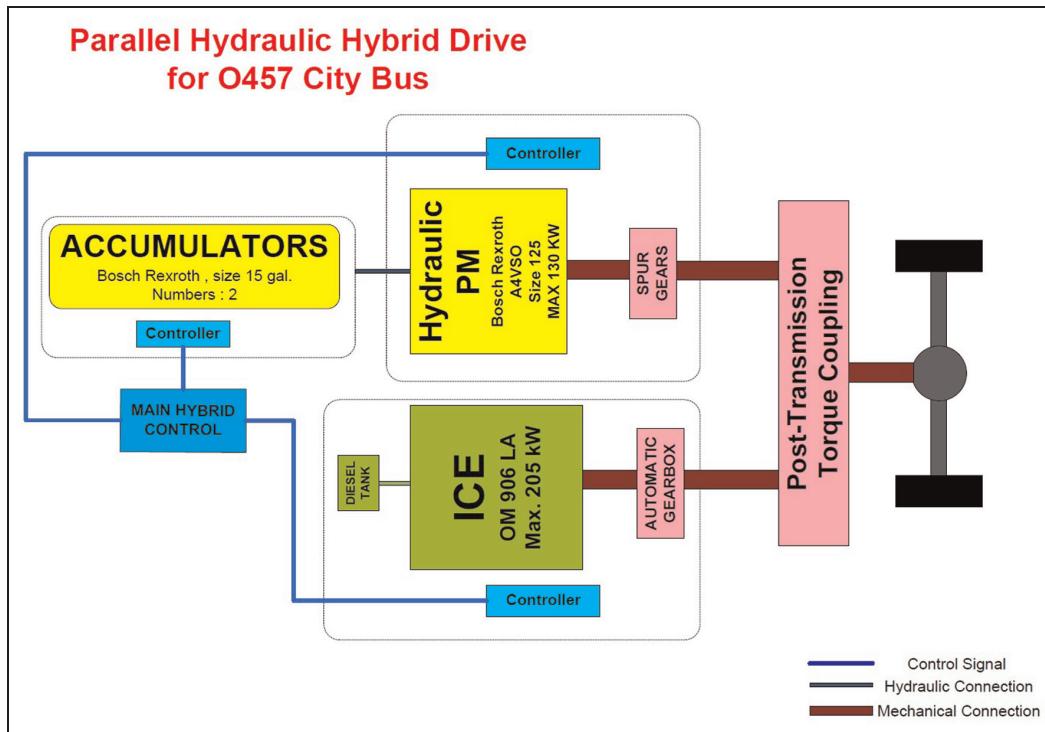
updated every second regarding the previous speed profile up to the current position.

### Simulation study

Here, a parallel hydraulic hybrid powertrain (Figure 12) for a city bus is used to assess the performance of the proposed intelligent control strategy. Several efforts have been made to develop an optimal control strategy for hydraulic hybrid powertrains in the literature.<sup>9,26–30</sup> The powertrain contains an ICE as the primary power source and a hydraulic pump–motor as the secondary power source. In addition, two hydraulic accumulators are used as the energy storage system. The specifications of the parallel hydraulic powertrain are presented in Table 4. A feedforward model for the powertrain was generated by the present authors in MATLAB/Simulink<sup>31</sup> and was used for running the simulations. In this model, the driver block creates the appropriate command signal according to the difference between the driving cycle speed and the actual vehicle speed. This signal is sent to the hybrid central control block which computes the torque command signals of each power source so as to satisfy the driver's demand. The performance of this block is based on the designed control strategy. The output signals are sent to the ICE and the hydraulic pump–motor blocks which generate the torque demands. Finally, both output torques are coupled in the torque coupling and the final signal is sent to the vehicle dynamics block. The fuel consumption of the bus is computed in the ICE block considering the brake specific fuel consumption map of the engine.

In the following simulation study, a rule-based control strategy is used as the benchmark. The driver's torque demand, the vehicle speed and the SOCs of the accumulators are the input variables in this control strategy. The control strategy has four operating modes.<sup>31</sup>

1. *First mode.* The bus is in acceleration mode, the driver's torque demand is below  $T_0$ , the vehicle speed is below  $V_0$  and the SOC is above the



**Figure 12.** The schematic diagram of the parallel hydraulic hybrid powertrain.<sup>31</sup>  
PM: pump–motor; MAX, Max.: maximum; ICE: internal-combustion engine.

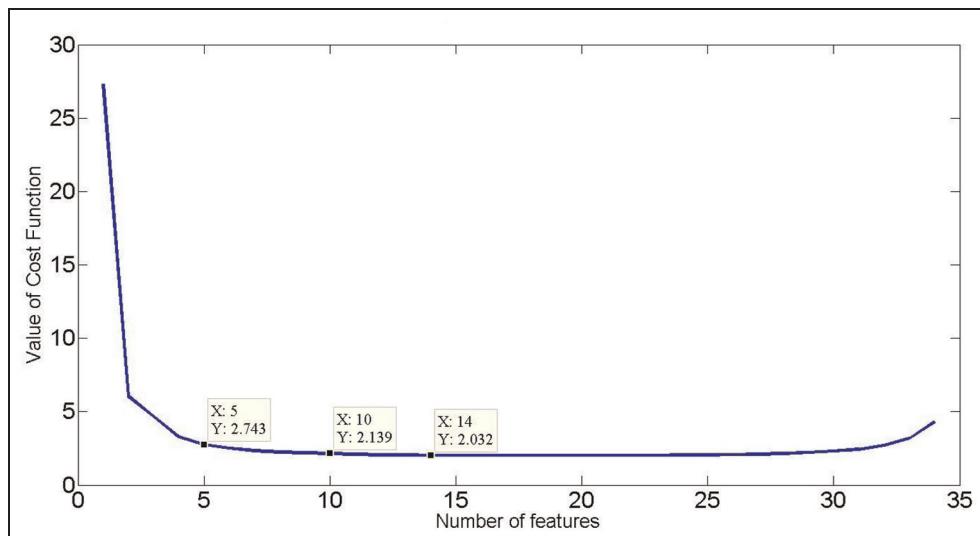
**Table 4.** Specifications of the parallel hydraulic hybrid bus.<sup>31</sup>

Gross mass	9820 kg
Mass including the passengers	14,720 kg
Internal-combustion engine	OM906LA. Maximum power, 205 kW. Maximum torque, 1100 N m.
Hydraulic pump–motor	A4VSO. Maximum power, 131 kW. Maximum torque, 696 N m. Maximum displacement, 125 cm <sup>3</sup> /rev. Automatic transmission with four speeds
Gearbox	Two bladder-type accumulators.
Accumulators	Maximum operating pressure, 345 bar. Nominal volume, 50 l (each).

minimum allowable value. The torque demand from the driver is provided by the hydraulic pump–motor alone and the ICE is off.

2. *Second mode.* The bus is in acceleration mode, the driver's torque demand is above  $T_0$ , the vehicle speed is above  $V_0$  and the SOC is above the minimum allowable value. In this mode the ICE turns on and operates as the main power source of the hybrid powertrain. Also, the ICE is helped by the hydraulic pump–motor in the situation when the driver's torque demand is above the maximum torque generated by the engine. Accumulator charging is not considered in this mode.

3. *Third mode.* The bus is in acceleration mode and the SOC of the accumulator is not above the minimum allowable value. In this mode the ICE is the only power source in the hybrid powertrain. In order to keep the ICE in its most efficient region, the torque generated by the ICE could be more than the driver's torque demand for most situations. In these situations, an additional engine torque is stored in the accumulators by the hydraulic pump–motor.
4. *Fourth mode.* The bus is in braking mode. In this mode the hydraulic pump–motor is working as a pump and generates the main braking torque for



**Figure 13.** The optimum set containing 10 driving features.

**Table 5.** Optimum set of driving features for the proposed hydraulic hybrid powertrain.

Number	Feature
1	Percentage of time that the speed is between 0 km/h and 10 km/h
2	Percentage of time that the speed is between 10 km/h and 20 km/h
3	Maximum acceleration ( $\text{m/s}^2$ )
4	Percentage of time that the acceleration is between $0.2 \text{ m/s}^2$ and $0.4 \text{ m/s}^2$
5	Percentage of time that the acceleration is between $0.6 \text{ m/s}^2$ and $0.8 \text{ m/s}^2$
6	Percentage of time that the deceleration is between $-0.2 \text{ m/s}^2$ and $0.4 \text{ m/s}^2$
7	Percentage of time that the deceleration is between $-0.6 \text{ m/s}^2$ and $0.8 \text{ m/s}^2$
8	Relative positive acceleration ( $\text{m/s}^2$ )
9	Percentage of time that the product of the speed and the acceleration is between $0 \text{ m}^2/\text{s}^3$ and $5 \text{ m}^2/\text{s}^3$
10	Positive kinetic energy ( $\text{m/s}^2$ )

the bus (regenerative braking). The mechanical braking system of the bus is used if the torque generated by the hydraulic pump–motor is not sufficient.

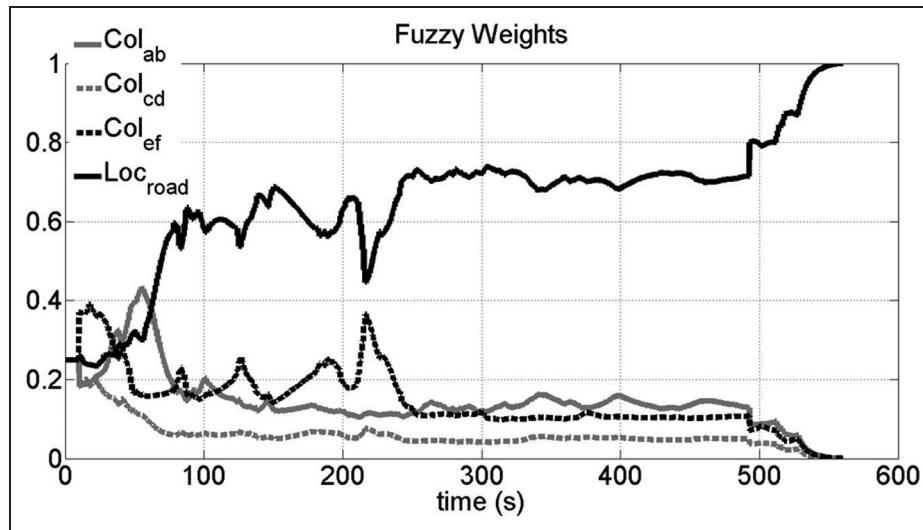
In the proposed rule-based control strategy,  $T_0$  and  $V_0$  are defined according to the specifications of the two power sources and to fit the driving requirements of the Tehran bus driving cycle.<sup>32</sup>

First, a simulation was run to see the performance of the proposed FDCIA. The optimum set of features for the proposed hydraulic hybrid powertrain is determined by the floating search method (Figure 13). As can be deduced from Figure 13, the optimum set contains 10 driving features. Since the value of the clustering cost function does not have any significant drop after using 10 driving features, this set is defined as the optimum set of driving features. The features are presented in Table 5.

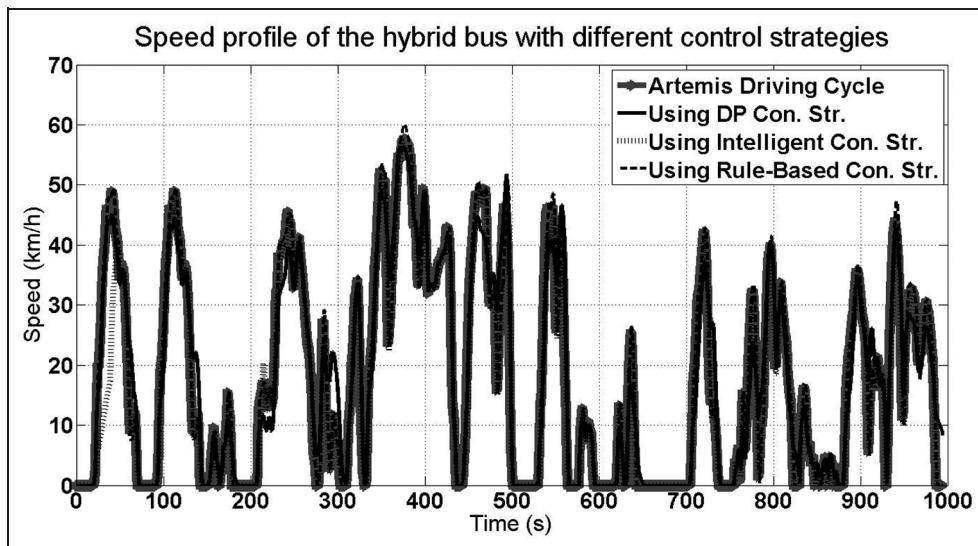
For the purpose of evaluation, one of the reference driving cycles, e.g. the local road, was considered as the input to the FDCIA. The outputs, i.e. the instantaneous similarity weights, are shown in Figure 14. Since the similarity weights are computed every second

regarding the past profile of the vehicle speed, they are called the instantaneous weights. As can be seen, the weight corresponds to when the exact driving cycle reaches 0.6 just after 100 s, while the other weights are below 0.3. Moreover, the exact driving cycle weight reaches 0.7 after 250 s, while the others remain below 0.2. The values of the driving features for the input and exact driving cycles are equal at the end of the time period of the driving cycle which makes the corresponding similarity weight equal to one.

Here, several simulations are organized to evaluate the capability of the intelligent control strategy to reduce the fuel consumption of the parallel hydraulic hybrid powertrain. The simulations are performed using the ode45 solver in MATLAB/Simulink software with a time step equal to 0.01 s. The results for the comparison of the fuel consumptions of the hybrid powertrain utilized with a rule-based control strategy, the proposed intelligent control strategy, the intelligent control strategy utilized with a non-fuzzy identification agent and also an optimum control strategy are presented in Table 6. The results are for simulations of the hybrid bus in seven distinct driving cycles. The rule-based control strategy<sup>31</sup> is developed by generating



**Figure 14.** The similarity weights for four reference driving cycles when one of these is the input to the FDCIA.  $\text{Col}_{ab}$ : collector A–B;  $\text{Col}_{cd}$ : collector C–D;  $\text{Col}_{ef}$ : collector E–F;  $\text{Loc}_{\text{road}}$ : local road.



**Figure 15.** Comparison of the hybrid bus speed profiles utilized with three proposed control strategies in the Artemis driving cycle. DP: dynamic programming; Con. Str.: control strategy.

rules in accordance with some expert knowledge on the parallel hydraulic hybrid powertrain. Also, the results for the simulations of the hybrid bus with the control strategy extracted from the DP algorithm are added to show the difference between the intelligent control strategy and the optimum control strategy. Moreover, in the non-fuzzy intelligent control strategy, the similarity weights are set to zero for all the standard driving cycles except for one of these standard driving cycles which is set to 1. This driving cycle is the most similar to the input driving cycle. This is the idea which was developed by others in previous investigations.<sup>10–20</sup> In a non-fuzzy intelligent control strategy, each period of the input driving cycle is considered to be exactly similar to that of the standard driving cycle. Consequently, the corresponding optimized control strategy is used for energy management of the hybrid powertrain. For

implementing the intelligent control strategy with non-fuzzy similarity weights, most values of the weights are set to one while the others are set to zero. To eliminate the effect of the difference between the initial SOC and the final SOC on the consumed fuel value, the results are gathered together for simulations of the model in five sequential series of each driving cycle. Figure 15 shows the driving cycle tracking performance for the hybrid bus with three proposed control strategies in the Artemis driving cycle. It should be noted that there is a speed tracking error in the model simulations, since the model is a feedforward type.

As can be seen in Table 6, while using non-fuzzy driving cycle identification in the intelligent control strategy leads to a similar fuel consumption to the case using fuzzy identification in some driving cycles, there is a huge difference within the results in some other

**Table 6.** Comparison of the fuel consumptions of the hybrid powertrain when different control strategies are incorporated.

Driving cycle	Control strategy	Fuel consumption (l/100 km)	Mean squared error of speed	Reduction (%) with respect to the hybrid with the rule-based strategy
Nuremberg cycle	Rule based	51.31	0.0017	—
	Dynamic programming	46.37	0.0062	9.63
	Intelligent utilized with fuzzy identification	47.57	0.002	7.29
	Intelligent utilized with non-fuzzy identification	57.18	0.0021	-11.44
Tehran cycle	Rule based	46.83	0.0007	—
	Dynamic programming	44.41	0.0030	5.17
	Intelligent utilized with fuzzy identification	45.33	0.0007	3.21
	Intelligent utilized with non-fuzzy identification	46.71	0.0007	0.26
New York bus cycle	Rule based	96.88	0.003	—
	Dynamic programming	82.66	0.0051	14.68
	Intelligent utilized with fuzzy identification	89.0	0.0044	8.13
	Intelligent utilized with non-fuzzy identification	89.67	0.0045	7.44
UK bus cycle	Rule based	55.4	0.0015	—
	Dynamic programming	45.7	0.0044	17.5
	Intelligent utilized with fuzzy identification	51.4	0.0017	7.22
	Intelligent utilized with non-fuzzy identification	54.3	0.0017	1.99
Artemis cycle	Rule based	56.67	0.0032	—
	Dynamic programming	46.39	0.0059	18.14
	Intelligent utilized with fuzzy identification	53.75	0.0078	5.15
	Intelligent utilized with non-fuzzy identification	53.99	0.0075	4.73
FTP cycle	Rule based	39.17	0.0017	—
	Dynamic programming	33.34	0.0062	14.9
	Intelligent utilized with fuzzy identification	38.55	0.0023	1.6
	Intelligent utilized with non-fuzzy identification	41.1	0.0024	-4.93
Japan midtown cycle	Rule based	42.54	0.0012	—
	Dynamic programming	39.21	0.0053	7.83
	Intelligent utilized with fuzzy identification	40.21	0.0012	5.48
	Intelligent utilized with non-fuzzy identification	40.69	0.0012	4.35

FTP: Federal Test Procedure.

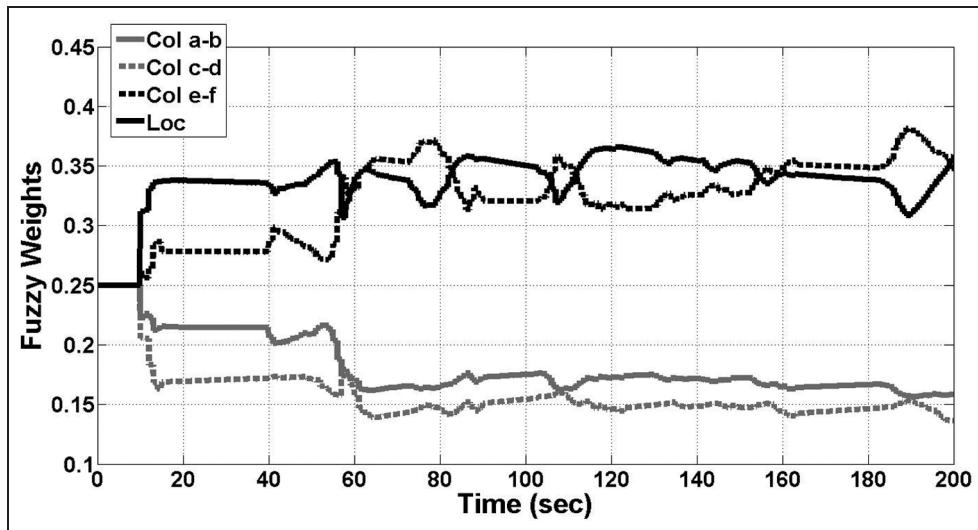
driving cycles. The similarity weights for a portion of the Nuremberg driving cycle and a portion of the New York bus driving cycle are presented in Figure 16 and Figure 17 respectively. As can be observed, there are two reference driving cycles with close similarity values in the Nuremberg driving cycle, while there is just a major similarity weight for one of the reference driving cycles in the New York bus driving cycle. By using non-fuzzy identification, one of the major patterns in the Nuremberg driving cycle is discarded, and consequently an energy management which is far from optimum is achieved. This limitation does not exist in the New York bus driving cycle. Therefore, the fuel consumptions are the same using the two above-mentioned control strategies.

In addition, it is evident that the optimal control strategy leads to lower fuel consumption than does the

intelligent control strategy with fuzzy identification agent in all driving cycles. As stated earlier in the introduction to this paper, there are several drawbacks to using the optimal control strategy generated by the DP algorithm, which makes use of the intelligent control strategy more reliable.

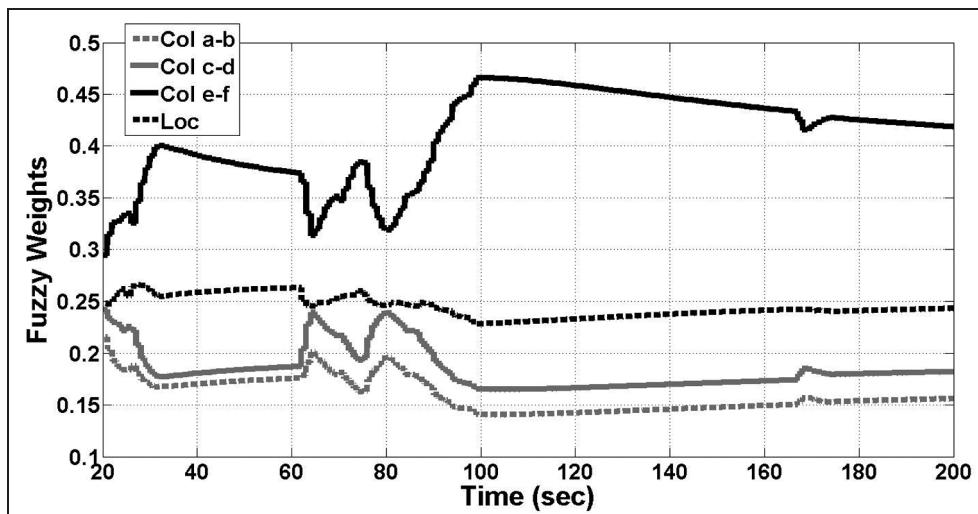
1. The control strategy is optimized for only one specific driving cycle and should be computed again for another driving cycle.
2. The control strategy is offline, and full information of the driving cycle is required prior to performing the DP algorithm and generating the optimal control strategy.

Like the other intelligent control strategies in the literature,<sup>12–16</sup> the proposed intelligent control strategy



**Figure 16.** Similarity weights in the Nuremberg driving cycle.

Col a-b: collector A–B; Col c-d: collector C–D; Col e-f: collector E–F; Loc: local road.



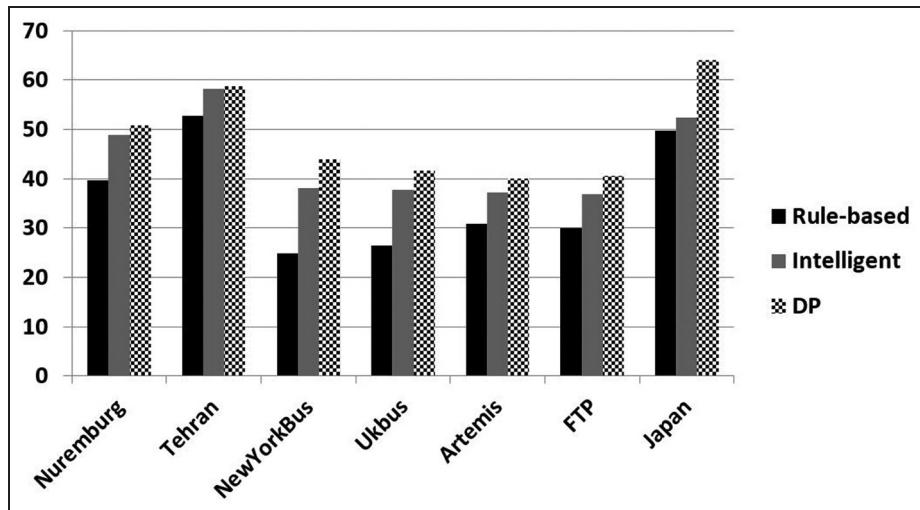
**Figure 17.** Similarity weights in the New York bus driving cycle.

Col a-b: collector A–B; Col c-d: collector C–D; Col e-f: collector E–F; Loc: local road.

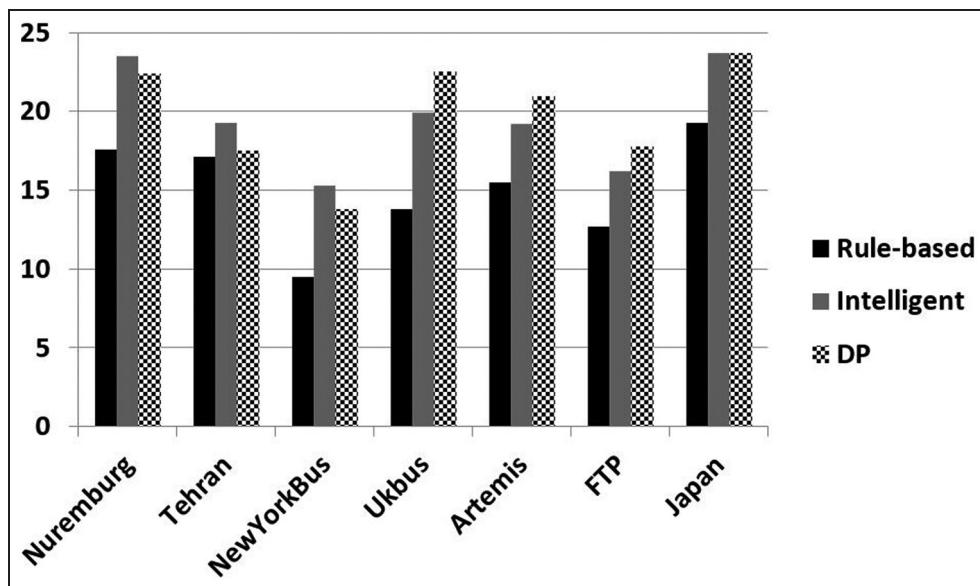
with the fuzzy identification agent is used to overcome the above drawbacks. In fact, the intelligent control strategy is near the optimal strategy for several driving cycles, and it can be implemented online.

In addition, the results in Table 6 state that the fuel consumption of the hybrid bus with the intelligent control strategy utilized with the fuzzy identification agent is lower than that for the hybrid bus with the rule-based control strategy. The differences between the fuel consumptions corresponding to these control strategies are different from one driving cycle to another. The rule-based control strategy acts uniquely in all driving cycles with different driving conditions. In fact, there is no adaptation in the rule-based control strategy to the

changes in the driving conditions. Rather, the intelligent control strategy can adapt itself to the driving conditions of each driving cycle. This is achieved by using the FDCIA. This awareness about the driving conditions is shown by the increase in the percentage of the braking energy regeneration. The values for these percentages corresponding to the simulations of the hybrid bus with three different control strategies in all the driving cycles are presented in Figure 18. As can be seen, the regeneration of the braking energy for the hybrid bus with the intelligent control strategy is higher than that for the hybrid bus with the rule-based control strategy. The increase in the regeneration of braking energy causes an increase in the usage of the stored energy in the



**Figure 18.** Percentages of braking energy regeneration.  
DP: dynamic programming; FTP: Federal Test Procedure.



**Figure 19.** Percentages of the required power for the hydraulic pump–motor in the vehicle.  
DP: dynamic programming; FTP: Federal Test Procedure.

accumulators. More use of the accumulators' stored energy results in less fuel consumed in the ICE. The percentages of the required power for the hydraulic pump–motor in the vehicle for each driving cycle are shown in Figure 19.

Finally, there are two special considerations in the simulation study of the proposed intelligent control strategy.

1. In this paper, the design process of the intelligent control strategy is proposed generally for hybrid powertrains and can also be implemented on electric hybrid vehicles. In fact, the implementation procedure is the same for an electric hybrid powertrain. The major difference is the dynamic relation between the state and the control variables in the

DP algorithm (Figure 5) which should be updated for an electric motor–generator and a battery pack. On the basis of the results achieved for a hydraulic hybrid powertrain in the present manuscript, it is expected that similar improvements would be achieved for the electric hybrid powertrains. The awareness about the driving environments would be helpful in energy management for all hybrid powertrains using either an electric power generation source or a hydraulic source.

2. As can be seen, the proposed intelligent control strategy is implemented as a higher-level controller to produce the control signals for the main components of the hybrid powertrain, i.e. the ICE, the second power generation source and also the mechanical brake system. The relation between the

higher-level controller and each of the lower-level controllers depends on the quality of these command signals, including the frequencies of the signals, the level of noise in the signals and also their upper and lower bands. The compatibility of the two higher-level controllers should be confirmed.

- (a) It can be expected that the upper-level controller is sufficiently rapid to transmit the driver's torque demand through the power-generating sources. This is due to the online nature of the proposed intelligent control strategy. Moreover, a powerful processor may be used if a more rapid processing rate is required.
- (b) The upper and lower bands for the generated torques of the two power generation components are implemented in the present simulations. This helps the lower-level controllers to be operated in conjunction with the proposed intelligent control strategy.

## Conclusion

In this study, the design procedure of an intelligent control strategy for hybrid powertrains is presented. The control strategy is adapted with the current driving conditions by generating the similarity weights for the unknown driving cycle to a set of reference driving cycles in an online process. This is achieved by using an FDCIA. The similarity weights are applied to a linear combination of the predefined optimized control strategies for the reference driving cycles in order to develop the control decision in every instance of the unknown driving cycle. Here, DP, which is an offline optimization method, is used to develop the optimized control strategies. Moreover, a hydraulic hybrid powertrain is simulated for seven different driving cycles to evaluate the performance of the control strategy. The simulation results show that the driving patterns, which are discarded using a non-fuzzy driving cycle identification agent, are taken into account by implementing the fuzzy identification agent during the generation of the control decisions. This leads to a higher fuel consumption reduction using an intelligent control strategy utilized with the FDCIA. Furthermore, the proposed intelligent control strategy reaches a performance close to those of the optimized control strategies in some driving cycles. It should be noted that, while the optimized control strategy is generated offline, the control decisions in the proposed intelligent control strategy are made online. For future study, the performance of the intelligent control strategy may be improved by considering more reference driving cycles representing different driving patterns.

## Declaration of conflict of interest

The authors declare that there is no conflict of interest.

## Funding

This work was supported by the Vehicle, Fuel and Environment Research Institute of the University of Tehran.

## References

1. Salmasi FR. Control strategies for hybrid electric vehicles: evolution, classification, comparison and future trend. *IEEE Trans Veh Technol* 2007; 56(5): 2393–2404.
2. Ehsani M, Gao Y and Emadi A. *Modern electric, hybrid electric and fuel cell vehicles: fundamentals, theory and design*. Boca Raton, Florida: CRC Press, 2010.
3. Farrall SD and Jones RP. Energy management in an automotive electric/heat engine hybrid powertrain using fuzzy decision making. In: *IEEE international symposium on intelligent control*, Chicago, Illinois, USA, 25–27 August 1993, pp. 463–468. New York: IEEE.
4. Cerruto E, Consoli A, Raciti A and Testa A. Energy flow management in hybrid vehicles by fuzzy logic controller. In: *7th Mediterranean electro-technical conference*, Antalya, Turkey, 12–14 April 1994, Vol 3, pp. 1314–1317. New York: IEEE.
5. Rajagopalan A, Washington G, Rizzoni G and Guezenneec Y. Development of fuzzy logic and neural network control and advanced emissions modeling for parallel hybrid vehicles. Report NREL/SR-540-32919, National Renewable Energy Laboratory, Golden, Colorado, USA, 2001.
6. Schouten NJ, Salman MA and Kheir NA. Fuzzy logic control for parallel hybrid vehicles. *IEEE Trans Control Systems Technol* 2002; 10(3): 460–468.
7. Matheson PL and Stecki JS. Modeling and simulation of a fuzzy logic controller for a hydraulic-hybrid powertrain for use in heavy commercial vehicles. SAE paper 2003-01-3275, 2003.
8. Lin C-C, Peng H, Grizzle JW and Kang J-M. Power management strategy for a parallel hybrid electric truck. *Trans Control Systems Technol* 2003; 11(6): 839–848.
9. Wu B, Lin C-C, Filipi Z et al. Optimal power management for a hydraulic hybrid delivery truck. *J Veh System Dynamics* 2004; 42(1–2): 23–40.
10. Jeon S, Jo S, Park Y and Lee JM. Multi-mode driving control of a parallel hybrid electric vehicle using driving pattern recognition. *Trans ASME, J Dynamic Systems, Measmt Control* 2002; 124: 141–148.
11. Lin CC, Peng H, Jeon S and Lee JM. Control of a hybrid electric truck based on driving pattern recognition. In: *6th international symposium on advanced vehicle control*, Hiroshima, Japan, 9–13 September 2002. Tokyo: JSAE.
12. Langari R and 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(3): 925–934.
13. Won J-S and Langari R. Intelligent energy management agent for a parallel hybrid vehicle – Part II: torque distribution, charge sustenance strategies, and performance results. *IEEE Trans Veh Technol* 2005; 54(3): 935–952.
14. Abdollahi AD, Nikravesh SK and Menhaj MB. An intelligent control strategy in a parallel hybrid vehicle. *J Iranian Assoc Electr Electron Engrs* 2007; 4(2): 43–52.
15. Park J, Chen Z, Kiliaris L et al. Intelligent vehicle power control based on machine learning of optimal control

- parameters and prediction of road type and traffic congestion. *IEEE Trans Veh Technol* 2009; 58(9): 4741–4756.
16. Chen Z, Kiliaris L, Murphrey YL and Masrur MA. Intelligent power management in SHEV based on roadway type and traffic congestion levels. In: *Vehicle power and propulsion conference*, Dearborn, Michigan, USA, 7–10 September 2009, pp. 915–920. New York: IEEE.
  17. Murphrey YL, Park J, Chen Z et al. Intelligent hybrid power control – Part I: machine learning of optimal vehicle power. *IEEE Trans Veh Technol* 2012; 61(8): 3519–3530.
  18. Huang X, Tan Y and He X. An intelligent multi-feature statistical approach for the discrimination of driving conditions of a hybrid electric vehicle. In: *International joint conference on neural networks*, Atlanta, Georgia, USA, 14–19 June 2009, pp. 1286–1293. New York: IEEE.
  19. Montazeri-Gh M and Asadi M. Intelligent control for HEV power management based on driving conditions. In: *2nd international conference on modeling, simulation and applied optimization*, Abu Dhabi, United Arab Emirates, 24–27 March 2007.
  20. Montazeri-Gh, M and Asadi M. Intelligent approach for parallel HEV control strategy based on driving cycles. *Int J Systems Sci* 2011; 42(2): 287–302.
  21. Theodoridis S and Koutroumbas K. *Pattern Recognition*. 3rd edition. New York: Elsevier, 2006.
  22. Carlson TR and Austin RC. Development of speed correction cycles. Report SR97-04-01, Sierra Research, Inc., Sacramento, California, USA, 1997.
  23. Pu J, Yin C and Zhang J. Fuel optimal control of parallel hybrid electric vehicles. *Frontiers Mech Engng China* 2008; 3(3): 337–342.
  24. Wegleiter H, Schweighofer B, Holler G and Brunnader R. Fast quasi optimal control of hybrid electric vehicles considering limiting conditions. In: *European electric vehicle congress*, Brussels, Belgium, 26–28 October 2011, paper number: EEVC-22650916.
  25. Ericson E. Independent driving pattern factors and their influence on fuel-use and exhaust emission factors. *Transpn Res Part D* 2001; 6(5): 325–341.
  26. Johri R and Filipi Z. Self-learning neural controller for hybrid power management using neuro-dynamic programming. SAE paper 2011-24-0081, 2011.
  27. Johri R, Salvi A and Filipi Z. Optimal energy management for a hybrid vehicle using neuro-dynamic programming to consider transient engine operation. In: *ASME 2011 dynamic systems and control conference and Bath/ASME symposium on fluid power and motion control*, Arlington, Virginia, USA, 31 October–2 November 2011, Vol 2, paper DSAC2011-6138, pp. 279–286. New York: ASME.
  28. Filipi Z, Louca L, Daran B et al. Combined optimization of design and power management of the hydraulic hybrid propulsion system for the 6 × 6 medium truck. *Int J Heavy Veh Systems* 2004; 11(3–4): 372–402.
  29. Sun H, Jiang J-H and Wang X. Optimal torque management strategy for a parallel hydraulic hybrid vehicle. *Int J Automot Technol* 2007; 8: 791–798.
  30. Taghavipour A and Foumani MS. Optimized fuzzy control strategy for a SPA hybrid truck. *Int J Automot Technol* 2012; 13: 817–824.
  31. Esfahanian V, Safaei A, Nehzati H et al. Design and modeling of a parallel hydraulic hybrid bus. In: *International conference on applications and design in mechanical engineering*, Penang, Malaysia, 27–28 February 2012, paper number EN06. Perlis: Universiti Malaysia Perlis.
  32. Montazeri-Gh M and Naghizadeh M. Development of the Tehran car driving cycle. *Int J Environ Pollution* 2007; 30(1): 106–118.
  33. Seinfeld JH and Lapidus L. Aspects of the forward dynamic programming algorithm. *Ind Engng Chem Process Des Dev* 1968; 7: 475–478.
  34. Bellman R. *Dynamic programming*. Princeton, New Jersey: Princeton University Press, 1957.
  35. Dreyfus SE. *Dynamic programming and calculus of variations*. New York: Academic Press, 1965.

## Appendix I

### Notation

$a^+$	acceleration of the vehicle
$\dot{m}_f$	fuel consumption rate of the engine
$q$	clustering factor
$T_{eng}$	torque of the engine
$T_{sec}$	torque of the secondary power source
$u_i$	fuzzy similarity weights
$v$	speed of the vehicle
$x$	distance driven by the vehicle
$x_i$	number of features of the driving cycles $i$
$\theta_j$	number of features of the reference clusters $j$

### Abbreviations

DP	dynamic programming
FDCIA	fuzzy driving cycle identification agent
ICE	internal-combustion engine
MLP	multi-layer perceptron
PKE	positive kinetic energy
RPA	relative positive acceleration
SOC	state of charge

## Appendix 2 Dynamic programming

DP is a recursive algorithm for solving sequential decision problems.<sup>15</sup> In this algorithm, the main problem is divided into several subproblems, and then the optimum decision in each of them is set. The subproblems are interconnected with respect to time. In fact, the optimum decision path in the entire time duration of the main problem is determined by connecting the optimum decision of all subproblems. This trick prevents implementation of an exhaustive search for finding the optimum solution, which consumes much more time. In 1957, Bellman<sup>34</sup> proposed and proved the optimality of the DP algorithm.<sup>16</sup> He developed the algorithm through the principle of optimality.

Consider a continuous dynamic system with the differential equation

$$\begin{aligned}\dot{x} &= f(x(t), u(t), t) \\ x(0) &= x_0\end{aligned}$$

where  $x(t)$  is the vector of state variables and  $u(t)$  is the vector of control variables. Regarding the routine procedure of solving optimal control problems,  $u(t)$  should be determined from the acceptable range of values in order to optimize a scalar cost function such as

$$R(u(t), t_f) = \phi(x(t_f)) + \int_0^{t_f} J(x(t), u(t)) dt$$

The problem for the discrete systems is stated similarly according to<sup>15</sup>

$$\begin{aligned} x_{n+1} - x_n &= h(x_n, u_n, n \delta t) \\ R(u_n, N) &= \phi(x_N) + \sum_{n=1}^{N-1} J(x_n, u_n) \delta t \end{aligned}$$

where  $N$  is the number of time steps for the discrete system. The Bellman optimality principle defines the recursive equation to find the optimal solution as

$$\begin{aligned} F_{N-n}(x_n) &= \min_{u_n} [J(x_n, u_n) \delta t + F_{N-(n+1)}(x_{n+1})] \\ F_0(x_N) &= \phi(x_N, N) \end{aligned}$$

where  $F_{N-n}(x_n)$  is the integrated value for the above cost function  $R(u_n, N)$  using the optimal path from step  $n$  to the final step. In other words, the optimum

decision in each step is determined by minimizing the integrated cost function from the current step to the last step of the discrete system. Generally, the solution starts from the last step and continues in a backward manner to the first step. This procedure, which is called backward DP, requires the value for the cost function at the last step of the system so as to be initialized.

Besides the procedure proposed above, there is another procedure for performing the DP algorithm. Dreyfus<sup>35</sup> in 1965 proposed forward DP.<sup>17</sup> The state of the system at the first step and its cost function value are known for initializing the algorithm. The algorithm starts from the first step and continuous to the last step in a forward manner. The recursive equation for forward DP is<sup>33</sup>

$$\begin{aligned} H_n(x_n) &= \min_{u_{n-1}} [J(x_{n-1}, u_{n-1}) \delta t + H_{n-1}(x_{n-1})] \\ H_0(x_0) &= 0 \end{aligned}$$

where  $H_n(x_n)$  is the value for the integrated cost function using the optimal path from the first step to the  $n$ th step. As can be seen, there is not any requirements for defining the values of the cost function at the last step. This is the greatest advantage that forward DP has over backward DP.