

Article

Development of an Advanced Rule-Based Control Strategy for a PHEV Using Machine Learning

Hanho Son , Hyunhwa Kim, Sungho Hwang and Hyunsoo Kim *

School of Mechanical Engineering, Sungkyunkwan University, Seobu-ro, Suwon-si 2066, Korea; hanho1014@naver.com (H.S.); yc4331@hanmail.net (H.K.); hsh0818@skku.edu (S.H.)

* Correspondence: hskim@me.skku.ac.kr; Tel.: +82-31-290-7911

Received: 18 December 2017; Accepted: 29 December 2017; Published: 1 January 2018

Abstract: This paper presents an advanced rule-based mode control strategy (ARBC) for a plug-in hybrid electric vehicle (PHEV) considering the driving cycle characteristics and present battery state of charge (SOC). Using dynamic programming (DP) results, the behavior of the optimal operating mode was investigated for city (UDDS $\times 2$, JC08 $\times 2$) and highway (HWFET $\times 2$, NEDC $\times 2$) driving cycles. It was found that the operating mode selection varies according to the driving cycle characteristics and battery SOC. To consider these characteristics, a predictive mode control map was developed using the machine learning algorithm, and ARBC was proposed, which can be implemented in real-time environments. The performance of ARBC was evaluated by comparing it with rule-based mode control (RBC), which is a CD-CS mode control strategy. It was found that the equivalent fuel economy of ARBC was improved by 1.9–3.3% by selecting the proper operating mode from the viewpoint of system efficiency for the whole driving cycle, regardless of the battery SOC.

Keywords: plug-in hybrid electric vehicle (PHEV); operating mode; driving cycle characteristics; battery state of charge (SOC); machine learning; rule-based control

1. Introduction

A plug-in hybrid electric vehicle (PHEV) drives mostly in the EV mode when the battery state of charge (SOC) is sufficient. However, when the battery SOC drops to the lower limit, the vehicle runs in the hybrid electric vehicle (HEV) mode to sustain the battery SOC using various operating modes such as power split, parallel mode and etc. To achieve improved system efficiency as well as SOC balancing, it is essential to control the operating mode for a given vehicle speed, wheel power, and battery SOC [1–3].

In addition, in the HEV mode, load leveling is generally performed to operate the engine on the optimal operating line, and the power is distributed to the engine and motor depending on the magnitude of the load leveling [4,5]. Since the PHEV efficiency varies depending on the power distribution between the engine and motor, it is important to obtain the optimal power management strategy for the PHEV for the given driving conditions.

For the optimal control strategy of a PHEV, rule-based control for the CD and CS modes has been widely used based on heuristic data [6,7]. Rule-based control has the advantage of easy on-line application, but cannot guarantee optimality for real driving cycles [8]. To solve this problem, control strategies were proposed to ensure local optimality, including rule-based control using the mode control algorithm [3], fuzzy logic [9,10], and the equivalent consumption minimization strategy based on equivalent fuel consumption of the battery energy [11,12]. However, since these control strategies do not consider the effect of the present battery SOC, they cannot guarantee global optimality for the CD and CS mode. To ensure global optimality, rule-based control was proposed by implementing an optimization tool, such as dynamic programming (DP) [6,13,14] or Pontryagin's minimum principle

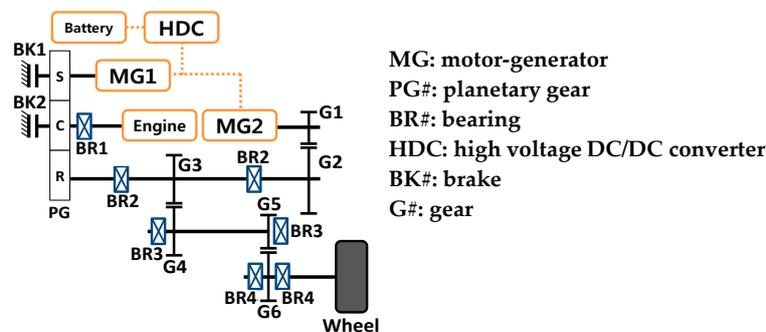
(PMP) [15–17]. In a previous study [2], a rule-based mode control (RBC) strategy was obtained for the CS mode using DP with power electronics (PE) and drivetrain losses. The RBC strategy was derived based on the operating points considering the number of mode occurrences for the HWFET and UDDS cycles. The mode shift boundary was determined by the vehicle speed and wheel power without consideration of the present battery SOC. An optimal rule-based control strategy was developed for non plug-in HEV based on machine learning [18,19]. In the literature [18], hybrid architectures were investigated that minimize the cost, fuel consumption and clustering optimization rule extraction method was proposed to identify the optimal control strategy. However, few works have been reported on rule-based optimal control strategy of PHEV considering the present SOC for the CD mode and CS mode which can be implemented in real time.

In this study, an advanced rule-based mode control strategy (ARBC) was proposed for a PHEV considering the driving cycle characteristics and present battery SOC for CD and CS mode. To develop a mode control map, a backward simulator was developed by considering the losses of the power electronic system and drivetrain elements. Using the predictive mode control map that is obtained from machine learning, an ARBC was suggested, and its performance was evaluated for a real driving cycle.

2. PHEV Model for Backward Simulator

2.1. Vehicle Configuration and Specification

In this study, a power split/parallel-type PHEV was selected as the target vehicle. The target vehicle implements four operating modes: (1) EV#1, (2) EV#2, (3) power split, and (4) parallel. Figure 1 shows the configuration and operating modes for the target PHEV with the engagement states of the friction elements. In the EV#1 and power split modes, brake 1 (BK1) and brake 2 (BK2) are disengaged. By engaging BK2, the vehicle drives in the EV#2 mode using motor-generator 1 (MG1) and motor-generator 2 (MG2) together. When BK1 is engaged and the speed of MG1 remains zero, the parallel mode is realized, where the engine speed directly depends on the vehicle speed. The vehicle specifications are shown in Table 1.



| Operating mode | BK1 | BK2 | MG1 | MG2 | Engine |
|----------------|------------|------------|-----|-----|--------|
| EV#1 | Disengaged | Disengaged | Off | On | Off |
| EV#2 | Disengaged | Engaged | On | On | Off |
| Power split | Disengaged | Disengaged | On | On | On |
| Parallel | Engaged | Disengaged | Off | On | On |

Figure 1. Configuration and operating mode of the power split/parallel-type plug-in hybrid electric vehicle (PHEV) with the engagement states of the friction elements [20].

Table 1. Vehicle specification of the power split/parallel-type PHEV.

| Vehicle Specification | | |
|-----------------------|----------------------|--------------------|
| Engine | Max power/torque | 115 kW/185 Nm |
| MG1 | Max power/torque | 70 kW/50 Nm |
| MG2 | Max power/torque | 90 kW/270 Nm |
| Battery | Max power/capacity | 50 kW/25 Ah |
| Vehicle | Mass | 1800 kg |
| | Tire radius | 0.32 m |
| Gear ratio | PG/G1–G2/G3–G4/G5–G6 | 2.6/2.478/1.0/3.54 |

2.2. Speed and Torque Analysis

The speed and torque analysis was performed for the power split mode. In the EV#1 mode, the planetary gear rotates freely because the vehicle is driven only by MG2. The planetary gear acts as a reduction gear in the EV#2 and parallel modes. The speed relationships for the EV#1, EV#2, and parallel modes are shown in Figure 2. In the power split mode, the speed and torque equations can be derived via lever analysis as follows [21]:

$$\omega_{MG1} = (N_{PG} + 1)\omega_e - (N_{PG})\omega_r \quad (1)$$

$$\omega_{MG2} = N_{MG2}\omega_r = N_{MG2}N_{FRG}\omega_o \quad (2)$$

$$T_r = \left(\frac{N_{PG}}{N_{PG} + 1} \right) T_e \quad (3)$$

$$T_{MG1} = -\left(\frac{1}{N_{PG} + 1} \right) T_e \quad (4)$$

$$T_o = T_r + N_{MG2}T_{MG2}, \quad (5)$$

where ω is the rotational speed, T is the torque, and N is the gear ratio. The subscript MG represents the motor and generator (MG), PG the planetary gear, e the engine, r the ring, and o the output shaft.

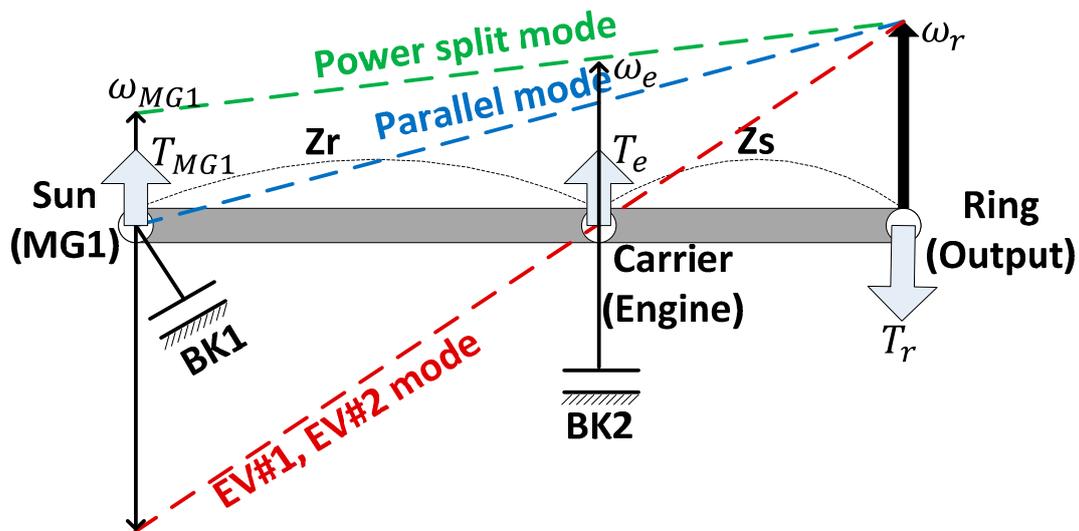


Figure 2. Speed and torque analysis using the lever analogy in each operating mode.

2.3. Backward Simulator with Component Losses

A backward simulator was developed to extract the optimal mode control strategy using DP. The process of the backward simulation was described in the literature [2,20]. For the target PHEV,

the control and state variables were selected as the battery power and battery SOC. The instantaneous optimal operating points of the engine, MG1, and MG2 were determined at each time step for the given driving information. In the global horizontal plane, i.e., the time-SOC plane, the minimum fuel consumption was calculated by adding the minimum consumption rate for each time step. The backward simulation was performed using the following recursive equation and battery SOC constraint:

$$J^*(P_b(t), t) = \int_{t_0}^{t_f} g\{P_b(t), t\} dt \tag{6}$$

$$SOC_{\text{initial}} - SOC_{\text{final}} = \text{constant}, \tag{7}$$

where t_0 is the initial time, t_f is the final time, J^* is the minimum fuel consumption, P_b is the battery power and g is the fuel consumption rate.

In the backward simulation, the drivetrain component loss and PE system loss were considered. The PE system loss was obtained from the efficiency map of the motor and generator (MG) and high voltage DC/DC converter (HDC). The HDC loss was determined according to the voltage for a boost converter. The drivetrain component losses were modeled using the mathematical governing equations and experimental results [2,20]. The loss models of the PE system and drivetrain component are shown in Table 2.

Table 2. Mathematical governing equation and experimental map for the power electronics (PE) system and drivetrain component loss model.

| | | | |
|--------------------------|---|--|---|
| Power electronics system | MG1 torque, Nm MG1 speed, rpm < MG1 > | MG2 torque, Nm MG2 speed, rpm < MG2 > | HDC efficiency Voltage Power < HDC > |
| Drivetrain components | Torque loss, relative value Speed, rpm < Clutch > | Torque loss, relative value Speed, rpm < MG1 unloaded > | |
| | Torque loss, relative value Speed, rpm < Seal ring > | Relative value Pressure, bar Speed, rpm < Oil pump > | |
| | Gear [22] | $T_{\text{loss_Gear}} = C_{\text{Gear}} \times T_{\text{in}}$ | |
| | Planetary gear (PG) [23,24] | $T_{\text{loss_PG_carrier}} = C_{\text{PG}} \times T_{\text{in_PG}}$ $T_{\text{loss_PG_ring}} = C_{\text{PG}} \times \left(\frac{Z_r}{Z_r + Z_s} \right) \times T_{\text{in_PG}}$ $T_{\text{loss_PG_sun}} = C_{\text{PG}} \times \left(\frac{Z_s}{Z_r + Z_s} \right) \times T_{\text{in_PG}}$ | |
| Bearing [22,25] | $T_{\text{loss_BRGload}} = f_1 \times P_1^a \times d_m^b \times 10^{-3}$ | | |
| Churning [26] | $T_{\text{loss_churning}} = \frac{1}{2} \rho \omega^2 R_p^3 S_m C_m$ | | |

3. Advanced Rule-Based Mode Control Strategy

3.1. Dynamic Programming to Obtain the Optimal Operating Mode According to the Battery SOC and Driving Cycle Characteristics

In a previous study [2], a RBC strategy was proposed, which can be applied in real-time environments. RBC was obtained based on the mode control map constructed from the DP by considering the PE and drivetrain components losses. RBC is a type of CD-CS mode control strategy. In RBC, vehicle drives in the EV mode until the battery SOC reaches the SOC lower limit. When the battery SOC reaches the lower limit, the vehicle drives in the charge sustaining (CS) mode using the RBC map.

In this study, an advanced RBC strategy was proposed, which can be applied in both the CD and CS modes. To develop the advanced RBC strategy (ARBC), investigation of the behavior of the optimal operating mode when the battery SOC is higher than the lower limit (CD mode) as well as in CS mode is required.

To develop the mode control map considering the battery SOC variation and driving cycle characteristics, DP was performed for the UDDS $\times 2$, HWFET $\times 2$, NEDC $\times 2$, and JC08 $\times 2$ cycles when the initial battery SOC changes from 30–60%. The battery SOC trajectories and operating time of each mode from the DP are shown in Figure 3. It is seen from the DP results in Figure 3 that the operating time of each mode varied according to the initial battery SOC. For instance, for the UDDS $\times 2$ cycle, the operating time of the EV#1 mode increased as the initial SOC increased. As a result, the SOC decreased rapidly to the final SOC, 30%. For the JC08 $\times 2$ cycle, since the final SOC did not drop to the lower limit of 30% for the initial SOC of 60%, DP was performed for SOC = 30–50%. From Figure 3, it is found that the mode control map needs to be derived by considering the initial battery SOC.

Figure 4 shows the optimal operating mode for the UDDS, HWFET, NEDC, and JC08 cycles obtained by the DP when the initial battery SOC varies. It is seen that the optimal operating mode selected by DP is changed by the driving cycle characteristics as well as the initial battery SOC. When the initial SOC is 50%, the EV mode is mostly used for the vehicle speed $v < 46$ kph because the battery energy is sufficient. In contrast, the power split mode is seldom selected. When the initial SOC is 30%, the power split mode is used for SOC balancing (regions A, C, D). When the vehicle speed is faster than 46 kph, the parallel mode is used since it can be implemented only after the engine is connected to the wheel when the engine speed is higher than the idling speed, $\omega_{idling} = 1000$ rpm. When the initial SOC is equal to 50% for the HWFET cycle, the power split mode is selected in the high vehicle speed and low power region (B), but it can be seen that its operating area varies according to the driving cycle characteristics. For the JC08 cycle, the power split mode was not used in the high speed and low power region. Since the demanded wheel power is relatively small in the JC08 cycle, the EV#1 mode was selected instead of the parallel and power split modes. From the DP results, it is seen that the optimal operating mode map varies depending on the present battery SOC and driving cycle characteristics. Since the battery SOC and driving characteristics are always changing in the real driving environment, it is necessary to develop an optimal mode control map considering the battery SOC change and road characteristics.

| Cycle | Optimal battery SOC trajectories from dynamic programming (final SOC = 30%) | Operating time of each mode (sec) | | | | |
|----------|--|-----------------------------------|-----|------|------|------|
| | | Initial SOC | 30% | 40% | 50% | 60% |
| UDDS ×2 | | EV#1 | 849 | 1051 | 1256 | 1467 |
| | | EV#2 | 4 | 6 | 6 | 18 |
| | | Power split | 335 | 183 | 28 | 0 |
| | | Parallel | 332 | 280 | 230 | 35 |
| | | Regen | 702 | 702 | 702 | 702 |
| HWFET ×2 | | EV#1 | 306 | 478 | 689 | 881 |
| | | EV#2 | 60 | 74 | 84 | 102 |
| | | Power split | 120 | 135 | 100 | 58 |
| | | Parallel | 864 | 663 | 477 | 309 |
| | | Regen | 170 | 170 | 170 | 170 |
| NEDC ×2 | | EV#1 | 844 | 944 | 1160 | 1282 |
| | | EV#2 | 0 | 51 | 62 | 80 |
| | | Power split | 102 | 2 | 5 | 10 |
| | | Parallel | 462 | 411 | 181 | 36 |
| | | Regen | 366 | 366 | 366 | 366 |
| JC08 ×2 | | EV#1 | 517 | 731 | 979 | - |
| | | EV#2 | 0 | 2 | 2 | - |
| | | Power split | 191 | 20 | 0 | - |
| | | Parallel | 324 | 279 | 51 | - |
| | | Regen | 662 | 662 | 662 | - |

Figure 3. The optimal battery state of charge (SOC) trajectories for the UDDS ×2, HWFET ×2, NEDC ×2, and JC08 ×2 cycles from the dynamic programming results (final battery SOC = 30%).

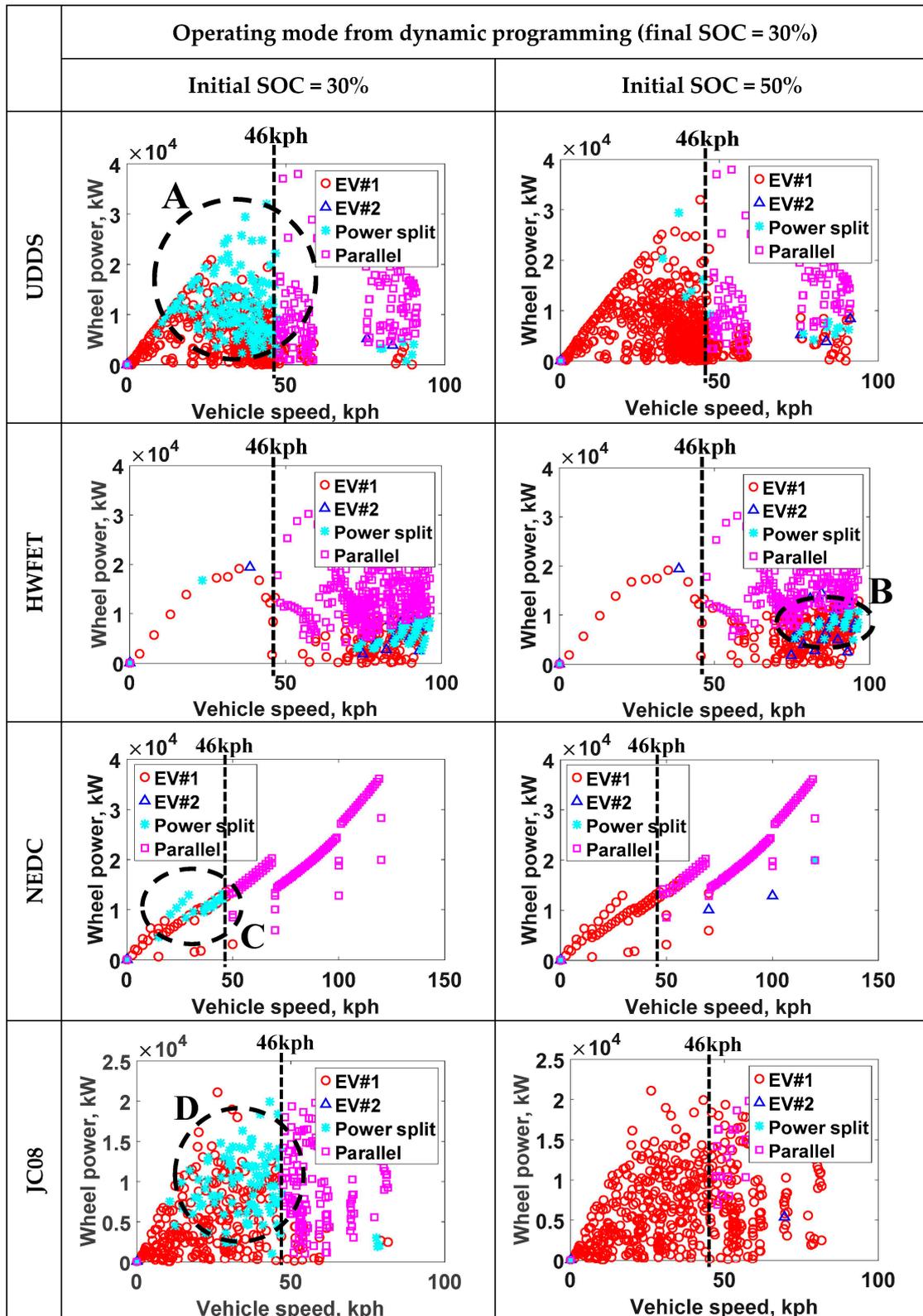


Figure 4. The points for the optimal operating mode that are obtained from the dynamic programming according to the driving cycle characteristics and initial battery SOC.

3.2. Advanced Rule-Based Mode Control Strategy Using a Predictive Mode Control Map

To develop an optimal mode control map with consideration of the present battery SOC and driving cycle characteristics, predictive models were built using machine learning, and a predictive mode control map was constructed based on the selected predictive model.

To build the predictive model, the vehicle speed, wheel power, and battery SOC were defined as the input variables, and the optimal operating mode was defined as an output variable, i.e., the response variable. Using the input and output variables, machine learning was performed to build the predictive model. Since the selected operating mode is a discrete data type, the following three classification algorithms were used: nearest neighbor classification (NNC), decision trees (DTs), and support vector machines (SVMs).

NNC is a machine learning algorithm that finds the nearest neighbors from known data for prediction [27,28]. NNC has the characteristic of simple and good predictive performance. DTs is also a classification method to predict the output response using the sequential binary decisions. It is easy to predict the response output, which does not require normalization or preprocessing data [29,30]. SVMs creates a boundary between each response output using the support vector to build a predictive model. SVMs works well for both low-dimensional and high-dimensional data, but requires a large computational load [31].

Using the data set from the dynamic programming results, predictive models were built to predict the optimal operating mode. Since the learning and validation of a predictive model cannot guarantee the reliability of the model, learning and validation of the data set were divided into the training set and the test set. The proportion of the training and test data sets used was 70% and 30%, respectively, which is generally considered to be highly reliable [32].

Table 3 shows the prediction accuracy for each predictive model that was built using the training set. The resubstitution accuracy was obtained by validating the predictive model using the training set. The test accuracy was obtained by validating the predictive model using the test set. When NNC predicts the new response output to the new input variable, the predictive model accuracy depends on the number of nearest neighbor data, k [27]. In general, since k has a high model accuracy with a value from 1 to 6, NNC was built according to k . The resubstitution and test accuracy for each NNC model was evaluated. When k is 1, it was found that the resubstitution accuracy is 100%, which means that the predictive model is over-fitted from the training set. When the predictive model is over-fitted, it may result in a problem where the noise data of the training set in the over-fitted model is also learned and causes a generalization error [33]. Therefore, $k = 3$ was selected for NNC, which can avoid the over-fitting problem with a relatively high test accuracy. In DTs, the performance of the predictive model is determined depending on the pruning step. Since the pruning step $p = 0$ leads to over-fitted learning, $p = 15$ was selected, which has the highest test accuracy among the six pruning levels. In SVMs, Gaussian SVMs showed the highest resubstitution and test accuracy, 92.2% and 90.2%, respectively, compared to the linear SVMs. However, it is seen that the accuracy of Gaussian SVMs is lower than that of DTs and NNC.

Using the selected predictive models, NNC with $k = 3$, DTs with $p = 15$, and Gaussian SVMs, predictive mode control maps were obtained. In Figure 5, the predictive mode control map for the selected NNC, DT, and SVM models are compared with the optimal operating mode by the DP for the present battery SOC, 30% and 50%. As shown in Figure 5c,d, the predictive mode control map by NNC shows an unclear boundary between the operating modes (region P) because the NNC model is over-fitted due to noise. The unclear boundary may cause a frequent mode shift. The predictive mode control map by DTs (Figure 5e,f) provides a clear boundary, which means that it has a robustness against the noise data of the training set. In contrast, in the predictive mode control map by SVMs, we can see that the EV#1 mode is used for almost all of the driving range, which does not make sense in the real driving environment.

Table 3. Resubstitution and test accuracy for the predictive model including nearest neighbor classification (NNC), decision trees (DTs), and support vector machines (SVMs). (*Resubstitution accuracy* = $(1 - \text{resubstitution error}) \times 100$, *test accuracy* = $(1 - \text{test error}) \times 100$).

| | | Predictive Model (Classifier) | | | | |
|--|---------|---------------------------------------|----------------------|-------------|-------------------------------|--------------|
| Training set | | 70% | | | | |
| Test set | | 30% | | | | |
| | | Nearest neighbor classification (NNC) | Decision trees (DTs) | | Support vector machine (SVMs) | |
| Resubstitution accuracy /Test accuracy [%] | $k = 1$ | 100% /93.3% | $p = 0$ | 97.3%/93.5% | Linear | 61.5% /61.6% |
| | $k = 2$ | 96.4% /93.2% | $p = 5$ | 97.2%/93.5% | Gaussian | 92.2% /90.2% |
| | $k = 3$ | 96.5% /92.4% | $p = 10$ | 97.0%/93.6% | | |
| | $k = 4$ | 94.8% /92.7% | $p = 15$ | 96.5%/93.7% | | |
| | $k = 5$ | 94.5% /92.5% | $p = 20$ | 95.9%/93.5% | | |
| | $k = 6$ | 94.1% /92.6% | $p = 25$ | 95.1%/93.2% | | |

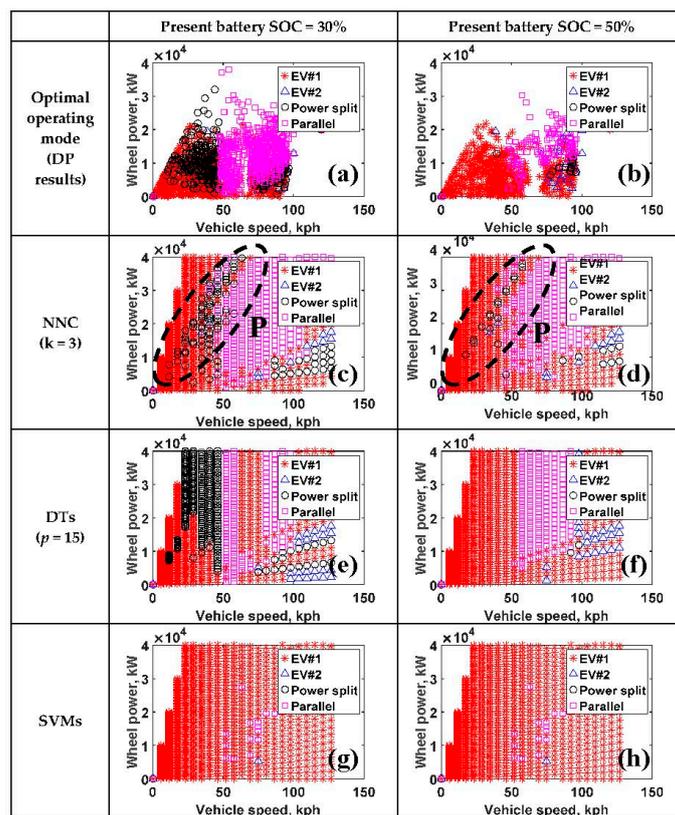


Figure 5. Predictive mode control map using NNC, DTs, and SVMs when the present battery SOC is 30% and 50%. (a,b) DP results; (c,d) NNC; (e,f) DTs; (g,h) SVMs.

From the comparison of the predictive mode control maps in Figure 5, the mode control map by DTs was selected by considering the performance for robustness, resubstitution accuracy, and test accuracy.

Using the predictive mode control map by DTs, an advanced RBC strategy (ARBC) was proposed, which is shown in Figure 6. ARBC is rule-based control since it uses a predictive mode control map that

is constructed from the off-line environment. For the given wheel power, vehicle speed, and present battery SOC, the demanded operating mode is determined from the predictive mode control map. Since the predictive mode control map was constructed with respect to the various SOC and driving cycle characteristics by machine learning, the optimal operating mode can be predicted from ARBC when the vehicle encounters driving conditions outside the training set.

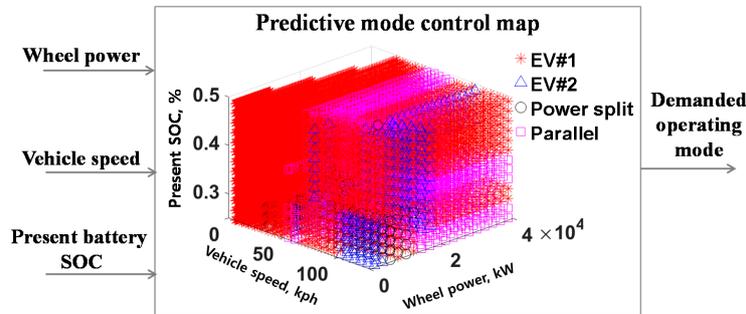


Figure 6. Advanced rule-based mode control (RBC) strategy.

4. Performance of the Advanced Rule-Based Mode Control Strategy

4.1. Forward Simulator

To evaluate the performance of ARBC, a forward simulator was developed, which is shown in Figure 7. The forward simulator consists of the advanced rule-based control strategy and vehicle model. In the vehicle model, dynamic models of the engine, MG1, MG2, planetary gear, brake, HDC, battery, and vehicle longitudinal motion were constructed using MATLAB/Simulink and SimDriveline, including the PE system and drivetrain component losses.

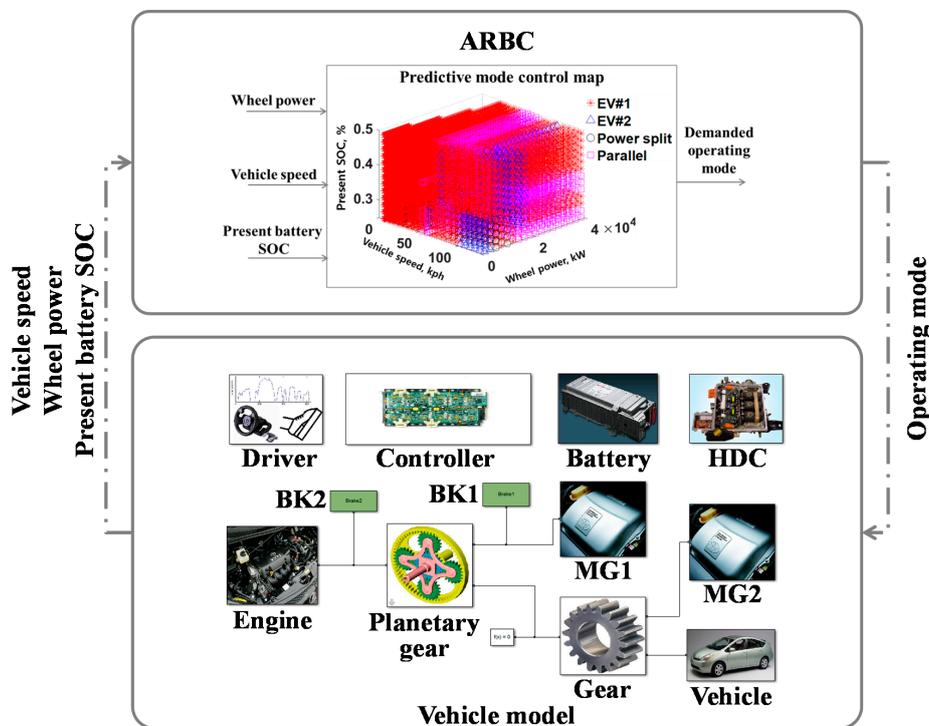


Figure 7. Forward simulator.

4.2. Performance of ARBC by Comparing with RBC

In Figure 8, the performance of ARBC is compared with that of RBC [2] using the forward simulator in Figure 7. The simulations were performed for two HWFET cycles (a), HWFET $\times 2$, to evaluate the performance of the proposed mode control strategy. Figure 8b shows the operating mode. Since the initial battery SOC is 50%, which is much higher than the lower limit of 30%, RBC selects the EV#1 and EV#2 modes for region P until the battery SOC drops to the lower limit. Since the engine is off, the PHEV was driven using the battery energy. As a result, the battery SOC rapidly decreased while the engine speed (e) and fuel consumption (d) remained at zero. In contrast, ARBC selects the parallel mode and EV#1 mode (b) in region P from the predictive mode control map. Whenever the parallel mode was used, the engine speed (e) and torque (f) were generated. The reason the parallel mode was selected in region P is that it is more efficient to use the engine rather than the battery to supply the demanded vehicle power required for highway driving. Owing to the frequent usage of the parallel mode, i.e., the engine, the battery SOC (c) decreased more slowly, which resulted in reduced PE losses (k) compared with those of RBC. The fuel consumption (d) of ARBC increased due to the parallel mode operation. After region P, RBC performed charging and discharging of the battery repeatedly for SOC balancing (c). Whenever the engine was operated to charge the battery, the SOC increased (c) and the fuel consumption (d) increased. As a result, the fuel consumption by RBC increased rapidly and the accumulated fuel consumption of RBC becomes larger than ARBC after point Q. The MG2 speed (g) is proportional to the vehicle speed for both controls because MG2 is mechanically connected to the wheel. The MG2 torque (h) was generated to meet the required wheel torque via the load leveling control. The MG2 torque (h) shows a negative value when regenerative braking is performed. Since MG1 is used to drive the vehicle in the EV#2 mode, MG1 shows a negative speed (i), referring to Equation (1), and negative torque (j) to generate the positive power to propel the vehicle. In ARBC, the MG1 torque and speed remain at zero in the parallel mode since it does not work. When the EV#1 mode is selected, the MG1 speed also shows a negative value. It is seen that MG1 generates a negative torque at approximately $t = 1200 - 1290$ s when the power split mode is used. In the power split mode, the MG1 torque was generated from the relationship in Equation (4). The amount of drivetrain loss (l) for ARBC and RBC are similar during the whole driving cycle.

Since the simulation results in Figure 8 were obtained by the forward simulation, a difference exists between the final SOC and the target SOC, 30%. Therefore, to evaluate the fuel economy, the SOC difference was transformed into the equivalent fuel economy (EFE) as follows [34]:

$$FE = \frac{D}{\frac{(SOC_{initial} - SOC_{final}) \times Q}{E_g} + \frac{\Delta m_{fuel}}{\rho_{fuel}}} \quad (8)$$

where E_g is the gasoline-equivalent energy content of electricity, D is the driving distance, SOC_{final} is the final battery SOC, $SOC_{initial}$ is the initial battery SOC, Q is the battery capacity, Δm_{fuel} is the fuel consumption, and ρ_{fuel} is the fuel density.

Comparing the equivalent fuel economy using Equation (8), EFE of ARBC was obtained as 24.9 km/L, which is an improvement of 3.3% compared with 24.1 km/L of RBC.

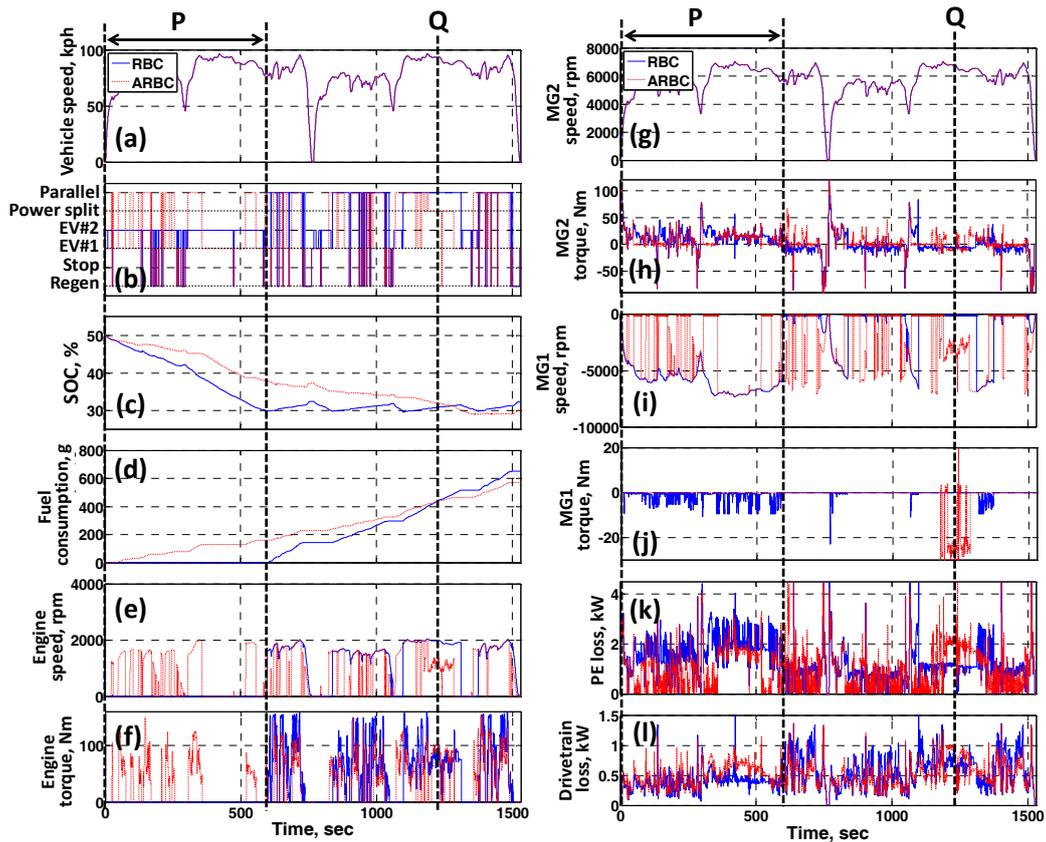


Figure 8. Simulation results of ARBC and RBC for HWFET $\times 2$ cycle. (a) Vehicle speed; (b) Operating mode; (c) Battery SOC; (d) Fuel consumption; (e) Engine speed; (f) Engine torque; (g) MG2 speed; (h) MG2 torque; (i) MG1 speed; (j) MG1 torque; (k) PE loss; (l) Drivetrain loss.

4.3. Performance of ARBC for a Real Driving Route

The performance of ARBC was evaluated for a real driving route. In Figure 9a, the SKKU route is shown, which consists of a city (A-B-C-D-E) and a highway (E-A). The vehicle speed profile collected from GPS is shown in Figure 9b for the SKKU $\times 2$ cycle.

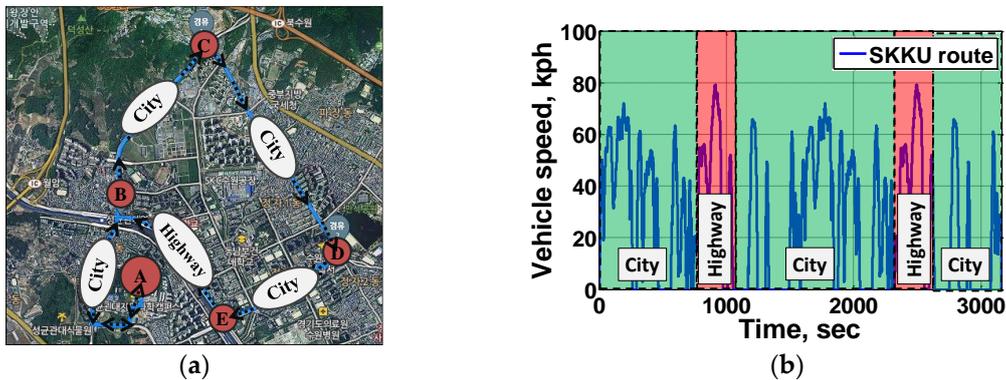


Figure 9. Real driving cycle: (a) SKKU route and (b) vehicle speed profile for the SKKU $\times 2$ cycle.

In Figure 10, simulation results by ARBC were compared with RBC for the SKKU $\times 2$ cycle. In the simulation, the initial and final battery SOC were set as 50% and 30%, respectively. It is seen from Figure 10 that in region R, the parallel and EV#1 modes were used by ARBC, while the EV#1 and EV#2 modes were selected for RBC. Whenever the parallel mode was used, the engine speed (e) and torque

(f) were generated. It is seen that the battery SOC (c) by ARBC decreased slowly through the whole driving cycle, while the SOC by RBC showed the typical behavior of CD-CS control. The equivalent fuel economy of ARBC was obtained as 26.8 km/L, which is an improvement of 1.9% compared with the 26.3 km/L of RBC.

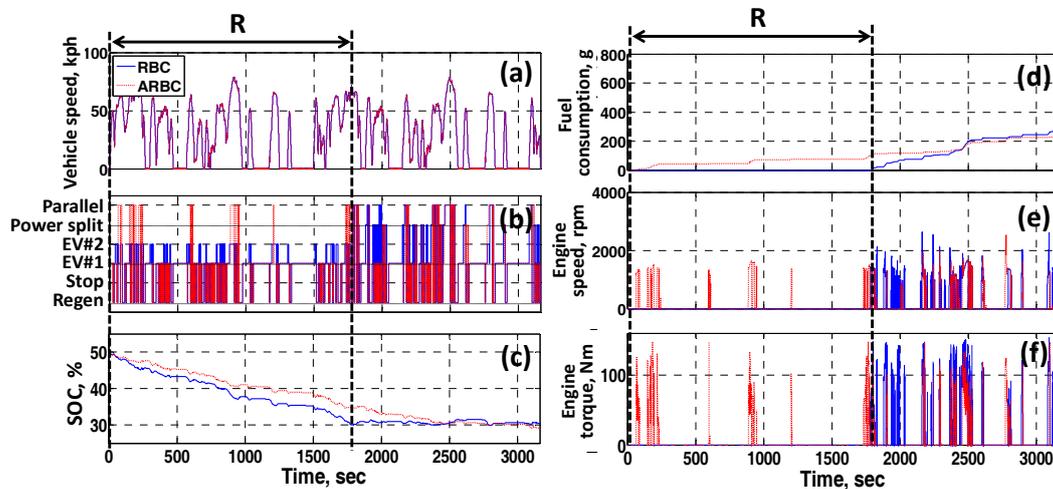


Figure 10. Simulation results of ARBC and RBC for the SKKU $\times 2$ cycle. (a) Vehicle speed; (b) Operating mode; (c) Battery SOC; (d) Fuel consumption; (e) Engine speed; (f) Engine torque.

5. Conclusions

In this study, an advanced RBC strategy was proposed for a PHEV by considering the driving cycle characteristics and present battery SOC. First, the speed and torque analysis was performed for the target PHEV using the lever analogy, and a backward simulator was developed based on DP. Using DP, the operating mode that has the maximum fuel economy potential was investigated for city (UDDS $\times 2$, JC08 $\times 2$) and highway (HWFET $\times 2$, NEDC $\times 2$) driving cycles. From the DP results, it was found that the operating range of the EV mode becomes wider as the initial battery SOC increases. In addition, it is seen that the selection of the operating mode varies according to the driving cycle characteristics, even if the initial and final battery SOC are the same. To consider these characteristics, a predictive mode control map was developed using machine learning. Among various machine learning algorithms, DTs was selected in building the predictive model since it showed the highest resubstitution and test accuracy. Using the predictive mode control map obtained from DTs, an advanced RBC strategy (ARBC) was proposed, which selects the operating mode considering the present battery SOC and driving cycle characteristics. The performance of ARBC was evaluated by comparing with a RBC (RBC), which is a CD-CS mode control strategy. From the simulation results, it was found that ARBC drives in the HEV mode (power split and parallel) even when the battery SOC is higher than the lower limit, which provided more efficient engine operation. As a result, the battery SOC by ARBC decreased slowly throughout the whole driving cycle, and this prevented unnecessary battery charging by the engine, which forces the engine to operate with low efficiency. The equivalent fuel economy (EFE) of ARBC was improved by 3.3% for the highway (HWFET $\times 2$) cycle. For a real driving route (SKKU $\times 2$), which consists of the city and highway, the EFE of ARBC was improved by 1.9% compared to that of RBC.

The ARBC proposed in this study provides the improved fuel economy by selecting the proper operating mode from the viewpoint of system efficiency for the whole driving cycle regardless of the battery SOC, and can be implemented in a real-time environment using the rule-based mode shift map constructed from machine learning.

Acknowledgments: This material is based upon work supported by the Ministry of Trade, Industry, and Energy (MOTIE, Korea) under Industrial Technology Innovation Program. No. 10062742.

Author Contributions: Hanho Son developed the advanced rule-based mode control strategy of PHEV considering present battery SOC and driving cycle characteristics and wrote the paper. Hyunhwa Kim contributed in development of a backward simulator using dynamic programming. Hyunsoo Kim and Sungho Hwang supervised this research and wrote the paper.

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

References

1. Kim, N.; Choi, S.; Jeong, J.; Vijayagopal, R.; Stutenberg, K.; Rousseau, A. Vehicle level control analysis for Voltec powertrain. In Proceedings of the EVS30 Conference, Stuttgart, Germany, 9–11 October 2017.
2. Son, H.; Kim, H. Development of near optimal rule-based control for plug-in hybrid electric vehicles taking into account drivetrain component losses. *Energies* **2016**, *9*, 420. [[CrossRef](#)]
3. Ma, C.; Kang, J.; Choi, W.; Song, M.; Ji, J.; Kim, H. A comparative study on the power characteristics and control strategies for plug-in hybrid electric vehicles. *Int. J. Autom. Technol.* **2012**, *13*, 505–516. [[CrossRef](#)]
4. Kim, J.; Kim, N.; Hwang, S.; Hori, Y.; Kim, H. Motor control of input-split hybrid electric vehicles. *Int. J. Autom. Technol.* **2009**, *10*, 733–742. [[CrossRef](#)]
5. Kang, J.; Choi, W.; Kim, H. Development of a control strategy based on the transmission efficiency with mechanical loss for a dual mode power split-type hybrid electric vehicle. *Int. J. Autom. Technol.* **2012**, *13*, 825–833. [[CrossRef](#)]
6. Lee, H.; Jung, J.; Im, W.; Cha, S. Optimization of rule-based power management control strategies of HEV based on dynamic programming. In Proceedings of the Korean Society of Automotive Engineers, Spring Conference, Gwang-ju, Korea, 28–29 May 2015; p. 953.
7. Lin, C.; Peng, H.; Grizzle, J.W. Power management strategy for a parallel hybrid electric truck. *IEEE Trans. Control Syst. Technol.* **2003**, *11*, 839–849.
8. Kim, N. Energy Management Strategy for Hybrid Electric Vehicles Based on Pontryagin’s Minimum Principle. Ph.D. Thesis, Seoul National University, Seoul, Korea, 2009.
9. Schouten, N.J.; Salman, M.A.; Kheir, N.A. Fuzzy logic control for parallel hybrid vehicles. *IEEE Trans. Control Syst. Technol.* **2002**, *10*, 460–468. [[CrossRef](#)]
10. Li, S.G.; Sharkh, S.M.; Walsh, F.C.; Zhang, C.N. Energy and battery management of a plug-in series hybrid electric vehicle using fuzzy logic. *IEEE Trans. Veh. Technol.* **2011**, *60*, 3571–3585. [[CrossRef](#)]
11. Paganelli, G.; Delprat, S.; Guerra, T.M.; Rimaux, J.; Santin, J.J. Equivalent consumption minimization strategy for parallel hybrid powertrains. In Proceedings of the Vehicular Technology Conference, Birmingham, AL, USA, 6–9 May 2002.
12. Zhang, C.; Vahidi, A. Route preview in energy management of plug-in hybrid vehicles. *IEEE Trans. Control Syst. Technol.* **2012**, *20*, 546–553. [[CrossRef](#)]
13. Yuan, Z.; Teng, L.; Fengchun, S.; Peng, H. Comparative study of dynamic programming and Pontryagin’s minimum principle on energy management for a parallel hybrid electric vehicle. *Energies* **2013**, *6*, 2305–2318. [[CrossRef](#)]
14. He, H.; Tang, H.; Wang, X. Global optimal energy management strategy research for a plug-in series-parallel hybrid electric bus by using dynamic programming. *Math. Probl. Eng.* **2013**, *2013*. [[CrossRef](#)]
15. Kim, N.; Cha, S.; Peng, H. Optimal control of hybrid electric vehicles based on Pontryagin’s minimum principle. *IEEE Trans. Control Syst. Technol.* **2011**, *19*, 1279–1287.
16. Kim, N.; Rousseau, A. Sufficient conditions of optimal control based on Pontryagin’s minimum principle for use in hybrid electric vehicles. *J. Autom. Eng.* **2012**, *226*, 1160–1170. [[CrossRef](#)]
17. Onori, S.; Tribioli, L. Adaptive Pontryagin’s minimum principle supervisory controller design for the plug-in hybrid GM Chevrolet Volt. *Appl. Energy* **2015**, *147*, 224–234. [[CrossRef](#)]
18. Finesso, R.; Spessa, E.; Venditti, M. An unsupervised machine-learning technique for the definition of a rule-based control strategy in a complex HEV. *SAE Int. J. Altern. Powertrain* **2016**, *5*, 308–327. [[CrossRef](#)]
19. Jeon, S.; Jo, S.; Park, Y.; Lee, J. Multi-mode driving control of a parallel hybrid electric vehicle using driving pattern recognition. *J. Dyn. Syst. Meas. Control* **2002**, *124*, 141–149. [[CrossRef](#)]
20. Son, H.; Park, K.; Hwang, S.; Kim, H. Design methodology of a power split type plug-in hybrid electric vehicle considering drivetrain losses. *Energies* **2017**, *10*, 437. [[CrossRef](#)]

21. Benford, H.; Leising, M. The lever analogy: A new tool transmission analysis. *SAE Pap.* **1981**, 810102. [[CrossRef](#)]
22. *Gears—Thermal Capacity—Part 2: Thermal Load-Carrying Capacity*; ISO/TR 14179-2:2001; International Organization for Standardization: Geneva, Switzerland, 2001.
23. Haka, R.J. Determination of efficiency (torque related losses) in planetary gearsets-generalized theory for simple and compound gearsets. In Proceedings of the ASME 2003 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Chicago, IL, USA, 2–6 September 2003; pp. 1085–1097.
24. Chen, J.Y.; Borgerson, J.B. Analytical and test evaluation of planetary gear train efficiency (torque related losses) with multiple power flow arrangements. In Proceedings of the ASME 2003 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Chicago, IL, USA, 2–6 September 2003; pp. 1057–1065.
25. SKF General Catalogue. Available online: <http://www.worldcat.org/title/skf-general-catalogue/oclc/225258275> (accessed on 13 January 2016).
26. Changenet, C.; Velex, P. A model for the prediction of churning losses in geared transmissions-preliminary results. *J. Mech. Des.* **2006**, *129*, 128–133. [[CrossRef](#)]
27. Cover, T.; Hart, P. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **1967**, *13*, 21–27. [[CrossRef](#)]
28. Short, R.; Fukunaga, K. The optimal distance measure for nearest neighbor classification. *IEEE Trans. Inf. Theory* **1981**, *27*, 622–627. [[CrossRef](#)]
29. Safavian, S.R.; Landgrebe, D. A survey of decision tree classifier methodology. *IEEE Trans. Syst. Man Cybern.* **1991**, *21*, 660–674. [[CrossRef](#)]
30. Muralidharan, V.; Sugumaran, V. Feature extraction using wavelets and classification through decision tree algorithm for fault diagnosis of mono-block centrifugal pump. *Energies* **2013**, *46*, 353–359. [[CrossRef](#)]
31. Hsu, C.; Lin, C. A comparison of methods for multiclass support vector machines. *IEEE Trans. Neural Netw.* **2002**, *13*, 415–425. [[PubMed](#)]
32. Kilian, Q.W.; John, B.; Lawrence, K.S. Distance metric learning for large margin nearest neighbor classification. *J. Mach. Learn. Res.* **2009**, *10*, 207–244.
33. Nadeau, C.; Bengio, Y. Inference for the generalization error. *Mach. Learn.* **2003**, *52*, 239–281. [[CrossRef](#)]
34. Department of Energy, U.S. Electric and hybrid vehicle research, development, and demonstration program; petroleum-equivalent fuel economy calculation. *Federal Regist.* **2000**, *65*, 113.



© 2018 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 (<http://creativecommons.org/licenses/by/4.0/>).