

Article

Powertrain Control for Hybrid-Electric Vehicles Using Supervised Machine Learning

Craig K. D. Harold * , Suraj Prakash * and Theo Hofman * 

Control Systems Technology Group, Mechanical Engineering Department, Eindhoven University of Technology, 5612 AZ Eindhoven, The Netherlands

* Correspondence: c.k.d.harold@student.tue.nl (C.K.D.H.); s.prakash@tue.nl (S.P.); t.hofman@tue.nl (T.H.); Tel.: +31-40-247-2827 (T.H.)

Received: 26 February 2020; Accepted: 11 May 2020; Published: 14 May 2020



Abstract: This paper presents a novel framework to enable automatic re-training of the supervisory powertrain control strategy for hybrid electric vehicles using supervised machine learning. The aim of re-training is to customize the control strategy to a user-specific driving behavior without human intervention. The framework is designed to update the control strategy at the end of a driving task. A combination of dynamic programming and supervised machine learning is used to train the control strategy. The trained control strategy denoted as SML is compared to an online-implementable strategy based on the combination of the optimal operation line and Pontryagin's minimum principle denoted as OOL-PMP, on the basis of fuel consumption. SML consistently performed better than OOL-PMP, evaluated over five standard drive cycles. The EUDC performance was almost identical while on FTP75 the OOL-PMP consumed 14.7% more fuel than SML. Moreover, the deviation from the global benchmark obtained from dynamic programming was between 1.8% and 5.4% for SML and between 5.8% and 16.8% for OOL-PMP. Furthermore, a test-case was conducted to emulate a real-world driving scenario wherein a trained controller is exposed to a new drive cycle. It is found that the performance on the new drive cycle deviates significantly from the optimal policy; however, this performance gap is bridged with a single re-training episode for the respective test-case.

Keywords: machine learning; powertrain control; automatic re-training; hybrid electric vehicles; dynamic programming; transmission; energy management

1. Introduction

In the late 1970s, the European Union (EU) established the link between air quality and automotive emissions, thereby setting in motion policies to reduce air pollution. In 1992, the Euro norm for passenger cars was introduced that set a ceiling for concentration of pollutants [1]. These norms are made more stringent with time [2] and enforces companies to adopt more efficient automotive powertrains. This can be illustrated with the growth in hybrid electric vehicle (HEV) market share and the estimated increase in sales over the next decade [3]. In line with the efforts to improve overall powertrain efficiency, significant strides have been made in transmission development. As a result, the continuously variable transmission (CVT) with a steel pushbelt is predicted to achieve an efficiency of 97% [4].

Based on these automotive trends and the superiority of HEV topology P2 over P1 [5], a P2 plug-in hybrid electric vehicle (PHEV) with a CVT is considered as the system, wherein an electric motor (EM) is directly connected to the drive shaft while the internal combustion engine (ICE) is connected in parallel via a clutch, depicted in Figure 1. An energy source, in this case a battery (BAT), supplies power to the EM. The combined power from the EM and the ICE is transmitted by the CVT to the wheels, with an intermediate speed reduction through a fixed differential gear. The addition of an

EM introduces a torque-split control variable that is strongly inter-coupled with the control of the transmission. Strategic control of the torque-split, transmission gear ratio, clutches, engine on/off, etc., can reduce the combined energy consumption of the EM and ICE. Several strategies exist that seek to minimize this energy consumption, and these strategies are reviewed in Section 1.1.

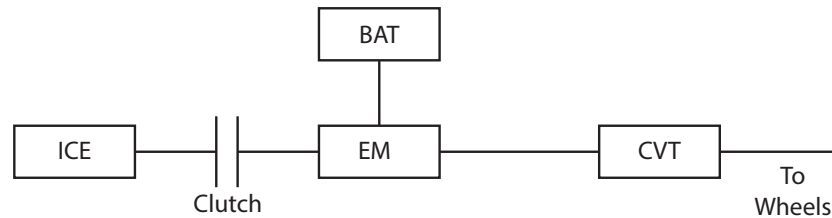


Figure 1. Schematic of a P2 hybrid layout with a CVT.

1.1. Literature Review

The clutch in the P2 configuration disengages the ICE from the powertrain to reduce engine drag; however, it is not considered as a control variable in this study. The clutch is assumed to be activated when there is no demand for ICE power and while activated the ICE is assumed to be idling. Therefore, in this study, only the gear ratio and torque split are considered as control variables and the state of charge of BAT (ζ) is considered as the state of the system.

Accounting for the system dynamics, boundary conditions, constraints on states and control spaces, the two-point boundary value problem can be solved using dynamic programming (DP) that guarantees optimality through an exhaustive search of all control and state grids based on the Bellman's principle of optimality [6,7]. With respect to the considered system, this implies that, for an a priori drive cycle with known boundary conditions, the optimal gear ratio and optimal torque split can be found offline using DP, thereby serving as a benchmark for all other control strategies. This optimal policy is drive cycle dependent, thereby rendering DP unsuitable for online implementation [8]. Online-implementable control strategies exist; however, they are sub-optimal and generally decouple the control variables. Therefore, these online implementable control strategies are reviewed separately as gear ratio control for CVTs in Section 1.1.1 and torque split control in Section 1.1.2.

1.1.1. Gear Ratio Control

CVTs offer a wide range of gear ratios. The classical optimal operating line (OOL) strategy is found to be the most economical method for the conventional drive-train [9]. Subsequently, a modified optimal operating line (M-OOL) strategy that accounts for the CVT system loss is shown to marginally improve over the OOL [10]. A similar method using the equivalent consumption of the electric energy to fuel energy is used to build a hybrid optimal operation Line (H-OOL) [11]. Apart from these standard rule-based methods, another method uses sub-optimal feedback controllers to approximate the optimal control policy and achieves almost optimal performance at a substantially reduced computational effort [12].

1.1.2. Torque Split Control

A comprehensive review of torque split control strategies is given in [13]. Common heuristic based control strategies can be divided into rule-based [14] and fuzzy-logic approaches [15–17]. Fuzzy-logic based approaches are preferred for their robustness and suitability to multi-domain, nonlinear, time-varying systems such as PHEVs [13]. Model predictive control (MPC) methods are also found to be computationally efficient for online implementation [18,19]. However, MPC is heavily dependent on the prediction accuracy and therefore online optimization methods based on Pontryagin's minimum principle (PMP) are preferred [20,21]. An added advantage of PMP is that it is governed by only one costate variable [22]. Similar control strategies, like equivalent consumption

minimization strategy (ECMS) first introduced in [23], utilizes the efficiency of the battery and the operating mode to determine the equivalence factor. This ECMS was adapted to PMP [24] and a comparative study with DP was done in [25]. In principle, a good estimation of the costate or equivalence factor can result in near optimal performance [26,27] and therefore online optimization methods are preferred to heuristics [28].

Meanwhile, machine learning (ML) techniques have gained popularity for their ability to control complex tasks by deriving patterns or rules from a data-set or through experience [29,30]. These techniques have also been extended to automotive applications, for example, drive cycle prediction [31], drive cycle recognition [32], training the torque split controller from DP using supervised machine learning (SML) [33], reinforcement learning (RL) for power distribution between the battery and the capacitor [34], etc. In certain tasks, controllers trained using ML have outperformed the controllers based on classical control theory [35].

In the case of supervisory control strategies for an HEV, certain learning based strategies are shown to be comparable to the commonly used control strategies [31]. For continuous-spaces, the actor–critic method was used for the power management in a PHEV [36]. A qualitative study on RL techniques on HEVs and PHEVs shows potential for RL controllers to replace rule based controllers [37]. Similarly, learning based techniques have been used to train neural networks to predict the driving environment and generate an optimal torque split, achieving fuel savings [33]. However, further improvement can be made by customizing the strategy to a specific driving behavior as driving behavior can influence vehicle fuel consumption [32,38]. This driving behavior could be based on the geographic-location, traffic congestion, personal style, etc. In practice, automotive companies offer driving modes such as eco, sport, normal, etc., to address these driver preferences but cannot fully encapsulate the driving behavior. Therefore, the potential of ML can be exploited to bridge the gap to the global optimal without human intervention and thus forms the basis of this research.

Research Question: How can ML be incorporated into supervisory powertrain control in order to adapt to a specific driving behavior?

1.2. Objectives

Apart from the main objective of minimizing overall energy consumption, ML can be extended to improve upon existing practices. In the existing practices for HEVs and conventional powertrains, the control strategy is tuned by experienced calibration engineers through iterative real-time vehicle tests (calibration time) before online implementation, resulting in a strategy that caters to the average driver. Therefore, the objectives (O1 and O2) of the study are to combat the drawbacks of conventional practices, i.e., calibration time and the inability to customize the control strategy to a specific driver. Furthermore, as suggested in literature [37], RL methods can improve fuel economy when compared to the rule based methods. However, these RL techniques come at the cost of learning time and this forms the third objective (O3), wherein the learning time must be minimized:

- O1: customize the control strategy to a specific driver,
- O2: reduce the time consumed for calibration,
- O3: improve learning efficiency.

In order to address O1, the controller must be able to account for the driver behavior. In this study, the vehicle velocity and its acceleration are considered as a representation of the driver behavior. In practice, the throttle position is considered; however, with a backward facing model, it is replaced with vehicle acceleration. In order to address O2, a learning algorithm must be present to adapt to this driver behavior. Several algorithms are available that can learn in real-time or from past data and are discussed in Section 1.1. Real-time learning algorithms like RL require an exploration phase (trial and error) to determine the optimal control respective to the vehicle state, which suggests that it needs to repeatedly encounter identical vehicle states in order to determine the best possible control. However, real-world driving will seldom encounter the identical vehicle states, i.e., identical combination of

velocity, acceleration, state of charge, etc. Hence, real-time learning solutions could require thousands of kilometers of driving in order to learn a good control strategy. Objective O3 is to improve learning efficiency thereby reducing training time. In order to address O3, it must be understood that for a given driving trajectory, with boundary conditions, there exists an optimal control policy. Therefore, utilizing this optimal control policy for training would reduce the training time drastically, as there is no requirement of an exploration phase. This would entail that the training occurs after the driving task is completed, in order to find and learn the optimal control policy.

1.3. Contributions

In this paper, a framework is presented that consists of three segments; in segment 1, the route planner analyzes the drive cycle and the end-point condition for the state of charge (ζ) is derived. Under the assumption that the drive cycle is representative of the driving behavior, O1 is addressed. In segment 2, based on the end-point condition on the state of charge (ζ_f), DP finds an optimal control policy for the a priori drive cycle. Finally, in segment 3, the input parameters from segment 1 and the optimal control policy from segment 2 are used to train a controller using SML algorithms. The absence of human intervention to learn a strategy that addresses O2 and utilizes the optimal control policy that addresses O3.

This trained controller is validated by comparing its performance in terms of fuel consumption, to the global optimal solution derived from DP and an online-implementable control strategy based on literature that uses a combination of OOL and PMP. It should be noted that DP in this study refers to the approximate dynamic programming, wherein the state and control spaces are discretized.

Organization: The paper is organized as follows, Section 2 describes the mathematical modeling of the system. Section 3 formulates the control problem and introduces the proposed framework to solve the problem. Section 4 elaborates on the experimental setup, discusses the results, and presents a test-case. Finally, Section 5 concludes this study and suggests future propositions.

2. Modeling of the System

In this section, the HEV powertrain components are described and the energy flow illustrated in Figure 1 is mathematically modeled. The system is modeled as backward quasi-static, which approximates the system to be static at a given time instance, depicted in Figure 2. Only longitudinal dynamics of the vehicle are considered, and it is assumed that the vehicle only moves forward or is stationary. The equations are taken from [39] and the parameter values are given in experimental design setup is Section 4. Energy losses within each component are taken from manufacturer specifications or modeled from test-bench data.

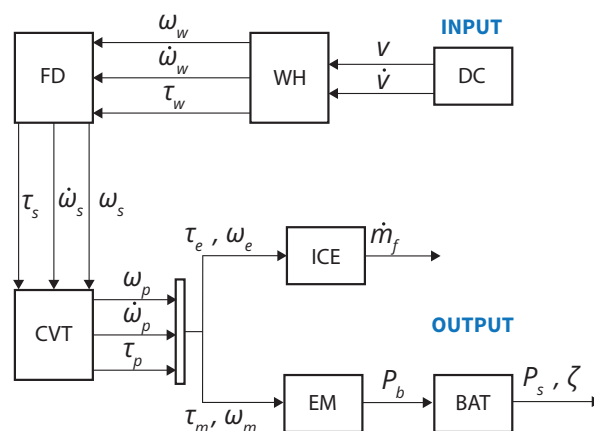


Figure 2. Model overview (DC—Drivecycle, WH—Wheel, FD—Fixed differential, CVT gearbox, ICE—Internal combustion engine, EM—Electric motor, BAT—battery).

Input parameters: The input to the system is the drivecycle, specifically the vehicle speed and the vehicle acceleration, recorded at 1 Hz. To physically achieve this acceleration at a given velocity, the resisting forces must be equal to the driving force applied by the wheel on the road. The resisting forces taken into account are aerodynamic drag (F_a), rolling resistance (F_r), gravity (F_g) and inertia (F_i), given respectively by Equations (1)–(4):

$$F_a = \frac{1}{2} \cdot \rho \cdot c_d \cdot A_f \cdot v^2 \quad (1)$$

$$F_r = m_v \cdot g \cdot \mu_r \cdot \cos(\alpha) \quad \text{for } (v > 0) \quad (2)$$

$$F_g = m_v \cdot g \cdot \sin(\alpha) \quad (3)$$

$$F_i = m_v \cdot (1 + m_r) \cdot \dot{v} \quad (4)$$

where ρ is the density of air, c_d is the aerodynamic coefficient, A_f is the frontal surface area, v is the vehicle speed, m_v is the mass of the vehicle, g is the acceleration due to gravity, μ_r is the static rolling coefficient, α is the road inclination, m_r is the mass of rotating parts, and \dot{v} is the acceleration of the vehicle.

Wheel: The driving force (F_w) required at the point of contact of the wheel with the road is the sum of the resisting forces. Subsequently, the torque of the wheel axle is calculated as a factor of the wheel radius. The wheel speed and wheel acceleration can be calculated from the vehicle speed and the vehicle acceleration respectively, as shown in Equations (6) and (7):

$$F_w = F_a + F_r + F_g + F_i$$

$$\tau_w = F_w \cdot r_w \quad (5)$$

$$\omega_w = \frac{v}{r_w} \quad (6)$$

$$\dot{\omega}_w = \frac{\dot{v}}{r_w} \quad (7)$$

where τ_w is the torque at the wheel axle, r_w is the radius of the wheel, ω_w is the rotational speed of the wheel, and $\dot{\omega}_w$ is the rotational acceleration of the wheel.

Differential: The fixed differential factors in the fixed gear ratio, resulting in the required torque and rotational speed at the secondary pulley:

$$\tau_s = \frac{\tau_w}{\gamma_{fd}} \quad (8)$$

$$\omega_s = \omega_w \cdot \gamma_{fd} \quad (9)$$

where γ_{fd} is the ratio of the fixed differential gear, τ_s is the torque at the secondary pulley and ω_s is the rotational speed of the secondary pulley.

Gearbox: The gearbox used in this study is a push-belt CVT type P920 [40], with an under-drive ratio of 0.416 and an over-drive ratio of 2.149. Transmission of speed and torque from the secondary pulley to the primary pulley is dependent on the selected gear ratio; this relation is shown in Equation (11). The loss in transmission of power is attributed to the mechanical loss and pumping loss, modeled from experimental data [40]. An example of the mechanical loss is illustrated in Figure 3 for seven different gear ratios for the vehicle speed of 40 kmph. These losses are measured at the test-bench for the full range of gear ratios at various vehicle speeds and stored in a lookup-table:

$$\tau_p = \tau_s \cdot \gamma_g + \tau_{lo} \quad (10)$$

$$\omega_p = \frac{\omega_s}{\gamma_g} \quad (11)$$

where τ_p is the torque at the primary pulley, γ_g is the selected gear ratio of the CVT, τ_{lo} is the torque loss within the CVT that is the sum of the mechanical and pumping losses, and ω_p is the rotational speed of the primary pulley.

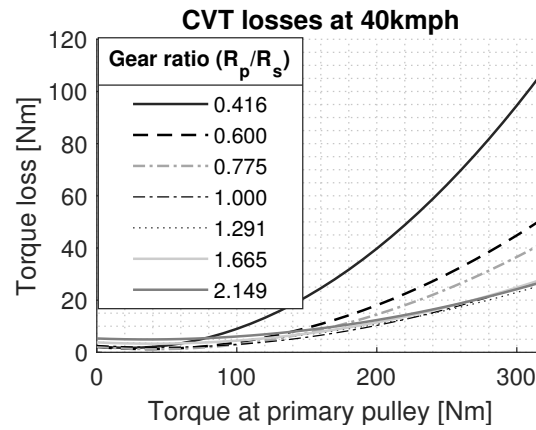


Figure 3. CVT mechanical loss at vehicle speed of 40 kmph [40].

Torque split: The torque at the primary pulley of the CVT is the combined torque delivered by the EM and ICE:

$$\tau_p = \tau_e + \tau_m \quad (12)$$

where τ_e is the ICE torque and τ_m is the EM torque

Engine: The ICE is a 1.6 L, 82-kW unit producing a maximum torque of 143-Nm and is taken from the Peugeot 206 model year 2005. The instantaneous ICE torque and speed are used to determine the fuel consumption from the brake specific fuel consumption (BSFC) map, depicted in Figure 4. The BSFC map expresses the fuel consumed in [g/kWh] that is taken from the manufacturer's specification and is converted to [g/s] using Equation (13):

$$\dot{m}_f = \begin{cases} \frac{\tau_e \cdot \omega_e \cdot BSFC(\tau_e, \omega_e)}{3600 \cdot 1000} & \text{if } \tau_e > 0 \\ m_{f,idle} & \text{if } \tau_e \leq 0 \end{cases} \quad (13)$$

where \dot{m}_f is the instantaneous fuel mass flow in [g/s], BSFC is the fuel consumed in [g/kWh], τ_e is the ICE torque, ω_e is the ICE rotational speed, and $m_{f,idle}$ is the fuel consumption at idling.

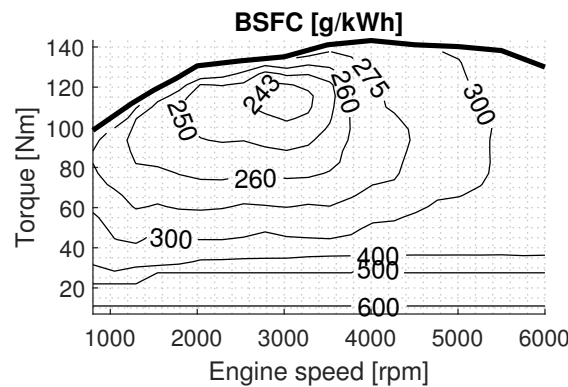


Figure 4. Engine BSFC map.

Electric motor: The 30-kW permanent magnet EM is taken from the 1999 Toyota Prius PHEV (Aichi, Japan). The efficiency of the EM ($\eta_m(\tau_m, \omega_m)$) can be determined from the efficiency map in

Figure 5 and the output power of the battery (P_b) can be calculated as in Equation (14). The data for the efficiency map are taken from test bench measurements [41]:

$$P_b = \begin{cases} \tau_m \cdot \omega_m \cdot \eta_m(\tau_m, \omega_m) & \text{generating} \\ \frac{\tau_m \cdot \omega_m}{\eta_m(\tau_m, \omega_m)} & \text{motoring} \end{cases} \quad (14)$$

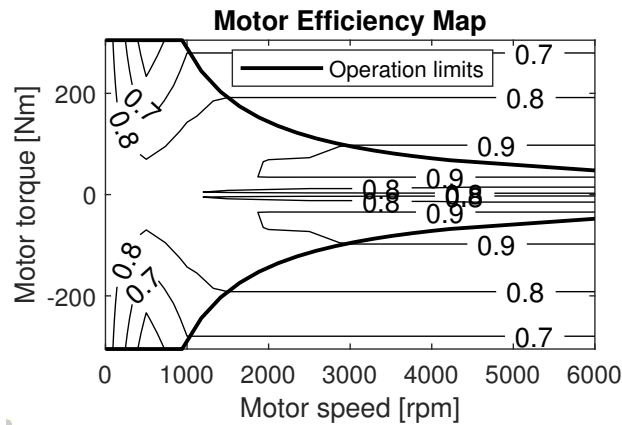


Figure 5. Motor efficiency map [41].

Battery: The 288-V and 6-Ah nickel metal hydride (NiMH) battery pack is taken from the Toyota Prius 2000 model. It is modeled as a voltage input with an internal resistance for simulation and as an equivalent resistance circuit to achieve convexity in the Hamiltonian. The test bench measurements from the Insight battery pack are scaled up to the Prius battery pack [42]. The coulombic efficiency (η_c) is assumed to be 0.905 for charging and discharging. The current within the battery can be calculated from the output power of the battery, and is given in Equation (15):

$$I_b = \frac{\left(U_{oc} - \sqrt{U_{oc}^2 - 4 \cdot \Omega \cdot P_b} \right) \cdot \eta_c}{2 \cdot \Omega} \quad (15)$$

Subsequently,

$$\zeta(t) = \zeta(t-1) - \frac{I_b}{Q_0 \cdot 3600}, \quad (16)$$

where I_b is the battery current, U_{oc} is the open circuit voltage, Ω is the internal resistance of the battery circuit, P_b is the power output of the battery, η_c is the coulombic efficiency of the battery, ζ is the state of charge, t is the time instance, and Q_0 is the nominal battery capacity.

With the system modelled, the next section describes the formulation and solution of the control problem.

3. Methodology

In this section, the control problem is formulated in Section 3.1. Subsequently, the solution for this control problem using the proposed SML framework is described in Section 3.2. In order to evaluate the performance of the proposed SML framework, conventional solutions such as DP is elaborated upon in Section 3.3 and an online implementable control strategy OOL-PMP is introduced and described in Section 3.3.

3.1. Control Problem

The objective of the control problem is to minimize energy consumption over the drive cycle, while satisfying the physical constraints of the system. Considering the practical application of a PHEV, the boundary conditions for state of charge (ζ_i, ζ_f) is fixed for a given drive cycle and the fuel consumption (m_f) is minimized:

$$J = \int_{t_0}^{t_f} \dot{m}_f dt \quad (17)$$

where t_0 is the initial time and t_f is the final time.

The control problem can be formulated as:

$$\min_{x,u} J(x,u)$$

subject to

$$\begin{aligned} h_1 &:= \dot{x}(t) - f(x(t), u(t), t) = 0 \\ h_2 &:= \zeta_i - \zeta(t_0) = 0 \\ h_3 &:= \zeta_f - \zeta(t_f) = 0 \\ h_4 &:= \gamma_{g,min} - \gamma_g(t_0) = 0 \\ g_{1,2} &:= \gamma_{g,min} \leq \gamma(t) \leq \gamma_{g,max} \\ g_3 &:= P_e(t) - P_{e,max} \leq 0 \\ g_{4,5} &:= \omega_{e,min} \leq \omega_e(t) \leq \omega_{e,max} \\ g_{5,6} &:= P_{m,min} \leq P_m(t) \leq P_{m,max} \\ g_{7,8} &:= \omega_{m,min} \leq \omega_m(t) \leq \omega_{m,max} \\ g_{9,10} &:= I_{b,min} \leq I_b(t) \leq I_{b,max} \\ g_{11,12} &:= \zeta_{min} \leq \zeta(t) \leq \zeta_{max} \\ g_{13,14} &:= -1 \leq u_{ts}(t) \leq 1 \\ g_{15} &:= -p(t_0) \leq 0 \end{aligned}$$

State space; $x = \{\zeta\}$ where $\zeta = [\zeta_{min}, \zeta_{max}] \subset R$ with $\zeta_{min} = 0.1$ (10% of battery capacity) and $\zeta_{max} = 0.9$ (90% of battery capacity). Control space; $u = \{u_{ts}, u_g\}$ where $u_{ts} = [-1, 1] \subset R$ and $u_g = [\gamma_{g,min}, \gamma_{g,max}] \subset R$ with $\gamma_{g,min} = 0.416$ and $\gamma_{g,max} = 2.149$. $p(t_0)$ is the costate that satisfies the boundary conditions for PMP.

The following subsection introduces the machine learning framework that is proposed as a solution for the control problem.

3.2. Solution Using Supervised Machine Learning

A few comments are in order, and it is assumed that a robust baseline strategy exists while training data are being accumulated. Secondly, in order to satisfy the objective (O1), the controller training is performed from scratch. Thirdly, in the specified framework, training occurs on completion of the driving task. Therefore, it is assumed that the vehicle is equipped with sufficient memory to store vehicle states.

Proposed Framework: The framework is divided into three segments, namely, Route Planner (RP), Dynamic Programming (DP), and Supervised Machine Learning (SML). The flow of events and parameters are illustrated in Figure 6:

- **Route Planner** records the drive cycle and the initial condition (ζ_i) for the respective drive cycle. The velocity trajectory depicted in the route planner segment in Figure 6 is an example of the recorded drive cycle. Based on this drive cycle, an end-point condition ζ_f is determined. In this study, ζ_f is calculated assuming 1.1% battery charge is available for the distance of 1 km. The assumption is made based on the average driving and charging cycles of HEVs in the Netherlands [43]. The requirement of the route planner is to set the boundary condition for the a priori drive cycle. There are more sophisticated planners based on traffic congestion, terrain, charging stations, etc. but do not add value to this study, hence neglected.
- Secondly, **Dynamic Programming** solves the two-point boundary value problem satisfying ζ_f resulting in the optimal control policy (u_{ts}^*, u_g^*) and optimal state trajectory (ζ^*, γ_g^*), for the given drive cycle. The discretized state and control spaces are elaborated in Section 3.3.
- Thirdly, **Supervised Machine Learning** segment develops a control strategy by mapping the input parameters from the drive cycle to the optimal control policy from DP, using SML algorithms. The rules derived from this mapping represent the control strategy and make predictions for a new input as shown in Figure 6. No universal algorithm exists to model the system; therefore, an SML algorithm is selected based on an exhaustive search. The SML algorithm is selected with a five-fold cross validation based on its accuracy of predictions, deviation of false predictions from the optimal value, and computational time for each prediction. The various algorithms are shown in Table 1 along with their prediction accuracy and the number of predictions the algorithm is capable of every second. Both characteristics are desired to be as high as possible and based on this, the selected algorithm is highlighted. Additionally, a memory module is used to store previously recorded data for the purpose of re-training the controller.

Table 1. Performance of Supervised Machine Learning Algorithms for the NEDC case.

Algorithm	Gear-Ratio Control		Torque-Split Control	
	Accuracy (%)	Prediction/Sec	Accuracy (%)	Prediction/Sec
Decision Tree-Fine	96.8	6700	98.0	68,000
Decision tree-medium	94.3	24,000	96.6	100,000
Linear discriminant	70	16,000	78.7	75,000
Quadratic discriminant	78.0	30,000	-	-
SVM-cubic	93.4	6500	94.9	3800
SVM-fine Gaussian	96.5	3500	93.8	3200
KNN-fine	94.8	15,000	93	41,000
KNN-medium	94.6	12,000	90.9	27,000
KNN-cubic	94.5	12,000	91.1	39,000
KNN-weighted	97.0	18,000	93.3	39,000
Ensemble-bagged trees	97.0	3300	98.0	7800
Ensemble boosted trees	96.7	3300	97.3	9100

The learning algorithms used for individual control strategies are discussed in Section 3.2.1 and the parameters with which the algorithm achieved the accuracy and prediction speed highlighted in Table 1 are introduced.

Proposed framework

Craig Harold

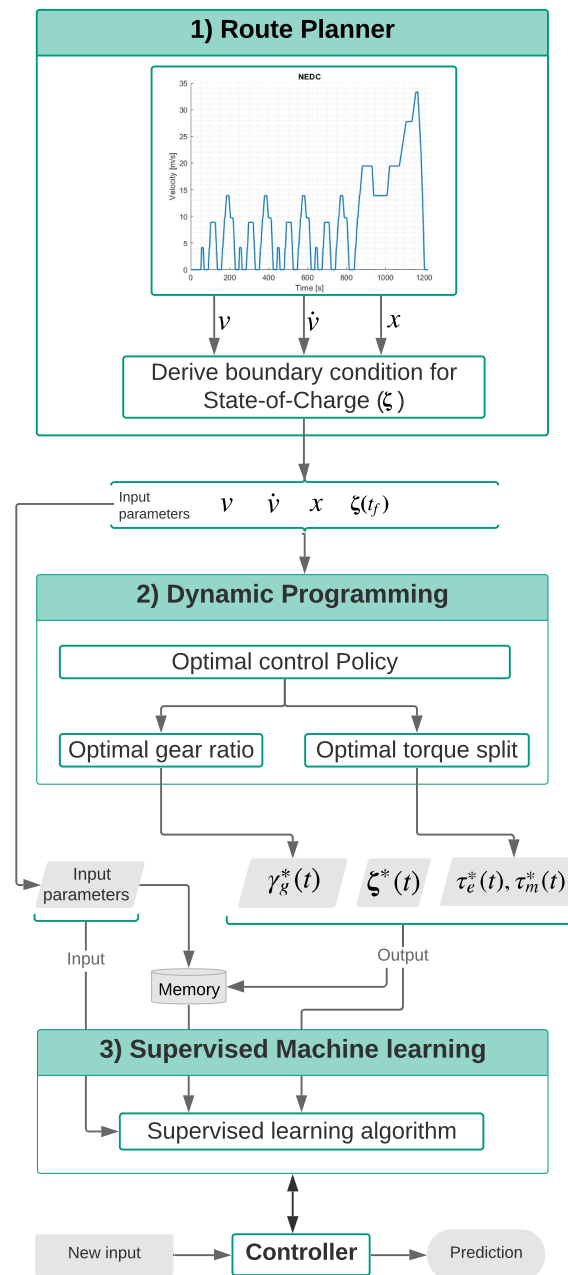


Figure 6. Proposed framework.

3.2.1. Supervised Machine Learning Algorithms

Decision Tree: As the name suggests, decision tree (DT) builds a model for the data to go from observation to prediction through branches. It is representative of a root system beneath a tree and is also representative of human decision-making. Each node represents binary logic and filters down to a

prediction based on this set of binary logic gates. The nodes of the tree are split based on impurity gain (δI), given by Equation (18):

$$\delta I = P(T)i_t - P(T_L)i_{t_L} - P(T_R)i_{t_R} \quad (18)$$

where $P(T)i_t$ is the probability of the splitting candidate or node t in the set of all observations T , $P(T_L)i_{t_L}$ is the probability that left child node (t_L) is present in the left observation set T_L and $P(T_R)i_{t_R}$ is the probability that the right child node (t_R) is present in the right observation set (T_R). In essence, a node is selected and all the observations are partitioned at the node. The impurity gain checks the number of instances of a class that are common on both sides of the partition, thereby the impurity of the class.

For this study, the primary pulley speed (ω_p), the torque demand at the primary pulley (τ_p), and the state of charge of the battery (ζ) were used as input features, and the optimal torque split was the desired output from the DT. The properties of the decision tree used for training are as follows: maximum number of splits was set to 100 and k-fold cross validation is set to 5. An example with relevance to the system model is depicted in Figure 7, wherein the decision tree is trained for torque split control, but limited to 10 nodes. It is intuitive that, for the negative torque demand (power flow from the wheels to the energy source), the resulting torque split is closer to 1 indicating complete regeneration.

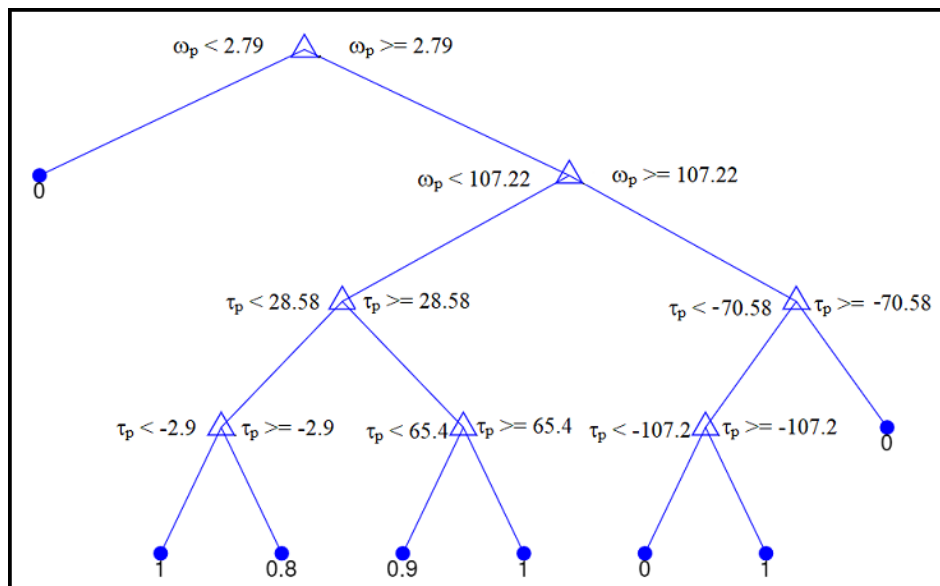


Figure 7. Decision tree trained for torque split control (u_{ts}) with 10 nodes.

K-Nearest Neighbors: The K-Nearest Neighbor (KNN) algorithm is non-parametric, i.e., no model is fitted to the data and all the work is done when a prediction is required. In principle, the KNN algorithm takes a vote of the closest neighbors to predict an output. The 'K' in KNN represents the number of neighbors to consider. Therefore, the 'K' should be an odd number to ensure a majority in the vote. KNN is used as the gear ratio control algorithm, wherein the input features are the vehicle speed (v) and vehicle acceleration (\dot{v}) and the desired output is the optimal gear ratio (γ_g^*). It must be noted that \dot{v} is used as a feature since the drivecycles considered do not include elevation profiles. In case of non-horizontal drivecycles, the torque required at the secondary pulley (τ_s) will substitute \dot{v} and in case of a forward facing model or practical applications, the throttle input will substitute \dot{v} . The properties of the KNN algorithm used are as follows; the closest neighbors are determined by the Euclidean distance, the number of votes accounted for is 7, equal weighting given to all neighbors and the k-fold cross validation is set to 5. It must be noted that to ensure effective learning with

the limited available data points from each drive cycle, the CVT was discretized into seven equally spaced classes for the SML case—ergo limiting the performance of the SML control strategy. However, with the abundance of data from real-world driving, this limitation can be overcome and in turn the full potential of the CVT can be exploited.

In order to evaluate the performance of the proposed framework, the SML controller is compared to conventional solutions that are elaborated in Section 3.3.

3.3. Solutions Using Classical Control

To validate the SML controller, its performance is compared to the global benchmark set by DP and the online implementable benchmark set by OOL-PMP.

Global Benchmark, DP: DP results in the global optimal control policy for the a priori drive cycle with boundary conditions ζ_i and ζ_f . A cost matrix is built with all possible state-action combinations at each timestep through the drive cycle. All infeasible states or actions are penalized with a high cost. Finding a path with minimal cost through the cost matrix results in the optimal control policy.

In order to build the cost matrix, the state and control space is discretized. The CVT is discretized into 100 gear ratios to exploit the full potential of the CVT. The torque split control ($u_{ts} \in [-1, 1]$) is discretized into intervals of 0.05, and the implication of the torque split variable is shown in Equations (19) and (20). For cases where $\omega_p > 0$,

$$\tau_e = (1 - u_{ts}) \cdot \tau_p \quad (19)$$

$$\tau_m = (u_{ts}) \cdot \tau_p \quad (20)$$

where ω_p is the rotational speed of the primary pulley and τ_p is the torque required at the primary pulley. The state space $x = \{\zeta\}$ is discretized as follows: $\zeta \in [0.1, 0.9]$ in intervals of 0.005 and $\gamma_g \in [0.416, 2.149]$ discretized to 100 ratios. The time interval of 1 s is chosen since the difference in ζ is very small for any smaller intervals of time.

Online Benchmark, OOL-PMP: The combination of OOL-PMP is based on a decoupled approach from the literature review in Section 1.1. The OOL is constructed by using the most efficient points of ICE operation over the entire ICE speed range and is used to control the gear ratio (u_g), while the PMP method is used to control the torque split (u_{ts}). Since the system is modeled as backward facing, the power required at the primary pulley (P_p) is estimated and subsequently a gear ratio is selected by OOL. It is counter-intuitive to use a torque split controller in combination with OOL, since OOL inherently determines the operating point of the ICE and ergo the operating point of the EM based on the power request at the primary pulley. However, to ensure that the torque split is optimal and online-implementable, a PMP approach is used to control the torque split. The formulation of the Hamiltonian and application of PMP is taken from [22] and given in Equation (21). The losses in the EM are modelled as a second degree polynomial, while the ICE losses and BAT open-circuit voltage are approximated by a linear fit. The control variable chosen is the power of the battery (P_b):

$$H = F - p \cdot \dot{x} = P_f + p \cdot P_s \quad (21)$$

where P_f is the fuel power, x is the state of charge (ζ), $\dot{x} = -P_s$ is the time derivative of ζ , and (p) is the costate:

$$H = \gamma_{p_1}(P_p - \frac{-\gamma_{m_1} + \sqrt{\gamma_{m_1}^2 + 4 \cdot \gamma_{m_2} \cdot (P_b - \gamma_{m_0})}}{2 \cdot \gamma_{m_2}}) + \gamma_{p_0} \dots \dots + p \cdot \frac{U_{oc} \cdot (U_{oc} - \sqrt{U_{oc}^2 - 4 \cdot \Omega \cdot P_b})}{2 \cdot \Omega} \quad (22)$$

where P_p is the power required at the primary pulley of the CVT, $(\gamma_{p_1}, \gamma_{p_0})$ are the coefficients of the linear fit that models the ICE losses, $(\gamma_{m_2}, \gamma_{m_1}, \gamma_{m_0})$ are the coefficients of the second degree polynomial

used to model the EM losses, U_{oc} is the open-circuit voltage of the BAT, and Ω is the resistance of the BAT.

The necessary conditions of PMP are as follows:

$$\frac{\partial H}{\partial P_b} = 0$$

$$\frac{\partial H}{\partial \zeta} = \dot{p}$$

Solving the first condition results in the optimal P_b^* , shown in Equation (23):

$$P_b^* = \frac{U_{oc}^2 (\gamma_{p1}^2 - p^2 \cdot \gamma_{m1}^2 + 4 \cdot p^2 \cdot \gamma_{m2} \cdot \gamma_{m0})}{4 \cdot (\Omega \cdot \gamma_{p1}^2 + p^2 \cdot U_{oc}^2 \cdot \gamma_{m2})} \quad (23)$$

Solving this as an initial value problem using a bisection algorithm results in the initial costate $p(t_0)$ that satisfies the boundary conditions ζ_i and ζ_f . In order to solve this boundary value problem, the exact velocity and acceleration profile from the respective drivecycle are considered.

With the proposed SML solution described along with the conventional solutions in order to compare the performance of the SML control strategy, the following Section 4 discusses the results of the study.

4. Results

This section outlines the results obtained by solving the control problem defined in Section 3.1. The experiment setup is outlined in Section 4.1, results of which are discussed in Section 4.2 wherein the performance of the SML controller is compared to the OOL-PMP and DP strategies. Furthermore, a test-case is elaborated in Section 4.3 wherein the effects of re-training the SML controller are observed.

4.1. Experimental Design

As mentioned in Section 3.2, the real-world drive cycle of the user is used in the framework. However, for experimental purposes, the real-world driving data are replaced by standard drive cycles and each drivecycle is assumed to represent the respective individual driving behavior. The drive cycles NEDC, EUDC, FTP75-highway, WLTP, and JP10-15 mode are used in simulation. To make a valid comparison between the control strategies, the boundary conditions are fixed respectively for each drive cycle as per the assumption made in the RP in Section 3.2. The parameters for the system modeled in Section 2 are given in Table 2.

4.2. Numerical Results

The results summarized in Table 3 show the fuel consumption in liters per 100 kilometers and the percentage loss in fuel when compared to the optimal solution. In Table 3, the proposed machine learning solution is denoted as SML, the globally optimal solution is denoted as DP, and the online implementable solution is denoted as OOL-PMP. It is evident from Table 3 that the SML control strategy consistently performs better than the OOL-PMP control, thereby bridging the gap to DP. The OOL-PMP performs well with steady state drive cycles while deviates significantly from the optimal solution with instantaneous drive cycles. The NEDC case is elaborated to give a better understanding of the performance figures achieved in Table 3 by illustrating the differences in control strategy operation over the drivecycle.

NEDC Results: The NEDC case is elaborated with graphical illustrations for the comparison of controllers. Due to legibility concerns with graphical representations, the more prominent drive cycle, WLTP, is not illustrated. However, numerical results for all the drive cycles are tabulated in Table 3. Figure 8 illustrates the working of the gear ratio controllers over the complete drive cycle while Figure 9

showcases the differences in the control strategies over the last quarter of the drive cycle. Subsequently, ICE and EM operating points are illustrated in Figures 10 and 11, and the resulting fuel consumption is illustrated in Figure 12. It is evident from this comparison that the SML controller almost perfectly tracks the DP control strategy, hence resulting in a 2.6% loss in optimality for the NEDC. While the OOL-PMP controller significantly deviates from the optimal control and results in 11.5% increase in fuel consumption.

Table 2. Vehicle parameters.

Parameter	Symbol	Value	Unit
Base vehicle weight	m_v	1300	kg
Wheel radius	r_w	0.316	m
Final drive ratio	γ_{fd}	4.695	-
Air drag coefficient	c_d	0.36	-
Rolling resistance coefficient	μ_r	0.01	-
Air density	ρ	1.18	kg/m ³
Frontal area	A_f	2.5813	m ²
Mass of rotating parts	m_r	0.05	-
Gravitational constant	g	9.81	m/s ²
CVT under-drive ratio	$\gamma_{g,min}$	0.416	-
CVT over-drive ratio	$\gamma_{g,max}$	2.149	-
Max. Engine power	$P_{e,max}$	82	kW
Max. Electric motor power	$P_{m,max}$	30	kW

Table 3. Fuel consumption in liters per 100 kms.

	DP	OOL-PMP	SML
NEDC (11.7 km)	4.25	4.74 (+11.5%)	4.36 (+2.6%)
EUDC (7.0 km)	4.68	4.95 (+5.77%)	4.93 (+5.34%)
JP10-15 (4.2 km)	4.05	4.33 (+6.9%)	4.21 (+3.8%)
FTP (17.7 km)	3.88	4.53 (+16.75%)	3.95 (+1.8%)
WLTP (23.3km)	4.75	5.31 (+11.8%)	4.84 (+1.9%)

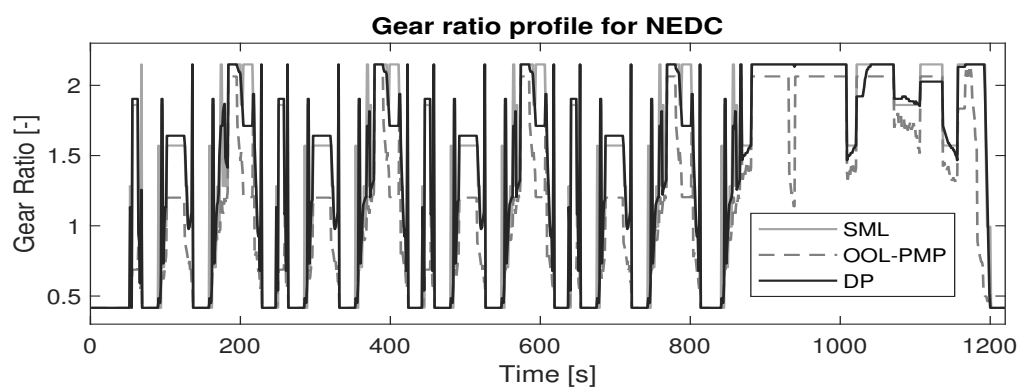


Figure 8. Gear Ratio Controller comparison on the NEDC.

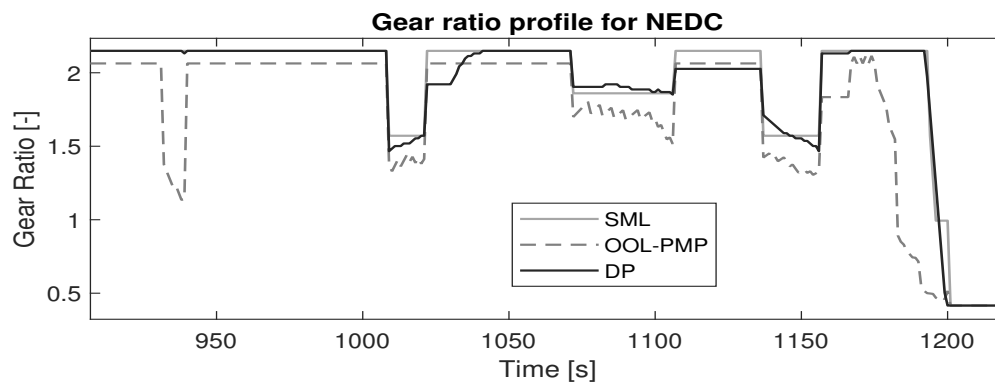


Figure 9. Gear ratio controller comparison over the last quarter of the NEDC.

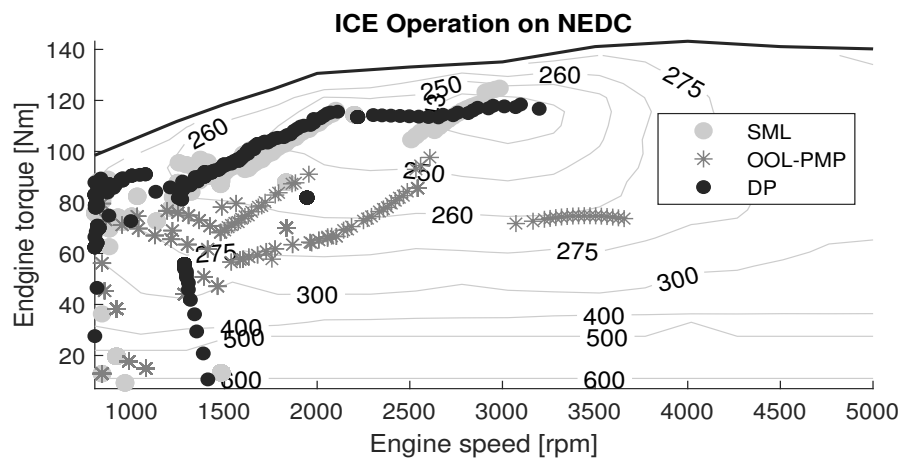


Figure 10. Engine operating points on the NEDC.

4.3. Test Case

Evident from Section 4.2, the near perfect tracking by the SML controller is dependent on the quality of the dataset. Additionally, the good performance is also attributed to the fact that the trained controller was tested with the same input drive cycle. However, it seldom occurs wherein the identical training conditions are experienced in practice. Therefore, it is critical for the proposed framework to efficiently adapt the control strategy to the newly experienced drivecycle. A test case is conducted wherein the control strategy trained with the EUDC is utilized on the FTP-75 (representing the newly experienced drivecycle), and the effect of re-training is observed. It is evident from Figure 13a that the SML controller trained with EUDC largely deviates from the optimal control strategy. In line with the principle of the proposed framework to adapt to driving behavior, the control strategy was then re-trained with the FTP-75 drivecycle—thereby significantly bridging the gap of SML control strategy to the optimal strategy with a single episode of re-training. Further re-training of the control strategy improved the performance and is illustrated in Figure 14a,b. However, as mentioned earlier, the exact combination of vehicle states and driving conditions is not likely to reoccur in practice and therefore the effect of re-training with a single episode is more relevant to real-world application. It is important to note that, in this test case, the new drivecycle forms a significant portion of the training dataset and hence resulted in a drastic change; this is analogous to early stages of re-training in the real-world. Intuitively, a less drastic change in the control strategy occurs when the new drivecycle is a small fraction of the training dataset, thereby capturing only the essence of the new drivecycle while maintaining the desired control from the previous learning episodes.

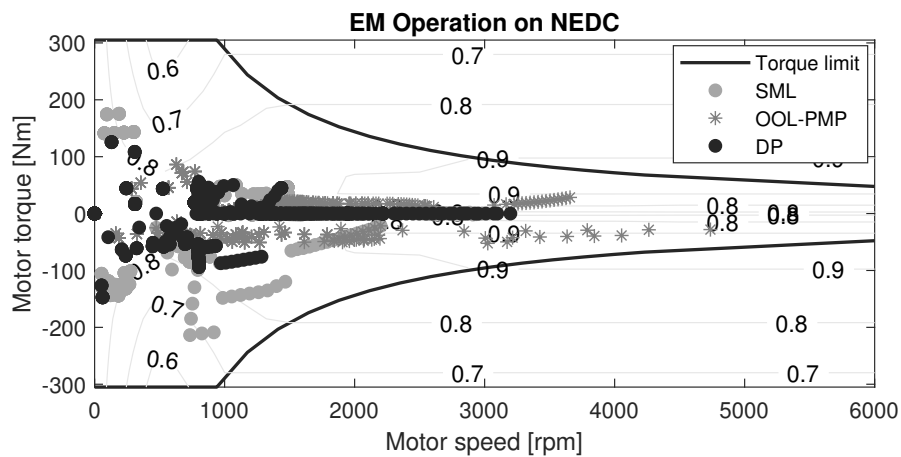


Figure 11. Electric Motor operating points on the NEDC.

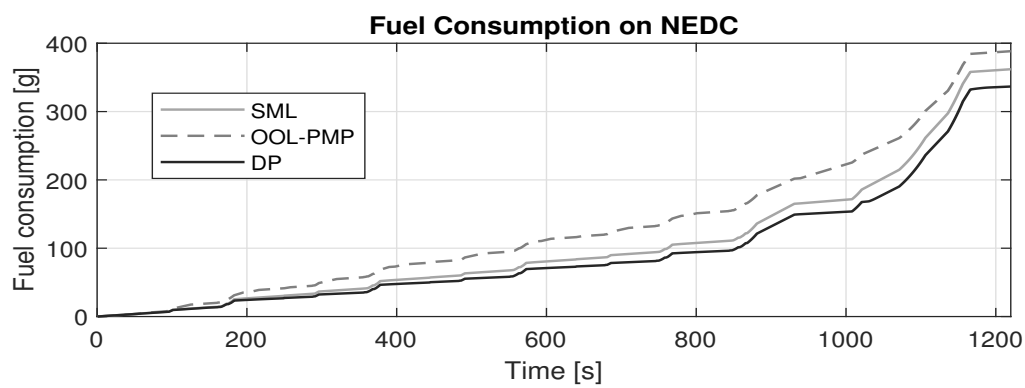
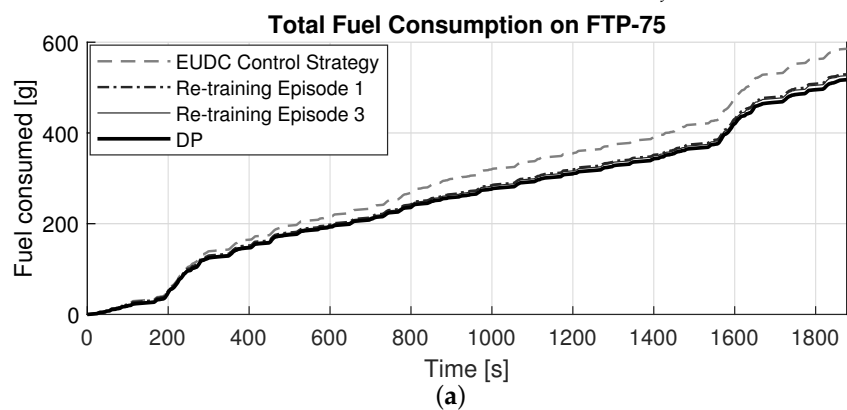
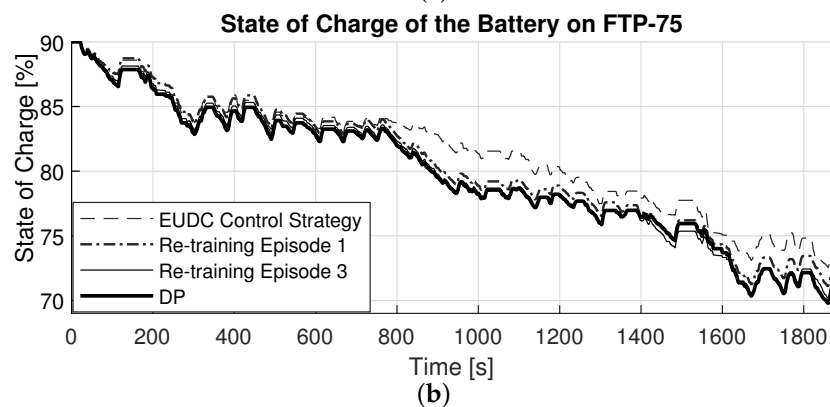


Figure 12. Fuel consumption comparison for identical ζ_f on the NEDC.



(a)



(b)

Figure 13. Comparison between EUDC trained controller and re-trained controllers on the FTP75 drive cycle. (a) fuel consumption; (b) state-of-charge.

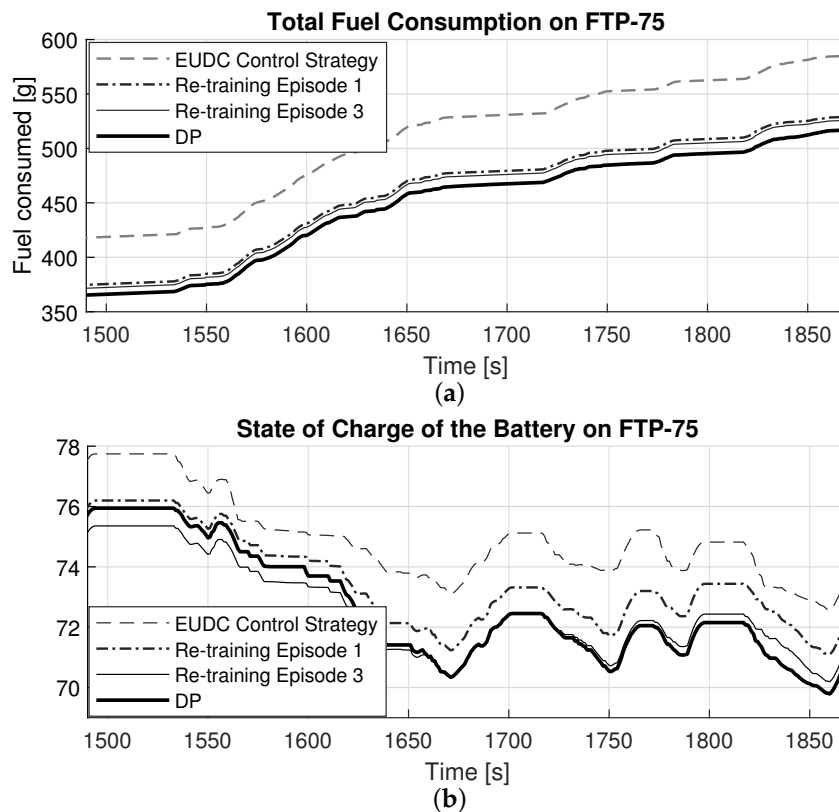


Figure 14. Controller comparison for the last 20% of FTP75 drive cycle. (a) fuel consumption; (b) state-of-charge.

5. Conclusions

The results show that the learned control strategy (SML) outperforms the conventional online implementable strategy (OOL-PMP), tested over several standard drivecycles in terms of fuel economy. This indicates that the proposed framework could effectively learn from past driving experiences to reproduce close to optimal results for the identical training conditions, i.e., the same drivecycle. However, since real-world driving will seldom experience identical training conditions, it is critical to study the efficacy of the proposed framework in adapting the control strategy to newly experienced drivecycles. Therefore, a test-case is conducted wherein the control strategy trained for EUDC is utilized on the FTP-75 and the effect of re-training the control strategy with the new drivecycle is observed. The EUDC strategy applied to the FTP-75 showed a large deviation from the optimal trajectory and a single episode of re-training significantly bridged the gap to optimality. Therefore, the proposed framework could be a viable alternative to the existing control strategies that adapts efficiently to a specific driver behavior.

Downsides of this proposed framework are briefly highlighted, the requirement of on-board computational power to perform DP is a real-world limitation but off-loading the computational burden is a prospective solution, since modern cars already record vehicle data and upload it to a cloud in the presence of internet connectivity. Furthermore, the algorithm properties, size of the dataset, and quality of dataset have shown to affect the performance and require more in-depth research.

As for future applications and improvements, the current framework using past data can be replaced with a predicted drive cycle based on geographic location, traffic congestion, terrain, weather, etc. that is easily available with current map technology. Additional states can be added, such as a slope sensor to consider the elevation of the road, in turn providing a more extensive control strategy capable of accounting for dynamic environments.

Author Contributions: Conceptualization, C.K.D.H., S.P. and T.H.; methodology, C.K.D.H. and T.H.; software, C.K.D.H.; writing—original draft preparation, C.K.D.H.; writing—review and editing, C.K.D.H. and T.H.; supervision, S.P. and T.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

Abbreviations

Acronym	Name
BAT	Battery
CVT	Continuously Variable Transmission
DC	Drive Cycle
DT	Decision Tree
DP	Dynamic Programming
ECMS	Equivalent Consumption Minimization Strategy
EM	Electric Motor
EU	European Union
EUDC	Extra Urban Driving Cycle
FD	Fixed Differential
FTP75	Federal Test Procedure 75
HEV	Hybrid Electric Vehicle
H-OOL	Hybrid Optimal Operating Line
ICE	Internal Combustion Engine
JP10-15	Japanese 10–15 driving cycle
KNN	K Nearest Neighbors
ML	Machine Learning
M-OOL	Modified Optimal Operating Line
NEDC	New European Driving Cycle
OOL	Optimal Operating Line
PHEV	Plugin Hybrid Electric Vehicle
PMP	Pontryagin’s Minimum Principle
RL	Reinforcement Learning
RP	Route Planner
SML	Supervised Machine Learning
SVM	Support Vector Machines
WH	Wheel
WLTP	Worldwide Harmonized Light Vehicle Test Procedure

References

1. Bennett, G. *Air Pollution Control in the European Community*; U.S. Department of Energy Office of Scientific and Technical Information: Oak Ridge, TN, USA, 1992.
2. Bielaczyc, P.; Woodburn, J.; Szczotka, A. An assessment of regulated emissions and CO₂ emissions from a European light-duty CNG-fueled vehicle in the context of Euro 6 emissions regulations. *Appl. Energy* **2014**, *117*, 134–141. [[CrossRef](#)]
3. IEA. *Global EV Outlook*; IEA: Paris, France, 2019.
4. Sluis, F.; Noll, E.; Leeuw, H. Key technologies of the pushbelt CVT. *Target* **2013**, *25*, 50.
5. Yang, Y.; Hu, X.; Pei, H.; Peng, Z. Comparison of power-split and parallel hybrid powertrain architectures with a single electric machine: Dynamic programming approach. *Appl. Energy* **2016**, *168*, 683–690. [[CrossRef](#)]
6. Bellman, R. Dynamic Programming. *Science* **1957**, *153*, 34–37. [[CrossRef](#)] [[PubMed](#)]
7. Bertsekas, D.; Tsitsiklis, J. *Parallel and Distributed Computation: Numerical Methods*; Prentice Hall: Englewood Cliffs, NJ, USA, 1989; Volume 23.
8. Kermani, S.; Delprat, S.; Guerra, T.; Trigui, R.; Jeanneret, B. Predictive energy management for hybrid vehicle. *Control Eng. Pract.* **2012**, *20*, 408–420. [[CrossRef](#)]

9. Bonsen, B.; Steinbuch, M.; Veenhuizen, P. CVT ratio control strategy optimization. In Proceedings of the 2005 IEEE Conference Vehicle Power and Propulsion, Chicago, IL, USA, 7 September 2005.
10. Ryu, W.; Kim, H. CVT ratio control with consideration of CVT system loss. *Int. J. Automot. Technol.* **2008**, *9*, 459–465. [\[CrossRef\]](#)
11. Kim, C.; NamGoong, E.; Lee, S.; Kim, T.; Kim, H. Fuel economy optimization for parallel hybrid vehicles with CVT. *SAE Trans.* **1999**, *108*, 2161–2167.
12. Pfiffner, R.; Guzzella, L. Optimal operation of CVT-based powertrains. *Int. J. Robust Nonlinear Control* **2001**, *11*, 1003–1021. [\[CrossRef\]](#)
13. Huang, Y.; Wang, H.; Khajepour, A.; Hongwen, H.; Ji, J. Model predictive control power management strategies for HEVs: A review. *J. Power Sources* **2017**, *341*, 91–106. [\[CrossRef\]](#)
14. Hofman, T.; Steinbuch, M.; Druten, R.; Serrarens, A. Rule-based energy management strategies for hybrid vehicle drivetrains: A fundamental approach in reducing computation time. *IFAC Proc. Vol.* **2006**, *39*, 740–745. [\[CrossRef\]](#)
15. Pusca, R.; Ait-Amirat, Y.; Berthon, A.; Kauffmann, J. Fuzzy-logic-based control applied to a hybrid electric vehicle with four separate wheel drives. *IEE Proc. Control Theory Appl.* **2004**, *151*, 73–81. [\[CrossRef\]](#)
16. Shi, G.; Jing, Y.; Xu, A.; Ma, J. Study and simulation of based-fuzzy-logic parallel hybrid electric vehicles control strategy. In Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications, Jinan, China, 11 December 2006; Volume 1, pp. 280–284.
17. Koo, E.; Lee, H.; Sul, S.; Kim, J. Torque control strategy for a parallel hybrid vehicle using fuzzy logic. In Proceedings of the Conference Record of 1998 IEEE Industry Applications Conference, Thirty-Third IAS Annual Meeting (Cat. No. 98CH36242), St. Louis, MO, USA, 12–15 October 1998; Volume 3, pp. 1715–1720.
18. Opila, D.; Wang, X.; McGee, R.; Grizzle, J. Real-time implementation and hardware testing of a hybrid vehicle energy management controller based on stochastic dynamic programming. *J. Dyn. Syst. Meas. Control* **2013**, *135*, 021002. [\[CrossRef\]](#)
19. Joševski, M.; Abel, D. Distributed predictive control approach for fuel efficient gear shifting in hybrid electric vehicles. In Proceedings of the 2016 European Control Conference (ECC), Aalborg, Denmark, 29 June–1 July 2016; pp. 2366–2373.
20. Zheng, C.; Xu, G.; Cha, S.; Liang, Q. Numerical comparison of ECMS and PMP-based optimal control strategy in hybrid vehicles. *Int. J. Automot. Technol.* **2014**, *15*, 1189–1196. [\[CrossRef\]](#)
21. Kim, N.; Cha, S.; Peng, H. Optimal Control of Hybrid Electric Vehicles Based on Pontryagin’s Minimum Principle. *IEEE Trans. Control Syst. Technol.* **2010**, *19*, 1279–1287.
22. Jager, B.; Keulen, T.; Kessels, J. *Optimal Control of Hybrid Vehicles*; Springer: Berlin, Germany, 2013.
23. Paganelli, G.; Delprat, S.; Guerra, T.; Rimaux, J.; Santin, J. Equivalent consumption minimization strategy for parallel hybrid powertrains. In Proceedings of the IEEE 55th Vehicular Technology Conference, VTC Spring 2002 (Cat. No. 02CH37367), Birmingham, AL, USA, 6–9 May 2002; Volume 4, pp. 2076–2081.
24. Serrao, L.; Onori, S.; Rizzoni, G. ECMS as a realization of Pontryagin’s minimum principle for HEV control. In Proceedings of the 2009 American Control Conference, St. Louis, MO, USA, 10–12 June 2009; pp. 3964–3969.
25. Serrao, L.; Onori, S.; Rizzoni, G. A comparative analysis of energy management strategies for hybrid electric vehicles. *J. Dyn. Syst. Meas. Control* **2011**, *133*, 031012. [\[CrossRef\]](#)
26. Musardo, C.; Rizzoni, G.; Guezennec, Y.; Staccia, B. A-ECMS: An adaptive algorithm for hybrid electric vehicle energy management. *Eur. J. Control* **2005**, *11*, 509–524. [\[CrossRef\]](#)
27. Onori, S.; Serrao, L. On Adaptive-ECMS strategies for hybrid electric vehicles. In Proceedings of the International Scientific Conference on Hybrid and Electric Vehicles, Malmaison, France, 6–7 December 2011; Volume 67.
28. Pisu, P.; Rizzoni, G. A Comparative Study Of Supervisory Control Strategies for Hybrid Electric Vehicles. *IEEE Trans. Control Syst. Technol.* **2007**, *15*, 506–518. [\[CrossRef\]](#)
29. Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; et al. Mastering the game of go without human knowledge. *Nature* **2017**, *550*, 354. [\[CrossRef\]](#)
30. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv* **2013**, arXiv:1312.5602.

31. Geulen, S.; Josevski, M.; Nellen, J.; Fuchs, J.; Netz, L.; Wolters, B.; Abel, D.; Ábrahám, E.; Unger, W. Learning-based control strategies for hybrid electric vehicles. In Proceedings of the 2015 IEEE Conference on Control Applications (CCA), Sydney, NSW, Australia, 21–23 September 2015; pp. 1722–1728.
32. 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* **2000**, *124*, 141–149. [[CrossRef](#)]
33. Murphey, Y.; Park, J.; Kiliaris, L.; Kuang, M.; Masrur, M.; Phillips, A.; Wang, Q. Intelligent Hybrid Vehicle Power Control—Part II: Online Intelligent Energy Management. *IEEE Trans. Veh. Technol.* **2013**, *62*, 69–79. [[CrossRef](#)]
34. Xiong, R.; Cao, J.; Yu, Q. Reinforcement learning-based real-time power management for hybrid energy storage system in the plug-in hybrid electric vehicle. *Appl. Energy* **2017**, *211*, 538–548. [[CrossRef](#)]
35. Juang, J.; Lin, R.; Liu, W. Comparison of classical control and intelligent control for a MIMO system. *Appl. Math. Comput.* **2008**, *205*, 778–791. [[CrossRef](#)]
36. Li, Y.; Hongwen, H.; Peng, J.; Zhang, H. Power Management for a Plug-in Hybrid Electric Vehicle Based on Reinforcement Learning with Continuous State and Action Spaces. *Energy Procedia* **2017**, *142*, 2270–2275. [[CrossRef](#)]
37. Hu, X.; Liu, T.; Qi, X.; Barth, M. Reinforcement Learning for Hybrid and Plug-In Hybrid Electric Vehicle Energy Management: Recent Advances and Prospects. *IEEE Ind. Electron. Mag.* **2019**, *13*, 16–25. [[CrossRef](#)]
38. Ericsson, E. Independent driving pattern factors and their influence on fuel-use and exhaust emission factors. *Transp. Res. Part D Transp. Environ.* **2001**, *6*, 325–345. [[CrossRef](#)]
39. Guzzella, L.; Sciarretta, A. *Vehicle Propulsion Systems: Introduction to Modeling and Optimization*; Springer: Berlin, Germany, 2013; Volume 1.
40. Vroemen, B. *Component Control for the Zero Inertia Powertrain*; Control Systems Technology: Osborne Park, WA, USA, 2001; pp. 85–89.
41. National Renewable Energy Laboratory. *NREL's Testing of Prius Japanese Motor at Unique Mobility 4/1999*; Technical Report; National Renewable Energy Laboratory: Golden, CO, USA, 2001.
42. National Renewable Energy Laboratory. *NREL Test Data From Testing Entire Insight Battery Pack Model Year 2000*; Technical Report; National Renewable Energy Laboratory: Golden, CO, USA, 2001.
43. Franke, T.; Krems, J. Understanding charging behavior of electric vehicle users. *Trans. Res. Part F Traffic Psychol. Behav.* **2013**, *21*, 75–89. [[CrossRef](#)]



© 2020 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/>).