Deep Reinforcement Learning for Advanced Energy Management of Hybrid Electric Vehicles

Roman Liessner, Christian Schroer, Ansgar Dietermann and Bernard Bäker
Dresden Institute of Automobile Engineering, TU Dresden, George-Bähr-Straße 1c, 01069 Dresden, Germany

Keywords: Energy Management, Deep Learning, Reinforcement Learning, Hybrid Electric Vehicle.

Abstract: Machine Learning seizes a substantial role in the development of future low-emission automobiles, as manufacturers are increasingly reaching limits with traditional engineering methods. Apart from autonomous driving, recent advances in reinforcement learning also offer great benefit for solving complex parameterization tasks. In this paper, deep reinforcement learning is used for the derivation of efficient operating strategies for hybrid electric vehicles. There, for achieving fuel efficient solutions, a wide range of potential driving and traffic scenarios have to be anticipated where intelligent and adaptive processes could bring significant improvements. The underlying research proves the ability of a reinforcement learning agent to learn nearly-optimal operating strategies without any prior route-information and offers great potential for the inclusion of further variables into the optimization process.

1 INTRODUCTION

Today, automobile manufacturers are confronted with major demands regarding the development of new drive trains. On the one hand, strict legislation and regulation requiring overall and continuous reduction of fuel consumption and emission levels of new vehicles. On the other hand, rising customer demands concerning vehicle dynamics, overall comfort and affordability. While purely electric vehicles often lack larger operation ranges and are still rather expensive due to high costs for production and development, hybrid electric vehicles (HEV) offer a good compromise to combine the benefits of conventional combustion engines and novel electric motors.

The use of an electric machine (EM) as a supplementary motor increases the degree of freedom of the driving unit and a so-called operating strategy has to be applied for the efficient coordination of both energy converters. Ideally, a variety of factors with effect on fuel consumption are taken into account by such a strategy. These factors range from the driver's influence through different driving styles and habits, environmental conditions like traffic, route and road information up to internal state information of the vehicle like fuel and battery levels. In general, these factors can be seen as highly stochastic, intercorrelated and dependent on the individual situation. For example, the driving style of a sporty driver operating a sports car in a large city is significantly different from the driving style of the same driver in an all-terrain vehicle in the mountains.

These scenarios are captured in velocity profiles, so-called driving cycles, which are not only used for calibration but also for certification of new vehicles. Formerly, emission levels were derived purely from deterministic cycles (e.g. "New European Driving Cycle") for which an efficient strategy could specifically be optimized. Nevertheless, the ability of these synthetic cycles to represent reality can and should be questioned as the variety of potential driving scenarios in real-world traffic is huge. For that reason, future certification tests (Real Driving Emissions) (European Commission, 2017) will be performed directly on the road rather than in fully controlled testing environments which poses major challenges for the manufacturers. Therefore, traditional approaches for the derivation of operating strategies with fixed rule-based strategies seem outdated, especially with inclusion of diverse driver, route and vehicle information. In order to bring significant fuel and energy savings beyond certification procedures to the actual customer with every-day use of the vehicle, new and innovative approaches are required for the energy management of HEVs.

In recent years, machine learning has increasingly gained momentum in the engineering context where highly non-linear problems have to be solved and

Liessner, R., Schroer, C., Dietermann, A. and Bäker, B.
Deep Reinforcement Learning for Advanced Energy Management of Hybrid Electric Vehicles.
DOI: 10.5220/0006573000610072
In *Proceedings of the 10th International Conference on Agents and Artificial Intelligence (ICAART 2018) - Volume 2*, pages 61-72
ISBN: 978-989-758-275-2

greater abstraction levels reduce valuable information. Reinforcement learning (RL) in particular offers significant benefit for planning and optimization tasks, mostly because an agent learns model-free, with great efficiency and through direct interaction with its environment. In contrast to traditional RL methods, which were bound to rather small and lowdimensional problems, the combination of RL and neural networks, called deep reinforcement learning, allows the application in very complex domains with highdimensional sensory input. In the context of autonomous driving for example, RL can play a vital role in fine-grained control of the vehicle in challenging environments (Mirzaei and Givargis, 2017)(Chae et al., 2017). Next to that, the energy management of modern vehicles appears to be a problem where state-of-the-art machine learning could dramatically improve current technology, bringing significant benefits to customers and the environment.
The main contributions of this paper are: (1) Derivation of a deep reinforcement learning framework capable of learning nearly-optimal operating strategies (2) The use of stochastic driver models for improved state generalization and preventing the strategy from overfitting. (3) Inclusion of the battery temperature with additional power limitation in to the optimization process.

The paper is structured into six sections. Section 2 establishes the fundamentals of hybrid vehicles, energy management and reinforcement learning. Section 3 introduces related work and a concrete formulation of the problem. In Section 4, the experimental setup for solving the energy management problem with RL is described, for which the results will be shown in section 5. Section 6 concludes the paper and gives a prospect of potential future work.

2 BACKGROUND

2.1 Hybrid Electric Vehicles

The term 'hybrid vehicle' is classified by the UN as vehicles which have at least two energy storages and two energy converters for locomotion (UN-ECE Transport Division, 2005). Today, the most widely spread implementation is the combination of a conventional internal combustion engine (ICE) with an electric machine. Biggest advantage of this combination is the ability to exploit vehicle deceleration for recuperation, meaning the conversion of braking energy into electric energy for recharging the electric energy storage. With the EM assisting the ICE in times of high loads up to 30% of fuel could be saved, especially in urban environments, where the efficiency of vehicles operated solely by ICE is typically rather low (Guzzella and Sciarretta, 2013).
If the combustion engine and the electric machine are both directly connected to the driving axle of the hybrid vehicle they form a parallel power train structure where the EM can be switched on and off as desired. The basic architecture of parallel hybrid drive train structures is shown in Figure 1.

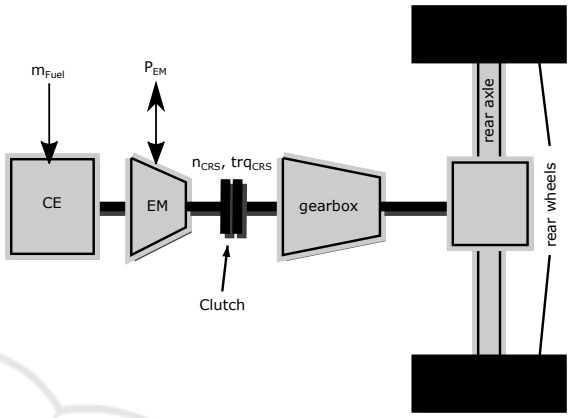

Figure 1: Structural scheme of a parallel mild hybrid.

Hybrid vehicles with a parallel power train are often designed as mild hybrids, meaning that the EM has a comparatively low power output against the ICE. In this case, the EM is mainly used for recuperation of braking energy and short-term support of the combustion engine in periods of high load. In contrast to plug-in hybrids, mild hybrids do not offer the opportunity to charge the electric storage externally and the entirety of the energy used for support of the ICE has to be gained from recuperation. Deliberately, purely electric driving with only the EM operated for locomotion of the vehicle is not intended (Guzzella and Sciarretta, 2013). Mild hybrids can be seen as a compromise between vehicles operated by a single combustion engine and full hybrids with the ability to drive fully electric for long routes, saving costs in the development and production of the vehicle as well as operational costs for fuel and energy.

2.2 Operating Strategy

Mainly, the control of HEVs can be split into two levels. The component level (low-level), which controls the components of the driving unit with traditional feedback-control procedures, and the supervisory level (high-level) for monitoring and control of any power flow within the vehicle. Latter one will also process any kind of vehicle and driving information (speed, torque, acceleration, slope), and thus can be described as the energy management

system (EMS). Primary goal of the EMS is deriving optimal control signals for the low-level in order to achieve best possible energy efficiency (Onori et al., 2016). The energy management is embedded in an overall operating strategy of the vehicle which, next to fuel saving, could also include any other aspects like vehicle dynamics or driving comfort as well as certain technical goals. For example, limitations of the maximum battery power could be desired in order to avoid damage or to extend the lifetime of the component.

Most derivation processes for energy management start with model-based simulations. Deriving an optimal strategy can then be described as a multi-goal optimization process. Various methods exist for solving this problem. On the one hand, global optimization methods like dynamic programming (DP) (Kirschbaum et al., 2002) which offer the chance to determine optimal solutions for any given driving cycle. However, since these methods require full knowledge of the route in advance and are computationally very expensive, the application in the vehicle for online control seams infeasible. In practice DP mostly serves for academic validation of other procedures. On the other hand there are local optimization methods like the Equivalent Consumption Minimization Strategy (ECMS) where fuel and electric energy used for driving is balanced through a weighting factor (Chasse and Sciarretta, 2011). Nevertheless, selection of an appropriate factor requires further effort for optimization with ECMS since it heavily depends on the state of the route being driven and the habits of the driver. Further details to traditional optimization methods for operating strategies can be found in (Guzzella and Sciarretta, 2013).

For adaptability towards different driver types, more innovative approaches are required. Next to stochastic dynamic programming shown in (Leroy et al., 2012) and (Tate et al., 2008), machine learning and especially reinforcement learning offer good concepts for the efficient inclusion of specific driving and driver information into the optimization process of operating strategies for hybrid vehicles.

2.3 Reinforcement Learning

The basic idea of reinforcement learning algorithms is strongly geared towards how humans or animals learn and can be described as a learning process by trial and error. The interactive nature makes RL particularly interesting for learning problems with no existing set of structured and labeled training examples.

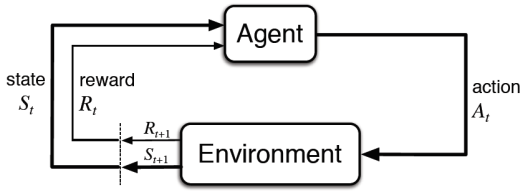

Figure 2: Scheme of the typical agent-environment-interaction for reinforcement learning (Sutton and Barto, 2012).

The fundamental structure of any RL algorithm is shown in Figure 2 and can be described as a structured interaction of an agent with its environment. At every discrete timestep t the agent is presented with a state s_t from the environment for which he must choose an action a_t. The agent determines his actions based on an internal policy π which maps an action to every observable state:

$$\pi : s \leftarrow a \tag{1}$$

For every chosen action the agent receives a reward r_{t+1} and a new state s_{t+1}. Goal of any RL algorithm is for the agent to adjust his policy in such a way that his return g_t, which is given as the weighted sum of the temporal rewards

$$g_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \tag{2}$$

is maximized at any given timestep. The discount factor $\gamma \in [0,1)$ indicates the relevance of future rewards. In general, the agents policy can be either stochastic or deterministic, latter will further be denoted with μ. In order for the agent to learn an optimal policy π_* which maximizes his return in the given environment, the problem statement has to fulfill the Markov-Property, meaning that at any point of time all of the environments historic state information is captured in the current state s_t and so any following state and future rewards solely depend on s_t and the agents chosen action a_t. A process satisfying the Markov-Property can be described as a Markov-Decision-Process (MDP) (Puterman, 2010). Since complete satisfaction of this condition for real-world problems is often unfeasible, a good approximation of an MDP is often sufficient for the applicability of reinforcement learning.

Different approaches exist for solving an MDP with RL, one of which is the concept of Temporal-Difference (TD) Learning. Here, the agent maintains an estimate of the achievable return which can iteratively be updated based on the truth found in the actually experienced state sequences and the received rewards during the interaction with the environment. One proven way of solving an MDP with TD-methods

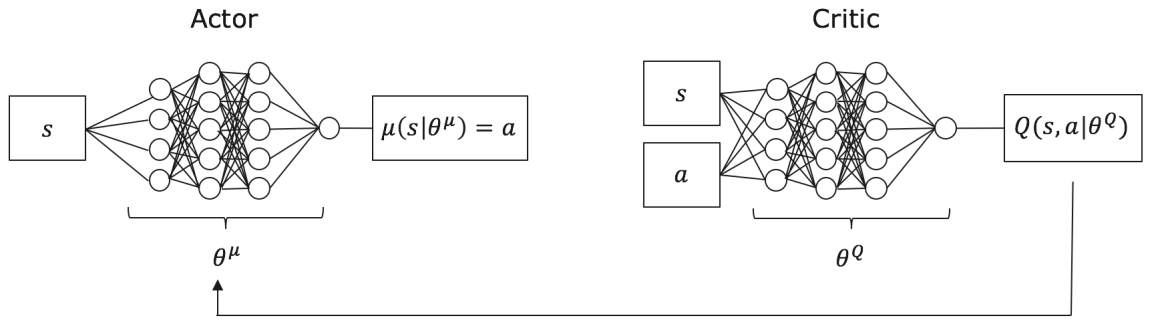

Figure 3: Basic structure of the DDPG actor-critic agent. Actor with parameters θ^μ, state s as input and action based on deterministic policy μ as output. Critic taking both state and chosen action of the actor as input and giving out an according Q-Value, based on parameters θ^Q. Update signal for higher rewards for actor as policy gradient, derived with the chain-rule of both networks (Timothy P. Lillicrap et al., 2015).

is Q-Learning where the agent estimates a Q-Value for every possible action, indicating the achievable future return when this action is chosen. After every interaction with the environment, an iterative update of the current estimation for the chosen action can be made by

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha\,[\tilde{y_t} - Q(s_t,a_t)] \tag{3}$$

with

$$\tilde{y_t} = \left[r_{t+1} + \gamma \max_{a_{t+1}} Q(s_{t+1},a_{t+1}) \right] \tag{4}$$

With a sufficiently small learning rate α, the iterative updates of the agents estimations are proven to converge towards the real Q-Values of the MDP. Knowledge of the Q-Values automatically results in following the optimal policy if the agent in every state picks the action with the highest Q-Value:

$$\pi_*(s) = arg\max_a Q(s,a) \tag{5}$$

which ultimately maximizes his return. Even though Q-Learning has proven to be very effective for rather small and low-dimensional control tasks, the need for discrete state and actions spaces limits the number of application options for real-world optimization problems.

2.4 Deep Reinforcement Learning

Deep reinforcement learning, as a combination of deep learning with artificial neural networks and the interactive learning structure of RL, has gained much attention in recent years. As the foundation of almost any major progress in artificial intelligence in the past decade, neural networks offer the essentials to make RL valuable for a whole new range of highdimensional real-world optimization problems. Hence, deep neural networks can be deployed as non-linear function approximators with trainable parameters θ^Q in the context of Q-learning, resulting in the Deep Q-Network (DQN) algorithm (Volodymyr Mnih et al., 2013). Instead of iteratively updating running estimates of the actions Q-values, the DQN is trained to output the correct values for any given state by minimizing the loss:

$$L(\theta^Q) = \left[y_t - Q(s_t,a_t \mid \theta^Q) \right]^2 \tag{6}$$

with

$$y_t = \left[r_{t+1} + \gamma \max_{a_{t+1}} Q(s_{t+1},a_{t+1} \mid \theta^{Q'}) \right] \tag{7}$$

For what was always believed to be inherently unstable, (Mnih et al., 2015) show effective learning of the DQN by introducing two main features: training of the network with uncorrelated minibatches of the past - called experience replay - and deriving target values y_t with a separate network with parameters $\theta^{Q'}$, called the target network. In this manner, a DQN-agent could be trained to achieve human level performance in many games of an Atari 2600 emulator only receiving unprocessed pixels as state-input and the raw score as a reward.

The processing of highdimensional state-vectors can be seen as a significant advantage for many engineering tasks where often data gathered from a large number of sensors has to be filtered elaborately for the use with conventional optimization methods. However, for many control tasks even bigger advantages are believed to arise from the use of continuous control parameters in order to avoid quality limiting discretization errors. Plain Deep-Q-Networks do not offer the chance of continuous output parameters. Thus, numerous evolutions of DQN have been

proposed in the last years, mostly categorized as policy gradient algorithms like the Deep Deterministic Policy Gradient (DDPG) (Timothy P. Lillicrap et al., 2015) or Trust Region Policy Optimization (TRPO) (John Schulman et al., 2015). A continuous variant of DQN is proposed in (Shixiang Gu et al., 2016) with the Normalized Advantage Function. Continuous policy gradient methods use an actor-critic architecture where an actor represents a currently followed policy with output of continuous action variables which are then evaluated by a second part called the critic. In case of DDPG both actor and critic are presented as deep neural networks, where the actor performance is evaluated through a deterministic policy gradient (David Silver et al., 2014) derived from the critic as a DQN. The DDPG architecture is shown in Figure 3.

3 REINFORCEMENT LEARNING FOR HEV CALIBRATION

3.1 Related Work

Compared to traditional and most commonly used methods for deriving efficient operating strategies in the automotive industry like DP, ECMS or fuzzy approaches, machine learning and in particular reinforcement learning present new opportunities with significant advantages. Most of all the agent would not require any prior information about the course of the route in order to decrease fuel consumption since he is fully trained to do so with only current state information.

In (X. Lin et al., 2014) a RL agent is given control over particular parameters of the control unit within the model of a HEV. Given a discrete state consisting of power demand, vehicle speed, and battery charging state the agent was able to determine a strategy to regulate battery output voltage and gear translation resulting in fuel savings up to 42%. In (C. Liu and Y. L. Murphey, 2014) the discrete state space of the constrained optimization is extended with specific trip information like the remaining traveling distance, which made the operating strategy even more efficient. In both cases traditional Q-learning methods were used for optimization which require discretization of the generally continuous state and control parameters within the vehicle.

Preliminary investigations (Patrick Wappler, 2016) have proven the high quality of the gained solutions of classic RL algorithms. The discrete representation of the results in a Q-table would also allow a straight forward approach for the integration in common energy management systems of HEV. However, the well-known curse of dimensionality exponentially increases the size of the table with inclusion of further variables which limits the problem to rather small state and action spaces in order to deal with restrictions concerning memory and computing time. Additionally, without the use of approximation or interpolation methods the agent is unable to value any state he has not processed during training. With severely different driving scenarios occuring in the real world considering traffic or driving styles, the derivation of a fully complete discrete representation is unfeasible. A rough discretization excludes valuable information within the optimization problem and limits the achievable fuel savings.

3.2 Problem Formulation

The central goal of the energy management is finding a control signal $action(t)$ for the minimization of the fuel consumption m_{Fuel} during a drive of time $t_0 \leq t \leq t_f$, expressed as the minimization of an integral value J:

$$J = \int_{t_0}^{t_f} \dot{m}_{Fuel}(action(t), t)\,\mathrm{dt} \tag{8}$$

wherein $\dot{m}_{Fuel}$ describes the fuel mass flow rate. The minimization of J is subject to physical limitations concerning performance measures, limitations of the energy storage and variance of the charging state of the battery (SOC). Thus, the energy management optimization problem is a temporally limited optimization problem with secondary and boundary conditions (Onori et al., 2016).

3.2.1 Boundary Conditions

Unlike plug-in hybrids, mild hybrids can not be charged externally so they must be operated in a charge-conserving way. Hence a boundary condition can be stated:

$$SOC(t_f) = SOC_{target} \tag{9}$$

which also allows for comparison of different solutions with an optimal solution found by dynamic programming. A desired target value could be:

$$SOC_{target} = SOC(t_0) \tag{10}$$

maintaining the initial charging state of the battery at the beginning of the driving cycle. However, for practical application a hard constraint does not seem necessary and ranging the terminal charging state $SOC(t_f)$ between threshold values is acceptable.

With a sufficiently small range, a deviation to the target value does not lead to interference of the vehicles functionality (Onori et al., 2016).

3.2.2 Secondary Conditions

Secondary conditions of the optimization problem can be stated for the state variables of the HEV as well as the control parameters. Limiting state variables like the *SOC* and the battery temperature ϑ_{bat} ensure safe use of the components, operation in regions of high efficiency and extended lifetime. Limitations of the control variables arise from physical limits of actuators and maximum performance (or power output *P*) of engines and battery.

Thus, it can be stated:

$$\begin{aligned} SOC_{min} \leq SOC(t) \leq SOC_{max}, \\ P_{bat,min} \leq P_{bat}(t) \leq P_{bat,max}, \\ \vartheta_{bat,min} \leq \vartheta_{bat}(t) \leq \vartheta_{bat,max}, \\ Trq_{x,min} \leq Trq_x(t) \leq Trq_{x,max}, \\ n_{x,min} \leq n_x(t) \leq n_{x,max} \end{aligned} \tag{11}$$

for

$$x = \text{ICE}, \text{EM}$$

where n and Trq denote rotational speeds and torque of the EM and ICE. The notations $(\cdot)_{min}$ and $(\cdot)_{max}$ represent the minimum and maximum of the respective entries at every point of time.

4 REINFORCEMENT LEARNING SETUP

The use of deep reinforcement learning requires the statement of the problem as the typical interaction of an agent with his environment. Concerning the energy management problem of HEV, the agent can be chosen to represent the control unit of the vehicle. Then the environment can include any outside part of the world containing information for the derivation of an efficient operating strategy. Here the environment is assembled by two models, one of the vehicle and one of the driver.

4.1 Vehicle Model

The mild hybrid power train, which is used for the vehicle model, is briefly described in section 2.2 and shown in Figure 1. Both energy converters are located on the crank shaft of the driving unit where the torque is split according to the operating strategy and physical limitations. The ICE gets fuel from a fuel tank and the EM is supplied with electric energy from a battery. The same battery stores the recuperated energy from vehicle deceleration, downhill driving or load point shifting of the combustion engine.

Since targets concerning speed and acceleration are specified by driving cycles, the vehicle model is orientated backwards with power demands calculated from the wheels to the driving unit with each intermediate component modelled individually. Figure 4 shows the signal flow of the vehicle model as a block diagram.

4.1.1 Vehicle Dynamics

With a predetermined speed v_{veh} and slope δ_{veh} from the driving cycle and the vehicle parameters par_{veh}, the required power for the vehicle P_{veh} can be determined by approximating the driving resistance caused by rolling friction and aerodynamic drag (Guzzella and Sciarretta, 2013):

$$P_{veh} = f(v_{veh}, \delta_{veh}, par_{veh}) \tag{12}$$

With the addition of the dynamic rolling radius of the wheel r, the required torque Trq_{whl} and the rotational speed at the wheels n_{whl} can be calculated to:

$$Trq_{whl} = \frac{P_{veh} \cdot r}{v_{veh}} \tag{13}$$

$$n_{whl} = \frac{v_{veh}}{2\pi \cdot r} \tag{14}$$

4.1.2 Consumption Model

An analytical description of chemical processes within the combustion engine is barely possible, hence the fuel consumption of the ICE was modelled empirically based on measured data from a power train test bench. With input of the speed and torque at the wheels, the selected gear G as well as the electric power of the electric motor $P_{EM,el}$ (derived by the chosen action of the agent), the model determines a fuel volume m_{Fuel} consumed in a time span of $\Delta t = 1s$:

$$m_{Fuel} = f(n_{whl}, Trq_{whl}, G, P_{EM,el}) \tag{15}$$

4.1.3 Electric Motor Model

The model of the EM determines the losses due to the conversion between electric and mechanical power. Based on the efficiency for a given crankshaft speed n_{crs} and the direction of the conversion, the empirical model outputs the power after losses:

$$P_{mech} = f(n_{crs}, P_{el}) \tag{16}$$

$$P_{el} = f(n_{crs}, P_{mech}) \tag{17}$$

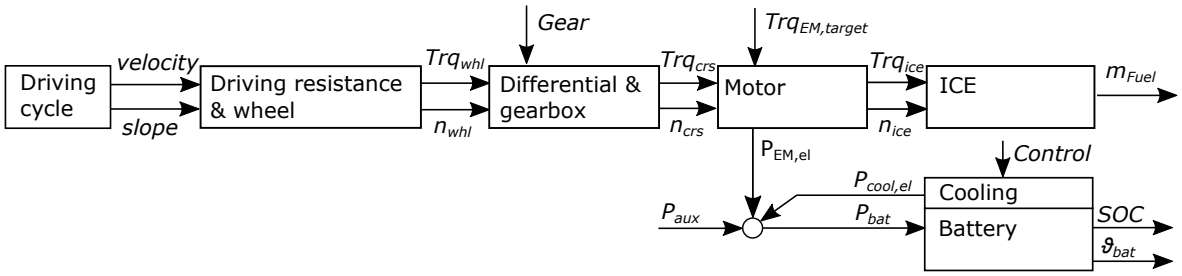

Figure 4: Signal flow diagram of the HEV model.

4.1.4 Battery Model

For the calculation of change in charging state, the battery was modelled as an equivalent two terminal network with simplification of the underlying physics of batteries. With neglection of any effect of temperature or loading cycles for the batteries capacitance, the change of charging state was purely determined from the previous state and any required power output from the battery, composing of the power for operation of the EM as well as any auxiliary users (e.g. the cooling system):

$$SOC_{t+1} = f(SOC_t, P_{bat}) \tag{18}$$

For results shown in section 5.4, a temperature model of the battery was implemented approximating the change of temperature due to the power output and the cooling control signal $Cool_{Control}$:

$$\vartheta_{bat,t+1} = f(\vartheta_{bat,t}, P_{bat}, Cool_{Control}) \tag{19}$$

The utilization of the cooling system causes an additional power consumption $P_{cool,el}$.

4.2 Driver Model

The driver is implemented as driving cycles implying specific velocity procedures the vehicle drives through simulatively. Next to standard procedures for the determination of emission levels, cycles generated by a stochastic driver model from (Liessner et al., 2017) are used for a more realistic reproduction of real-world traffic scenarios. Hereby, diverse characteristics can be considered within those cycles like different market features, specific driving styles and habits or traffic situations at different points during the day. The tuning of operating strategies towards special features in real-world driving scenarios can be seen as a major benefit for customers concerning fuel efficiency of their vehicles since it can be assumed that e.g. an operating strategy derived with european driving data will not work as efficient in India or China.

4.3 States, Actions and Reward

4.3.1 States

At each timestep during simulative driving the environment provides a state vector for the agent constituting the current internal state of the vehicle. Here, the state vector is defined by:

$$s_t = (n_{whl}, M_{whl}, SOC, \vartheta_{bat}, G) \tag{20}$$

In the case of the underlying vehicle model not taking into account the battery temperature and the according power limit, ϑ_{bat} can be rejected from the state vector. In this form the environmental state complies with the Markov-Property in terms of any following state of the environment (vehicle model) being only dependent on the current state and the chosen action by the agent. Hence reinforcement learning can be utilized to solve the underlying control problem.

4.3.2 Actions

For any given state the agent has to choose an action which he considers optimal for the maximization of his return. For the energy management problem actions can include any parameterizable metric with an effect on fuel and energy efficiency. Commonly parameterized metrics in the optimization of operating strategies are the split of torque between the electric motor and the combustion engine, the choice of the gear or the management of the battery temperature. Here we only consider the choice of power output for the electric motor as an action controlled by the RL agent. Choice of gear and temperature regulation are implemented as heuristics. As mentioned, the electric motor is subject to multiple restrictions concerning the maximum possible power output. Not only is this limited by the current crankshaft speed but also - if considered by the model - by the maximum power output of the battery regarding its temperature. In contrast to discrete Q-tables, where the use of certain actions in specific states can be restricted by assigning extremely low Q-values, a continuous

output of neural networks can hardly be limited without any additional feedback signal which the network can learn to cope with. Hence the selectable action of the agent was implemented as a percent value of the current maximally applicable torque with the output layer of the neural network as a tanh-layer restricting the action to $a_t \in [-1,1]$. The applicable power of the electric motor is then determined by:

$$P_{EM} = \begin{cases} a_t \cdot P_{EM,max} \; for \; a_t \geq 0 \\ a_t \cdot P_{EM,min} \; for \; a_t < 0 \end{cases} \tag{21}$$

where the agent can choose to operate the EM either as a motor or a generator respectively by choosing positive or negative action values.

4.3.3 Reward

Since the main goal of the operating strategy is minimizing fuel consumption but the RL agent aims at maximizing his return, a common practice is to state the reward with a negative sign. Here the agents reward based on his selected action is defined as the negative total energy usage per timestep by:

$$r_t = -(E_{che} + \kappa E_{el}) \tag{22}$$

with

$$E_{che} = m_{Fuel} \cdot \rho_{Fuel} \cdot H_{Fuel} \tag{23}$$

and

$$E_{el} = P_{bat} \cdot \Delta t \tag{24}$$

where the agent not only gets incentivized to minimize fuel consumption but to do so with the least amount of electric energy as possible. Thereby E_{el} is weighted by a factor $\kappa = f(SOC)$ which rates the cost of electric energy proportionally to the deviation of the SOC to a target value SOC_{target}. In doing so, the agent should learn to balance his SOC over the course of a driving cycle as described in section 3.2.1.

4.4 Training

With the stated RL specific formulation, a DDPG algorithm was implemented for solving the energy management problem. With a lower computational cost than TRPO and a straight forward implementation, DDPG has proven its usability and sample efficiency in other implementations (Duan Yan et al., 2016). Additionally, the actor critic architecture offers a good approach for possible application in real-world vehicle-hardware with rather low computational capabilities, since the resulting operating strategy is captured in a single neural network.

Training was conducted for one specific driving cycle which could include any type of information about driving styles or traffic situations. The current operating strategy was evaluated every 5 episodes on that specific cycle to track the learning progress. In training episodes however, as mentioned, the agent was confronted with a unique stochastic cycle based on the original one in order to prevent the strategy from overfitting to a specific velocity procedure. At the same time, the length of the stochastic cycles was varied in every episode. In order to increase the amount of possible states which the network is confronted with during training, the initial values of the network were varied each training episode on a random basis and within defined bounds. For evaluation the SOC and ϑ_{bat} were always initialized to 50% and $20°C$ in order to the track learning progress.

For general exploration of the state space, as in (Timothy P. Lillicrap et al., 2015), an Ornstein-Uhlenbeck-Process (Borodin and Salminen, 1996) was used for adding noise to the chosen actions of the agent. The exploration noise was slowly decayed over the first 1000 training episodes.

The training procedure of DDPG is shown as pseudocode in algorithm 1.

Algorithm 1: Pseudocode for DDPG.

```
initialize networks
initialize replay buffer
for episode = 1,M do
    initialize exploration process
    observe initial state s_0
    for t=1,T do
        choose action a_t with actor and add noise
        execute a_t
        observe r_{t+1} and s_{t+1}
        store [s_t, a_t, r_{t+1}, s_{t+1}] in buffer
        choose minibatch from buffer
        update critic with loss L from minibatch
        update actor with policy gradient
        update target networks
```

4.5 Hyperparameters

Similar to the original architecture, both actor and critic were implemented as deep neural networks with 2 hidden layers each with 400 and 300 neurons and Rectified-Linear-Unit activation. The actor receives a standardized state vector and outputs a continuous action as described. The critic additionally takes in the actions into the second layer of the network and outputs a continuous Q-Value through linear activation. An Adam-optimizer was chosen for the update-

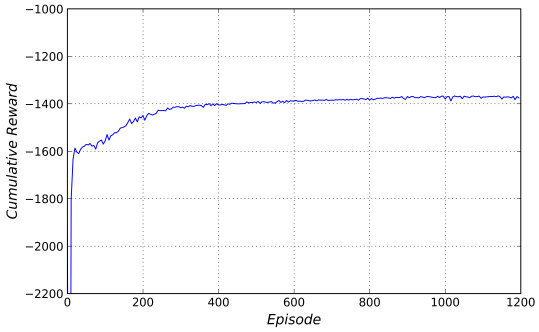

Figure 5: Cumulative reward (return) of the agent during training process for the New-York-City-Cycle.

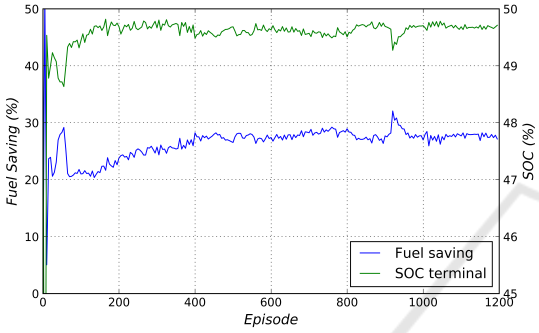

Figure 6: Fuel saving in % compared to driving solely with the combustion engine (blue) and terminal battery charging state (green) during the training process for the New-York-City-Cycle.

process of the neural networks with a learning rate of 10^{-5} for the critic and 10^{-4} for the actor. The replay buffer for sampling minibatches was initialized to a size of 10^5 where in every timestep a minibatch size of 32 was sampled to update the critic network.

For the training process a discount factor of $\gamma = 0$ was chosen, thus the agent optimized his strategy only towards local rewards. However, since the reward function contains a dynamic weighting factor dependent on the current charging state, the agent has turned out to adapt his strategy rather long-sighted. That way significantly better results were achieved compared to discounts $\gamma > 0$.

5 RESULTS

5.1 Learning

Figure 5 and 6 show exemplary training progress for simulative driving of the New-York-City-Cycle (NYCC), a characteristic cycle for low speed city driving with many start-stop maneuvers. As shown, the agent continuously increases his cumulative reward - the return - with progressive training. In doing so, according to the reward function, the resulting energy consumption is decreased consistently with a constant fuel saving of over 20% compared to driving with the combustion engine only, established very early in the training process. With the resulting battery charge at the terminal state of an episode very close to the initial value of 50%, the *SOC* can be described as balanced. In a low speed environment where the rather low-powered electric motor can be used very effectively, a positive deviation of the battery charge throughout the cycle can well be expected for any efficient operating strategy. Since the agent did not have any hard constraint towards the charging state but was only implicitly incentivized through the reward function, he is free to make any compromise between the use of chemical or electric energy which he considers most efficient.

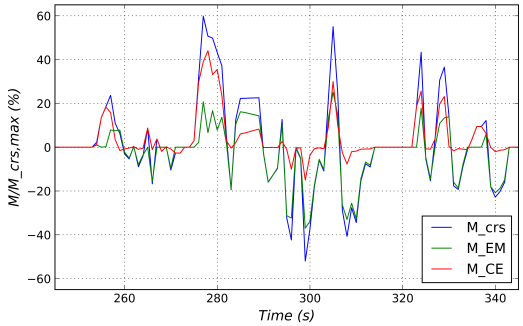

Figure 7: Excerpt of the split of overall torque (blue) between the combustion engine (red) and the electric motor (green) for the New-York-City-Cycle, controlled by the fully trained agent.

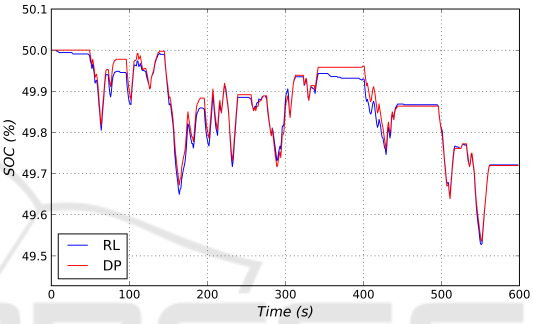

Figure 8: Comparison of battery loading trajectories for the New-York-City-Cycle with the strategy learned by the RL agent (blue) and the optimal strategy derived by Dynamic Programming (red).

Figure 7 shows an excerpt from the resulting strategy for the NYCC in terms of split of torque between the combustion engine and the electric motor which the agent establishes with the choice of his actions. As shown, the agent fully learns to use moments of vehi-

cle deceleration for recharging the battery with recuperational energy by choosing actions $a < 0$ which runs the electric motor in generator-mode. Mentionable is that after 20 episodes of training the level of this behavior is already nearly as sophisticated as after 1000 episodes of simulative driving.

It is recalled that the results for purely deterministic cycles only serve evaluation purposes. Since the agent is trained with random cycles similar to the one he is being evaluated on, he is able to deal with high amounts of stochasticity concerning the velocity profile of the drive and will control the electric motor just as efficient. Due to the lack of space this can not be fully presented here and will be briefly covered in section 5.3.

In general, training the agent for a variety of different driving cycles is exceptionally stable. For driving profiles including rather high velocity and acceleration rates the learning rate of the actor network had to be adjusted occasionally, since the unavoidable increase of the reward magnitude through higher momentary fuel consumptions leads to bigger gradients in the beginning of the training, where the neural networks are initialized with outputs close to zero and the loss increases accordingly. Clipping the rewards or gradients to certain maximum values would be a considerable option to avoid this, though useful information about the fuel consumption might be excluded at a later point in the training.

5.2 Quality Analysis

For analyzing the quality of the operating strategy learned by the agent, a comparison was made to a strategy derived by dynamic programming with discrete state and action spaces. With full knowledge of the cycle in advance, the result of DP can be seen as the globally optimal solution even though a small error will be unavoidable through the use of discrete parameters.

Table 1: Comparison of fuel savings for the strategy derived by the RL agent and a global optimum computed with DP in contrast to driving with ICE only.

Driving Cycle	RL	DP	Δ
NYCC	28.6%	29.5%	0.9%
WLTP	16.0%	16.3%	0.3%
US06	10.8%	11.2%	0.4%
FTP75	20.1%	20.7%	0.6%

Table 1 shows the resulting fuel savings for 4 different velocity profiles typically used in calibration and certification processes. The DP results are based on the terminal *SOC* of the RL strategy in order to assure comparability. As seen, only small deviations occur, where the optimal strategy achieves less than 1% more savings in fuel which makes the strategy learned by the agent nearly-optimal. A comparison of both battery charging trajectories resulting from the operating strategy for the NYCC is shown in Figure 8, confirming that the agent's strategy converges towards a global optimum and does not get stuck in a poor local optimum throughout the training process. In contrast to DP, the RL agent does not need any prior information about the driving route and he is trained to control the EM optimally with only momentary state information provided by the vehicle environment. A local strategy optimization with nearly globally optimal solutions can be seen as a major advantage of deep reinforcement learning over any other traditional method for the derivation of operating strategies for HEVs.

5.3 Stochastic Cycles

As mentioned, in every episode of the training process the agent is confronted with a different stochastic velocity profile with the same characteristics concerning speed and acceleration as the one he will be evaluated on. Next to preventing overfitting, this is mainly aimed at increasing the agent's ability to generalize his strategy towards potentially unknown states and driving profiles. Where traditional RL methods with discrete parameters require tedious approximation methods, generalizing knowledge is a major advantage of deep RL obtained through the use of neural networks.

Table 2: Comparison of a strategy trained with stochastic velocity profiles and the other trained solely deterministically. Results are shown as the deviation of fuel saving in % compared to a global optimal solution computed by dynamic programming.

Driving Cycle	Stochastic	Deterministic
NYCC	1.6%	1.5%
$Rand_{NYCC,1}$	0.95%	3.1%
$Rand_{NYCC,2}$	1.5%	6.6%
$Rand_{NYCC,3}$	1.7%	2.5%
$Rand_{NYCC,4}$	1.4%	3.1%
$Rand_{NYCC,5}$	1.6%	3.2%

Table 2 shows a comparison of two strategies, one trained with stochastic cycles based on the NYCC, the other trained purely with the deterministic NYCC which was repeatedly driven through simulatively. Where the deterministically derived strategy performs good on the original NYCC, it is shown that for any other stochastic velocity profile similar to the original NYCC the strategy performs significantly worse which clearly indicates overfitting. The stochastically

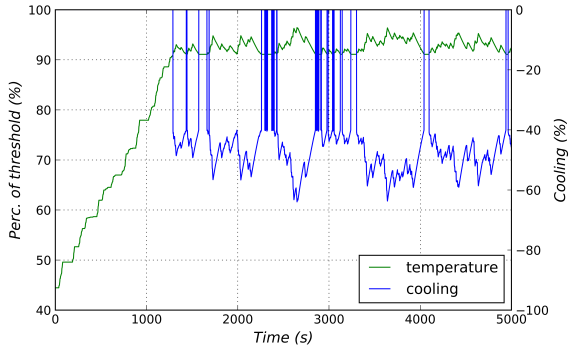

Figure 9: Battery temperature relative to the threshold for power limitation (green) together with the control of cooling power (blue) for simulative driving of a 5.000s long stochastic cycle based on NYCC with the operating strategy learned by the agent.

trained strategy in contrast shows constantly good results for any driven route similar to the original NYCC with a deviation to the global optimal solution of only $0.9 - 1.7\%$. Additionally, stochastic cycles could include any type of further information about driving habits or traffic scenarios which neural networks could easily take into account for the operating strategy. This results in a much more accurate illustration of realistic traffic potentially leading to increased fuel savings and emission reduction in the actual operation of the vehicle rather than deriving approximate values on a test bench which could hardly be achieved in the real-world.

5.4 Inclusion of Temperature Information

As mentioned, for this paper a temperature approximation of the battery was included into the vehicle model combined with a power restriction to prevent damage from overheating. Additionally, a cooling mechanism of the battery was implemented with a heuristic used for control. Figure 9 shows an exemplary development of the battery temperature for simulative driving of a 5000 s stochastic cycle, which the agent has optimized his strategy for. As it shows, the agent has learned to control the EM in such a manner that the temperature level of the battery remains below a threshold value over which the maximum power output of the battery would be restricted and the agent could not use the full potential of the electric motor. Additionally, a clear tendency can be seen to keep the temperature level close to the threshold level of the cooling heuristic, around 91%, in order to decrease the additional use of battery power for the cooling system. With the additional boundary condition implemented implicitly into the vehicle model and the agent not receiving any explicit information except the resulting reward based solely on the energy consumption, the agent still found nearly optimal solutions to the energy management problem. Here the major potential of deep reinforcement learning can be seen compared to traditional methods for the derivation of operating strategies with discrete variables as additional optimization goals can be implemented into the training process without any significant extra cost or loss in quality.

6 CONCLUSION

The energy management of hybrid electric vehicles poses major challenges for automobile manufacturers where traditional approaches show many deficiencies in processing additional information concerning real-world driving scenarios. In this paper, a deep reinforcement learning framework has been derived that offers great potential for solving many of those problems. It has been shown that a deep RL agent is capable of achieving nearly-optimal fuel consumption results with a locally trained strategy which can be applied online in the vehicle. In contrast to dynamic programming, no prior knowledge of the driving route is necessary and the training with stochastic driving cycles allows for greater generalization to variously different velocity profiles. Additionally, deep reinforcement learning allows the efficient inclusion of further optimization criteria.

Another advantage can be seen in the potential adaptability of the agent which would be able to learn from real driving data even during direct usage of the vehicle. A conceivable result would be an intelligent energy management system constantly adapting to specific driving habits, typically driven routes or other desirable characteristics of the vehicle owner or driver.

Future work can build up on the presented results and the arising potential for further improvement. This includes a broader examination of the agents capabilities of online generalization and adaptability, simulating alternating traffic scenarios or different driving habits while constantly updating the agents strategy. Furthermore, a big field of research arises from the integration of additional vehicle or future route information, e.g. from navigation data. A predictive strategy offers great potential for further reduction of fuel consumption and emission-levels.

In a next step, the algorithm could be applied on a power train test bench for assessing and refining the results learned with the simulation model. Similarly, the transfer into the real vehicle would be feasible.

REFERENCES

Borodin, A. N. and Salminen, P. (1996). Ornstein-uhlenbeck process. In Borodin, A. N. and Salminen, P., editors, *Handbook of Brownian Motion — Facts and Formulae*, pages 412–448. Birkhäuser Basel, Basel.

C. Liu and Y. L. Murphey (2014). Power management for plug-in hybrid electric vehicles using reinforcement learning with trip information. In *2014 IEEE Transportation Electrification Conference and Expo (ITEC)*, pages 1–6.

Chae, H., Kang, C. M., Kim, B., Kim, J., Chung, C. C., and Choi, J. W. (2017). Autonomous braking system via deep reinforcement learning. *CoRR*, abs/1702.02302.

Chasse, A. and Sciarretta, A. (2011). Supervisory control of hybrid powertrains: An experimental benchmark of offline optimization and online energy management. *Control Engineering Practice*, 19(11):1253–1265.

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller (2014). Deterministic policy gradient algorithms. In Tony Jebara and Eric P. Xing, editors, *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 387–395. JMLR Workshop and Conference Proceedings.

Duan Yan, Chen Xi, Houthooft Rein, Schulman John, and Abbeel Pieter (2016). Benchmarking deep reinforcement learning for continuous control. *International Conference on Machine Learning*, 2016:1329–1338.

European Commission (2017). Draft regulation: real-driving emissions in the euro 6 regulation on emissions from light passenger and commercial vehicles.

Guzzella, L. and Sciarretta, A. (2013). *Vehicle propulsion systems: Introduction to modeling and optimization*. Springer, Heidelberg, 3rd ed. edition.

John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel (2015). Trust region policy optimization. *CoRR*.

Kirschbaum, F., Back, M., and Hart, M. (2002). Determination of the fuel-optimal trajectory for a vehicle along a known route. *IFAC Proceedings Volumes*, 35(1):235–239.

Leroy, T., Malaize, J., and Corde, G. (2012). Towards real-time optimal energy management of hev powertrains using stochastic dynamic programming. In *2012 IEEE Vehicle Power and Propulsion Conference*, pages 383–388. IEEE.

Liessner, R., Dietermann, A., Bäker, B., and Lüpkes, K. (2017). Generation of replacement vehicle speed cycles based on extensive customer data by means of markov models and threshold accepting. *SAE International Journal of Alternative Powertrains*, 6(1).

Mirzaei, H. and Givargis, T. (2017). Fine-grained acceleration control for autonomous intersection management using deep reinforcement learning. *CoRR*, abs/1705.10432.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.

Onori, S., Serrao, L., and Rizzoni, G. (op. 2016). *Hybrid electric vehicles: Energy management strategies*. Springer, London.

Patrick Wappler (2016). Applikation von hybridfahrzeugen auf basis künstlicher intelligenz. Master's thesis, Technische Universität Dresden, Lehrstuhl für Fahrzeugmechatronik.

Puterman, M. L. (dr. 2010). *Markov decision processes: Discrete stochastic dynamic programming*. Wiley Series in Probability and Statistics. John Wiley & Sons, Hoboken, [dodr.] edition.

Shixiang Gu, Timothy P. Lillicrap, Ilya Sutskever, and Sergey Levine (2016). Continuous deep q-learning with model-based acceleration. *CoRR*, abs/1603.00748.

Sutton, R. S. and Barto, A. G. (2012). *Introduction to reinforcement learning*. MIT Press, 2 edition.

Tate, E. D., Grizzle, J. W., and Peng, H. (2008). Shortest path stochastic control for hybrid electric vehicles. *International Journal of Robust and Nonlinear Control*, 18(14):1409–1429.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra (2015). Continuous control with deep reinforcement learning. *CoRR*, abs/1509.02971.

UNECE Transport Division (2005). Vehicle regulations: Regulation no. 101, revision 2,.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller (2013). Playing atari with deep reinforcement learning. In *NIPS Deep Learning Workshop*.

X. Lin, Y. Wang, P. Bogdan, N. Chang, and M. Pedram (2014). Reinforcement learning based power management for hybrid electric vehicles. In *2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 33–38.