Institute for Design and Control of Mechatronical Systems

Master's Thesis:

Hybrid powertrain online control using reinforcement learning

Author: Thomas Leidinger

Supervisors: Univ.-Prof. DI Dr. Luigi del Re

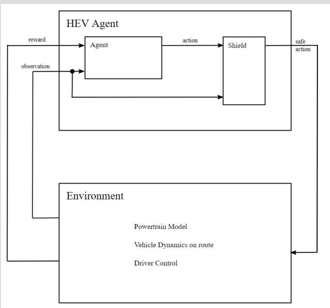
Finished: September 2021

Abstract

Nowadays, the general environmental problems of energy waste and greenhouse gas emissions represent an important topic in our world. Hence, the decrease of air pollution and fuel consumption of vehicles plays a major role to improve the current situation. In the field of transportation, the Hybrid Electric Vehicle (HEV) is an approach to reduce greenhouse gas emissions by selecting the power out of two sources. Therefore, the main task of the energy management of a HEV includes the power split between the electrical machine/battery and the engine. The recharging ability of the battery inappropriate parts of the driving cycle supports the HEV to improve the general goals. The best results are obtained if full information is known in advance and therefore a global solution can be calculated e.g., with Dynamic Programming (DP). However, the computational burden prevents this method to be used in changing environments (driving cycles) to recalculate a new solution.
The machine learning method Reinforcement Learning (RL) can obtain a comparable solution and reduce the computational burden to a large extent. This approach works by learning a policy due to visiting different states through executing different actions. In this thesis, a Deep Reinforcement Learning (DRL) method is developed to handle the case of 3 actions/inputs values for the HEV model. Therefore, a Neural Network is used as a function approximator in the considered complex problem which approximates a policy even for unvisited states. Further, an action shield is used to prevent the agent from selecting actions that would lead to infeasible solutions. In the beginning, the agents (controller) are trained in 4 different environments (4 different city driving cycles) respectively. The DRL is shown to be able to deliver very comparable results compared to the results of DP in all 4 cases. In the next step, a trained agent was applied to the other 3 driving cycles to test the performance of the agent running in changing/uncertain environment. It shows that the agents, although trained in other environments, can adapt to new environments flexibly and deliver good results within few seconds. This application simulates the online approach by computing solutions in unknown environments.

Method and Implementation

The implementation of the Deep Reinforcement Learning method, applied to the HEV problem, is made with a Deep Q-Learning algorithm. The main part of the implementation of the method in MATLAB® is made with a step-function and a reset-function. The step-function is called every time step in an episode (learning interval) and calculates the next observations, relevant signals and the reward based on the states of the model and fed actions. Therefore, the model supplies the specified signals for the considered time step. If the episode ends, the reset function is called and the observations and relevant signals were set to the initial value. After the reset, the agent begins to learn again and starts the new episode. The goal of the agent is to maximize the expected return of rewards.



Environment
The environment consists of all parts of the system that the agent does not include. Therefore, the environment of the HEV problem also includes the driving cycle. Since in this work, the agent is required to be able to control a driving cycle with changing conditions, different driving cycles were implemented to simulate the online behavior.

HEV Agent
The agent represents the controller of the Reinforcement Learning system, who has the task of selecting the optimal actions for the HEV problem. The DQN-agent operates in a trial-and-error manner to explore the environment due to the fact to maximize the total cumulative reward. In some cases of the training process, when the action selection leads to operating regions of the engine or the electrical machine which are infeasible, the agent calculates unacceptable solutions. Therefore, these actions have to be avoided by an additional shield block. To do so, the shield receives the values of the current state/observation and action and calculates a safe action based on this information.

Action Space and Observation Space
Three action variables are the input for the powertrain model: gear shift command, the clutch shift command and the coupler value between the engine and the electrical machine. The observations of an RL problem must fulfill the task of delivering information for the agent: State of charge of the battery (SoC), Gear-position (j_k), Clutch-position (c_k), Required torque (T), Required rotational speed (w).

Reward Function
To evaluate an executed action of an agent, this function provides a reward value. In this case, to reach the goals of the HEV problem, charge sustaining, minimizing the fuel consumption (qf), gear shifts (z_j) and clutch shifts (z_c) have to be considered: $r_{\text{total}} = r - z_j \cdot \text{fac}_j - z_c \cdot \text{fac}_c$

$$r = \begin{cases} -\text{qf}: & \text{SoC} \geq \text{SoC}_{\text{min}} \text{ and } \text{SoC} \leq \text{SoC}_{\text{max}} \\ -\text{qf}_{\text{max}}: & \text{SoC} < \text{SoC}_{\text{min}} \text{ or } \text{SoC} > \text{SoC}_{\text{max}} \end{cases}$$

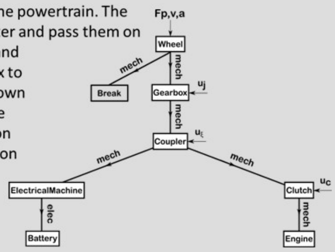
Hybrid Electric Vehicle Model

The HEV model is composed of a vehicle model and a powertrain model. The vehicle model also includes the parameters of the driving cycle in addition to the vehicle parameters. The powertrain model, which consists of an engine, an electric machine, a battery, etc., needs to provide the required power demand from the driving cycle and vehicle model.

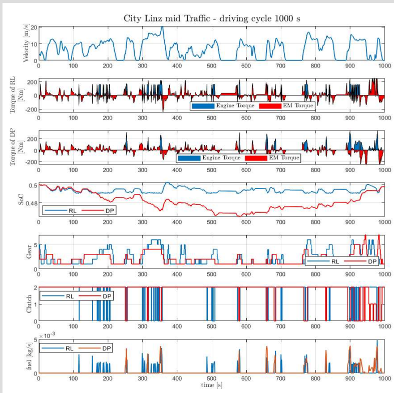
Vehicle Model
This model computes the power demand based on the vehicle and the driving conditions. Therefore, an assumption is made that the powertrain is stiff and there is no wheel slip. The traction force $F_P = f(F_r)$ and the resistance force F_r are used to calculate the power demand: $P = F_P \cdot v$

Powertrain Model
This model realizes the energy flow through the components of the powertrain. The individual parts of the powertrain calculate their specific parameter and pass them on to the next component. This model works with a break heuristic and does not use an extra coupler between the brake and the gearbox to control the mechanical energy flow provided by the wheel. As shown in the figure, the inputs of the gearbox, coupler and clutch are the three inputs of the powertrain model and the general optimization problem formulation. The required power demand is a combination of the battery P_b, the engine P_e and the loss power from the powertrain components P_L :

$$P + P_L = P_b + P_e$$



Results



Comparison with Dynamic Programming
The Dynamic Programming approach delivers the benchmark solution of the HEV problem. The computational time of the optimal solution depends heavily on the discretization of the considered HEV model parameter. The DP solution is a global calculated result (as seen by the course of the SoC of the battery of the DP solution) in contrast to the DRL method.

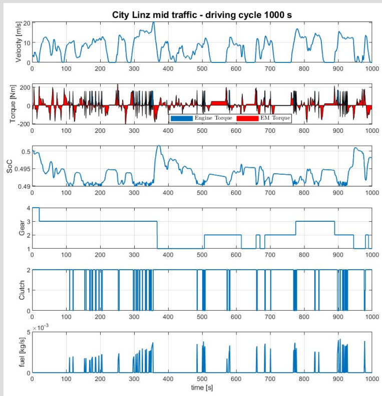
Method	DRL	DP
Total fuel consumption [kg]	0.1774	0.1749
Number of gear shifts	175	81
Number of clutch shifts	92	34
Total cost	0.2308	0.1979
Final SoC	0.4978	0.4979

The total fuel consumption computed by the DRL method reaches a value that is near the optimal fuel consumption value of the DP method. However, the number of gear shifts and the number of clutch shifts are thereby increased.

Online Application
In contrast to the previous results (where the agent is applied to the the same environment (driving cycle) as he learned in), an agent is applied to a driving cycle other than the one he is trained on to test the online capability of DRL. The DP solution, applied on a different driving cycle, does not deliver feasible solutions. Therefore, if the used model is selected to simulate an HEV problem, only the DRL method computes feasible results.

Method	DRL	online DRL
Total fuel consumption [kg]	0.1774	0.1946
Number of gear shifts	175	13
Number of clutch shifts	92	91
Total cost	0.2308	0.2154
Final SoC	0.4978	0.4980
Computational time [s]		4.5

The upcoming observations, in this case, are similar to the occurred observations in the training process and so the agent knows how to calculate a useful solution, which results in comparable numbers.



Conclusions and Outlook

In conclusion, the results of the Deep Reinforcement Learning approach, of course, can not deliver the optimal solution as the benchmark method Dynamic Programming. The results manifest that in offline cases (where the agent is trained in an environment and then simulated in the same environment), DRL can perform comparably as DP but takes a longer time. This is due to the computational effort of a machine learning method like DRL. However, in online cases (where a trained agent of one environment is then simulated in a new environment), DRL shows its advantages of delivering feasible solutions, calculated in short computational time, which is crucial in online applications. Therefore, the online DRL approach can be seen as a robust online approach under disturbances.

The implementation of the 3 actions DRL method in an HEV problem could be expanded to the existing applications of the 1 action approach stated in the literature. This means, for example, that the agent learns in an environment where the driving cycle is changed each or in a fixed period of episodes. Then, the agent learns different styles of routes and can handle a wider range of driving cycles in online applications. Further work could also be done by choosing different types of Deep Reinforcement Learning agents, like a Deep Deterministic Policy Gradient (DDPG) agent which is used in previous works for the 1 action approach.