

Setting the Equivalence Factor in Equivalent Consumption Minimization Strategy using a Reinforcement Learning Based Controller

Energy Management in Long Haul Hybrid Trucks

Master's thesis in Applied Physics

Ragnar Englund

MASTER'S THESIS 2017

Setting the Equivalence Factor in Equivalent Consumption Minimization Strategy using a Reinforcement Learning Based Controller

Energy Management in Long Haul Hybrid Trucks

RAGNAR ENGLUND



CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Signals and Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2017

Setting the Equivalence Factor in Equivalent Consumption Minimization Strategy
using a Reinforcement Learning Based Controller
Energy Management in Long Haul Hybrid Trucks
RAGNAR ENGLUND

© RAGNAR ENGLUND, 2017.

Supervisor: Jonas Hellgren, Volvo Trucks AB
Supervisor: Jonas Sjöberg, Signals and Systems, Chalmers
Examiner: Jonas Sjöberg, Signals and Systems, Chalmers

Master's Thesis 2017
Department of Signals and Systems
Chalmers University of Technology
SE-412 96 Gothenburg

Cover: Heat map constructed in Matlab showing showing the equivalence factor as a function of the state. For more details see section 7.2.3

Typeset in L^AT_EX
Gothenburg, Sweden 2017

Setting the Equivalence Factor in Equivalent Consumption Minimization Strategy
using a Reinforcement Learning based Controller
Energy management in Long Haul Hybrid trucks
RAGNAR ENGLUND
Department of Signals and Systems
Chalmers University of Technology

Abstract

In this master's thesis a novel approach for energy management in hybrid trucks is analyzed and presented. A reinforcement learning algorithm, based on tabular Q-learning, is used to dynamically set the equivalence factor in an ECMS (Equivalent Consumption Minimization Strategy) torque splitter in order to minimize the ownership cost of a long haul hybrid truck during a transport task. The ownership cost includes fuel consumption, battery degradation and disc brake degradation which is an element unique to this paper, and from the results it is apparent that reinforcement learning has a lot of potential as an energy management controller. In the best test case the developed controller lowers the ownership cost compared to a conventional truck by $\sim 5.7\%$ and has a lower ownership cost than when a constant equivalence factor is used. The RL agent is also shown to find good decision policies when truck hardware, such as the battery, changes in its size or cost.

Keywords: Hybrid truck, long haul hybrid truck, ECMS, reinforcement learning, Q-learning, machine learning, energy management system, EMS.

Acknowledgements

This thesis work has run a lot smoother thanks to my supervisor at Volvo AB Jonas Hellgren, who has given me advise and helped me a lot with structuring my work. I would also like to thank my examiner at Chalmers Jonas Sjöberg who, even though he had less insight in my work, still has managed to give a lot of valuable inputs during the course of the project.

I would like to thank Sven Ahlinder at Volvo AB who helped me with writing a very computationally efficient function for linear interpolation, which was heavily used in the thesis work. Also thanks to my friends Simon Johansson and Johan Särnbratt at Chalmers who helped me with optimization questions.

Finally I would like to thank the other master thesis workers Sudanwa Bhadd, Jincan Hi, Anton Lidbeck and Kazim Raza Syed and all other employees who brightened my stay at Volvo AB.

Ragnar Englund, Gothenburg, June 2017

Contents

List of Figures	xi
List of Tables	xv
1 Introduction	1
1.1 Background	1
1.2 Main contributions	4
1.3 Terms used in the report	4
2 Transport task and vehicle dynamics model	5
2.1 Transport task	5
2.2 Vehicle dynamics	6
3 Powertrain model	7
4 Economical modeling	9
5 Powertrain controllers	11
5.1 Gear controller	11
5.2 Torque splitter	13
5.2.1 Cost function	13
5.3 RL based controller	14
5.3.1 Action	14
5.3.2 Reward	14
5.3.3 State	15
5.3.4 Q-learning	16
6 Simulations with constant equivalence factor	19
6.1 Studied vehicle and transport task	19
6.2 Results - Default truck	20
7 Simulations with RL based equivalence factor	23
7.1 Studied vehicle and transport task	23
7.2 Results	24
7.2.1 Default truck	24
7.2.2 Decreased ESS size	26
7.2.3 Non-position based state	28

8 Sensitivity analysis	31
8.1 Results	31
9 Conclusion	33
10 Future work	35
A Appendix	I
A.1 Hybrid powertrain model	I
A.2 Conventional powertrain model	II
A.3 Battery model	II
B Appendix	V
B.1 ECMS	V
B.2 Constraints	VI
C Appendix	VII

List of Figures

1.1	Schematic of the hybrid powertrain model. Power can be supplied from both the battery and from the engine either to brake or propel the truck forward.	1
1.2	Schematic of the control architecture, and its different components, used in the virtual test bench. ECMS is used as the torque splitter and a RL algorithm as the equivalence factor controller.	2
1.3	Schematic view of the ownership cost of a long haul hybrid truck and its components. The components considered in this thesis project are costs from fuel consumption, battery degradation and disc brake degradation.	3
2.1	A truck with all forces considered in the model.	6
3.1	Schematic of the powertrain model used. The images illustrate where the different torques T , angular velocities ω and gear ratios used in equation (3.1) to (3.5) are positioned in the powertrain. The arrows show how the torque is transferred from the ENG and EM to the wheels.	7
4.1	Schematic view of the different components of the ownership cost of a long haul truck. The components considered in this report are costs from fuel consumption, battery degradation and disc brake degradation.	9
5.1	Schematic of the control architecture, and its different components, used in the virtual test bench. The RL based controller here acts as the Eq factor controller.	11
5.2	A flow chart that illustrates the logic of mapping a $(T_{prop}, \omega_{wheel})$ pair to a gear. The feasible gears are chosen as is depicted in Figure 5.3.	12
5.3	Illustration of how the four gears, G_1, G_2, G_3, G_4 , are chosen from a given $(T_{prop}, \omega_{wheel})$ tuple. The gear itself is chosen according to the logic in Figure 5.2. $(T_{prop}, \omega_{wheel})$ is shown as the red star and the possible gears that the logic can choose between are shown as the blue circles. T_i, ω_j are discretized propulsion torques and wheel angular velocities.	12

5.4	Illustration of how T_{em} , T_{eng} and T_{disc} are used. The figure shows a propulsion torque T_{prop} that decreases linearly as a function of time. The propulsion torque is supplied first by the EM and ENG, and when they reach their limits the rest of the propulsion torque is supplied by the disc brake.	14
5.5	Illustration showing the height differences used in state 2, see Section 5.3.3. The truck looks at the current altitude as well as the altitudes 250 m and 3 km ahead. Then the height differences are calculated as $dh_{3km} = H_3 - H_2$ and $dh_{250m} = H_2 - H_1$	16
6.1	Barplots comparing the constant eqf and a conventional truck. The leftmost barplot shows the ownership cost. The next plot to the right illustrates the fuel consumption, the third the battery degradation and the final plot the disc degradation. The conventional truck does not have any battery degradation since it does not have an ESS.	20
6.2	UPPER: Plot showing the propulsion torque the truck has to provide and how much that is provided by the EM, ENG and disc brakes. MIDDLE: Plot of the gear profile during the transport task. LOWER: Plot of the SoC profile during the transport task.	21
7.1	UPPER: Plot showing the propulsion torque the truck has to provide together with the torques provided by the EM, ICE and the disc brakes. MIDDLE: Plot of the equivalence factor during the transport task. LOWER: Plot of the SoC during the transport task.	24
7.2	Barplots comparing the RL agent, constant eqf and a conventional truck. The leftmost barplot shows the ownership cost. The next plot to the right illustrates the fuel consumption, the third the battery degradation and the final plot the disc degradation. The conventional truck does not have any battery degradation since it does not have an battery. Note that the constant eqf has the lowest OC since it is a very good energy management controller, as was mentioned in Section 7.1.	25
7.3	Barplots comparing the RL agent, constant eqf and a conventional truck. The leftmost barplot shows the ownership cost. The next plot to the right illustrates the fuel consumption, the third the battery degradation and the final plot the disc degradation. The conventional truck does not have any battery degradation since it does not have a battery.	26
7.4	UPPER: Plot showing the propulsion torque the truck has to provide together with the torques provided by the EM, ICE and the disc brakes. MIDDLE: Plot of the equivalence factor during the transport task. LOWER: Plot of the SoC during the transport task.	27

7.5	Barplots comparing the RL controller, constant eqf and a conventional truck. The leftmost barplot shows the ownership cost. The next plot to the right illustrates the fuel consumption, the third the battery degradation and the final plot the disc degradation. The conventional truck does not have any battery degradation since it does not have a battery.	28
7.6	UPPER: Plot showing the propulsion torque the truck has to provide together with the torques provided by the EM, ICE and the disc brakes. MIDDLE: Plot of the equivalence factor during the transport task. LOWER: Plot of the SoC during the transport task. Note that the constant eqf has the lowest OC since it is a very good energy management controller.	29
7.7	Three heat maps illustrating the policy of the RL agent during different driving conditions. Each area represents the greedy action, the eqf, taken from that state. Black areas are states the RL agent did not encounter during the training simulations and thus has no data about. UPPER: Heat map of policy with $SoC = 0.4$. LOWER LEFT: Heat map of policy with $SoC = 0.5$. LOWER RIGHT: Heat map of policy with $SoC = 0.6$	29
A.1	The structure of the ESS. One ESS consists of one or several pack that are connected in parallel and each pack consists of many cells that are serially connected. For the details on how many packs and cells that were used please see Appendix C.	II
A.2	Plot of how the battery power output varies as a function of the SoC. Negative power represents charging the battery and positive represents a discharge.	III
C.1	Figure illustrating the altitude and gear profile during the transport task. Note that both of these profiles are the same for the RL agent, constant equivalence factor and the conventional truck.	IX

List of Tables

1.1	Table describing the abbreviations used in the report.	4
6.1	Specifications and initial setting for the truck used in the simulations.	19
7.1	Specifications and initial setting for the truck used in the simulations.	23
8.1	Parameter values and costs of different truck components and how they are changed for the different test cases, which can be seen in table 8.2	31
8.2	Comparison of the ownership cost between the RL based controller, the constant eqf and the conventional truck.	32
C.1	Specifications and initial setting for the truck used in the simulations.	VIII

1

Introduction

1.1 Background

The transport industry accounted for 23% of the greenhouse gases emitted in the USA during 2010 [4]. As governments starts to set emission goals for vehicles pressure is building on companies like Volvo AB to manufacture more efficient trucks which consumes less fossil fuel and contribute with less pollution.

One solution, which Volvo AB is heavily researching, are long haul hybrid trucks, which are considered in this report. A hybrid truck is a vehicle that utilizes the capacity of an energy storage system (ESS), in this case a battery, that modulate the use of a primary energy source, in this report a combustion engine (ENG). Together with the ESS comes an electric machine (EM) which can either work as a generator, recovering braking energy and storing it in the ESS, or as a motor aiding the ENG in propelling the truck.

The flexibility of using the ESS together with the ENG is what gives the hybrid its fuel saving properties compared to a conventional truck. However, this also creates a great challenge with hybrids. There is a need for a controller that decides how much power that should be supplied by the ESS and the ENG at each instant. The controller needs to take hardware, e.g. battery, EM and ENG (seen in Figure 1.1) into consideration. Also it has to consider some future information about the road and traffic situation so that the controller is able to plan ahead how to best utilize the different power sources. This makes the design of such a controller a complex task and a lot of research has been done on this topic where controllers have been developed that can use past, present and future driving conditions when optimizing the power usage of the hybrid truck [3]. In this report a new such controller using both future and current driving conditions is presented.

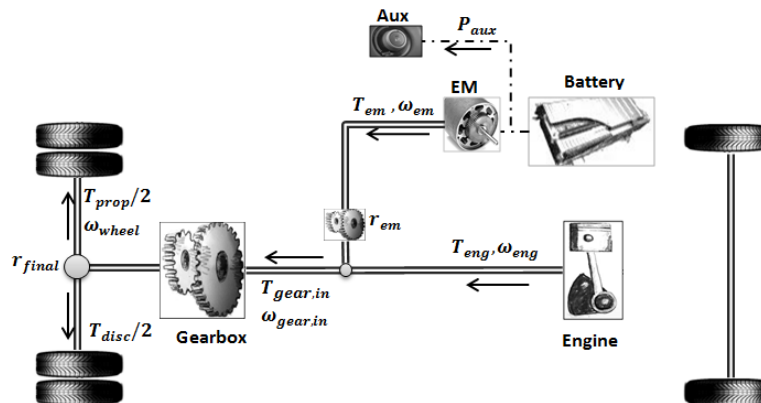


Figure 1.1: Schematic of the hybrid powertrain model. Power can be supplied from both the battery and from the engine either to brake or propel the truck forward.

This controller is based on the torque splitter Equivalent Consumption Minimization Strategy (ECMS) and a machine learning method called reinforcement learning (RL). The said controller has been developed and evaluated in a virtual test bench, as seen in Figure 1.2.

In the figure the different modules of the virtual test bench is shown. The *transport task* expresses the road slope and the velocity of the truck as a function of time. The *vehicle dynamics model* uses this information in order to calculate the torque at the wheel, T_{prop} , that is needed in order to reach said velocity and the angular velocity of the wheels, ω_{wheel} .

The *gear controller* uses T_{prop} and ω_{wheel} to calculate the current gear of the truck and this is sent to the *torque splitter*, where ECMS has been used, which decides how much torque that should be provided by the ENG and EM. However, the ECMS needs a parameter called the equivalence factor (eqf) in order to make the torque split and a RL algorithm has been used as an eqf controller, which sets the equivalence factor for the ECMS.

Lastly the torque split from the ECMS is sent to the *powertrain model*, which calculates how the torque split affects the truck, e.g. fuel consumption, battery degradation etc.

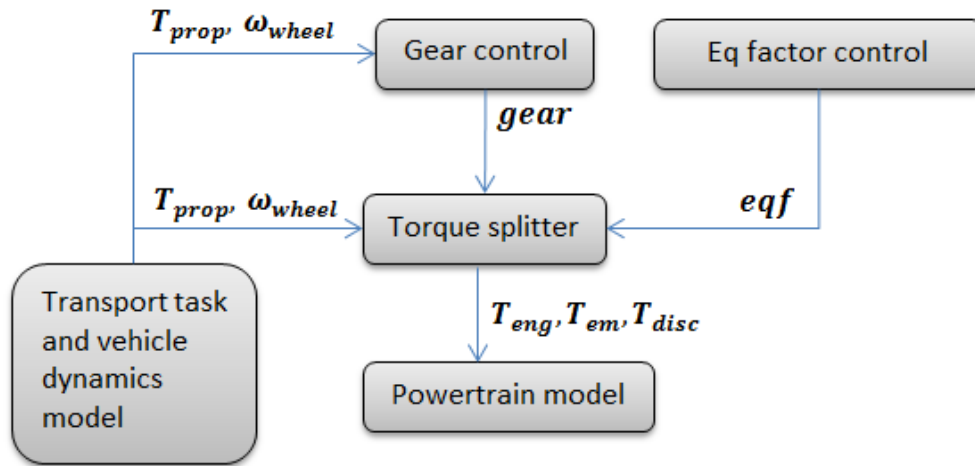


Figure 1.2: Schematic of the control architecture, and its different components, used in the virtual test bench. ECMS is used as the torque splitter and a RL algorithm as the equivalence factor controller.

The ECMS finds a torque split between the EM and the ENG that minimizes the current power used by the truck. If additional torque is required when braking then the disc brakes are used in order to supply this extra torque, otherwise they are not used. As mentioned earlier ECMS has a parameter called the equivalence factor (eqf) which states a relative cost between fuel and battery power, a small eqf making battery power more favorable to use and vice versa. Because of this the value of the eqf plays a crucial part in how the ECMS uses fuel and battery power, and thus the RL based controller plays an important role in setting the value for this parameter.

Since the RL is a machine learning method, unlike the ECMS, it needs to train on training/example data in order to approximate a certain function called the *Q-function*, which is used to set the eqf. The input to the Q-function is the current driving situation of the truck and an eqf. The output of the Q-function is how 'good' the eqf is in that driving situation, which is discussed more in Section 5.3. Using the virtual test bench a long haul hybrid truck is simulated on a transport task and training data is acquired for the RL algorithm. Eventually the RLs approximation of the Q-function converges and then it can be evaluated and compared to other strategies.

The driving situation is expressed as a 'state' that the RL algorithm can use and two different states are evaluated. One is the battery State of Charge (SoC) and the trucks position along the transport task and one is the battery SoC and the altitude change the upcoming 250m and 3km of the road. In the simulations the position and altitude changes can easily be calculated since the transport task is known. In practise such information about the road can be obtained by mapping the road ahead of time and then use the Global Positioning System (GPS) to track the position of the truck.

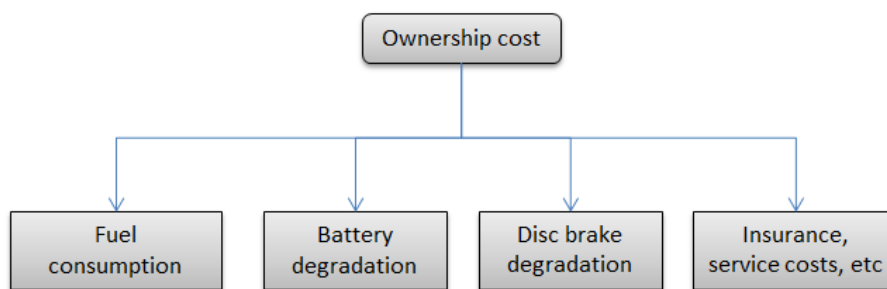


Figure 1.3: Schematic view of the ownership cost of a long haul hybrid truck and its components. The components considered in this thesis project are costs from fuel consumption, battery degradation and disc brake degradation.

The goal of the RL algorithm is to set the eqf dynamically throughout a transport task in order to minimize the ownership cost (OC) of the truck. Due to the nature of RL it could in theory find optimal strategies for setting the eqf with respect to anything the user wants [5] e.g. the *entire* OC of the truck. Some examples of what is included in the OC of a long haul hybrid truck can be seen in Figure 1.3. Looking at the whole OC is however outside the scope of this thesis work and thus the following components are considered:

- Fuel consumption
- Battery degradation
- Disc brake degradation

The OC achieved when evaluating the RL based eqf controller is compared to the OC of a conventional truck, as well as a controller where the eqf is constant throughout the entire transport task. The eqf is set so that the SoC of the battery at the end

of the transport task is the same as the SoC at the beginning of the transport task, so that the battery power used is perfectly balanced. The constant eqf controller is known to be an optimal controller for some truck hardware[1], e.g. large battery sizes and thus it is interesting to see how the RL based controller compares against this and if it learns a similar strategy.

1.2 Main contributions

1. A new approach for setting the eqf in ECMS using a RL based controller has been developed and evaluated.
2. When using the default hardware set up, where the specifications on the ENG, EM and battery were provided by Volvo AB, the RL based controller lowered the OC compared to a conventional truck but not compared the constant eqf controller.
3. When using a smaller battery or more expensive disc brakes in comparison to the hardware used in (2) the RL based eqf controller lowered the OC compared to both the conventional truck and the constant eqf controller.
4. The RL based eqf controller, for the two different states evaluated, in both cases find strategies for setting the eqf factor rivalling the constant eqf controller.

1.3 Terms used in the report

Below a table showing abbreviations used throughout the report is presented.

Abbreviation	Description
SoC	State of Charge, current charge of the battery.
SoH	State of Health, measures degradation of component.
ENG	Internal Combustion Engine.
EM	Electric machine.
RL	Reinforcement learning.
ESS	Energy Storage System.
eqf	Short for Equivalence Factor.
ECMS	Equivalent Consumption Minimization Strategy, torque splitter.

Table 1.1: Table describing the abbreviations used in the report.

2

Transport task and vehicle dynamics model

In this chapter the transport task and vehicle dynamics model, mentioned in Section 1.1, are described in more detail. The transport task is needed in order to provide the simulated truck with information about the road slope and the velocity of the truck. The vehicle dynamics model uses this information in order to calculate the angular velocity of the wheels ω_{wheel} and the propulsion torque T_{prop} that is needed at the wheels in order to maintain or reach the velocity given. ω_{wheel} and T_{prop} are then used by other modules in the virtual test bench, see Figure 1.2.

2.1 Transport task

As described earlier in Section 1.1 a transport task is a function where the slope of the road and the velocity of the truck is expressed as a function of time. Since a transport task can be several hours long it is inconvenient to save the slope and velocity at every instant, instead they are saved in segments:

$$f_v(t_i) = v_i \quad (2.1)$$

$$f_\alpha(t_i) = \alpha_i \quad (2.2)$$

$$i \in [1, N] \quad (2.3)$$

where t_i denotes the time at the beginning of the i th segment, N is the total number of segments while α_i and v_i is the slope of the road and the velocity of the truck at time t_i .

In order to process the information of the transport task as a continuous function $f(t)$, where $t_i \leq t < t_{i+1}$, the value for $f(t)$ is linearly interpolated between the timesteps t_i and t_{i+1} as:

$$f(t) = f(t_i) + \frac{f(t_{i+1}) - f(t_i)}{t_{i+1} - t_i} \cdot t. \quad (2.4)$$

The transport task used is Borås-Landvetter-Borås and is provided by Volvo AB.

2.2 Vehicle dynamics

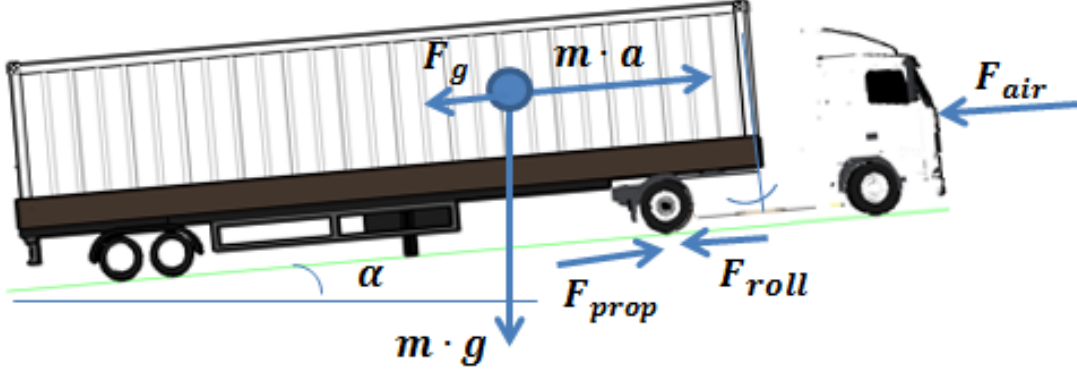


Figure 2.1: A truck with all forces considered in the model.

Considering the truck with the forces shown in Figure 2.1 the trucks acceleration may be written as:

$$m \cdot a(t) = F_{prop}(t) - F_g(t) - F_{roll}(t) - F_{air}(t) \quad (2.5)$$

where $F_{prop}(t)$ is the propulsion force, $F_g(t)$ is the gradient force, $F_{roll}(t)$ is the rolling resistance and $F_{air}(t)$ is the air drag. These forces are modelled as:

$$F_g(t) = mg \sin(\alpha(t))$$

where m is the truck weight, g the gravitational acceleration and α is the slope of the road, as seen in Figure 2.1. The rolling resistance is modelled as:

$$F_{roll}(t) = c_r mg \cos(\alpha(t))$$

where c_r is the rolling resistance coefficient. Lastly the air drag is modelled as:

$$F_{air}(t) = \frac{1}{2} \rho_a A_f c_d v(t)^2$$

where ρ_a is the density of air, A_f is the frontal area of the truck, c_d is the drag coefficient and v is the velocity of the truck.

F_{prop} is the propulsion force necessary in order to meet a specific acceleration a . As was specified in Section 2.1 a set speed of the vehicle is given from the transport task hence $a(t)$ is set so that this velocity is reached in one time step of the simulation. With only F_{prop} being unknown the equation (2.5) is rewritten as:

$$F_{prop}(t) = m \cdot a(t) + F_g(t) + F_{roll}(t) + F_{air}(t) \quad (2.6)$$

$$T_{prop}(t) = (m \cdot a(t) + F_g(t) + F_{roll}(t) + F_{air}(t)) \cdot r_w \quad (2.7)$$

Where $T_{prop}(t)$ is the propulsion torque needed to accelerate with $a(t)$ at time t and r_w is the radius of the wheel. Finally the angular velocity of the wheel is calculated as:

$$\omega_{wheel}(t) = \frac{v(t)}{r_w} \quad (2.8)$$

3

Powertrain model

Using the model for the vehicle dynamics, described in Section 2.2, a powertrain model of the truck is needed in order to describe how the torques and forces propagate throughout the truck. A schematic of the powertrain model used is again for convenience shown in Figure 3.1.

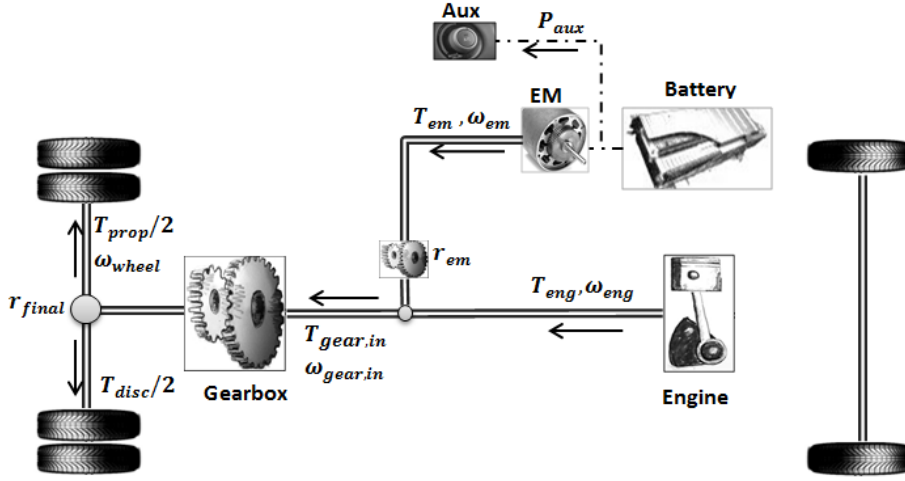


Figure 3.1: Schematic of the powertrain model used. The images illustrate where the different torques T , angular velocities ω and gear ratios used in equation (3.1) to (3.5) are positioned in the powertrain. The arrows show how the torque is transferred from the ENG and EM to the wheels.

A complete description of the powertrain model can be found in Appendix A, e.g. how the fuel consumption and battery degradation are calculated, and only the necessary equations are listed here:

$$T_{prop}(t) = (m \cdot a(t) + F_g(t) + F_{roll}(t) + F_{air}(t)) \cdot r_w \quad (3.1)$$

$$T_{prop}(t) = T_{gear,in}(t) \cdot r_{gearbox}(t) \cdot r_{final} \cdot \varphi(T_{prop}(t), \eta_{gearbox}) + T_{disc}(t) \quad (3.2)$$

$$T_{gear,in}(t) = T_{eng}(t) + T_{em}(t) \cdot r_{em} \quad (3.3)$$

$$\omega_{wheel}(t) = \frac{v(t)}{r_w} \quad (3.4)$$

$$\omega_{wheel}(t) = \frac{\omega_{gear,in}(t)}{r_{gearbox}(t) \cdot r_{final} \cdot \varphi(T_{prop}(t), \eta_{gearbox})} \quad (3.5)$$

$$\omega_{gear,in}(t) = \omega_{eng}(t) = \frac{\omega_{em}(t)}{r_{em}} \quad (3.6)$$

$$\varphi(x, \eta) = \begin{cases} \eta & x \geq 0 \\ \frac{1}{\eta} & x < 0. \end{cases} \quad (3.7)$$

Here $T_{prop}(t)$ is the propulsion torque needed to accelerate with $a(t)$ at time t , implicitly given from the transport task. Equation (3.1) and (3.4) are derived in Section 2.2. In Equation (3.3), $T_{gear,in}(t)$ is the shaft torque going into the gearbox, $r_{gearbox}$ is the gear ratio of the gearbox and r_{final} is the final ratio between the gearbox and the wheels. φ is the function described in Equation (3.7) and T_{disc} , T_{eng} and T_{em} are the torques produced by the disc brakes, combustion engine and electric machine respectively.

Lastly, v is the speed of the truck and ω_{wheel} is the rotation speed of the wheels, $\omega_{gear,in}$ is the rotation speed of the shaft before entering the gearbox while ω_{eng} and ω_{em} are the angular velocities of the ENG and the EM respectively, see Figure 3.1. With T_{prop} given from Equation (3.1), and ω_{wheel} given from Equation (3.4), the only other unknowns are T_{eng} , T_{em} and T_{disc} , which are set by the torque splitter described in section 5.2.

The powertrain is also able to act as a powertrain for a conventional truck. In this case the only logic that is needed is the split between T_{eng} and T_{disc} , which is described in Section 5.2. Details for how the powertrain needs to be changed in order to be run as a conventional truck is presented in Appendix B.

4

Economical modeling

In order to evaluate the hybrid truck a function is needed that gives a measure on how well the truck did on a transport task, see Section 2.1. From the standpoint of someone investing in a hybrid truck the ownership cost of the truck should be as low as possible, thus the truck is evaluated based on its ownership cost during a transport task.

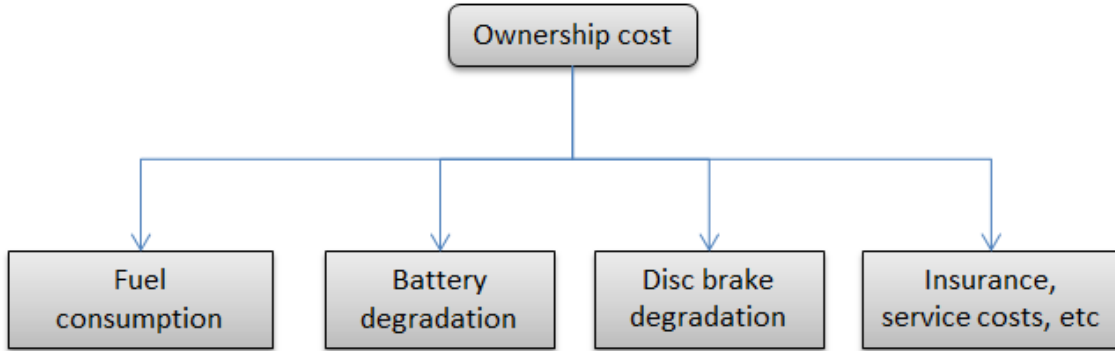


Figure 4.1: Schematic view of the different components of the ownership cost of a long haul truck. The components considered in this report are costs from fuel consumption, battery degradation and disc brake degradation.

The ownership cost of a hybrid truck has a myriad of components which was illustrated in Figure 1.3, and here again for convenience in Figure 4.1. Since the problem quickly becomes more complex as additional terms are considered the conventional approach so far has been to only consider the cost from fuel consumption. However, due to the nature of reinforcement learning (RL) any and all components of the ownership cost can now be considered, more details on this in Section 5.3.2. The focus has been on the costs originating from three different sources:

- Fuel consumption
- Battery degradation
- Disc brake degradation

Considering this many terms is something unique to this thesis project. The function that describes the ownership cost, which ought to be minimized, thus becomes:

$$OC = \Delta m_{fuel} \cdot C_{fuel} + \Delta \text{SoH}_{batt} \cdot C_{batt} + \Delta \text{SoH}_{disc} \cdot C_{disc} \quad (4.1)$$

4. Economical modeling

Here OC is the ownership cost, C_{fuel} is the fuel cost per unit volume while C_{batt} and C_{disc} are the costs of the battery and disc brakes. Δm_{fuel} is the amount of fuel consumed during the transport task, ΔSoH_{batt} is the change in the battery State of Health (SoH) and ΔSoH_{disc} is the change in the SoH of the disc brakes. In this report SoH is a variable that measures the condition of some hardware, e.g. the battery or disc brakes. If SoH= 1 the hardware is in perfect condition and if SoH= 0 it needs to be replaced.

For details on how ΔSoH_{batt} and ΔSoH_{disc} are defined see Appendix A and for the different costs C_{fuel} , C_{batt} and C_{disc} see Appendix C.

5

Powertrain controllers

This chapter regards how the gear controller, the RL based controller and the torque splitter works. These three components play a crucial role in how the EM and ENG are used and thus directly influences the OC of the truck, explained in Chapter 4. For convenience Figure 1.2 is again here shown in Figure 5.1, which illustrates how the different components fit into the virtual test bench.

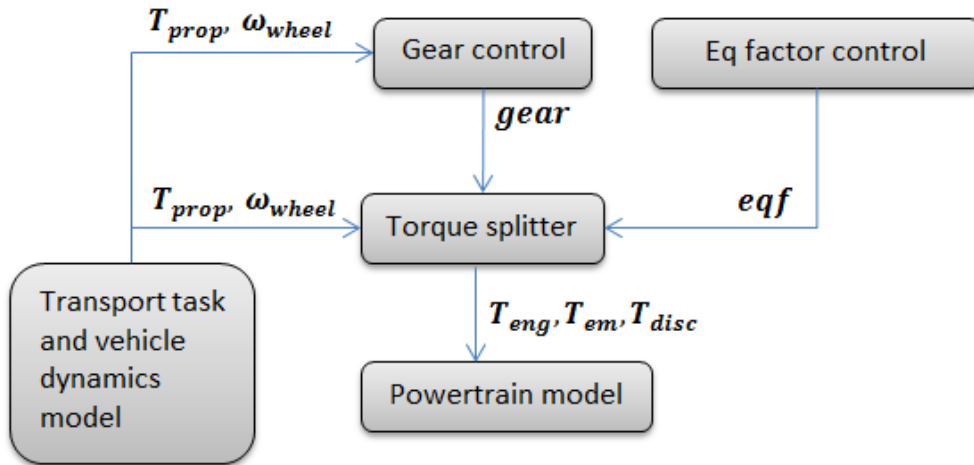


Figure 5.1: Schematic of the control architecture, and its different components, used in the virtual test bench. The RL based controller here acts as the Eq factor controller.

5.1 Gear controller

Here the gear controller, a part of the virtual test bench shown in Figure 5.1, is described in detail. The gear controller calculates the current gear of the truck which in turn determines a certain gear ratio $r_{gearbox}$. $r_{gearbox}$ is used in order to calculate $T_{gear,in}$ and $\omega_{gear,in}$ (see Equation (3.2) and (3.4)) and since both these parameters are used in order to calculate the torque split, see Section 5.2, using the correct gear is crucial in a hybrid truck.

However, in order to mainly focus on the reinforcement learning a simple, rule based, gear controller was chosen. The result was a function that mapped a $(T_{prop}, \omega_{wheel})$ tuple (see Chapter 3 for a description of these variables) to the gear that maximizes the efficiency of the ENG.

The function used a gear map that contained, for some discretized propulsion torques T_{prop}^i and wheel angular velocities ω_{wheel}^j , the gear that maximized the efficiency of the ENG. The rest of the function was a logic that mapped a $(T_{prop}, \omega_{wheel})$ tuple to a gear in the gear map and for any given tuple the logic could choose between four possible gears, as shown in Figure 5.3.

The developed logic is illustrated in Figure 5.2. In short: if T_{prop} is lower than the lowest mapable torque $T_{min} < 0$ the logic chooses the gear that maximizes $T_{gear,in}$ (see Equation (3.3)), which will be the lowest gear allowed by the ENG and EM. If not the logic checks the sign of the propulsion torque and if it is positive it chooses the lowest of the nearby gears, as seen in Figure 5.3, and otherwise it chooses the highest.

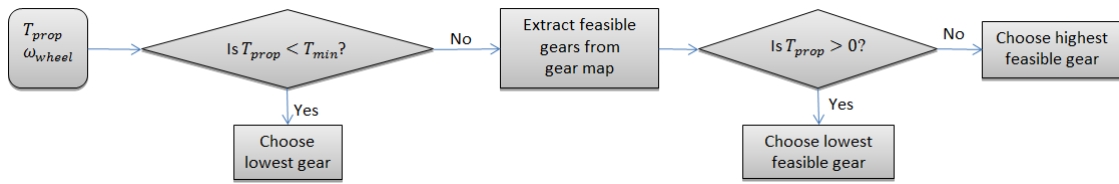


Figure 5.2: A flow chart that illustrates the logic of mapping a $(T_{prop}, \omega_{wheel})$ pair to a gear. The feasible gears are chosen as is depicted in Figure 5.3.

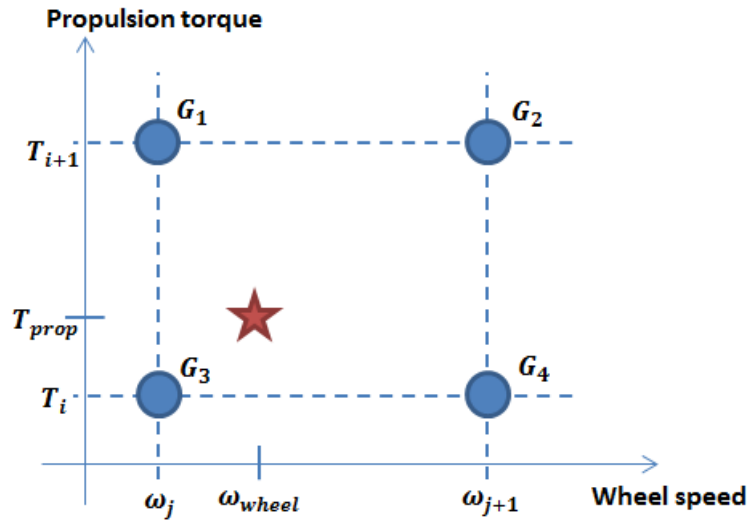


Figure 5.3: Illustration of how the four gears, G_1 , G_2 , G_3 , G_4 , are chosen from a given $(T_{prop}, \omega_{wheel})$ tuple. The gear itself is chosen according to the logic in Figure 5.2. $(T_{prop}, \omega_{wheel})$ is shown as the red star and the possible gears that the logic can choose between are shown as the blue circles. T_i , ω_j are discretized propulsion torques and wheel angular velocities.

5.2 Torque splitter

Here the torque splitter, introduced in Section 1.1, is described in more detail. The torque splitter used is called Equivalent Consumption Minimisation Strategy (ECMS) and finds the torque split that minimizes the current power used by the truck, given a parameter λ called the equivalence factor (eqf). The eqf sets a price of using battery power compared to using fuel power and it is this parameter that is set by the RL based controller, described in Section 5.3.

5.2.1 Cost function

The cost function, which the ECMS minimizes, is described below together with some other important equations:

$$f_{cost}(T_{gear,in}, \omega_{gear,in}) = P_{fuel}(T_{eng}, \omega_{eng}) + \lambda \cdot P_{batt}^{inner}(T_{em}, \omega_{em}) \quad (5.1)$$

$$T_{gear,in} = T_{eng} + r_{em} \cdot T_{em} \quad (5.2)$$

$$\omega_{gear,in} = \omega_{eng} = \frac{\omega_{em}}{r_{em}} \quad (5.3)$$

$$T_{prop} = T_{gear,in} \cdot r_{gearbox} \cdot r_{final} \cdot \varphi(\text{sign}(T_{prop}), \eta_{gearbox}) + T_{disc} \quad (5.4)$$

Here $P_{fuel}(T_{eng}, \omega_{eng})$ is the fuel power needed to meet a specific torque T_{eng} , with the ENG rotation speed ω_{eng} , while $P_{batt}^{inner}(T_{em}, \omega_{em})$ is the necessary battery power needed to meet the torque T_{em} with EM rotation speed ω_{em} . $\omega_{gear,in}$ is calculated given the truck velocity and the current gear (see Equation (3.5)) so the cost function is minimized with respect to T_{eng} and T_{em} .

λ is a parameter of the cost function, called the equivalence factor (eqf). The eqf sets the cost of using battery power compared to fuel power. As can be seen in Equation (5.1) a small λ will result in a low cost for using battery power and a high λ it will be more expensive to use battery power, compared to fuel power.

T_{disc} is also a variable set by the torque splitter but it is not used in the cost function but as a part of Equation (5.4). The disc brakes are only used as a last resort if the ENG and EM cannot produce enough braking torque, which is illustrated in Figure 5.4. As illustrated by the red line the propulsion torque T_{prop} is provided by the EM and ENG until they both reach their limit $T_{em,min}$ and $T_{eng,min}$. After that the disc brakes help by providing the necessary extra torque T_{disc} in order to meet T_{prop} .

A problem with evaluating Equation (5.1) however is that P_{fuel} and P_{batt}^{inner} are not given as continuous functions but are discrete maps, which are calculated from loss maps of the EM and ENG. In order to evaluate the cost function (Equation (5.1)) for torques and velocities between data points linear interpolation was used which still ensures that the global minimum of the cost function is at a data point of either of the power maps, or at the boundary given by the constraints. Then, by evaluating the cost function at these points, one can calculate which T_{em} , T_{eng} and T_{disc} that minimizes the cost function while satisfying all constraints, and use that as the torque split. For further details on ECMS, as well as the constraints used in the minimization, see Appendix B.

If the truck for some reason is unable to reach the acceleration $a(t)$ (described in Section 2.2) the ECMS gives an emergency torque split (ENG supplies as much

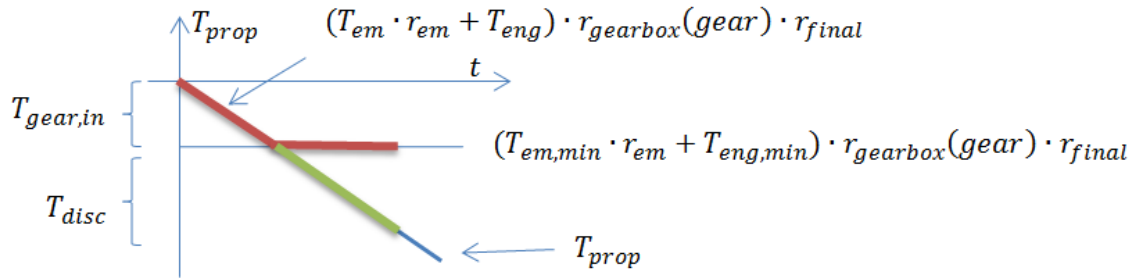


Figure 5.4: Illustration of how T_{em} , T_{eng} and T_{disc} are used. The figure shows a propulsion torque T_{prop} that decreases linearly as a function of time. The propulsion torque is supplied first by the EM and ENG, and when they reach their limits the rest of the propulsion torque is supplied by the disc brake.

torque as it can) and then the transport task continues. This is later utilized in Section 5.3 in order to remove unwanted behaviour from the RL agent.

5.3 RL based controller

Reinforcement learning (RL) is a machine learning method used for finding optimal solutions to a certain class of problems [5]. In order to use RL the problem has to be expressed in terms of actions, rewards and states, which are described in more detail in Sections 5.3.1 to 5.3.3. The RL algorithm (or RL agent) sets the equivalence factor (eqf) in the ECMS dynamically throughout the transport task. The strategy the RL agent uses to set the eqf along the transport task is called a policy, which eventually will converge to an optimal (or close to optimal) policy. The OC when following such an optimal policy is compared against the OC of a conventional truck and the constant eqf controller in Section 7.

For more information about RL in general see Richard S Sutton and Andrew G Bartos book Reinforcement Learning: an Introduction [5].

5.3.1 Action

As described in Section 1.1 the RL algorithm sets the eqf dynamically during the transport task. Thus the action of the RL agent is the eqf used in the ECMS torque splitter.

5.3.2 Reward

The cumulative reward, or return, throughout the transport task is what the RL agent maximizes and thus the reward expresses what the agent should learn. This is expressed as different terms in the reward function. The first term is the ownership cost OC , defined in Chapter 4, which expresses that the agent should minimize the ownership cost. As a reminder it is again written here in Equation (5.5):

$$OC = \Delta m_{fuel} \cdot C_{fuel} + \Delta \text{SoH}_{batt} \cdot C_{batt} + \Delta \text{SoH}_{disc} \cdot C_{disc} \quad (5.5)$$

The second term discourages the RL agent from leaving the battery with less charge than a certain reference $SoC_{end,ref}$. This is so that the learnt policy easier can be compared to other energy management systems. This reward is only given at the end of a transport task and is defined as

$$R_{term} = R_{terminal} \cdot term(s) \quad (5.6)$$

where:

$$term(s) = \begin{cases} 0 & \text{State } s \text{ not terminal} \\ 1 & \text{State } s \text{ terminal} \end{cases} \quad (5.7)$$

$$R_{terminal} = \begin{cases} k_1 \cdot (SoC_{end} - SoC_{end,ref}) & SoC_{end} \geq SoC_{end,ref} \\ k_2 \cdot (SoC_{end} - SoC_{end,ref}) & SoC_{end} < SoC_{end,ref} \end{cases} \quad (5.8)$$

where a terminal state is a state where the transport task ends and the simulations stops. $SoC_{end,ref}$ is a reference for the final SoC, SoC_{end} is the SoC at the terminal state and $k_1, k_2 \geq 0$.

The last term is set so that a penalty is given to the RL agent every time step the ECMS fails and cannot find a torque split that satisfies the current T_{prop} . Because of the nature of the transport task chosen, Borås-Landvetter-Borås (given by Volvo AB), the only reason this can happen is if the battery is close to completely discharged so that the EM cannot be used. If this happens the ECMS yield an emergency torque split and then the simulation continues, see Section 5.2. This reward is modelled as:

$$R_{fail} = -n_{fails} \quad (5.9)$$

where n_{fails} is the number of times the ECMS failed to find a torque split since the last action was taken.

All these terms together create the reward function:

$$R = OC + R_{terminal} \cdot term(s) + R_{fail} \quad (5.10)$$

Note that the OC is negative, which means that if the ownership cost increases the reward function will decrease and vice versa.

5.3.3 State

A state in a RL setting is all the information available to the RL algorithm. Two different states has been examined:

- State 1: SoC and position along transport task.
- State 2: SoC, altitude difference 250m forward ($dh250m$) and altitude difference between 250m and 3km forward ($dh3km$).

where SoC is the state of charge of the battery and the position is defined as

$$\text{position} = \frac{d}{d_{max}} \quad (5.11)$$

where d is the distance traveled by the truck along the transport task and d_{max} is the length of the entire transport task. The second state is defined just as written above but is also illustrated in Figure 5.5.

The states in RL should include all past information in order for the RL agent to learn what action that is optimal from each state. Mathematically this can be expressed as

$$\Pr \{R_{t+1} = r, S_{t+1} = s' \mid S_t, A_t\}. \quad (5.12)$$

That is, the probability of receiving reward r and ending up in state s' only depends on the previous state S_t and the action taken from that state A_t . A state, and the RL problem as a whole, is said to have the Markov property if Equation (5.12) is satisfied. Since the reward and next state, using either state 1 or 2, in our RL problem can be expressed as Equation (5.12) our problem also has the Markov Property.

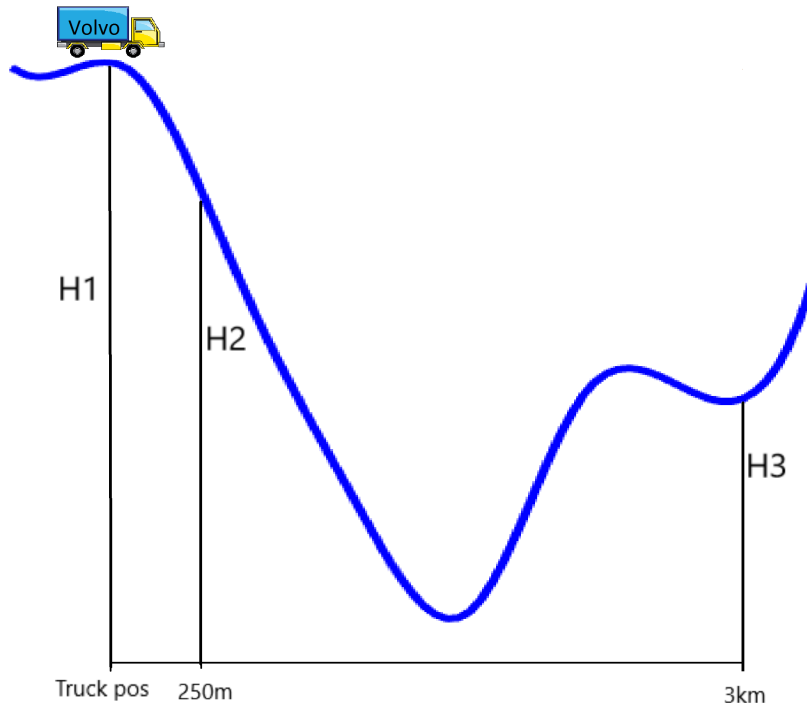


Figure 5.5: Illustration showing the height differences used in state 2, see Section 5.3.3. The truck looks at the current altitude as well as the altitudes 250 m and 3 km ahead. Then the height differences are calculated as $dh_{3km} = H3 - H2$ and $dh_{250m} = H2 - H1$.

5.3.4 Q-learning

In order to learn what action that is optimal from every state a state-action function is used, called the Q-function, written as $Q_\pi(s, a)$. Here π defines a policy, where an ϵ -greedy policy has been used. The ϵ -greedy policy chooses an action a according to:

$$a = \underset{a'}{\operatorname{argmax}} [Q_\pi(s, a')] \quad (5.13)$$

but with a probability ϵ instead chooses an action randomly. In Equation (5.13) the policy π also is ϵ -greedy.

The output of the Q-function $Q_\pi(s, a)$ is the expected return (see Section 5.3.2) of the RL agent, given that after taking the action a , from state s , the agent will follow the policy π . Since the Q-function is not immediately known to the agent it has to be learned, which is done with the Q-learning algorithm shown in Algorithm 1. By alternating between taking random and greedy actions, with respect to $Q_\pi(s, a)$, the agent eventually learns the Q-function and can then, with $\epsilon = 0$, choose the optimal action according to Equation 5.13.

Lastly a tabular Q-function is used, which means that the Q-function is only evaluated for some discrete, finite number of states and actions. For details on how the states and actions are discretized see Appendix C

Algorithm 1 Pseudo code for the Q-learning with experience replay. To learn more about different parameters see Sutton and Bartos book on reinforcement learning [5]

```
1: Initialize  $Q(s, a), \forall s \in \mathcal{S}, a \in \mathcal{A}(s)$ , and  $Q(\text{terminal state}) = 0$ 
2: Initialize  $memory = \{\}$ 
3: for each simulation do
4:   Initialize  $S$ 
5:   while  $S$  is not terminal do
6:     Choose  $A$  from  $S$  using policy derived from  $Q$  (e.g.  $\epsilon$ -greedy)
7:     Take action  $A$ , observe  $R, S'$ 
8:     Append  $(S, A, R, S')$  to  $memory$ 
9:      $S \leftarrow S'$ 
10:  end while
11:   $n = 0$ 
12:  while  $n < n_{max}$  do
13:    Sample random  $(S, A, R, S')$  from  $memory$ 
14:     $Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma \max_a Q(S', a) - Q(S, A)]$ 
15:     $n = n + 1$ 
16:  end while
17: end for
18: Return  $sum$ 
```

6

Simulations with constant equivalence factor

This chapter presents simulation results where the equivalence factor is held constant throughout the transport task. The motivation for this is to achieve benchmark values with which the RL based controller can be compared against later in Chapter 7.

It should be noted however that this approach is difficult to implement in practise since all future knowledge about the transport task is needed, which is explained in Section 6.1.

6.1 Studied vehicle and transport task

The parameter values used when evaluating the truck using a constant equivalence factor can be seen in Table 6.1.

Parameter	Description	Value
ESS size	Size of battery	9.24 kWh
m_{truck}	Mass of truck	$3.5 \cdot 10^4$ kg
dt	Timestep of the simulation	1 s
C_{fuel}	Cost of the diesel fuel	1.4 €/liter
C_{batt}	Cost of the battery	$1.5 \cdot 10^4$ €/pack
C_{disc}	Cost of the disc break system	$5 \cdot 10^3$ €/set
Eqf	Equivalence factor	2.058
SoC_{init}	Initial SoC	0.5

Table 6.1: Specifications and initial setting for the truck used in the simulations.

The transport task used was Borås-landvetter-Borås, provided by Volvo AB. The eqf was set to the value which perfectly balanced the use of fuel and battery energy, leaving the SoC at the end of the transport task at the same value as the initial SoC, SoC_{init} . In order to do this in practise one needs to know $T_{gear,in}(t)$ (described in Chapter 3) before hand for every time t in order to predict the battery power used throughout the transport task. The eqf can however easily be set empirically to the right value in the simulations, just by trial and error. Since the eqf is set in this way the controller knows every $T_{gear,in}(t)$ throughout the transport task, which gives it an advantage towards the RL based controller. Thus the results presented here should be regarded as an optimistic benchmark that is difficult to beat.

Simulations were run for other cost of the fuel, battery, disc brakes and also for a smaller ESS size. The results from all these simulations are displayed in Table 8.2 in Chapter 8.

6.2 Results - Default truck

An comparison between the constant equivalence factor and a conventional truck, where the default truck settings in Table 6.1 were used, can be seen in Figure 6.1. In Figure 6.2 an overview of truck behaviour during the transport task can be seen. It is clear that the hybrid truck outperforms the conventional truck, lowering the ownership cost by $\sim 6.2\%$ in comparison. In Figure 6.2 it can be seen that the gear controller, for the most part, chooses a high gear except when the truck needs to brake (indicated by a negative power need at the wheel), which is an artefact from how the transmission map was made. Since the gear chosen at any time only depends on the T_{prop} and ω_{wheel} the gear profile is the same for the results presented in Chapter 7 as well.

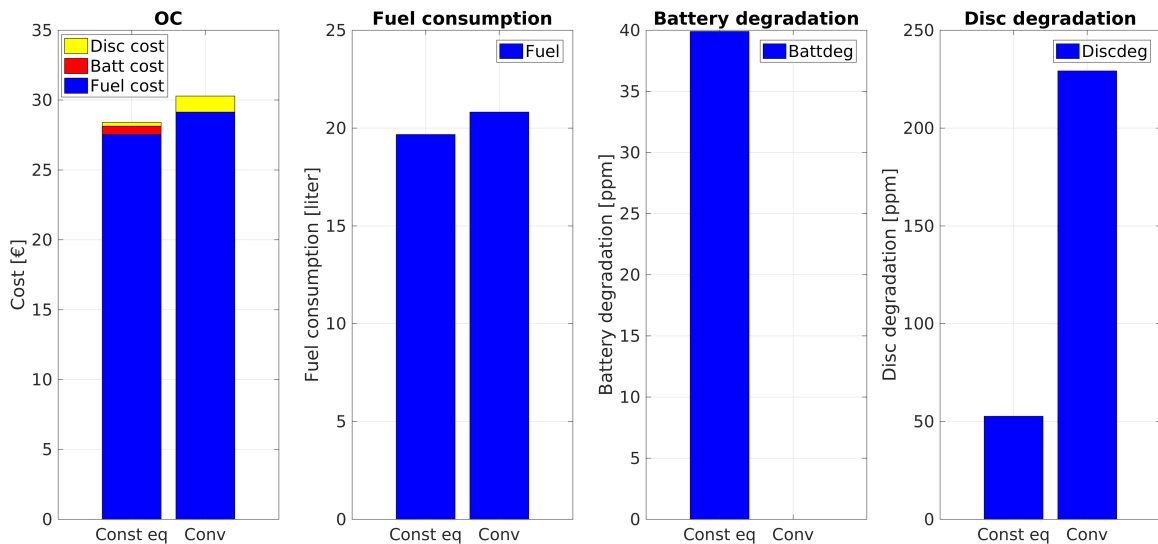


Figure 6.1: Barplots comparing the constant eqf and a conventional truck. The leftmost barplot shows the ownership cost. The next plot to the right illustrates the fuel consumption, the third the battery degradation and the final plot the disc degradation. The conventional truck does not have any battery degradation since it does not have an ESS.

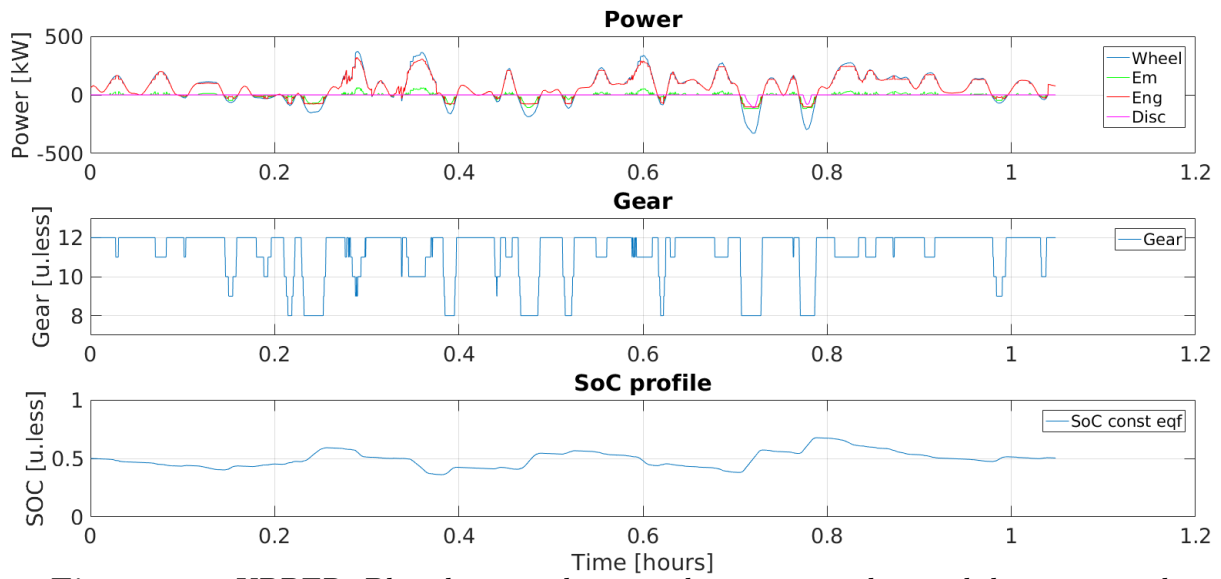


Figure 6.2: UPPER: Plot showing the propulsion torque the truck has to provide and how much that is provided by the EM, ENG and disc brakes. MIDDLE: Plot of the gear profile during the transport task. LOWER: Plot of the SoC profile during the transport task.

7

Simulations with RL based equivalence factor

This chapter presents the results when a RL based controller is used to dynamically set the equivalence factor in ECMS during the course of the transport task. The results from the RL based controller are compared against the results from Chapter 6.

7.1 Studied vehicle and transport task

The parameter values used in the truck simulations, using the RL based controller, can be seen in Table 7.1

Parameter	Description	Value
ESS size	Size of battery	9.24 kWh
m_{truck}	Mass of truck	$3.5 \cdot 10^4$ kg
dt	Timestep of the simulation	1 s
dt_{train}	Timestep between actions of RL agent	50 s
α	Learning rate of RL algorithm	0.5
γ	Discount factor of RL algorithm	1
SoC_{init}	Initial SoC when evaluating	0.5
$SoC_{end,ref}$	Referens for final SoC	0.5
fuel cost	Cost of the diesel fuel	1.4 €/liter
ESS cost	Cost of the battery	$1.5 \cdot 10^4$ €/pack
Disc brake cost	Cost of the disc brake system	$5 \cdot 10^3$ €/set

Table 7.1: Specifications and initial setting for the truck used in the simulations.

The RL agent used an ϵ -greedy policy, where the parameter ϵ was set as:

$$\epsilon = \min\left(1, \frac{25}{i}\right) \quad (7.1)$$

where i is i th training simulation. The transport task used was Borås-Landvetter-Borås, provided by Volvo AB. Simulations were run for other cost of the fuel, battery, disc brakes and also for a smaller ESS size. The results from all these simulations are displayed in Table 8.2 in Chapter 8. Also, a plot illustrating the altitude and gear profile for all these cases can be seen in Appendix C.

7.2 Results

The RL agent was trained for 12000 simulations on a transport task and then evaluated on the same transport task with a greedy policy, as shown in Equation (5.13). The exact ownership cost for each simulation can be found in chapter 8, where the following sections focus more on the comparison between the RL based controller, constant eqf and the conventional truck.

7.2.1 Default truck

The truck and RL settings were set as described in Section 7.1 and State 1 was used, as described in Section 5.3. For a more thorough list on all truck parameters, as well as how State 1 and the RL actions were discretized, see Appendix C.

A comparison between the RL agent, the constant equivalence factor and a conventional long haul truck is presented in Figure 7.2. From the figure it is clear that the hybrid truck using a RL based controller lowers the ownership cost compared to the conventional truck with $\sim 4\%$. It is however beaten by the constant equivalent factor when it comes to energy management, having a higher fuel consumption and battery degradation. This is hardly a surprise since, as mentioned in Section 1.1, using a constant eqf is proven to be optimal for larger ESS sizes.

An overview of the truck behaviour during the transport task is shown in Figure 7.2. Here the SoC dynamics during the transport task for the constant equivalence factor is also included and it is clear that the RL agent uses its battery energy in a very different way.

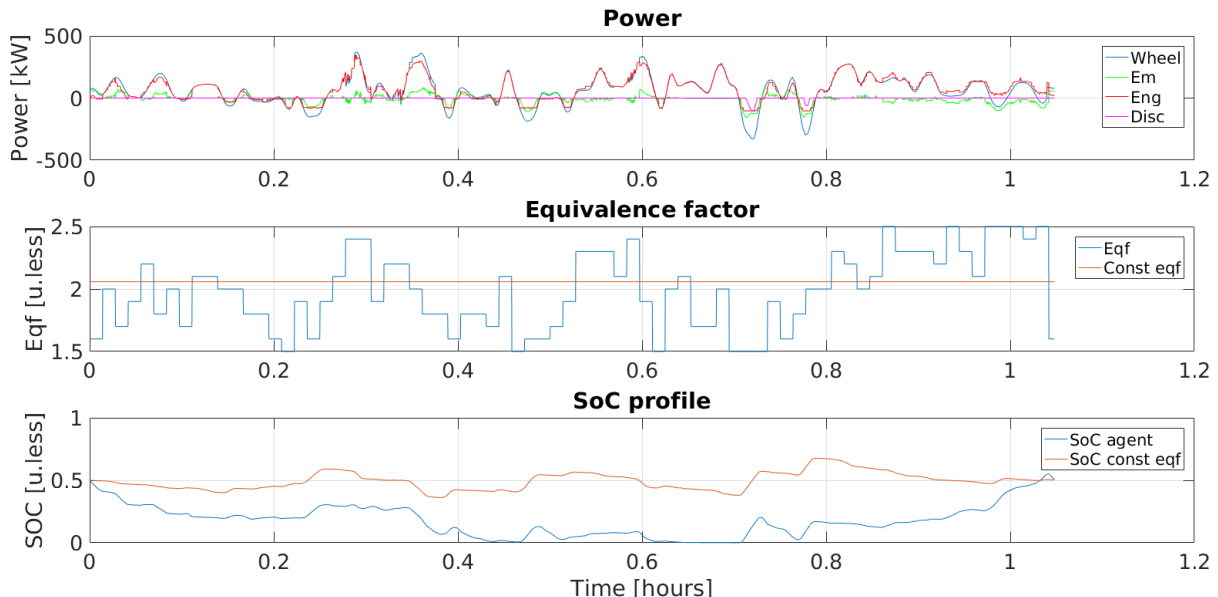


Figure 7.1: UPPER: Plot showing the propulsion torque the truck has to provide together with the torques provided by the EM, ICE and the disc brakes. MIDDLE: Plot of the equivalence factor during the transport task. LOWER: Plot of the SoC during the transport task.

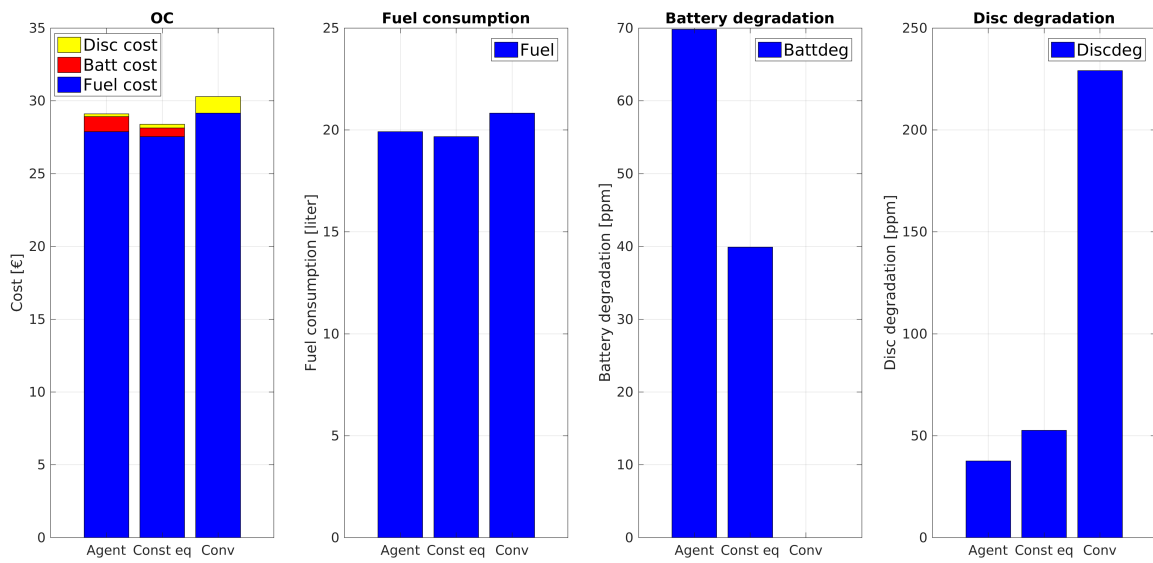


Figure 7.2: Barplots comparing the RL agent, constant eqf and a conventional truck. The leftmost barplot shows the ownership cost. The next plot to the right illustrates the fuel consumption, the third the battery degradation and the final plot the disc degradation. The conventional truck does not have any battery degradation since it does not have an battery. Note that the constant eqf has the lowest *OC* since it is a very good energy management controller, as was mentioned in Section 7.1.

7.2.2 Decreased ESS size

A second design example was evaluated with a smaller ESS, decreased from 9.24 kWh in Section 7.2.1 to 2.32 kWh here, and all other parameters and RL settings were the same as in Section 7.2.1. The results are presented in Figure 7.3 and 7.4 where it is clear that the RL agent adapted very well to a smaller battery, lowering the ownership cost by $\sim 5.7\%$ in comparison to the conventional truck and receiving a lower ownership cost than the constant equivalence factor.

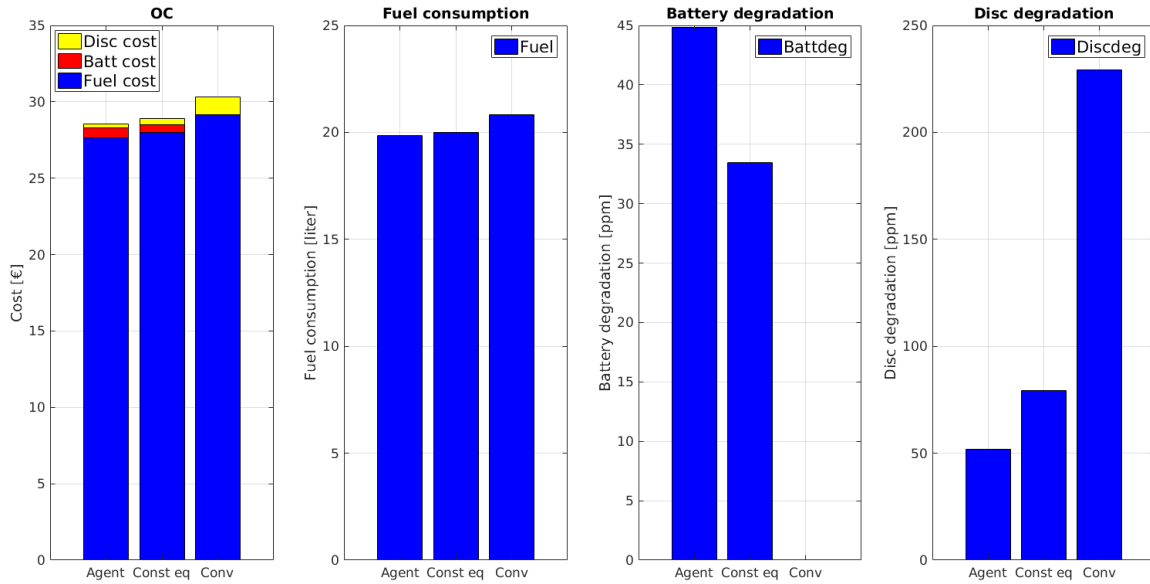


Figure 7.3: Barplots comparing the RL agent, constant eqf and a conventional truck. The leftmost barplot shows the ownership cost. The next plot to the right illustrates the fuel consumption, the third the battery degradation and the final plot the disc degradation. The conventional truck does not have any battery degradation since it does not have a battery.

That the RL based controller lowers the ownership cost with 5.7% instead of 4%, as was the case in Section 7.2.1, is counter intuitive since it is using an ESS with four times lower capacity. However, the RL agent uses the same amount of discretized states as in Section 7.2.1, but with a smaller battery, and thus the resolution of the SoC increases. A higher resolution gives the RL agent better knowledge of the actual SoC and is then able to take better actions, which could explain this effect.

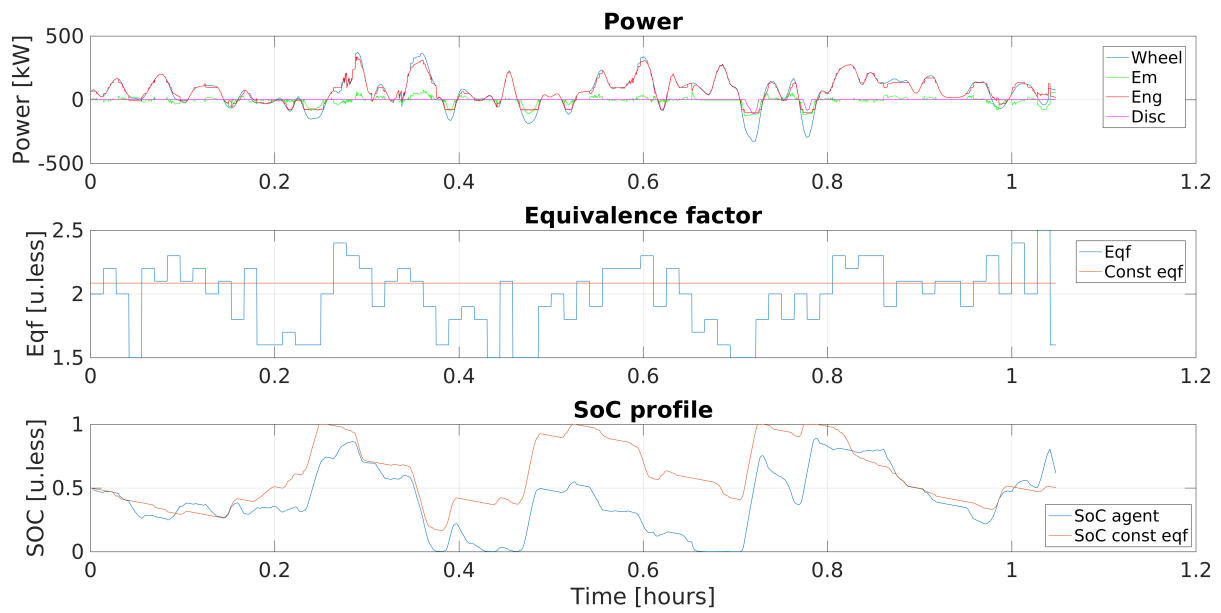


Figure 7.4: UPPER: Plot showing the propulsion torque the truck has to provide together with the torques provided by the EM, ICE and the disc brakes. MIDDLE: Plot of the equivalence factor during the transport task. LOWER: Plot of the SoC during the transport task.

7.2.3 Non-position based state

In this simulation the same settings as in Section 7.2.1 were used, with the exceptions that dt_{train} was decreased from 50 to 10 seconds and state 2, described in Section 5.3.3, was used instead of state 1. For details on how State 2 and the RL actions were discretized see Appendix C.

As can be seen in Figure 7.5 using this state also lowered the ownership cost compared to the conventional truck with approximately 2.6%. State 2 thus yields a higher ownership cost than state 1 does. One reason for this could be that state 2 does not have the Markov property, described in Section 5.3.3.

Looking at Figure 7.6 it is clear that this state representation uses its battery energy very similarly to the case with constant eqf, contrary to the results shown in Section 6.2. This might be the case since State 1, basing its state on the position of the truck, can learn more about the future parts of the transport task and thus can learn when it can have a low SoC.

Three additional plots can be seen in Figure 7.7, where the policy of the RL agent is presented as a function of the state. The different plots are shown for SoC= 0.4, 0.5 and 0.6 which is the SoC interval that the SoC stays in during the evaluation. It is clear from the figure that the RL controller sets the equivalence factor to a higher value when there is a steep down- or uphill or when the SoC is low. A lower equivalence factor tends to be set if the truck is currently in an uphill with an upcoming longer downhill (represented by the states in the upper left) or when the SoC is high. This result is intuitive since if there shortly will be a downhill the battery can be recharged and if the SoC is high there is battery energy to use.

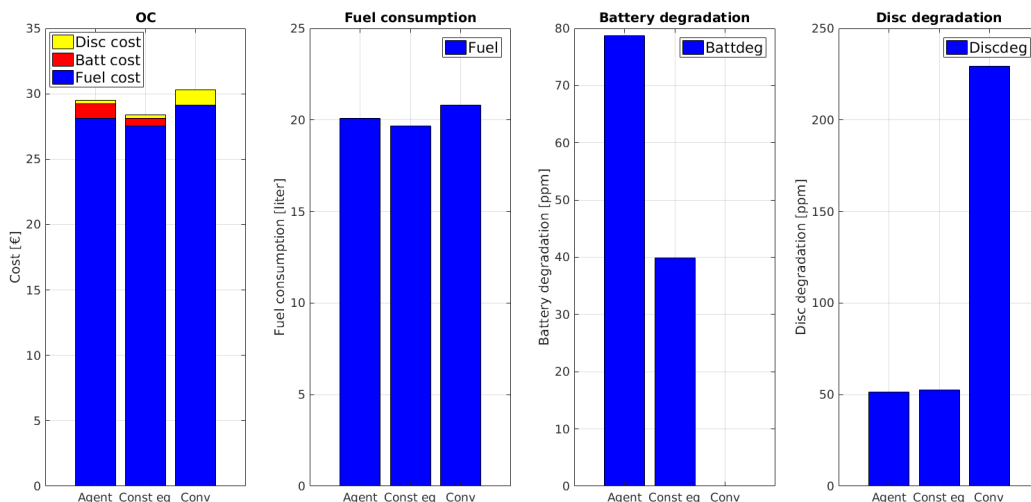


Figure 7.5: Barplots comparing the RL controller, constant eqf and a conventional truck. The leftmost barplot shows the ownership cost. The next plot to the right illustrates the fuel consumption, the third the battery degradation and the final plot the disc degradation. The conventional truck does not have any battery degradation since it does not have a battery.

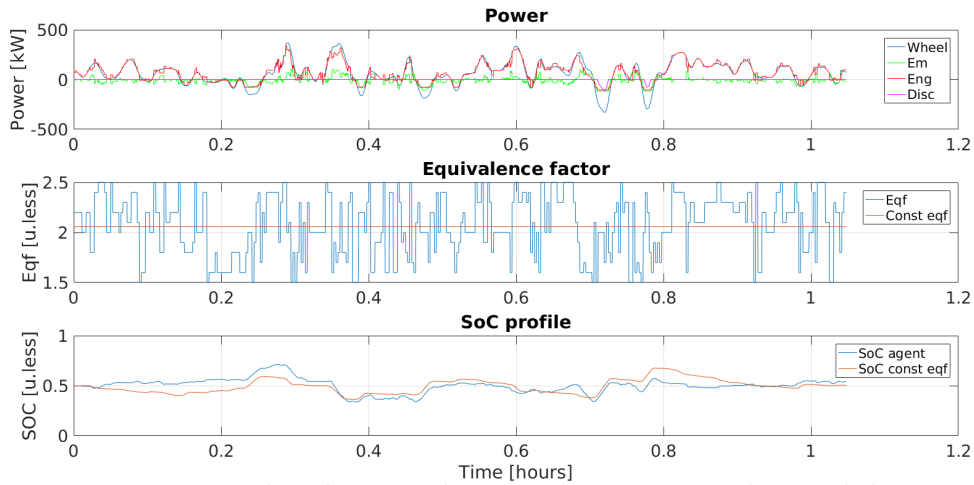


Figure 7.6: UPPER: Plot showing the propulsion torque the truck has to provide together with the torques provided by the EM, ICE and the disc brakes. MIDDLE: Plot of the equivalence factor during the transport task. LOWER: Plot of the SoC during the transport task. Note that the constant eqf has the lowest *OC* since it is a very good energy management controller.

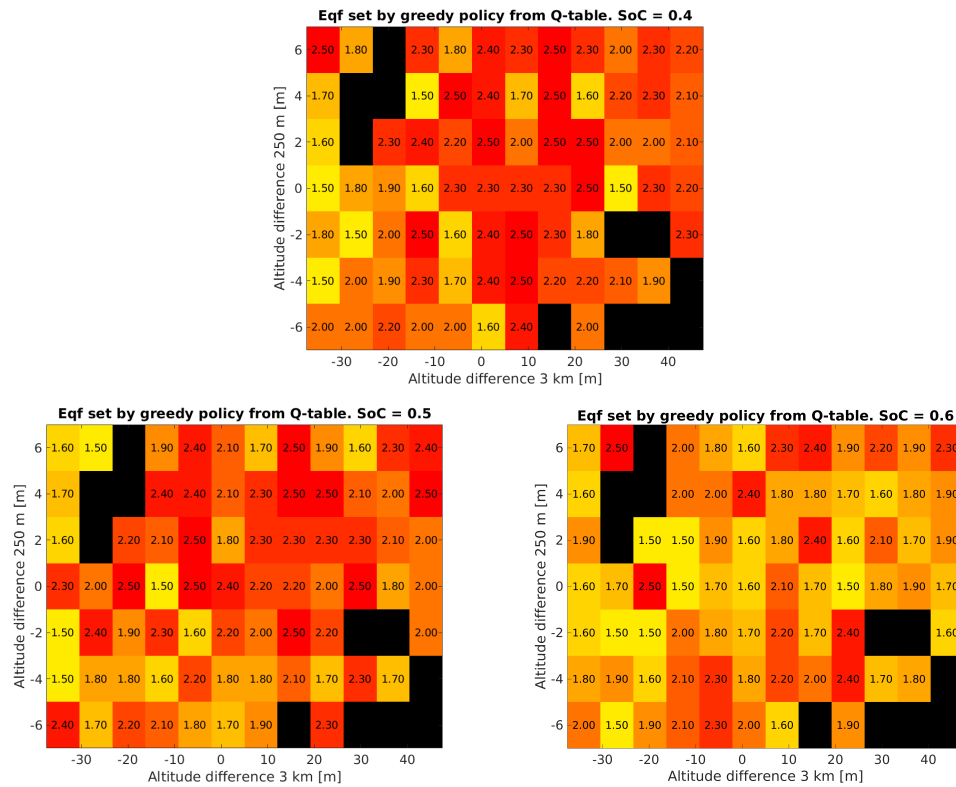


Figure 7.7: Three heat maps illustrating the policy of the RL agent during different driving conditions. Each area represents the greedy action, the eqf, taken from that state. Black areas are states the RL agent did not encounter during the training simulations and thus has no data about. UPPER: Heat map of policy with $SoC = 0.4$. LOWER LEFT: Heat map of policy with $SoC = 0.5$. LOWER RIGHT: Heat map of policy with $SoC = 0.6$.

8

Sensitivity analysis

8.1 Results

A sensitivity analysis was carried out in order to see how the RL based controller handled changes of the truck. For all simulations State 1 was used, as described in Section 5.3, and the settings from Section 7.2.1 were used. The hardware changes that were tested, as well as the default hardware settings, are presented in Table 8.1. By increasing the cost of certain components the different terms in the ownership cost, see Equation (4.1), will be of different importance for the RL controller to minimize. E.g. if fuel is very expensive it might be good to minimize the fuel cost more so than the battery degradation.

Component	Cost
ESS capacity	9.24 kWh
C_{fuel}	1.4 €/liter
C_{batt}	$1.5 \cdot 10^4$ €/pack
C_{disc}	$5 \cdot 10^3$ €/set
Expensive battery	$10 \cdot (\text{Battery cost})$
Expensive discs	$24 \cdot (\text{Disc break cost})$
Expensive fuel	$2 \cdot (\text{Diesel cost})$
Small ESS	$(\text{ESS capacity})/4$

Table 8.1: Parameter values and costs of different truck components and how they are changed for the different test cases, which can be seen in table 8.2

The results can be seen in Table 8.2, where it is clear that the RL agent rival the ownership cost of the constant eqf in all test cases. The only case where the conventional truck has the lowest ownership cost is when the battery is very expensive, which of course is because the conventional truck does not have a battery. The fact that the RL agent has a hard time beating the ownership cost of the constant equivalence factor is because this method is, as mentioned earlier in Section 1.1, very difficult to beat. It has in fact been shown that the constant equivalence factor is unbeatable for larger batteries [1], but that there is room for improvement for smaller battery sizes. This is confirmed by these results where the agents final policy beats the ownership cost of the constant equivalence factor when the battery size is decreased.

8. Sensitivity analysis

Test case	RL based controller	Constant eqf	Conventional truck
Default	29.10 €	28.43 €	30.30€
Small battery	28.57 €	28.89 €	30.30€
Expensive discs	33.73 €	34.50 €	56.65€
Expensive battery	34.10 €	33.78 €	30.30€
Expensive fuel	57.30 €	55.96 €	59.45€

Table 8.2: Comparison of the ownership cost between the RL based controller, the constant eqf and the conventional truck.

9

Conclusion

Reinforcement learning is clearly a powerful machine learning technique that not only rivals the energy management when using a constant equivalent factor but in some scenarios even surpasses it in minimizing the ownership cost. It also lowers the ownership cost in comparison to the conventional truck with $\sim 4\%$ using the default truck and with $\sim 5.7\%$ when using the smaller battery. The reason that it finds a better solution with the smaller battery pack could be because the SoC resolution becomes higher in this case, suggesting that using a continuous state space could greatly improve the results. Further the agent successfully finds good energy management policies for all test cases shown in Chapter 8.

10

Future work

Major improvement can be made on the RL algorithm. In this thesis project a tabular Q-learning approach was used but as the resolution is increased the curse of dimensionality rapidly becomes a problem. In order to circumvent this the RL agent should use a continuous state and action space, something that is achievable using function approximators such as a neural network or tile coding. Also the RL agent should be trained on several transport tasks since the final policy then becomes much more generic, instead of being tailored to one transport task as in this project. It is unclear how large of an effect the gear controller had on the results but since it is created in a rule based manner the human intuition sets a limit on how optimal this kind of gear controller can be. In order to really have an optimal gear controller the most straight forward way would be to let the RL agent also choose the gear of the truck.

Improvements could also be made with the powertrain model used. In this report the charge and discharge power of the battery was a function of the SoC but in a more realistic model it would also be a function of the battery temperature. Likewise ENG and EM temperatures would be relevant to model in the future as well.

Bibliography

- [1] Leo. Hellgren Jonas Ghandriz, Toheed. Laine and Bengt Jacobson. Sensitivity analysis of optimal energy management in plug-in hybrid heavy vehicles. 2017.
- [2] Simona Onori, Lorenzo Serrao, and Giorgio Rizzoni. Hybrid electric vehicles energy management strategies. *SpringerBriefs in electrical and computer engineering, Control, automation and robotics* (, 2016.
- [3] Antonio Sciarretta and Lino Guzzella. Control of hybrid electric vehicles. *IEEE Control systems*, 27(2):60–70, 2007.
- [4] F. Creutzig X. Cruz-Núñez M. D’Agosto D. Dimitriu M. J. Figueroa Meza L. Fulton S. Kobayashi O. Lah A. McKinnon P. Newman M. Ouyang J. J. Schauer D. Sperling Sims R., R. Schaeffer and G. Tiwari. Climate change 2014: Mitigation of climate change. contribution of working group iii to the fifth assessment report of the intergovern- mental panel on climate change. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, 2014.
- [5] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

A

Appendix

This appendix describes details in the powertrain and battery model, which was omitted in Chapter 3 and 5.

A.1 Hybrid powertrain model

Here all equations used in the hybrid powertrain is shown. Note that P_{eng} , P_{em} and ΔSoH_{batt} cannot be fully described due to confidentiality. $P_{batt,losses}$ is derived in more detail in Appendix A.3.

$$T_{prop} = (m \cdot a + F_g + F_{roll} + F_{air}) \cdot r_w \quad (\text{A.1})$$

$$T_{prop} = T_{gear,in} \cdot r_{gearbox} \cdot r_{final} \cdot \varphi(\text{sign}(T_{prop}), \eta_{gearbox}) + T_{disc} \quad (\text{A.2})$$

$$T_{gear,in} = T_{eng} + T_{em} \cdot r_{em} \quad (\text{A.3})$$

$$\omega_{prop} = \frac{\omega_{gear,in}}{r_{gearbox} \cdot r_{final} \cdot \varphi(\text{sign}(T_{prop}), \eta_{gearbox})} \quad (\text{A.4})$$

$$\omega_{gear,in} = \omega_{eng} = \frac{\omega_{em}}{r_{em}} \quad (\text{A.5})$$

$$P_{em}(T_{em}, \omega_{em}) = P_{em,mech} + P_{em,loss} = f(T_{em}, \omega_{em}) \quad (\text{A.6})$$

$$P_{eng}(T_{eng}, \omega_{eng}) = P_{eng} + P_{eng,loss} = f(T_{eng}, \omega_{eng}) \quad (\text{A.7})$$

$$P_{fuel}(T_{eng}, \omega_{eng}) = \frac{P_{eng}(T_{eng}, \omega_{eng})}{\eta_{eng}(T_{eng}, \omega_{eng})} \quad (\text{A.8})$$

$$P_{batt,inner}(T_{em}, \omega_{em}) = P_{batt,losses} + P_{batt,output} = P_{batt,losses} + P_{em} + P_{aux} \quad (\text{A.9})$$

$$\Delta SoH_{batt} = \frac{|P_{cell}|}{f(P_{cell}, \dots)} \quad (\text{A.10})$$

$$\Delta SoH_{disc} = -k \cdot P_{disc} \cdot t = -k \cdot T_{disc} \cdot \omega_{wheel} \cdot t \quad (\text{A.11})$$

$$\dot{m}_{fuel} = \frac{P_{fuel}}{Q_{fuel}} \quad (\text{A.12})$$

$$(\text{A.13})$$

$$\varphi(\text{sgn}, \eta) = \begin{cases} \eta & \text{sgn} \geq 0 \\ \frac{1}{\eta} & \text{sgn} < 0 \end{cases} \quad (\text{A.14})$$

A.2 Conventional powertrain model

The powertrain for the conventional truck re-uses most of the hybrid powertrain model, with a few exceptions. The major difference is that there is no ESS or EM, which yields:

$$P_{batt,inner} = P_{em} = T_{em} = 0 \quad (\text{A.15})$$

Further the power needed for the auxillary systems P_{aux} is modeled as an extra torque that is put on the ICE, so the conventional truck won't get an unfair advantage over the hybrid truck. This means Equation (A.3) needs to be rewritten as:

$$T_{gear,in} = T_{mech} + T_{aux} = T_{eng} \quad (\text{A.16})$$

$$T_{aux} = \frac{P_{aux}}{\omega_{eng}} \quad (\text{A.17})$$

A.3 Battery model

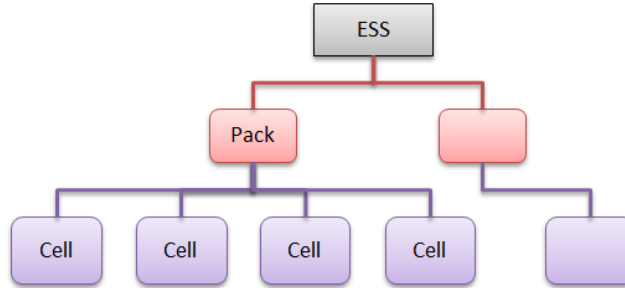


Figure A.1: The structure of the ESS. One ESS consists of one or several pack that are connected in parallel and each pack consists of many cells that are serially connected. For the details on how many packs and cells that were used please see Appendix C.

The ESS is built out of many smaller components, which can be seen as the heuristic layers in Figure A.1. The first layer is constructed with parallelly coupled packs, which all include serially coupled cells. A description of how the voltage, current and resistance of the cells influence the battery as a whole can be seen in Equation (A.18) to (A.23), together with details about how $P_{batt,losses}$ is calculated from Equation (A.9). Figure A.2 illustrates the upper and lower limit of the battery power output

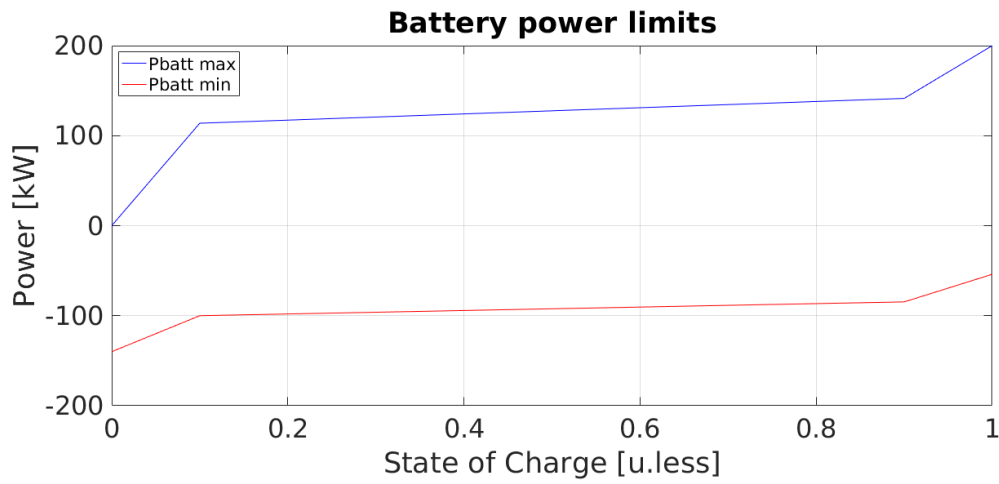


Figure A.2: Plot of how the battery power output varies as a function of the SoC. Negative power represents charging the battery and positive represents a discharge.

as a function of the SoC.

$$P = U \cdot I = I^2 \cdot R \quad (\text{A.18})$$

$$U_{batt} = U_{pack} = U_{cell} \cdot n_{cells} \quad (\text{A.19})$$

$$R_{batt} = \frac{R_{pack}}{n_{packs}} = \frac{R_{cell} \cdot n_{cells}}{n_{packs}} \quad (\text{A.20})$$

$$i_{batt} = i_{pack} \cdot n_{packs} = i_{cells} \cdot n_{packs} \quad (\text{A.21})$$

$$P_{batt,losses} = i_{batt}^2 \cdot R_{batt} = i_{cell}^2 \cdot R_{cell} \cdot n_{cells} \cdot n_{packs} \quad (\text{A.22})$$

$$(\text{A.23})$$

B

Appendix

This appendix considers details about how the ECMS cost function is defined, as well as the constraints used. ECMS is a quite well studied torque splitter where the book *Hybrid Electric Vehicles: Energy Management Systems* has been used as the major source of information regarding this subject. [2]

B.1 ECMS

The original cost function of ECMS, as described in Hybrid Electric Vehicles Energy Management Strategies, is defined as below in Equation (B.1):

$$\dot{m}_{f,eqv} = \dot{m}_{fuel} + \lambda \cdot \dot{m}_{vir} \quad (\text{B.1})$$

where \dot{m}_{fuel} is the instantaneous fuel consumption and \dot{m}_{vir} is called the *virtual fuel* which the electric machine consumes. λ is the *equivalence factor* as described in Section 5.2. From Equation (B.1), each term is described in detail below:

$$\dot{m}_{fuel} = \frac{P_{eng}(T_{eng}, \omega_{eng})}{\eta_{eng}(T_{eng}, \omega_{eng}) \cdot Q_{fuel}} \quad (\text{B.2})$$

Here $P_{eng}(T_{eng}, \omega_{eng})$ is the mechanical power produced by the ICE, $\eta_{eng}(T_{eng}, \omega_{eng})$ is the efficiency of the ICE and Q_{fuel} is the lower heating value of the fuel. T_{eng} is the torque produced by the ENG and ω_{eng} is the rotation speed of the ENG. Further the second term can be written as:

$$\dot{m}_{vir} = \frac{P_{batt,inner}}{Q_{fuel}} \quad (\text{B.3})$$

where $P_{batt,inner}$ is the inner power output of the battery, as described in Equation (A.9).

Now by using Equation (A.8), and multiplying with Q_{fuel} on both sides, Equation (B.1) can now be rewritten as:

$$cost = P_{fuel} + \lambda \cdot P_{batt,inner} \quad (\text{B.4})$$

and hence the problem of minimizing the equivalent fuel consumed (Equation (B.1)) is identical to minimizing the immediate power used, as shown in Equation (B.4).

B.2 Constraints

Several constraints were put on the possible minimas in Equation (B.4) due to limitations to the ICE, EM, battery pack and disc breaks. The constraints are listed in Equation (B.5) to B.8.

$$T_{em,min} \leq T_{em} \leq T_{em,max} \quad (\text{B.5})$$

$$T_{eng,min} \leq T_{eng} \leq T_{eng,max} \quad (\text{B.6})$$

$$P_{batt,min} \leq P_{batt} \leq P_{batt,max} \quad (\text{B.7})$$

$$T_{disc} \leq 0 \quad (\text{B.8})$$

C

Appendix

In this appendix many relevant simulation parameters are included were many of the values are supplied by Volvo Trucks. The different simulation parameters can be seen in Table C.1.

The states and actions in the RL algorithm was discretized as mentioned in Chapter 6, in state 1 there was:

- 11 SoC values, linearly spaced between 0 and 1
- 76 position values, linearly spaced between 0 and 1
- 11 actions, linearly spaced between 1.5 and 2.5

while in state 2 there was:

- 11 SoC values, linearly spaced between 0 and 1
- 7 dh_{250m} values, linearly spaced between -6 and 6
- 12 dh_{3km} values, linearly spaced between -26 and 46
- 11 actions, linearly spaced between 1.5 and 2.5

Parameter	Description	Value
ESS size	Size of battery	9.24 kWh
m_{truck}	Mass of truck	$35 \cdot 10^3$ kg
dt	Timestep of the simulation	1 s
dt_{train}	Timestep between actions of RL agent	50 s
α	Learning rate of RL algorithm	0.5
γ	Discount factor of RL algorithm	1
SoC_{init}	Initial SoC when evaluating	0.5
$SoC_{end,ref}$	Referens for final SoC	0.5
C_{fuel}	Cost of the diesel fuel	1.4 €/liter
C_{batt}	Cost of the battery	$1.5 \cdot 10^4$ €/pack
C_{disc}	Cost of the disc break system	$5 \cdot 10^2$ €/set
r_{em}	Ratio between electric machine and gear box	3
r_w	Radius of truck wheels	0.5 m
r_{final}	Ratio between gear box and wheels	2.6429
$\eta_{gearbox}$	efficiency of the gear box	1
n_{packs}	Number of packs in the ESS	1
n_{cells}	Number of cells in each pack, in the ESS	200

Table C.1: Specifications and initial setting for the truck used in the simulations.

Figure C.1 shows the altitude and gear profile during the transport task, which is the same for all truck configurations.

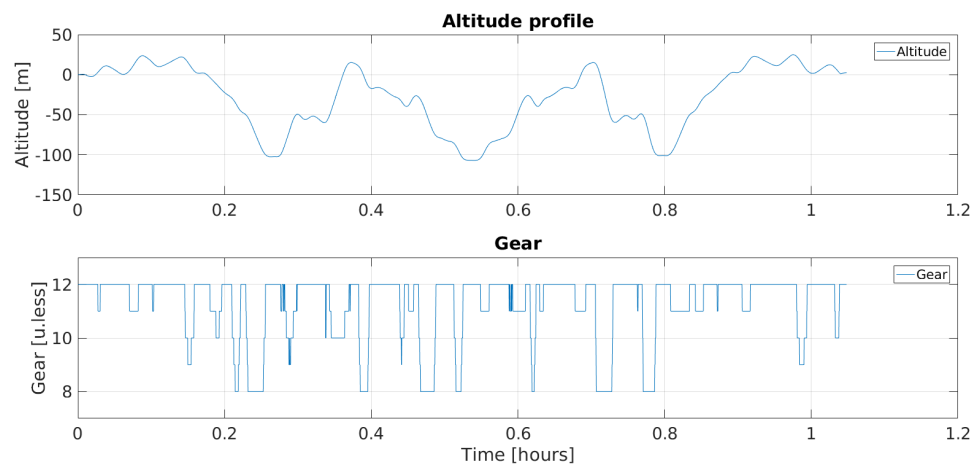


Figure C.1: Figure illustrating the altitude and gear profile during the transport task. Note that both of these profiles are the same for the RL agent, constant equivalence factor and the conventional truck.