Master of Science Thesis in Electrical Engineering
Department of Electrical Engineering, Linköping University, 2017

Route Based Optimal Control Strategy for Plug-in Hybrid Electric Vehicles

Johan Almgren and Gustav Elingsbo

LINKÖPING
UNIVERSITY

Master of Science Thesis in Electrical Engineering

Route Based Optimal Control Strategy for Plug-in Hybrid Electric Vehicles

Johan Almgren and Gustav Elingsbo

LiTH-ISY-EX--17/5081--SE

Supervisor: **Kristoffer Ekberg**
ISY, Linköpings University
Martin Sivertsson
Volvo Cars Corporation

Examiner: **Lars Eriksson**
ISY, Linköpings University

Division of Vehicular Systems
Department of Electrical Engineering
Linköping University
SE-581 83 Linköping, Sweden

Abstract

More restrictive emission legislations, rising fuel prices and the realisation that oil is a limited resource have lead to the emergence of the hybrid electric vehicles. To fully utilise the potential of the hybrid electric vehicles, energy management strategies are needed. The main objective of the strategy is to ensure that the limited electric energy is utilised in an efficient manner. This thesis develops and evaluates an optimisation based energy management strategy for plug-in hybrid electric vehicles. The optimisation methods used are based on a dynamic programming and ECMS approach. The strategy is validated against Vsim, Volvo Cars' performance and fuel consumption analysis tool as well as against strategies where parts of the optimisation is replaced by logic. The results show that the developed strategy consumes less fuel both compared to the corresponding Vsim strategy and the logic strategies.

Acknowledgments

We would like to express our warmest gratitude to Volvo Cars for this thesis opportunity and for the warm reception. People especially worth mentioning are Christoffer Strömberg, Håkan Larsson and Viktor Larsson. A special thanks to Martin Sivertsson who has been an invaluable support throughout the thesis. All your contributions, tutoring and guidance has been greatly appreciated.

We would also like to thank our pool-table colleagues Carl Bredberg and John Stjernrup for the company and all the laughs.

From Linköping University we would like to thank our supervisor Kristoffer Ekberg for all the valuable input. Also Lars Eriksson for giving us a solid foundation in the studied area.

Finally, we would like to to thank our families and friends for all the love and support.

Göteborg, June 2017
Johan Almgren and Gustav Elingsbo

Contents

Notation

General notations

Variable	Representing
F	Force
T	Torque
P	Power
v	Velocity
a	Acceleration
θ	Angle
ω	Angular velocity
J	Inertia
η	Efficiency

Vehicle model notations

Variable	Representing
m_v	Vehicle mass
r_{wh}	Wheel radius
g	Gravitational acceleration
c_d	Drag coefficient
A_f	Vehicle frontal area
α	Road inclination
ρ_{air}	Air density
i	Transmission ratio
c_{r0}	Rolling resistance coefficient
c_{r1}	Rolling resistance coefficient

Battery notations

Variable	Representing
I	Current
U	Voltage
ξ	State of charge
R_i	Battery internal resistance
Q	Battery capacity
Q_0	Battery nominal capacity

Optimal control notations

Variable	Representing
N_t	Number of discretised timesteps
N_{om}	Number of operating modes
λ	Costate (equivalence factor)
π^0	Optimal control sequence
μ	Adjoint state
u	Control signal
x	State variable

Other notations

Variable	Representing
Q_{LHV}	Fuel lower heating value
Δt_{sync}	ERAD syncronisation time

Abbreviations

Abbreviation	Full form
AER	All electric range
BATT	Battery
CD	Charge depleting
CS	Charge sustaining
DP	Dynamic programming
DDP	Deterministic dynamic programming
ECMS	Equivalent consumption minimisation strategy
EGB	Electric gearbox
EM	Electric motor
EMS	Energy management strategy
ERAD	Electric rear axle drive
FD	Final drive (transmission)
GB	Gearbox
HEV	Hybrid electric vehicle
ICE	Internal combustion engine
ISG	Integrated starter generator
PHEV	Plug-in hybrid electric vehicle
PMP	Pontryagin's minimum principle
SPA	Scalable product architecture

1 Introduction

1.1 Background

More restrictive emission legislations, rising fuel prices and the realisation that oil is a limited resource have lead to a radical change in the automotive industry. At the same time the consumer expectations of vehicle performance are growing. The hybridisation of drive lines is the current trend leader in the transition away from non renewable resources for passenger cars.

In hybrid electric vehicles (HEVs) the energy used in the propulsion system is combined from two energy sources, fossil fuel and electric energy. The electric energy is cheaper and has the potential to reduce emissions, which could make HEVs cleaner than conventional vehicles.

One type of HEV is the plug-in hybrid electric vehicle (PHEV) which contains a battery that can be recharged from an external power source. The opportunity of external charging enables a net discharge over the itinerary and thus a greater possibility to save fuel by using more electric power. PHEVs offers a good compromise between the long range of a conventional vehicle and high efficiency on commute routes.

1.2 Problem description

To be able to utilise the hybridisation in an efficient way an energy management strategy (EMS) is needed. The EMS ensures that the limited electric energy is used in an efficient way, allowing a good fuel economy.

An EMS can be designed in numerous ways depending on, for example, complexity, optimisation type or prior knowledge of driving mission. What they all have in common is that the objective is to determine the torque split between the different propulsion actuators. This in order to improve vehicle efficiency by optimising power utilisation and thus improving fuel economy and/or reducing emissions [12].

This thesis aims to develop, implement and evaluate a optimisation-based EMS for PHEVs to use as a benchmark for other control strategies.

1.3 EMS and related research

An extensive research has previously been made by several contributors in the area of EMS for HEVs. In general, the HEV control strategies can be divided in two different categories, the rule based- and the optimisation based control strategies. The optimisation based strategies contain two subcategories, the global- and real time optimisation strategies [12, 18, 19].

A classification overview of the control strategies is presented in Figure 1.1.

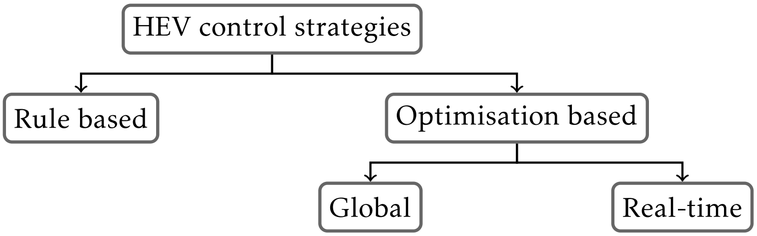


Figure 1.1: Categorisation of HEV control strategies. The control strategies contains two main categories, the rule based and optimisation based. The optimisation based strategies contains two subcategories, the global- and real-time optimisation strategies.

The earliest EMS were often of a rule based control type in which the controller follows a predefined set of rules. The rules are often based on heuristics or mathematical models where the design and parameterisation has to be tuned with specific drive missions and vehicle models. These controllers have the main objective of maximising the efficiency of the individual components by operating them in efficient working points. Their advantage is the fact that they are easily implemented, understood and that they can be trimmed to a desired behaviour. As the components are optimised individually, the result is a local optimal solution. In order to obtain the global optimal solution, the system should be optimised as a whole. This, together with the fact that the controllers can have poor performance for drive missions which have not undergone calibration started the development of the mathematical, optimisation based control strategies [12, 15, 18].

The optimisation based control strategies focuses on minimising a objective function while satisfying physical constraints on the system components. The objective function vary depending on the application, for HEVs they often include fuel consumption, emissions or torques. In contrast to the rule based control strategies, the optimisation based control strategies focus on optimising the system as a whole [18]. Optimisation based control strategies contain two subcategories, global and real-time optimisation stategies.

The global optimisation strategies focus on optimising the problem as a whole, thus yielding the optimal EMS. However, these methods require exact knowledge of the driving mission in advance. In reality, this is seldom the case as disturbances such as the driver, intersections and traffic gives an uncertainty in the data. Therefore, the global optimisation strategies are usually used offline to benchmark against vehicle-implementable strategies [15]. One of the more commonly used global optimisation strategy is the dynamic programming (DP).

In real-time optimisation strategies the problem is divided into a series of subproblems where each subproblem is optimised locally. These strategies do not necessarily require knowledge of the driving mission in advance why they are suitable for online implementation in vehicles. Some of the more common real time strategies for EMS are the Equivalent consumption minimisation strategy (ECMS), model predictive control, Pontryagin's minimum principle (PMP) and neural networks [12, 15]. Among these, the ECMS and PMP have gained the most attention [2].

Several researchers have been investigating a DP-based EMS, however the strategy is computationally demanding in the presence of several state variables why many researches utilises it in collaboration with a real-time optimisation strategy. For example, in [9] and [15] an optimal control strategy using DP and PMP was investigated.

Due to their potential of online implementation and the computational reduction, several pure, real-time optimisation strategies has been proposed. The authors of

[2] investigate a proportional ECMS control strategy. In [4] an approximate PMP strategy is proposed which resulted in a real-time, close to optimal controller.

The EMS presented in this thesis is a DP/ECMS combination designed to work as an offline benchmark tool. As the aim is to yield close to global optima there are more characteristics similar to the ones of a global optimisation strategy than a real-time. The EMS is not developed to be used for real-time applications.

1.4 Approach

The approach proposed in this thesis requires that the driving mission is known in advance. This means that parameters such as velocities, accelerations and road inclination are treated as inputs to the EMS.

The problem can be seen as an electric range exceeding itinerary, meaning that the destination can not be reached solely with the available electric energy. The available energy sources for propulsion are fuel and electric energy. The question raised is where along the trip the electric energy should be used in order to benefit the most from it. Operating the electric motor in effective operating regions allows effective battery usage which reduces the amount of fuel needed for propulsion. To ensure that as much electric energy as possible is used, the battery should be discharged to a minimum level upon arrival to the destination.

To summarise, the objective of the developed EMS is to optimise the power distribution among the available actuators with the purpose of minimising fuel consumption. Drive mission data is assumed to be known a priori and the goal is to investigate electric range exceeding drive missions. Aspects such as driveability and emissions will not be studied in detail. A flat road is assumed and the effects of road slope is not further investigated.

The proposed EMS consist of a mixture between the optimisation strategies Dynamic programming and Equivalent consumption minimisation strategy. The motivation for this selection is discussed in Section 5.1. A brief introduction to the two strategies is presented in Section 3.1 and Section 3.2.

The studied vehicle system is the scalable product architecture (SPA), Volvo Cars' hybrid platform with a parallel configuration. A schematic overview of the configuration can be seen in Figure 1.2. The system consists of an automatic, eight speed gearbox (GB), a final drive transmission (FD), an integrated starter generator (ISG), an internal combustion engine (ICE), a fuel tank (FT), a battery (BATT), an electric gearbox (EGB) and an electric rear axle drive (ERAD). The configuration is further explained in Section 2.4.

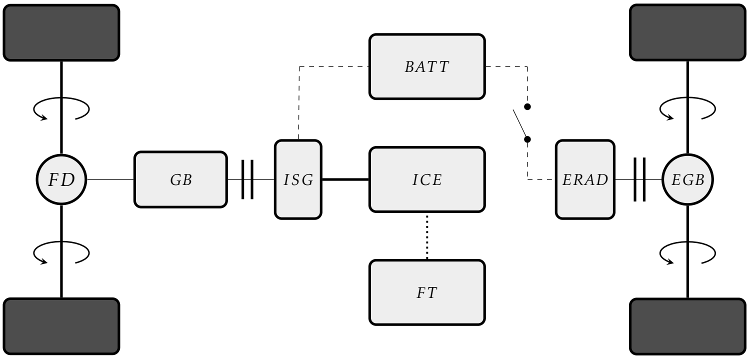


Figure 1.2: *An illustration of the studied SPA configuration. It contains an automatic, eight speed gearbox (GB), a final drive transmission (FD), an integrated starter generator (ISG), an internal combustion engine (ICE), a fuel tank (FT), a battery (BATT), an electric gearbox (EGB) and an electric rear axle drive (ERAD). Clutch and switches are engaged and disengaged to set different power transfer paths. The configuration is further treated in Section 2.4.*

1.5 Thesis goal

This thesis aims to develop a EMS for PHEVs that given driving data (velocities and acceleration) finds a close to optimal control sequence in order to minimise fuel consumption.

One of the investigated aspects is how much impact the optimisation of the different control signals has. Particularly, how much fuel is saved by replacing the different parts of Volvo's current control strategy with the developed strategy. To investigate the individual effects, the different parts are replaced one at a time. The parts investigated are:

- How much fuel can be saved by using an optimisation based control strategy for the ICE status (ON/OFF).
- How much fuel can be saved by using an optimisation based control strategy for the ERAD status (ON/OFF).
- How much fuel can be saved by using an optimisation based control strategy for the ERAD status (ON/OFF) and ERAD torque.
- How much fuel can be saved by using an optimisation based control strategy for the gear selection.

Furthermore, the EMS will be implemented in Volvo Cars' fuel economy and performance simulation environment. This to evaluate models and to see if the currently implemented control strategy can be improved with the developed EMS.

1.6 Outline

The thesis contains the chapters listed below,

- **The hybrid electric vehicle** - Contains an explanation of the most essential concepts of the existing HEV configurations. It also contains a thorough description of the the studied configuration.
- **Optimisation introduction** - A brief introduction to the optimisation methods dynamic programming and ECMS.
- **Model** - A presentation of the vehicle- and battery model used. Also explains the quasistatic approach.
- **Method** - Gives a full explanation and motivation to the developed EMS.
- **Validation** - Explains the validation methods to ensure optimality and a correct implementation of the EMS.
- **Results** - The chapter presents the obtained results.
- **Analysis of results** - Contains an analysis of the results.
- **Conclusions and future work** - Here, the main conclusions are presented together with a suggestion for future work.

2

The hybrid electric vehicle

The main difference between a HEV and a conventional vehicle or a battery electric vehicle is that the hybrid vehicles consists of at least two power sources and two propulsion actuators. For HEVs, a battery and an electric motor is added to the conventional ICE configuration. The added components allows regenerative braking, which is a concept of storing the braking energy as electric energy that otherwise would have been converted to unwanted heat losses in the brakes. The stored energy can later be used for propulsion.

In addition to regenerative braking, there are several fuel economy advantages of hybridisation. Some hybrid configurations allows the propulsion actuators to propel the vehicle in collaboration (hybrid driving). This allows a better dimensioning of the entire driveline since the power request peaks can be split between the actuators. The engine can as such be downsized while still maintaining the performance requirements of the vehicle. With the collaborative propulsion, the power distribution between the actuators can be optimised to ensure efficient operating regions. One major drawback of the hybridisation is the increased vehicle weight due to the added electrical components.

There are three main types of HEVs, which mainly differ in the combinational configuration of the actuators. The three types are the parallel hybrids, the series hybrids and the combined hybrids [3].

2.1 Parallel hybrid electric vehicle

An illustration of the parallel hybrid concept is shown in Figure 2.1.

The configuration contains an electric motor and an internal combustion engine as the propulsion actuators and a battery and fuel tank as energy sources. Both propulsion actuators are connected to the same drive shaft through a torque coupler and as such the actuators can propel the vehicle either individually or in collaboration. As earlier mentioned, this means that the power distribution between the actuators can be optimised to ensure efficient operating regions. It also allows downsizing of the ICE. As the actuators are coupled with the drive shaft, their rotational speed is set by the velocity of the wheels and the gear. The electric motor can be used as a generator, allowing the battery to be charged either through regenerative braking or from the ICE [3].

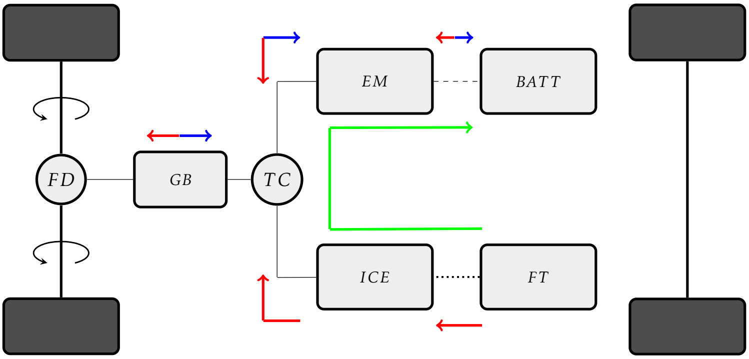


Figure 2.1: *A schematic illustration of a parallel HEV. The system consists of a final drive transmission (FD), a gearbox (GB), a torque coupler (TC), an internal combustion engine (ICE), a fuel tank (FT), an electric motor (EM) and a battery (BATT). The red arrows indicate the power transfer paths during hybrid driving. The blue arrows indicate the power path during regenerative braking. The green arrow indicate the power path when charging the battery with the ICE. Dashed lines represent electrical power flows and solid lines represent mechanical power flows.*

2.2 Series hybrid electric vehicle

A conceptual schematic of the series configuration is illustrated in Figure 2.2.

In contrast to the parallel hybrid, only the electric motor propels the vehicle in the series HEV. The electric motor receives power either from the battery or from a generator, which in turn is powered by the ICE. As the engine is decoupled from the drive shaft the engine can be chosen to work in efficient operating points. The series HEV require two electric machines which results in an increased vehicle weight. As in the case of the parallel concept, the battery can be charged either through regenerative braking or from the ICE [3]. The series configuration performs best in low speed applications.

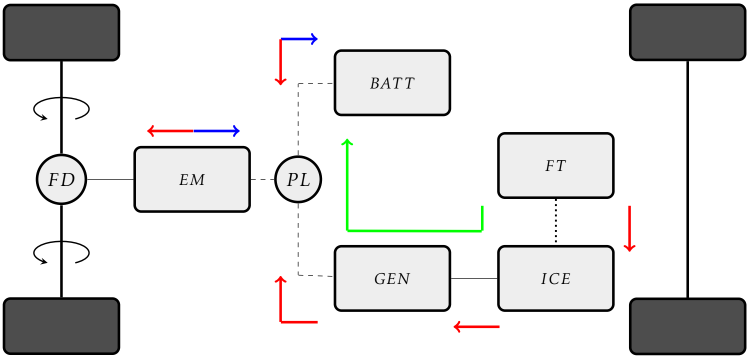


Figure 2.2: *A schematic illustration of the series HEV. The system contains a final drive transmission (FD), an electic motor (EM), a power link (PL), an internal combustion engine (ICE), a fuel tank (FT), a generator (GEN) and a battery (BATT). The red arrows indicate directions of possible power transfer during propulsion. The blue arrows indicate the power transfer during regenerative braking. The green arrow indicate the power path when charging the battery with the ICE. Dashed lines represent electrical power flows and solid lines represent mechanical power flows.*

2.3 Combined hybrid electric vehicle

The combined HEVs take advantage of both the series and the parallel hybrid concepts. In Figure 2.3, one of the combined HEV concepts is illustrated. The configuration contains two electric machines, namely an electric motor and a generator. The electric motor is used as in the parallel hybrid for propulsion and regenerative braking while the generator is mainly used for charging the battery from the ICE. As in the case of the parallel configuration, the EM and the ICE can propel the vehicle in collaboration. The configuration contains a power split device, which usually consists of a planetary gear set [3].

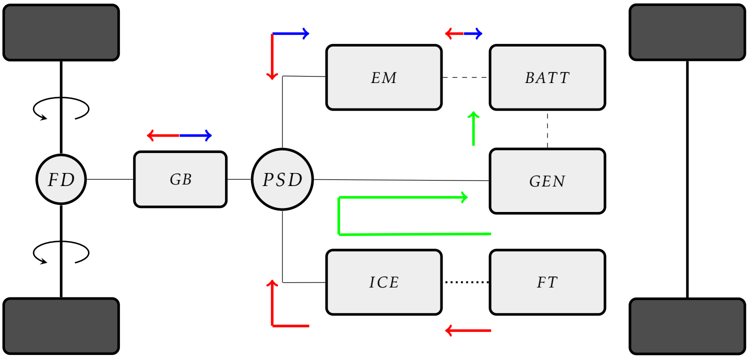


__Figure 2.3:__ A schematic illustration of a combined HEV. The system contains a final drive transmission (FD), a gearbox (GB), an electic motor (EM), an internal combustion engine (ICE), a fuel tank (FT), a generator (GEN), a battery (BATT) and a power split device (PSD). The red arrows indicate directions of possible power transfer during hybrid driving. The blue arrows indicate the power transfer during regenerative braking. The green arrow indicate the power path when charging the battery with the ICE. Dashed lines represent electrical power flows and solid lines represent mechanical power flows.

2.4 Studied configuration

The studied SPA platform is based on a parallel configuration with a added electric rear axle drive. The ERAD is an electric motor connected to the rear axle. The ERAD has a electric, one speed gearbox (EGB) that can be disengaged from the rear axle to reduce losses when the ERAD is inactive.

An overview of the studied platform is presented in Figure 2.4. In total, the configuration contains eight main components: an automatic, eight speed gearbox (EGB), a final drive transmission (FD), an integrated starter generator (ISG), an internal combustion engine (ICE), a fuel tank (FT), a battery (BATT), an electric gearbox (EGB) and an electric rear axle drive (ERAD).

The ISG replaces the conventional starting motor and is a fully capable electric motor with the possibility to serve as a generator. The ISG is directly coupled to the drive shaft and is thus mechanically coupled to the ICE. This enables charging the battery directly from the ICE and simultaneously running the ICE and ISG. However, since there is no opportunity to decouple the ICE and the ISG, it is not possible to run the ISG individually. For an all electric drive the ERAD can be used instead.

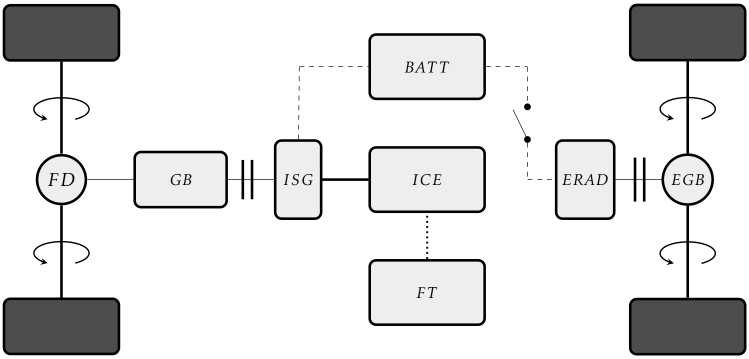


Figure 2.4: *A schematic illustration of the studied SPA model. Clutches and switches are engaged and disengaged to set a specified power transfer path. The configuration consists of a final drive transmission (FD), a gearbox (GB), an integrated starter generator (ISG), an internal combustion engine (ICE), a fuel tank (FT), a battery (BATT), an electric rear axle drive (ERAD) and an electric gearbox (EGB).*

Using the switches and clutches three combinations of power paths can be set. One where only the ICE and ISG are the active actuators, one where only the ERAD is active and one when all three actuators are active.

2.4.1 Active ICE and ISG

Figure 2.5 shows the available power transfer paths in the case of a turned off ERAD and a running ISG and ICE. The red arrows indicate the power flow in case of hybrid driving, the blue arrows indicate the power flow in regenerative braking. There is a possibility of charging the battery via the ICE, which is indicated by the green arrows. With this actuator combination the propelling or regenerative braking is done via the front wheels.

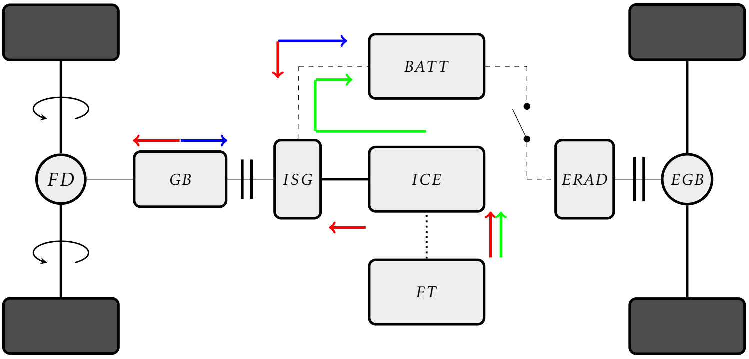


Figure 2.5: A schematic illustration of the available power flows in the case of a active ICE and ISG and turned off ERAD. Arrows indicate directions of possible power transfer. The red arrows indicate the power paths during hybrid drive, the blue arrows indicate the power paths during regenerative braking. The green arrows indicate the power flow when charging the battery via the ICE.

2.4.2 Active ERAD

When the ICE and ISG in inactive, the only actuator available for propulsion is the ERAD. The power flows in this scenario is illustrated in Figure 2.6. Battery energy is consumed or gained depending on if the vehicle is propelling (red arrows) or regenerative braking (blue arrows). In this case, the vehicle is propelled only using the rear wheels.

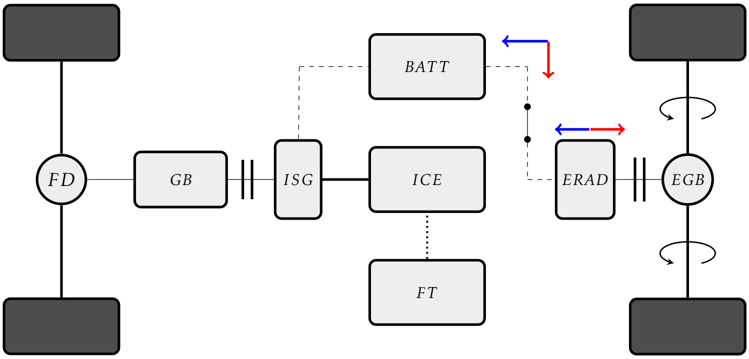


Figure 2.6: A schematic illustration of the available power flows in the case of a active ERAD and turned off ICE and ISG. Arrows indicate directions of possible power transfer. The red arrows indicate the power paths during electric propulsion, the blue arrows indicate the power paths during regenerative braking.

2.4.3 All propulsion actuators active

The scenario when all actuators are active can be seen in Figure 2.7. Propelling and regenerative braking can occur both at the front and rear wheels. From an optimisation point of view, this is the most complicated case as torques can be requested from all three actuators.

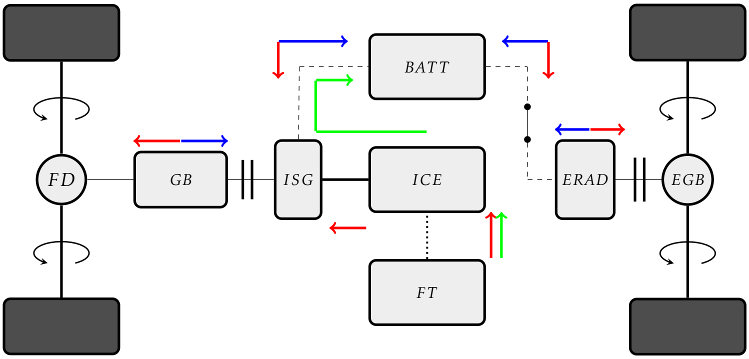


Figure 2.7: A schematic illustration of the available power flows in the scenario where all actuators are active. Arrows indicate directions of possible power transfer. The red arrows indicate the potential power paths during hybrid drive, the blue arrows indicate the potential power paths during regenerative braking. The green arrows indicate the power path when charging the battery with the fuel as energy source.

2.5 PHEV drive modes

PHEVs can be operated in three different drive modes: in charge depleting mode (CD), in charge sustaining mode (CS) or in blended mode which is a combination of the two. CD means that the vehicle runs purely on electrical energy from the battery. CS means that the vehicle is controlled similar to a HEV where the battery state of charge (SoC) is kept in a narrow band around a constant reference. The SoC is a measurement of the current battery capacity in relation to its nominal capacity.

A commonly used energy management strategy in today's PHEVs on electric range exceeding distances is to run the vehicle in CD mode and transfer to CS when the all electric range (AER) is reached and the battery is depleted. This CDCS strategy ensures that the battery is depleted upon arrival to the destination but has proven not to be the optimal strategy regarding fuel economy [10].

A more fuel efficient control strategy is to deplete the battery using a blended strategy. In blended mode the battery is discharged at a more even rate over the itinerary. In this way the electrical energy can be used at the most beneficial parts of the trip and the ICE can be used at efficient working points.

Figure 2.8 shows a graphical explanation of the how the battery is discharged with the CDCS and blended strategiy.

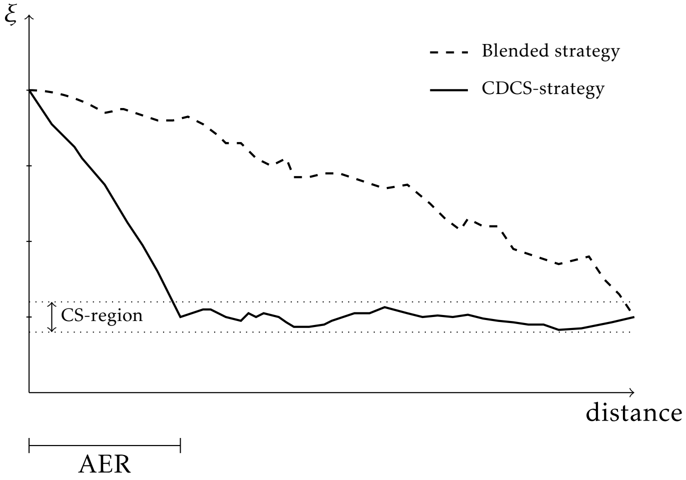


__Figure 2.8:__ Illustration of two discharge strategies. The solid line shows how the SoC (ξ) varies with the CDCS-strategy. CD is used until the AER is reached, where the strategy is changed to CS. The dashed line illustrates a blended strategy, in which the battery is discharged at a more even rate.

3

Optimisation introduction

3.1 Introduction to Discrete Dynamic Programming

Dynamic programming is a common technique for optimal control problems. Its main advantage is that it finds the optimal solution to the problem but has the drawback that the computational complexity increases exponentially with the number of states (n) and control inputs (m). The complexity increases linearly with time (N_t). The complexity can be expressed as,

$$\mathcal{O}(N_t \cdot p^n \cdot q^m) \tag{3.1}$$

This is why dynamic programming is not suitable for complex systems containing several states [3].

The main idea of dynamic programming is to divide the problem into several subproblems. These subproblems are solved independently and through their internal relations the solution to the original problem can be obtained. The strategy has a close relation to a shortest path problem and its solution process is based on Bellman's principle of optimality. The principle states that upcoming optimal decisions must not depend on previous decisions. An analogy would be that if the fastest way from point A to point C would pass through point B, then the partial route from B to C is the optimal for a journey starting in point B and ending in point C. [1, 3, 8]

The main objective is to minimise a certain cost function. How this cost function is defined depends on the application.

For more information on dynamic programming, see [1, 8, 17].

3.2 Introduction to ECMS

A common real-time optimisation method used in the area of EMS is the ECMS. The ECMS is derived from Pontryagin's minimum principle, which for dynamical systems provides the required conditions for optimal control [7].

The PMP defines a Hamiltonian function that is to be minimised,

$$H(x(t), u(t), \mu(t), t) = g(u(t), t) + \mu(t) \cdot f(x(t), u(t), t) \tag{3.2}$$

where $x(t)$ represents the state variables and $u(t)$ the control signals. The variable $\mu(t)$ represents the adjoint state in optimal control theory. Under the assumption that the battery internal resistance (R_i) and open circuit voltage (U_{oc}) is independent of the SoC (ξ), the adjoint state can be considered constant along the optimal trajectory [3, 7]. Under the same assumption the Hamiltonian function can be modified. By introducing a costate (λ),

$$\lambda = -\mu \cdot \frac{Q_{LHV}}{U_{oc}Q_0} \tag{3.3}$$

the function in equation (3.2) can be rewritten into terms of fuel power (P_f) and fuel equivalent electrical power (P_{ech}) [3].

$$H = P_f + \lambda P_{ech} \tag{3.4}$$

$$P_f = Q_{LHV} \cdot \dot{m}_f \tag{3.5}$$

$$P_{ech} = -\dot{\xi} U_{oc} Q_0 = I_{batt} U_{oc} \tag{3.6}$$

The variable Q_{LHV} represents the lower heating value of the fuel, Q_0 the battery's nominal capacity and $\dot{m}_f$ the fuel mass flow rate. As the fuel power and fuel equivalent electrical power are not directly comparable, the costate serves as a equivalence factor. The value of the costate will affect the level of battery depletion, a low value will result in a high depletion as electric energy is considered cheap in relation to fuel energy. On the opposite, a high value results in a low level of depletion.

The optimal costate value is heavily dependent on the driving mission and if it is known, the optimal controls can be determined. Therefore, one key aspect of the controller is to approximate the optimal costate for a specific drive cycle [13]. One practical issue is the fact that drive cycles are never perfectly known in advance, which implicates difficulties in approximating the optimal costate.

More about the ECMS can be found in [3, 7, 11].

4 Model

4.1 Drive cycle discretisation

The presented approach utilises a quasistatic approach, meaning that in small time intervals parameters such as velocity, acceleration and required wheel torque are considered constant. In other words, the problem is assumed to be static in each time interval. Hence, the drive cycle is discretised into N_t time steps of constant length Δt.

Drive cycle data contains information of the vehicle velocities with corresponding time instances. From drive cycle data the velocity and acceleration in each discretised time interval are calculated as,

$$v_i = v(t_i) \tag{4.1}$$

$$v(t) = \bar{v}_i = \frac{v_i + v_{i-1}}{2} \qquad \forall t \in [t_{i-1}, t_i) \tag{4.2}$$

$$a(t) = \bar{a}_i = \frac{v_i - v_{i-1}}{\Delta t} \qquad \forall t \in [t_{i-1}, t_i) \tag{4.3}$$

An illustration of the discretisation of a drive cycle is shown in Figure 4.1.

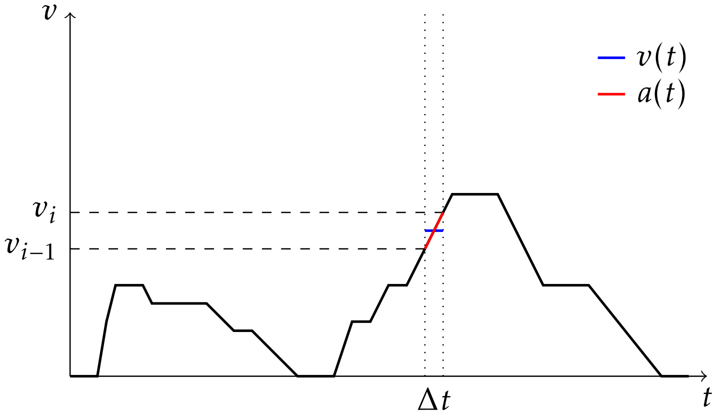


Figure 4.1: *Example of a drive cycle, the cycle is discretised into N_t time steps of length Δt. The problem is assumed to be static between two consecutive time instances t_{i-1} and t_i.*

4.2 Vehicle model

With the known velocity and acceleration in each time interval, the corresponding required wheel torque ($T_{req,wh}$) and power ($P_{req,wh}$) can be calculated using a longitudinal vehicle model [13],

$$T_{req,wh} = a\left(m_v r_{wh} + \frac{J_{wh}}{r_{wh}}\right) + T_{rld} \tag{4.4}$$

$$T_{rld} = r_{wh}(F_{air} + F_{roll} + F_{grav}) \tag{4.5}$$

$$F_{air} = \frac{1}{2}\rho_{air} c_d A_f v^2 \tag{4.6}$$

$$F_{roll} = m_v g(c_{r0} + c_{r1} v)cos(\alpha) \tag{4.7}$$

$$F_{grav} = m_v g sin(\alpha) \tag{4.8}$$

$$P_{req,wh} = T_{req,wh}\omega_{wh} \tag{4.9}$$

$$\omega_{wh} = \frac{v}{r_{wh}} \tag{4.10}$$

where m_v is the vehicle mass, r_{wh} the wheel radius, ρ_{air} the air density, c_d the air drag coefficient, A_f the frontal area, g the gravitational acceleration, c_{r0} and c_{r1} are rolling resistance coefficients and α represents the road inclination. J_{wh} is the resulting inertia from the entire vehicle model and depends on the active paths of power transfer.

T_{rld} is the torque required to cover the air-, rolling- and gravitational losses. These losses are modelled according to Equations (4.6)-(4.8).

The drive line is considered stiff, meaning no wind-up is considered in the drive shafts. It is modelled according to Equation (4.11-4.23). Transmission ratios are denoted by i.

$$J_{ICE}\ddot{\theta}_{ICE} = T_{ICE} - T_{ICE,loss} - T_{GB,ICE} \tag{4.11}$$

$$\dot{\theta}_{ICE} = \dot{\theta}_{GB} i_{GB} \tag{4.12}$$

$$J_{ISG}\ddot{\theta}_{ISG} = T_{ISG} - T_{ISG,loss} - T_{GB,ISG} \tag{4.13}$$

$$\dot{\theta}_{ISG} = \dot{\theta}_{GB} i_{GB} \tag{4.14}$$

$$\big(J_{GB} + \frac{J_{FD}}{i_{FD}}\big)\ddot{\theta}_{GB} = T_{GB} - T_{GB,loss} - T_{wh,front} \tag{4.15}$$

$$T_{GB} = T_{GB,ICE} + T_{GB,ISG} \tag{4.16}$$

$$\dot{\theta}_{GB} = \dot{\theta}_{wh} i_{FD} \tag{4.17}$$

$$J_{ERAD}\ddot{\theta}_{ERAD} = T_{ERAD} - T_{ERAD,loss} - T_{EGB} \tag{4.18}$$

$$\dot{\theta}_{ERAD} = \dot{\theta}_{EGB} i_{EGB} \tag{4.19}$$

$$J_{EGB}\ddot{\theta}_{EGB} = T_{EGB} - T_{EGB,loss} - T_{wh,rear} \tag{4.20}$$

$$\dot{\theta}_{EGB} = \dot{\theta}_{wh} \tag{4.21}$$

$$J_{wh}\ddot{\theta}_{wh} = T_{req,wh} - F_{wh} r_{wh} \tag{4.22}$$

$$T_{req,wh} = T_{wh,front} + T_{wh,rear} \tag{4.23}$$

The coupling between the ISG and the ICE can be seen in the torque Equation for the GB (4.16) as the torque consists of two contributing factors, $T_{GB,ICE}$ and $T_{GB,ISG}$. Since the front and rear axis have separate actuators the total wheel torque is the sum of the axle torques according to Equation (4.23).

4.3 Battery model

To accurately model a PHEV battery a complex partial differential equation model would be needed [7]. It is undesirable to introduce such computational complexity why a Thevenin equivalent circuit is used. An illustration of the battery model can be seen in Figure 4.2. The battery considered is of a Lithium-Ion type.

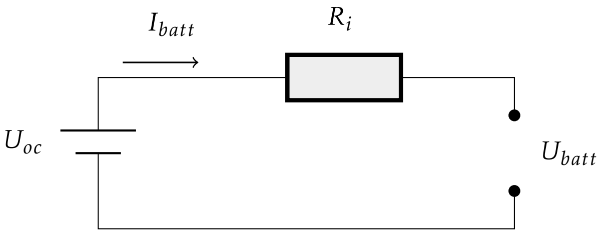


Figure 4.2: *Thevenin equivalent circuit model of a battery. U_{oc} represents the open-circuit voltage, R_i the internal resistance of the battery, I_{batt} the battery current and U_{batt} the battery voltage.*

This simplified model only has one dynamic state namely the battery SoC (ξ). It is defined as the ratio between the capacity of the battery (Q) and its nominal capacity (Q_0).

$$\begin{gathered}\xi(t) = \frac{Q(t)}{Q_0} \\ \xi \in [0,\ 1]\end{gathered} \tag{4.24}$$

The SoC is defined in the range $\xi \in [0, 1]$ but is seldomly charged or discharged to the limits as it may damage the battery.

In reality, the open circuit voltage and internal resistance is dependent of the SoC. In the normal region of SoC usage these vary only slightly [7]. Here, a small dependence is assumed and is modelled as an affine relationship.

Through charge balance, the variation in battery charge can be approximated as:

$$\dot{Q}(t) = -I_{batt}(t) \tag{4.25}$$

Which means that the change in state of charge can be expressed as:

$$\dot{\xi}(t) = -\frac{I_{batt}(t)}{Q_0} \tag{4.26}$$

With the quasistatic approach, the current is considered constant in each time interval why the change in SoC can be calculated according to Equation (4.27).

$$\Delta\xi(t) = -\frac{I_{batt}(t)}{Q_0}\Delta t \tag{4.27}$$

Using Kirschoff's law, the output voltage from the battery can be calculated as:

$$U_{batt}(t) = U_{oc}(t) - I_{batt}(t)R_i(t) \tag{4.28}$$

The battery power is obtained by multiplying the battery voltage with the battery current,

$$P_{batt}(t) = I_{batt}(t)(U_{oc}(t) - I_{batt}(t)R_i(t)) \tag{4.29}$$

Solving Equation (4.29) for the current yields:

$$I_{batt}(t) = \frac{U_{oc}(t) - \sqrt{U_{oc}^2(t) - 4R_i(t)P_{batt}(t)}}{2R_i(t)} \tag{4.30}$$

If the battery power is known the change in SoC can be calculated according to Equations (4.27) and (4.30).

5

Method

5.1 Motivation

Utilising a pure DP optimisation strategy would result in a global optimal EMS. However, both the fact that the PHEVs require optimisation on long routes and the fact that modern PHEVs contain large batteries makes the problem grow quickly in complexity. This as both time and state (SoC) has to be discretised. Having a fine discretised grid would yield an accurate result but imply heavy computational load. On the other hand, having a coarse grid would reduce the computational burden but increase the discretisation error.

The ECMS is a less computationally demanding method but would require separate logic to handle discrete control variables.

To obtain a good tradeoff between optimality and complexity, the presented strategy utilises both DP and ECMS. A similar approach was done in [9] and [15] where an optimal control strategy using dynamic programming and PMP was investigated. The latter concluded that using the two methods in collaboration resulted in a fast and close to optimal method.

The problem at hand is of mixed integer-type as it contains both continuous and discrete control variables. The ECMS requires separate logic for discrete control which is undesirable from an optimisation point of view. This is why the ECMS is allocated with the continuous control variables, in other words, solving the power split problem. The power split problem consists of determining the torques that each of the available actuators should output to meet the required wheel torque. The DP algorithm is allocated with the discrete control variables which are the selection of gear and whether the ICE and ERAD should be active or off.

5.2 Operating modes

To minimise the computational load of the DP algorithm a concept called *operating modes* is introduced. Instead of evaluating all the possible combinations of discrete control variables, the obvious non-optimal combinations are disregarded. For example, keeping a gear engaged while the ICE is off would generate extra losses without any gain. Therefore only the neutral is allowed while the ICE is off.

The remaining feasible combinations are referred to as operating modes. A specification of the operating modes are found in Table 5.1. By introducing the operating modes the combinations of discrete control variables are reduced to 20, which is almost half of the original candidates.

The ISG and ICE cannot be run separately and are thus combined into the state, $ICE_{on/off}$. The state $ERAD_{on/off}$ represents the ERAD state where a engaged EGB is represented by $ERAD_{on}$ and vice versa.

Depending on which operating mode is active, different paths of possible power transfer will be active or inactive, see Section 2.4. This raises a problem concerning how much power should be generated from each available actuator to meet the required traction power, i.e. the power split problem.

Table 5.1: *Specification of the 20 operating modes. Regarding the ICE and ERAD, the status 0 implicates a turned off actuator and 1 represents a turned on actuator. The list of gears represent the gear number associated with the corresponding operating mode. Gear 0 represents neutral.*

Operating mode	**ERAD**	**ICE**	**Gear**
1	0	1	0
2	0	1	1
3	0	1	2
4	0	1	3
5	0	1	4
6	0	1	5
7	0	1	6
8	0	1	7
9	0	1	8
10	1	1	0
11	1	1	1
12	1	1	2
13	1	1	3
14	1	1	4
15	1	1	5
16	1	1	6
17	1	1	7
18	1	1	8
19	1	0	0
20	0	0	0

5.3 Deterministic dynamic programming

The DP strategy presented is said to be of a deterministic dynamic programming (DDP) type. This as the cost function does not contain any random disturbances [3].

Due to the earlier mentioned computational complications associated with dynamic programming, the approach only has one discrete state, the operating modes. The DDP algorithm can here be interpreted as an method to solve a shortest path problem. The problem is set up of $N_t \times N_{om}$ nodes, where N_t represents the number of discretised timesteps and N_{om} the number of operating modes. This can be illustrated as a rectangular grid. Each node in a time instance $t = t_i$ is associated with the vehicle velocity and acceleration between time instances t_i and t_{i+1}. The main purpose of the dynamic programming algorithm is to determine which operating mode that should be active during each time interval. That is, finding the optimal control sequence:

$$\pi^0(x_0) = \{u_0, u_1, \ldots, u_{N_t-1}\} \tag{5.1}$$

where π^0 is the optimal control sequence given the initial state x_0. The control signals are denoted by u_n, $n = [0 \ldots N_t - 1]$. These corresponds to the active operating modes.

The problem can mathematically be stated as,

$$J_\pi(x_0) = \sum_{n=0}^{N_t-1} g_n(x_n, u_n) \tag{5.2}$$

$$x_{n+1} = f(x_n, u_n) \tag{5.3}$$

$$g_n = H(x_n) + c_t(u_n) \tag{5.4}$$

Where $J_\pi(x_0)$ represents the total cost of using control sequence π on the problem, given the initial state x_0.

The cost function g_n contains two terms, the *Hamiltonian* (H) and transition costs (c_t). In short, the Hamiltonian relates to the amount of fuel required to propel the vehicle, this is further explained in section 5.5. Transition costs represents the fuel cost for changing operating mode in two subsequent time instances. The transition costs are covered in section 5.4.

Often a terminal cost $g_N(x_N)$ is added to the total cost in Equation 5.2, this to fulfil state boundary conditions. In this application, it is undesirable to set the active operating mode at terminal time why the cost is set to zero.

The problem is solved in an recursive manner where the algorithm proceeds backwards in time, starting in time $N_t - 1$ proceeding towards time 0. Below, the DDP algorithm is explained.

DDP algorithm procedure:

Step1: Set $n = N_t - 1$.

Step2: For all the N_{om} nodes in the current time instance, find the minimum cost to go

$$J_n(x_n) = \min g_n(x_n, u_n) + J_{n+1}(f(x_n, u_n)) \tag{5.5}$$

where g_n represent the stage cost and J_{n+1} the cost to go to the previous stage.

Step3: If $n = 0$ then return the solution, otherwise set $n = n - 1$ and proceed to *Step 2*.

An step-by-step illustration of the full procedure can be seen in Figures 5.1-5.4. For simplification only six operating modes and five time instances are represented.

In this example, the procedure is initialised in operating mode 6 and time $t = t_{N-1}$ according to Figure 5.1. The costs for transitioning from node 6 to the nodes in the subsequent time instance are evaluated. The control which yields the lowest cost to go is stored and the cost is associated with node 6 and time t_{N-1}.

The same is done for the remaining nodes in the current time instance, see Figure 5.2. When all nodes have been evaluated in the current time instance, all nodes have been associated with a cost J_{N-1} and a control decision. This is illustrated in Figure 5.3.

Now, the same procedure is repeated for t_{N-2}, t_{N-3} and t_0. When t_0 has been evaluated there are six trajectories, the optimal trajectory chosen is the one with the lowest initial cost. In Figure 5.4, such a trajectory (marked red) can be seen. The result is interpreted as the green trajectory in Figure 5.4, that is u_i is the active mode between time instances t_i and t_{i+1}.

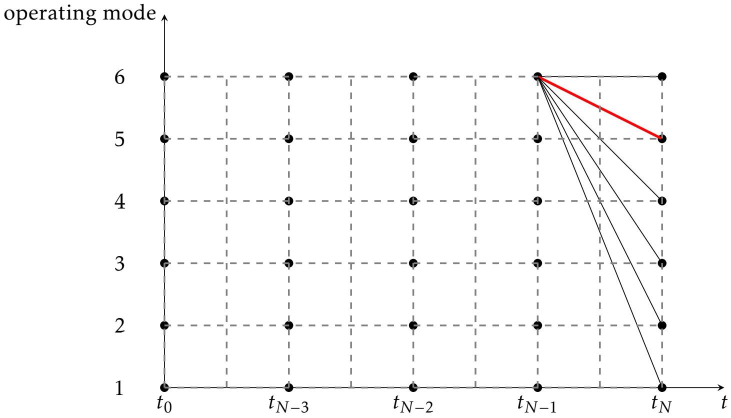


Figure 5.1: *Explanation of the DDP-algorithm. The cost of transitioning from node 6 to all the other nodes in the subsequent time instance are evaluated. The arc with the lowest cost to go (marked red) is chosen*

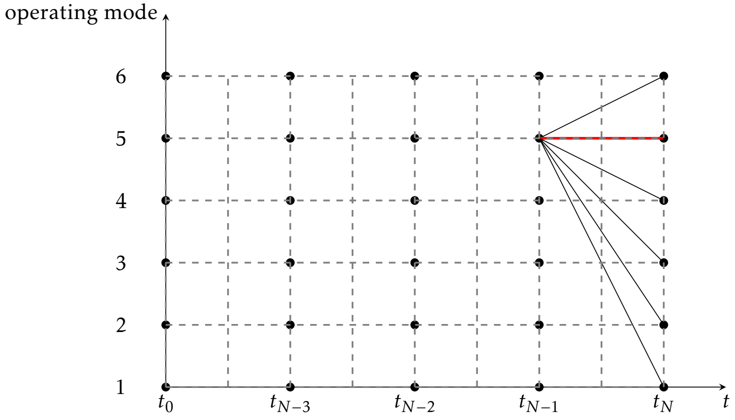


Figure 5.2: *Explanation of the DDP-algorithm. The same procedure is repeated for mode 5. Again, the lowest cost to go is stored and associated with the evaluated operating mode.*

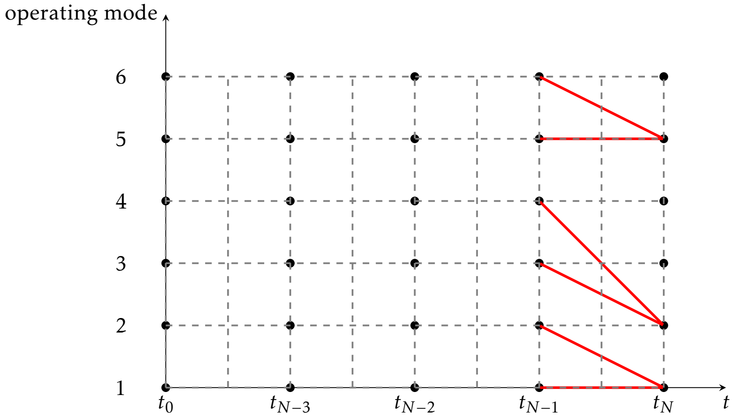


Figure 5.3: *Explanation of the DDP-algorithm. After all nodes have been evaluated in time t_{N-1} all the nodes have been updated with an associated cost to go. The procedure is then repeated for all preceding time instances.*

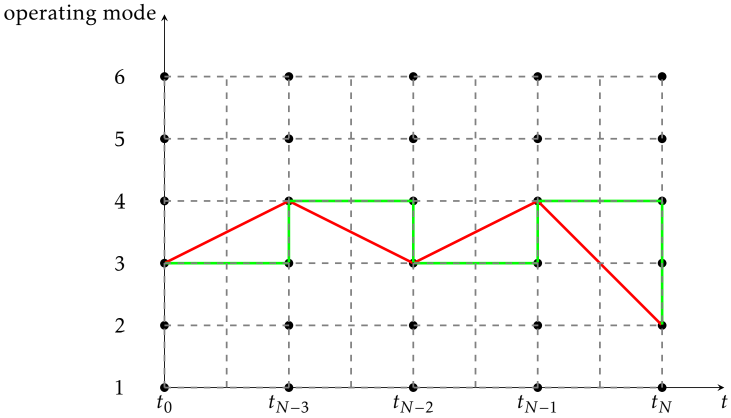


Figure 5.4: *Explanation of the DDP-algorithm. The optimal control trajectory can be seen in red. The green line corresponds to the interpretation of the solution, mode 3 is active in* $t \in [t_0, t_{N-3})$, *mode 4 in* $t \in [t_{N-3}, t_{N-2})$ *etc.*

5.4 Transition costs

Transition costs are included in the DDP cost function with the purpose of representing the fuel consumed when transitioning between operating modes. Three different transition costs are taken into account: *ICE starts/stops*, *gear shifts* and *ERAD starts/shutdowns*.

Engine starts/stops:

The engine start/stop costs are to take into consideration the energy required to start the ICE. During engine starts both fuel as well as battery energy is consumed. In reality, the power required to start the engine would depend on several factors, here an approximation of constant costs are considered. An engine start is approximated to consume a constant specified mass of fuel and battery energy. The amounts used are based on analyses made at Volvo.

Gear shifts:

Shifting gear entails a speed alteration of the engine. Upshifts results in a deceleration of the engine while downshifts results in an acceleration of the ICE. The kinetic energy required to accelerate the ICE is assumed to consume a constant mass of fuel per shift and is increased linearly with the number of gears shifted. No transition costs are added to upshifts as kinetic energy from the ICE is gained. As in the case with engine starts the amount of fuel consumed per gear shift are based on analyses at Volvo.

ERAD starts/shutdowns:

The effects of turning on or off the ERAD is based on rotational kinetic energy calculations. The power required to start the ERAD is supplied from the battery and in the case of ERAD shutdown battery energy is assumed to be gained. Calculations are made according to Equation 5.6.

$$P_{ERAD,syn} = -\sigma_{ERAD} \cdot \frac{J_{ERAD}}{2\Delta t_{sync}} (\omega_{wh} \cdot i_{egb})^2 \cdot \eta_{ERAD,syn}{}^{\sigma_{ERAD}} \tag{5.6}$$

$$\sigma_{ERAD} = \begin{cases} 1 & \text{if ERAD shutdown} \\ -1 & \text{if ERAD starting} \end{cases} \tag{5.7}$$

$P_{ERAD,syn}$ represents the required or gained battery power during ERAD start or shutdown.

The synchronisation time (Δt_{sync}) varies depending on the angular velocity of the wheels. To obtain the lowest possible time for synchronisation, the highest possible deliverable torque from the ERAD is requested ($T_{ERAD,max}$).

Equation (5.8) describes the maximum angular acceleration at a certain time step [3].

$$J_{ERAD}\dot{\omega}_{ERAD} = T_{ERAD,max}(\omega_{ERAD}) \tag{5.8}$$

From Equation (5.8) the ERAD angular velocity can be expressed using Euler forward,

$$\omega_{ERAD_n} = \omega_{ERAD_{n-1}} + h\frac{T_{ERAD,max}(\omega_{ERAD_{n-1}})}{J_{ERAD}} \tag{5.9}$$

where h is a small step size and n the current time step.

Equation (5.9) is computed in the interval $\omega_{ERAD} \in [0 \quad \omega_{ERAD,max}]$, with the results according to Figure 5.5. From this curve the ERAD synchronisation time can be interpolated by using the angular velocity of the ERAD.

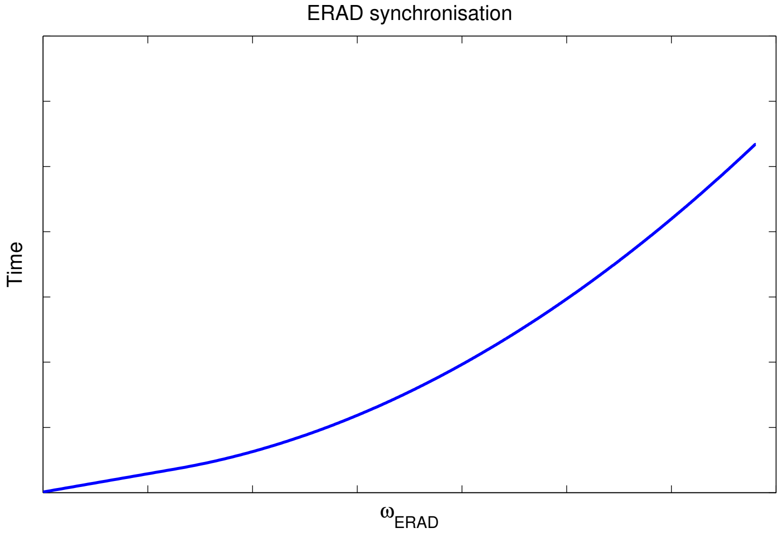


Figure 5.5: *ERAD synchronisation time as a function of ERAD angular velocity. The time for synchronisation is interpolated from the curve.*

5.5 Equivalent Consumption Minimisation Strategy

The implemented ECMS has the objective to determine the continuous control variables. The continuous control variables are the required torques from the three available propulsion actuators. Given that the status of the ICE (on/off), status of the ERAD (on/off) and the current gear is given by the DDP algorithm, the problem for the ECMS strategy is finding the actuator torques that minimises the Hamiltonian. That is, finding:

$$\left[T_{ICE,mech}, T_{ISG,mech}, T_{ERAD,mech}\right] = \text{argmin}\, H \tag{5.10}$$

where the subscript *mech* corresponds to the useful torques delivered by the actuators.

The required wheel power is calculated according to Equation (4.9). It can be split into a provided rear power (P_{rear}) and provided front power (P_{front}).

$$P_{req,wh} = P_{rear} + P_{front} \tag{5.11}$$

Where the front power (P_{front}) is provided by the ISG and ICE,

$$P_{front} = P_{ICE,mech} + P_{ISG,mech} - P_{GB,loss} \tag{5.12}$$

$$P_{GB,loss} = f(P_{ICE,mech} + P_{ISG,mech}) \tag{5.13}$$

The rear power (P_{rear}) is provided by the ERAD,

$$P_{rear} = P_{ERAD,mech} - P_{EGB,loss} \tag{5.14}$$

$$P_{EGB,loss} = f(P_{ERAD,mech}) \tag{5.15}$$

The Hamiltonian to be minimised is expressed according to Equation (5.16).

$$H = P_f(T_{ICE,mech}) + \lambda \cdot P_{ech}(T_{ISG,mech}, T_{ERAD,mech}) \tag{5.16}$$

where,

$$P_f = \max(0, P_{ICE,mech} + P_{ICE,loss}) \tag{5.17}$$

$$P_{ech} = U_{oc} I_{batt} = \frac{U_{oc}^2 - \sqrt{U_{oc}^4 - 4R_i P_{batt} U_{oc}^2}}{2R_i} \tag{5.18}$$

$$P_{batt} = P_{aux} + P_{ISG,tot} + P_{ERAD,tot} \tag{5.19}$$

$$P_{ISG,tot} = P_{ISG,mech} + P_{ISG,loss} \tag{5.20}$$

$$P_{ERAD,tot} = P_{ERAD,mech} + P_{ERAD,loss} \tag{5.21}$$

$$P_{ICE,loss} = f(P_{ICE,mech}, \omega_{ICE}) \tag{5.22}$$

$$P_{ISG,loss} = f(P_{ISG,mech}, \omega_{ISG}) \tag{5.23}$$

$$P_{ERAD,loss} = f(P_{ERAD,mech}, \omega_{ERAD}) \tag{5.24}$$

subject to,

$$P_{batt,min} \leq P_{batt} \leq P_{batt,max} \tag{5.25}$$

$$P_{ICE,min} \leq P_{ICE} \leq P_{ICE,max} \tag{5.26}$$

$$P_{ISG,min} \leq P_{ISG} \leq P_{ISG,max} \tag{5.27}$$

$$P_{ERAD,min} \leq P_{ERAD} \leq P_{ERAD,max} \tag{5.28}$$

P_{aux} in equation (5.19) represent auxiliary losses and is assumed to be constant.

The optimisation all comes down to dividing the required wheel propulsion power among the available actuators in a way that they operate in effective working points. In other words, maximising the efficiency of the system.

Depending on which operating mode is active, some actuators might not be available which limits the possible power transfer paths, see section 2.4. This affects the procedure of calculating the Hamiltonian. There are six possible calculation scenarios depending on what operating mode is evaluated:

Scenario 1: Operating mode 1 *(ERAD off, ICE on and gear in neutral)*

In this case of a neutral gear and a inactive ERAD, no propelling power is available. The option is to charge the battery via the ICE. This leaves $T_{ERAD,mech} = 0$, why only the optimal ISG and ICE torques needs to be found.

Scenario 2: Operating mode 2-9 *(ERAD off, ICE on and gear 1-8)*

Scenario 2 is similar to scenario 1 in the aspect that the optimal torques only needs to be determined for the ICE and ISG. The difference is that with an active gear propulsion is available via the ICE and ISG. This scenario corresponds to Figure 2.5.

Scenario 3: Operating mode 10 *(ERAD on, ICE on and gear in neutral)*

This scenario gives the option to propel the vehicle with the ERAD and to charge the battery with power from the ICE. As the gear is in neutral, no traction power can be delivered to the front wheels.

Scenario 4: Operating mode 11-18 *(ERAD on, ICE on and gear 1-8)*

From an optimisation point of view, this is the most complex case as all three actuators are available to generate tractive power. This scenario corresponds to Figure 2.7.

Scenario 5: Operating mode 19 *(ERAD on and ICE off)*

Here, the ERAD is the only actuator propelling the vehicle, thus $T_{ICE,mech} = T_{ISG,mech} = 0$. The required ERAD torque can be either positive or negative depending on if propelling or regeneratively braking. This scenario corresponds to Figure 2.6.

Scenario 6: Operating mode 20 *(ERAD off and ICE off)*

In this scenario none of the actuators are active, hence:

$$T_{ICE,mech} = T_{ISG,mech} = T_{ERAD,mech} = 0$$

Typically, this mode would be desirable at standstill.

5.6 Bisection method

As mentioned in the Section 3.2, the value of the costate (λ) will affect how much battery charge that will be consumed during an itinerary. As the DDP algorithm solves the problem backwards, the value of the costate will affect the starting SoC. A high costate value will yield a low initial SoC as battery depletion is considered expensive, on the opposite a low costate value will result in a high initial SoC.

The relation between the costate and the initial SoC is nonlinear and in order to obtain a desired battery depletion, the costate value has to be properly tuned. This is done using a root finding algorithm called the bisection algorithm. In [5], a comparison is made between different root finding algorithms treating this particular problem. It was found that the bisection method performed well in the aspects of accuracy, complexity and robustness.

Root finding algorithms are numerical methods intended to find a solution to the general equation $f(x) = 0$. Here, it is used to find the costate value that gives an initial SoC close to a predefined reference. Since the algorithm is numerical and iterative, the condition is relaxed according to:

$$\begin{aligned} \Delta\xi &= \xi_0(\lambda) - \xi_{0,ref} \leq \epsilon_{tol} \\ \xi_0 &= \xi(t = 0) \end{aligned} \tag{5.29}$$

where ϵ_{tol} is a small convergence tolerance. If a costate fulfils the condition in Equation 5.29 it is considered optimal [13].

The bisection method is a iterative method that after each iteration bisects the searching interval, slowly converging towards the solution. The iteration is repeated until one of the exiting conditions are met.

There are three exiting conditions for when the algorithm is aborted: A specified maximum number of iterations, a convergence tolerance (ϵ_{tol} in Equation 5.29) and a tolerance for the minimum update magnitude of the costate.

Below, the bisection algorithm is explained.

Bisection algorithm:

Step1: Solve the DDP/ECMS problem using a predefined upper boundary costate (λ_{UB}) and check if $\Delta\xi > 0$. If the condition holds, set $a = \lambda_{UB}$ and proceed to *Step 2*, otherwise lower λ_{UB} and repeat *Step 1*.

Step2: Solve the DDP/ECMS problem using a predefined lower boundary costate (λ_{LB}) and check if $\Delta\xi < 0$. If the condition holds, set $b = \lambda_{LB}$ and proceed to *Step 3*, otherwise increase λ_{LB} and repeat *Step 2*.

Step3: Set $\lambda = \frac{a+b}{2}$ and solve the DDP/ECMS problem.

- If $\Delta\xi > 0$, set $a = \lambda$ and repeat *Step 3*.
- If $\Delta\xi < 0$, set $b = \lambda$ and repeat *Step 3*.
- Abort if any of the exiting conditions are met.

5.7 Solution split

As a high or low SoC might damage the battery there are limits on how much charge the battery is allowed to contain. The implemented strategy does not handle the SoC as a state which is why the SoC constraints needs to be controlled separately.

In [9] the battery capacity is assumed to be dimensioned so that the SoC constraints never are exceeded. However, in [15, 16] any SoC constraint exceeding solution is solved by finding the point with the largest deviation from the constraint and splitting the problem in the corresponding time step. The sub-problems are solved separately where the reference SoC in the split time step is set to the exceeded limit. This is repeated until the entire SoC trajectory is within the constraint limits. As the sub-problems are solved individually with separate SoC references, each sub-problem obtains a separate optimal costates value.

To explain this further, Figure 5.6-5.8 shows a SoC constraint exceeding solution and explains the steps in the strategy. In Figure 5.6 the highest deviation of SoC occurs at time $t = \tau_1$. The costate value is constant during the entire driving mission.

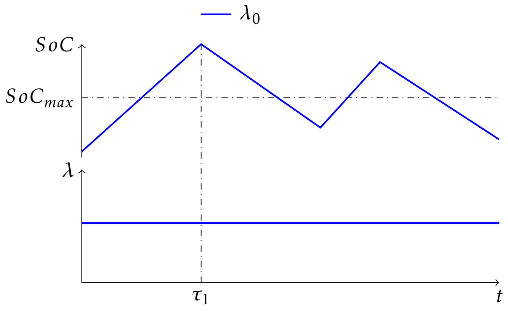


Figure 5.6: *Illustration of the solution split algorithm. The figure contains two graphs, a SoC trajectory and a graph showing the costate value for the trajectory. The upper blue line represent the SoC trajectory when $\lambda = \lambda_0$, it exceeds the SoC constraints with the highest deviation at τ_1. To handle this the problem is split into two sub-problems.*

The problem is split into two sub-problems in time $t = \tau_1$ and the problems are solved separately with the additional constraint $SoC(\tau_1) \approx SoC_{max}$. The result is shown in Figure 5.7. A new costate value for each interval is optimised to fit the new constraints (λ_{11}, λ_{12}). This is presented in comparison to the original costate value, λ_0, in the lower part of Figure 5.7. The new solution still exceeds the SoC constraint at $t = \tau_2$ which means that an additional solution split is required.

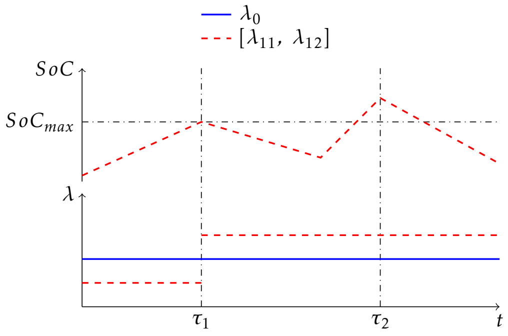
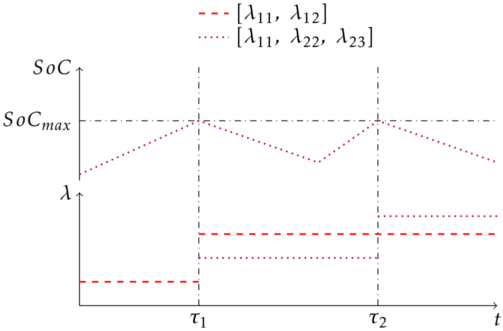


Figure 5.7: *Illustration of the solution split algorithm after the first iteration. The figure contains two graphs, the SoC trajectory (upper) and a graph showing the costate values for the trajectory (lower). The red dashed line represents the SoC trajectory when $\lambda = \lambda_{11}$ for $0 \leq t \leq \tau_1$ and $\lambda = \lambda_{12}$ for $t > \tau_1$. As comparison, the original costate (λ_0) is included. The solution still exceeds the SoC constraint with the highest deviation at τ_2. To handle this another solution split is performed.*

The new constraint $SoC(\tau_2) \approx SoC_{max}$ is introduced and the two sub-problems are solved yielding two new costate values, λ_{22} for the time interval $\tau_1 < t \leq \tau_2$ and λ_{23} for time interval $t > \tau_2$. The solution for the interval $0 \leq t \leq \tau_1$ is unchanged. The result is presented in Figure 5.8 where the entire SoC trajectory is within the constraints meaning that the solution is feasible.



- - - $[\lambda_{11}, \lambda_{12}]$
······ $[\lambda_{11}, \lambda_{22}, \lambda_{23}]$
SoC
SoC_{max}
λ
τ_1
τ_2
t

Figure 5.8: *Illustration of the solution split algorithm after the second iteration. The figure contains two graphs, the SoC trajectory (upper) and a graph showing the costate values for the trajectory (lower). The upper dotted line represents the SoC trajectory when $\lambda = \lambda_{11}$ for $0 \leq t \leq \tau_1$, $\lambda = \lambda_{22}$ for $\tau_1 < t \leq \tau_2$ and $\lambda = \lambda_{23}$ for $t > \tau_2$. The solution is within the SoC constraints meaning that the solution is feasible.*

In a more realistic scenario this would look more like Figure 5.9 where the solid line represent the solution before the solution split, the dashed line after the first iteration and the dotted line the final solution. The costate value for the time interval $0 \leq t \leq \tau_1$ is λ_1, λ_2 for $\tau_1 < t \leq \tau_2$ and when $t > \tau_2$ the costate value is λ_3.

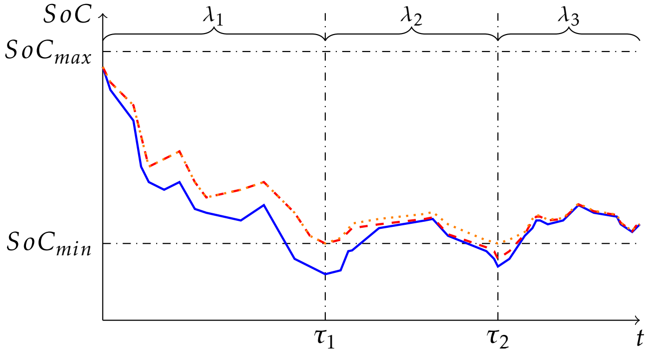


Figure 5.9: *Conceptual graph of the solution split algorithm. From the initial solution (solid line), the solution split algorithm is performed in two iterations. The first iteration handles the largest SoC constraint exceeding point which occurs at time τ_1. The result is represented by the dashed line. Since the SoC constraint is still exceeded a second iteration is performed. The result is presented as the dotted line. Since it is within the constraints during the entire drive cycle the solution is feasible.*

5.8 Complete algorithm

An overview of the full algorithm can be seen in Figure 5.10. It describes the connection between the different parts described earlier in the chapter.

The algorithm is initialised in the block Drive cycle data, where velocities and accelerations are calculated as described in Section 4.1. From the velocities and accelerations, the corresponding required wheel torque and power are calculated according to Equations 4.4 and 4.9. This is represented by the block RLP.

Next, the bisection method is initialised (explained in Section 5.6). A costate iteration is performed where the DDP/ECMS problem is solved. The DDP and ECMS algorithms are explained in Section 5.3 and 5.5 respectively. The iteration proceeds until one of the exiting conditions explained in Section 5.6 are met.

The final step consists of ensuring that the battery SoC does not exceed any limits. If the limit is exceeded the algorithm transitions into a solution split, explained in Section 5.7.

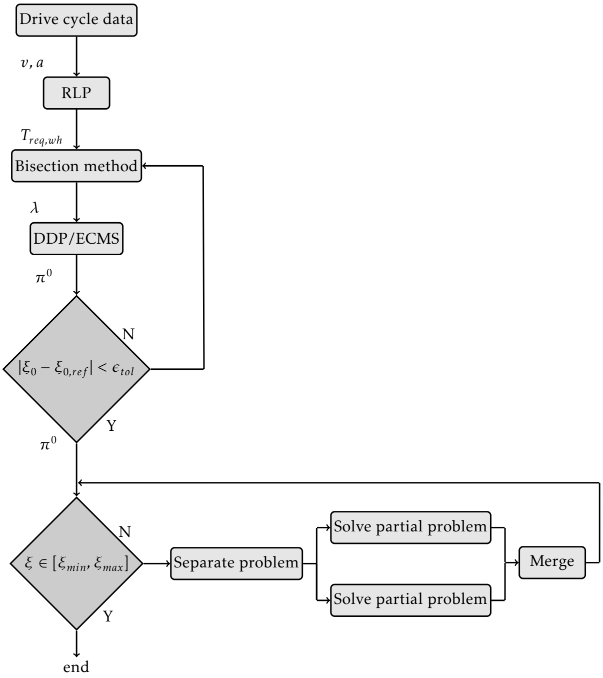


Figure 5.10: *Complete algorithm overview. The different parts are individually explained earlier in the chapter. Rectangular blocks represent operations and diamonds represent decisions. The two blocks at the top represents the vehicle and drive cycle data that gives the requested wheel torque which serves as input to the bisection method. The bisection method gives a costate value to the DDP/ECMS algorithm. An initial SoC is delivered from the DDP/ECMS block to the first decision block that decides whether the solution is good enough or if the costate needs to be updated. This decision is based on the exiting conditions described in Section 5.6. The next decision block is a SoC constraint check and if the SoC trajectory from the first loop exceeds the SoC constraint a solution split is perfomed according to the stages in Section 5.7.*

6 Validation

The developed EMS needs to be validated to ensure optimality and correct implementation.

One part of the validation consists of a comparison with strategies where the DDP/ECMS algorithm has been modified. The modifications consists of replacing a set of optimally decided control variables with logic. Also, the developed EMS is compared against a pure ECMS solution and a more complex method refered to as extended Hamiltonian. All this is covered in Section 6.1.

Furthermore, the developed EMS is compared with VSim, Volvo Cars' simulation environment for fuel economy and vehicle performance analysis. This to get a measurement of model accuracy and controller performance. This is further discussed in Section 6.2.

6.1 EMS

The investigated control signals replaced by logic are: **ERAD starts controlled by logic**, **ERAD starts and torque controlled by logic**, **ICE starts controlled by logic** and **gear controlled by logic**. To examine the gain of using a DP based optimiser, a comparison is made against a **pure ECMS** solution. Lastly, a comparison is done against a concept called the **Extended Hamiltonian**, which is an extension of the developed EMS.

6.1.1 ERAD controlled by logic

Two types of ERAD logic controllers are investigated.

The first one is **ERAD starts controlled by logic**, where the ERAD engagement is controlled by logic and the power split is optimised between all active actuators. The ERAD is engaged if the required wheel power is lower than a certain threshold or if the vehicle velocity is lower than a velocity threshold. The thresholds are obtained from VSim. The ERAD engagement also always occur when the ICE is turned off or if the requested wheel power exceeds the mechanical limit of the ICE.

The other ERAD logic uses the same logic for ERAD engagement, but instead of optimally distributing the power distribution over all available actuators this is done partially with logic. The logic states that the ICE and ISG provides all the power up to the maximum ICE limit, and optimally distributes the power between them. The ERAD provides any exceeding required power. This is the earlier mentioned **ERAD starts and torque controlled by logic strategy**.

Since ICE and gear still are solved by the optimal control strategy the possible operating modes for both the ERAD logic strategies are reduced to 10 which are listed in Table 6.1.

Table 6.1: *Specification of the 10 operating modes when the ERAD is controlled by logic. Regarding the ICE, status 1 represents the engine being turned on and 0 turned off. The list of gears represent the gear number associated with the corresponding mode. Gear 0 represents neutral.*

Operating mode	ERAD	ICE	Gear
1	X	1	0
2	X	1	1
3	X	1	2
4	X	1	3
5	X	1	4
6	X	1	5
7	X	1	6
8	X	1	7
9	X	1	8
10	X	0	0

6.1.2 ICE starts controlled by logic

This strategy uses logic to decide whether the ICE should be on or off. The logic is based on the vehicle velocity, required wheel power and the battery SoC. Using a look-up table, interpolation is used to determine if an ICE start is needed. Using this strategy the original operating modes are reduced to 18 according to Table 6.2.

Table 6.2: *Specification of the 18 available operating modes when the ICE starts are controlled by logic. Regarding the ERAD, status 1 represents it being enabled and 0 disabled. X means that the ICE is controlled by logic. The list of gears represents the gear number associated with the corresponding mode. Gear 0 represents neutral.*

Operating mode	ERAD	ICE	Gear
1	0	X	0
2	0	X	1
3	0	X	2
4	0	X	3
5	0	X	4
6	0	X	5
7	0	X	6
8	0	X	7
9	0	X	8
10	1	X	0
11	1	X	1
12	1	X	2
13	1	X	3
14	1	X	4
15	1	X	5
16	1	X	6
17	1	X	7
18	1	X	8

6.1.3 Gear controlled by logic

Like the ICE starts, the logic controlled gear is interpolated from look-up tables based on vehicle velocity, requested wheel torque and SoC. This limits the number of operating modes to 4, according to Table 6.3.

Table 6.3: *Specification of the 4 operating modes when the gear is controlled by logic. Regarding the ICE and ERAD, the status 1 represents an actuator being turned on and 0 represents a turned off actuator. Gear X represents gears 0-8 where 0 represents the neutral.*

Operating mode	ERAD	ICE	Gear
1	0	1	X
2	1	1	X
3	1	0	0
4	0	0	0

6.1.4 Pure ECMS

To investigate the benefits of utilising a DP algorithm, the developed EMS is compared to a pure ECMS strategy. With the ECMS strategy, the operating mode with the lowest Hamiltonian is chosen in each time interval. No transition costs are taken into consideration for the decision.

6.1.5 Extended Hamiltonian

The battery losses are dependent of how much current that is flowing through the battery. Therefore the transition costs between modes should be included in the battery loss calculation in the ECMS to increase the accuracy of the algorithm. However, this means that the upcoming mode must be known in prior which increases the computational burden of the ECMS algorithm from $\mathcal{O}(N_t \cdot N_{om})$ to $\mathcal{O}(N_t \cdot N_{om}^2)$ since all combinations of modes needs to be explicitly considered in each time step. N_t represents the number of discretised time steps and N_{om} the number of operating modes. This also means that the SoC value for the investigated operating mode can be used instead of the mean value as in the developed EMS.

The impact of the transition cost and SoC simplifications are investigated by comparing the results from the developed algorithm with the algorithm that uses extended Hamiltonian computations.

6.2 Comparison with VSim

One of the tools used for fuel economy and performance analysis at Volvo Cars is their self-developed, SIMULINK-based simulation environment VSim. The simulation tool contains a full vehicle model with corresponding subsystems and a driver model.

Implementing the developed EMS in VSim can be interesting in several aspects. Two of them, which are here investigated:

- Can the currently used EMS in VSim be improved with the DDP/ECMS-based EMS developed in this thesis in the aspect of reducing fuel consumption.
- How well does the developed EMS, based on simple models, perform when simulated in an advanced, complete vehicle model.

The DP-algorithm is an offline optimisation method that evaluates several potential solutions in each time step. Thus implementing it in SIMULINK would be complex and would implicate long simulation times. Therefore, some modifications of the algorithm has to be made before it can be implemented. The ECMS algorithm on the other hand can be simulated in real time why it is implementable in its original state.

The modification done to the DP-algorithm is how the discrete decision variables are implemented. The discrete decision variables are generated from a DDP/ECMS simulation and stored together with information of the time interval in which they are active. The information is treated as an input to Vsim and replaces the original strategy for gear selection, ICE status (ON/OFF) and ERAD status (ON/OFF) in Vsim.

Due to the model differences, simulating results that in DDP/ECMS resulted in a SoC close to the predefined reference does not ensure that the same is guaranteed in VSim. Consequently, a look-up table is generated by doing several DDP/ECMS simulations where the costate is slightly increased and decreased from the optimal costate. The costates and their corresponding discrete control variables are stored in the look-up table. The Vsim simulations are iterated using the bisection method, where the costate is updated until any of the exiting conditions described in Section 5.6 are met. The set of discrete variables that are chosen in each iteration are the ones associated with the costate that is closest to the currently evaluated.

7 Results

The chapter consists of several sections that presents results when comparing the final algorithm with the other strategies discussed in Chapter 6. Two different drive cycles are analysed, the NEDC and the SHC. Their velocity profiles can be found in Appendix A.

7.1 EMS

The result from the EMS in comparison with the simplifications and verification methods of that strategy can be seen in Table 7.1 - Table 7.6. The result covers CS simulations for several drive cycles as well as simulations with a battery depletion.

The presented results are fuel consumption, difference in initial SoC, number of ICE starts, number of gear shifts and number of ERAD starts. The difference in initial SoC is presented as $\Delta\xi_0 = \xi_{0,DDP/ECMS} - \xi_{0,logic}$, which means that a negative $\Delta\xi_0$ means that more electric energy has been used by the logic based strategy and vice versa. The other results are presented relative to the original DDP/ECMS strategy.

7.1.1 ERAD controlled by logic

The following results are a comparison between the developed EMS and the logic controlled ERAD strategy. Table 7.1 shows the results with logic controlled ERAD starts and Table 7.2 when ERAD starts and torque are controlled by logic.

ERAD starts controlled by logic

Table 7.1: *Difference in fuel consumption and actuator behaviour between the developed EMS and the strategy when ERAD starts are controlled by logic. Charge sustaining cases as well as a full battery depletion for the SHC are investigated. The values are presented relative to the results for the developed EMS. The difference in initial SoC is defined so that $\Delta\xi_0 < 0$ means that more electric energy has been used by the logic based strategy and vice versa.*

Charge Sustaining					
Drive cycle	**Fuel cons.**	$\Delta\xi_0$, **[%]**	**ICE starts**	**Gear shifts**	**ERAD starts**
NEDC	1.012	-0.04	1.00	1.025	0.14
SHC	1.006	0.08	1.225	1.25	1.00
Full Depletion					
SHC	1.01	0.015	1.22	1.36	0.632

Worth to notice is the difference in ERAD starts in the NEDC cycle. The NEDC consists of mostly low velocities and the logic based controller keeps the ERAD engaged below a certain velocity threshold. One big difference is that the developed EMS disengages the ERAD during stand-still while the logic based strategy does not. From the NEDC it can be seen that the fuel consumption is higher than the developed EMS at the same time as more electric energy is used. This indicates that there is a potential fuel saving in optimising ERAD starts.

ERAD starts and torque controlled by logic

Table 7.2: Difference in result between the developed EMS and the strategy where ERAD starts and torque are controlled by logic. Charge sustaining cases as well as a full battery depletion for the SHC are investigated. The values are normalised with the results for the developed EMS. The difference in initial SoC is defined so that $\Delta\xi_0 < 0$ means that more electric energy has been used by the logic based strategy and vice versa.

Charge Sustaining					
Drive cycle	**Fuel cons.**	**$\Delta\xi_0$, [%]**	**ICE starts**	**Gear shifts**	**ERAD starts**
NEDC	1.0397	0.23	1.00	1.10	0.14
SHC	1.011	0.556	1.20	1.223	0.98
Full Depletion					
SHC	1.008	-0.038	1.33	1.389	0.605

In this logic strategy it often occurs that the ERAD is engaged but does not have any power output when the ICE is turned on. This is due to the fact that the ERAD start logic engages the ERAD but since the requested power at the wheels is still lower than the power limit of the ICE, the ERAD will still not be used for propulsion.

The similarity with the ERAD start logic can be seen in the NEDC where the actuator behaviour is very similar. However, the fuel consumption is higher when the torque is also controlled by logic.

7.1.2 ICE starts controlled by logic

Table 7.3 shows the results from the strategy where the ICE starts are controlled by logic.

Table 7.3: *Difference in fuel consumption and actuator behaviour between the developed EMS and the strategy when the ICE starts are controlled by logic. Charge sustaining cases as well as a full battery depletion for the SHC are investigated. The difference in initial SoC is defined so that $\Delta\xi_0 < 0$ means that more electric energy has been used by the logic based strategy and vice versa.*

Charge Sustaining					
Drive cycle	**Fuel cons.**	$\Delta\xi_0$, **[%]**	**ICE starts**	**Gear shifts**	**ERAD starts**
NEDC	1.033	0.24	1.00	0.95	0.95
SHC	1.19	-0.1	7.10	1.85	1.15
Full Depletion					
SHC	1.43	-0.003	15.78	1.06	1.00

Notice the increase in number of ICE starts when using the logic based strategy in the SHC. The logic controlled strategies are normally used with hysteresis meaning that they have a lower time limit of how long the actuator must be active. Such an implementation is problematic here, due to the optimisation method used. This explains the high number of ICE starts.

During the NEDC the logic ICE controller performs in a similar manner to the developed EMS. This could be because of how the logic look-up table used by the logic strategy is tuned. There is still a 3.3% potential fuel saving by optimising the ICE starts for the NEDC.

For the SHC, the fuel consumption is 19% higher during charge sustaining and 43% higher during blended mode than for the developed EMS. One of the reasons for this is the big difference in number of ICE starts.

7.1.3 Gear controlled by logic

The comparison between the developed EMS and the strategy when the gearshifts are controlled by logic is shown in Table 7.4.

Table 7.4: *Difference in result between the developed EMS and the strategy when gearshifts are controlled by logic. Charge sustaining cases as well as a full battery depletion for the SHC are investigated. The results are presented relative to the results from the developed EMS. The difference in initial SoC is defined so that $\Delta\xi_0 < 0$ means that more electric energy has been used by the logic based strategy and vice versa.*

Charge Sustaining					
Drive cycle	**Fuel cons.**	$\Delta\xi_0$, [%]	**ICE starts**	**Gear shifts**	**ERAD starts**
NEDC	1.026	0.127	1.00	0.90	1.05
SHC	1.0033	-0.40	0.975	1.044	0.92
Full Depletion					
SHC	1.013	0.045	0.94	1.15	1.00

Notice that for the SHC in charge sustaining the logic controlled strategy has a higher fuel consumption than the developed EMS at the same time as it uses more electric energy.

7.1.4 Pure ECMS

The result from the ECMS simulation, seen in Table 7.5, shows how selecting the local optima in each time step performs in comparison to selecting a close to global optima over the entire drive cycle.

Table 7.5: *Difference in fuel consumption and actuator behaviour between the developed EMS and the ECMS strategy. Charge sustaining cases as well as blended mode with a full battery depletion for the SHC are investigated. The results are presented relative to the results from the developed EMS. The difference in initial SoC is defined so that $\Delta\xi_0 < 0$ means that more electric energy has been used by the logic based strategy and vice versa.*

Charge Sustaining					
Drive cycle	**Fuel cons.**	$\Delta\xi_0$, [%]	**ICE starts**	**Gear shifts**	**ERAD starts**
NEDC	1.82	0.06	4.00	4.93	1.43
SHC	1.83	0.11	6.40	5.66	6.88
Full Depletion					
SHC	2.98	0.017	20.17	21.29	10.39

It can be noted that the ECMS strategy changes the states of the actuators very frequently. This gives an indication of the importance of implementing transition costs.

7.1.5 Extended Hamiltonian

The results from the DDP/ECMS simulations with an extended Hamiltonian calculation can be seen in Table 7.6.

Table 7.6: *Difference in fuel consumption and actuator behaviour between the developed EMS and the EMS with extended Hamiltonian computations. Charge sustaining cases as well as a full battery depletion for the SHC are investigated. The results are presented relative to the results from the developed EMS. The difference in initial SoC is defined so that $\Delta\xi_0 < 0$ means that more electric energy has been used by the logic based strategy and vice versa.*

Charge Sustaining					
Drive cycle	**Fuel cons.**	$\Delta\xi_0$	**ICE starts**	**Gear shifts**	**ERAD starts**
NEDC	0.998	-0.0349	1.00	1.00	1.00
SHC	1.001	0.14	1.05	1.03	1.08
Full Depletion					
SHC	0.999	-0.379	1.05	1.03	1.08

Notice the similarity in the actuator behaviour between the two algorithms.

The NEDC indicates a lower fuel consumption for the strategy with extended Hamiltonian computations while the SHC in charge sustaining indicates a higher consumption. This can be explained by the difference in initial SoC and that the effect of more precisely calculated costs during transitions between operating modes makes a very small difference to the decision of active operating mode.

7.2 Comparison with Vsim

As discussed in Section 6.2, implementing the developed DDP/ECMS strategy in Vsim could provide some insight about its performance. Here, the total fuel consumption is compared using three strategies. One where the DDP/ECMS (denoted f) is simulated in its original state. The other two are based on Vsim simulations where the first one (denoted $f_{Vsim,mod}$) is the DDP/ECMS strategy implemented in Vsim as described in Section 6.2. The second is when using the original control strategy used in Vsim (denoted f_{Vsim}).

The main reason for the Vsim comparison is to see if the currently implemented control strategy can be improved with the developed EMS. As such, the f_{Vsim} strategy will be used as a baseline strategy. The fuel consumption for the other strategies will be presented relative to this strategy. The costate will be presented relative to the f strategy.

As the Vsim simulations are simulated in a forward manner, the deviation from the SoC reference occurs at final time. The comparison between the electric energy usage is presented in an absolute level of depletion difference to the f_{Vsim} strategy ($\Delta\xi_{Vsim} - \Delta\xi$), where $\Delta\xi = \xi_0 - \xi_f$ for the compared strategy.

Four scenarios are presented, all simulated on the SHC. The main difference between the scenarios is the initial battery charge and referenced level of depletion. For all the scenarios, the f_{VSim} strategy has been simulated where only the inital SoC has been modified. The resulting final SoC together with the initial SoC has been set as a reference to the other strategies. This to see how they perform under the same circumstances.

7.2.1 Scenario 1

The results from Scenario 1 can be seen in Table 7.7. The corresponding SoC trajectories are presented in Figure 7.1.

From the table and figure it can be seen that the f and $f_{Vsim,mod}$ strategy have similar costate and SoC trajectory. The $f_{Vsim,mod}$ strategy consumes slightly more fuel and electric energy than the f strategy.

Looking at the SoC trajectory for the f_{Vsim} strategy it can be seen that the strategy here has a CDCS behaviour. Comparing the fuel consumption for the $f_{Vsim,mod}$ and f_{Vsim} strategy a fuel reduction of 5.9% when using the $f_{Vsim,mod}$ strategy can be observed, this while consuming approximately the same amount of electric energy.

Table 7.7: *Results from Scenario 1 for the f,$f_{Vsim,mod}$ and f_{Vsim} strategies on the SHC. The total fuel consumption (m_f) is presented relative to the f_{Vsim} strategy. The costate is presented relative to the f strategy. The amount of electric energy consumed is presented in absolute level of depletion difference to the f_{Vsim} strategy.*

Scenario 1			
Strategy	m_f	λ	$\Delta\xi_{Vsim} - \Delta\xi$ [%]
f_{Vsim}	1.000	-	0
f	0.931	1.000	0.276
$f_{Vsim,mod}$	0.941	0.996	-0.014

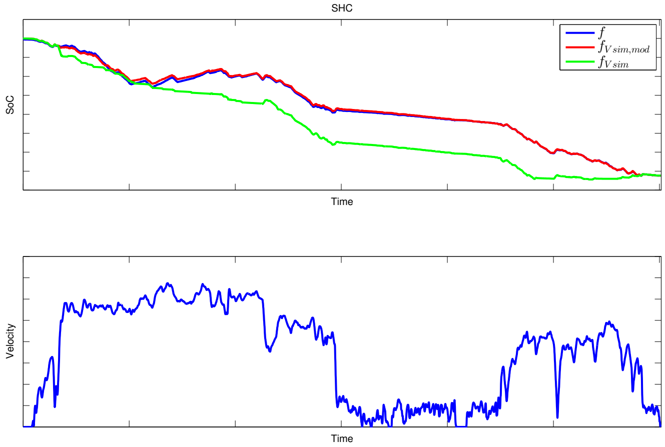


Figure 7.1: *State of charge trajectories for Scenario 1 on the SHC cycle. The blue line corresponds to the f strategy, the red line to the $f_{Vsim,mod}$ strategy and the green line to the f_{Vsim} strategy. All strategies have the same depletion reference. The velocity profile for the drive cycle is included in the figure.*

7.2.2 Scenario 2 and Scenario 3

In Table 7.8, the results from Scenario 2 and Scenario 3 is presented. The f_{Vsim} strategy does not deplete the battery to a minimum level in Scenario 2, why Scenario 3 is analysed for the f and $f_{Vsim,mod}$ strategy to see how much additional fuel can be saved by depleting the battery. The SoC trajectories for Scenario 2 can be seen in Figure 7.2.

The results from Scenario 2 shows a good agreement between the f and the $f_{Vsim,mod}$ strategy as both the SoC trajectory and costate are similar. Regarding the fuel consumption, the f strategy has a slightly higher consumption while consuming more electric energy. In Scenario 3, the $f_{Vsim,mod}$ has a higher depletion which explains the lowered fuel consumption and lower costate.

Comparing the f_{Vsim} and $f_{Vsim,mod}$ strategies, the fuel consumption is 5.6% lower for the $f_{Vsim,mod}$ in Scenario 2. Both strategies uses approximately the same amount of electric energy. By depleting the battery and using the $f_{Vsim,mod}$ strategy, a reduction of 24.4% in fuel consumption is obtained compared to the f_{Vsim} strategy which does not fully deplete the battery.

Table 7.8: *Results from Scenario 2 and Scenario 3 for the f,$f_{Vsim,mod}$ and f_{Vsim} strategies on the SHC. The total fuel consumption (m_f) is presented relative to the f_{Vsim} strategy. The costate is presented relative to the f strategy. The amount of electric energy consumed is presented in absolute level of depletion difference to the f_{Vsim} strategy.*

Scenario 2			
Strategy	m_f	λ	$\Delta\xi_{Vsim} - \Delta\xi$ [%]
f_{Vsim}	1.000	-	0
f	0.965	1.000	-0.244
$f_{Vsim,mod}$	0.944	0.989	0.002
Scenario 3			
Strategy	m_f	λ	$\Delta\xi_{Vsim} - \Delta\xi$ [%]
f	0.845	0.991	-5.891
$f_{Vsim,mod}$	0.756	0.927	-8.029

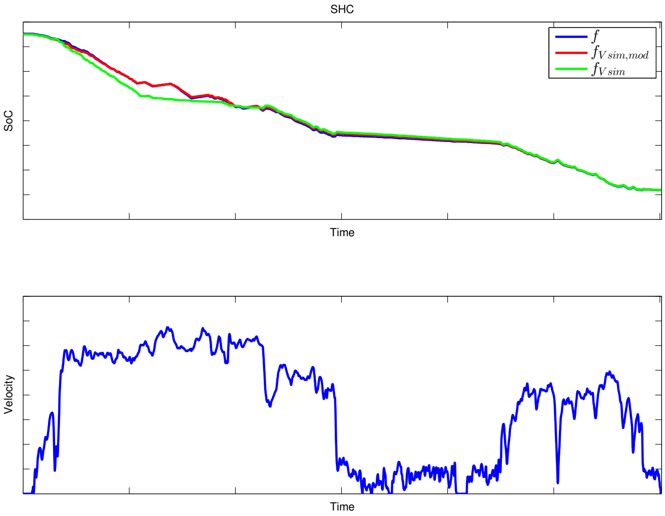


Figure 7.2: *State of charge trajectories for Scenario 2 on the SHC cycle. The blue line corresponds to the f strategy, the red line to the $f_{Vsim,mod}$ strategy and the green line to the f_{Vsim} strategy. All strategies have the same depletion reference. The velocity profile for the drive cycle is included in the figure..*

7.2.3 Scenario 4

In Scenario 4, a CS simulation is studied. The results are summarised in Table 7.9 and the corresponding SoC trajectories are presented in Figure 7.3.

Comparing the results between the f strategy and the $f_{Vsim,mod}$ strategy, the f strategy consumes less fuel while consuming more electric energy. The SoC trajectories between the same strategies show a similar behaviour.

With the $f_{Vsim,mod}$ strategy, a 3.6% reduction in fuel consumption is obtained compared to the f_{Vsim} strategy. This while using slightly less electric energy.

Table 7.9: *Results from Scenario 4 for the f,$f_{Vsim,mod}$ and f_{Vsim} strategies on the SHC. The total fuel consumption (m_f) is presented relative to the f_{Vsim} strategy. The costate is presented relative to the f strategy. The amount of electric energy consumed is presented in absolute level of depletion difference to the f_{Vsim} strategy.*

Scenario 4			
Strategy	m_f	λ	$\Delta\xi_{Vsim} - \Delta\xi$ [%]
f_{Vsim}	1.000	-	0
f	0.911	1.000	-0.227
$f_{Vsim,mod}$	0.964	1.205	0.026

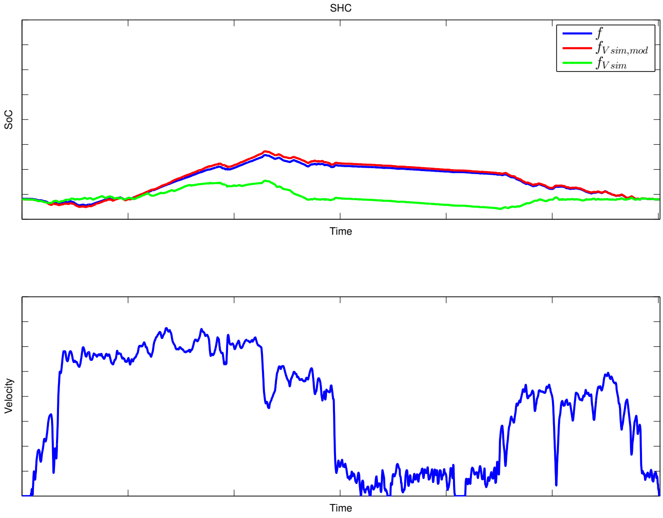


Figure 7.3: *State of charge trajectories for Scenario 4 on the SHC cycle. The blue line corresponds to the f strategy, the red line to the $f_{Vsim,mod}$ strategy and the green line to the f_{Vsim} strategy. All strategies have the same charge sustaining reference. The velocity profile for the drive cycle is included in the figure..*

7.3 Reoccuring behaviour of the DDP/ECMS strategy

7.3.1 Choice of operating mode

At some points during the simulations all actuators are enabled at the same time. During some of those occurrences both the ISG and the ERAD have a negative torque. This means that the ICE propels the vehicle while both the ISG and ERAD charge the battery. In Figure 7.4 this behaviour can be seen during a time interval in the later part of the NEDC.

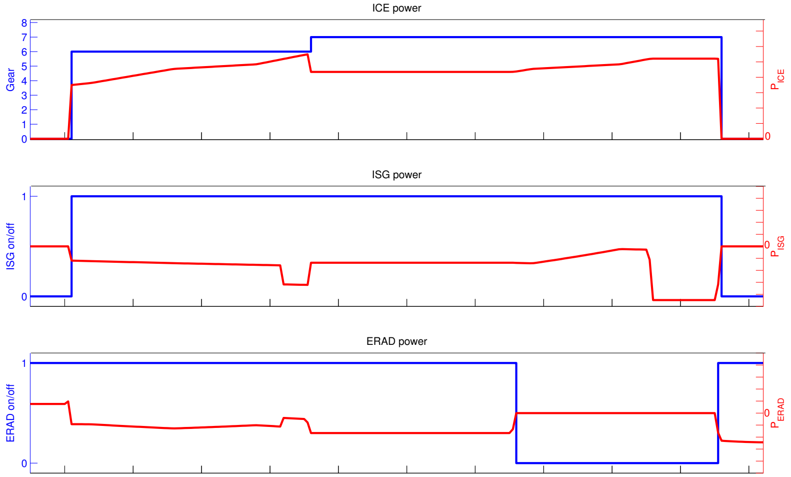


Figure 7.4: *Graphs showing the state of the actuators* $(1 = on,\ 0 = off)$ *together with corresponding actuator output power during an interval in the later part of the NEDC. For a period of time both the ISG and the ERAD have negative power.*

7.3.2 ERAD synchronisation

During low velocity parts of some decelerations the algorithm disengages the ERAD and enables it again shortly afterwards, see Figure 7.5. This means that the ERAD is turned off instead of taking advantage of regenerative braking. During ERAD disengagement the kinetic energy is transformed into electric energy which charges the battery.

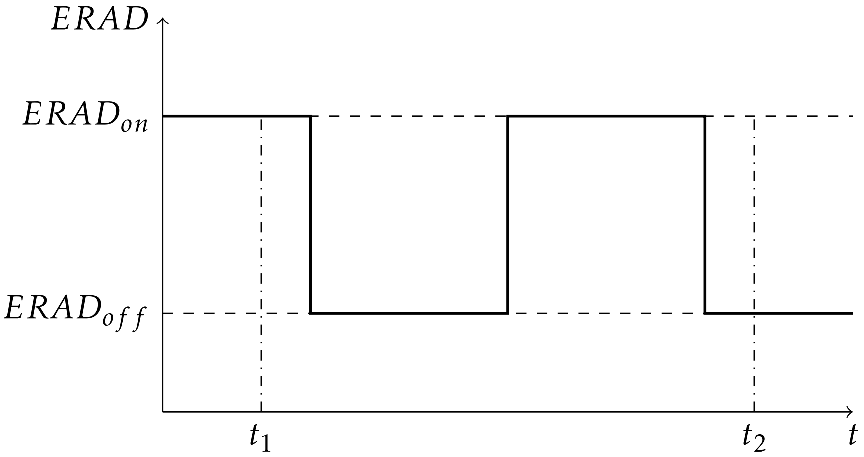


Figure 7.5: *Behaviour of the ERAD low velocities in some decelerations. The ERAD disengages during a short period of time just to be engaged again short thereafter.*

7.4 Solution split

All the results previously presented does not exceed the SoC constraints and are therefore not affected by the solution split algorithm. To validate the solution split a test with manipulated SoC limits is performed to make sure that the SoC constraints are exceeded. For this test the charge sustaining NEDC is used and the SoC trajectory before and after the solution split is shown as the blue and red line in Figure 7.6. The fuel consumption, initial SoC and costate values are presented in Table 7.10.

	f	f_{split}
m_f	1.00	1.03
SoC_0	1.00	0.988
λ	1.00	[0.986, 1.011]

Table 7.10: *Comparison of the result from the developed EMS, denoted f, and the manipulated SoC exceeding solution split, denoted f_{split}. The numbers are normalised with the results from the developed EMS.*

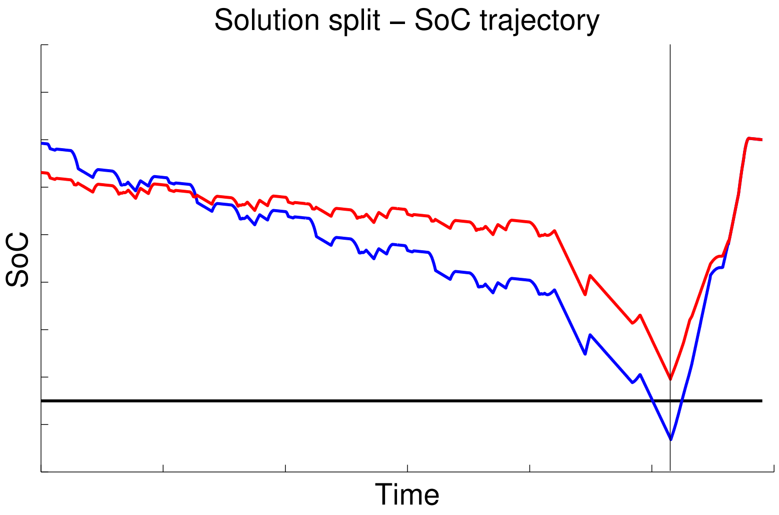


Figure 7.6: *The SoC trajectory during the solution split algorithm in the NEDC. The blue line is the SoC trajectory before the solution split and the red line after the split. The thick black line represents the manipulated lower SoC constraint and the thin vertical line represent the split time.*

The reason to why the red line does not end up exactly on the SoC limit is that the bisection method is used to iterate a new costate value that satisfies the new SoC constraints. Since the bisection method is numerical and iterative and the exit condition is set as a tolerance, it will yield a SoC value close to the new references.

8

Analysis of results

8.1 Bisection convergence

As seen from the results, the amount of battery energy used varies between the compared strategies even though they are set to have the same level of battery depletion. This is partly due to the fact that one of the exiting criterias of the bisection method is a tolerance. Another effect, discussed in [9], is that when the functionality to start/stop the ICE is implemented a non-linearity is potentially introduced in the relation between initial SoC and costate.

This might implicate problems in finding a initial SoC close to the reference as a small change in the costate can lead to a drastic change in the initial SoC. This can be explained by the fact that the small change in the costate can alter the optimal discrete control sequence. An illustration of this phenomenon is shown in Figure 8.1, where λ_{switch} is the costate value where the optimal discrete control sequence change.

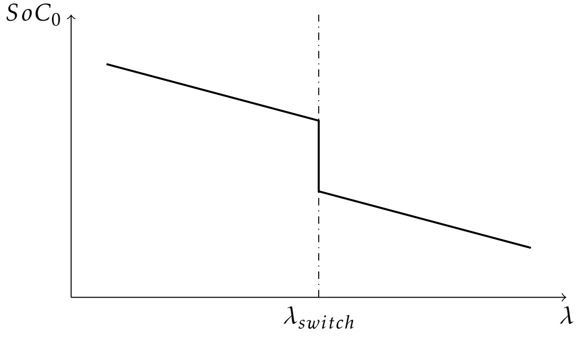


Figure 8.1: *Illustration of a non-linear effect between the reference SoC and the costate. At certain costate values (λ_{switch}) the optimal discrete control sequence is altered which leads to a large change in the initial SoC.*

As such, there are some intervals of an initial SoC that can not be reached, disregarded of how fine-tuned the costate is. Which is the other reason for the difference in electric energy usage seen in the results.

This entails some problems in the comparison of fuel consumption between different strategies and cycles. A higher electric energy consumption could mean that some parts of energy demand is covered by the excessive use of electric energy, potentially leading to a lower fuel consumption and vice versa. The authors of [15] transform the SoC deviation into equivalent fuel consumption. Such an approach is not treated here, this as the assumption is based on that the discrete control variables are unaltered. To avoid drawing misleading conclusions, the presented fuel consumption has not been corrected with respect to SoC reference deviation.

8.2 EMS

One of the goals of this thesis is to investigate potential fuel savings by optimising actuator behaviour. The results that validates this can be seen in Table 7.1 - 7.4 and will be discussed further in this section.

From the presented results it is seen that some of the logic controlled strategies are shifting modes more frequently than the developed EMS. This behaviour may not be desired in terms of driveability which is why most of the logic algorithms in Vsim have restrictions of a minimum time an actuator has to be active. However, the thesis aim is to find the optimal control in terms of fuel consumption which is why no such heuristics has been investigated.

8.2.1 ERAD controlled by logic

ERAD starts controlled by logic

Table 7.1 indicates that there is a potential improvement in fuel consumption when optimising the ERAD behaviour compared to the logic controlled. For the NEDC the obtained fuel saving is around 1.2% with the developed EMS.

ERAD starts and torque controlled by logic

To control ERAD starts and torque with logic seems to result in a higher fuel consumption than if just ERAD starts are logically controlled. This can be justified with the difference of an optimal power split between all active actuators, for the ERAD start logic, and a partially optimal power split between the ICE and the ISG and a logic controlled ERAD power, for the ERAD starts and torque logic.

8.2.2 ICE starts controlled by logic

The results in Table 7.3 indicate that there is a big potential fuel saving in the optimisation of the ICE starts. Since a large number of ICE starts could affect the fuel consumption and driveability negatively this logic should be used in hysteresis with other constraining controllers that, for example, controls the time between engine starts. Since the logic based strategy does not consider the costs for ICE starts, such a hysteresis could potentially even lower the fuel consumption.

The influence of the additional fuel due to the extra ICE starts is investigated to determine the credibility of the result. This is done for the charge sustaining SHC by only including the fuel cost for the same number of ICE starts as for the developed EMS. The result shows that without extra fuel costs the logic based controller still result in approximately 2.9% higher fuel consumption.

The results indicates that the ICE starts is an important discrete variable to be optimised.

8.2.3 Gear controlled by logic

The developed EMS takes the fuel cost associated with down shifts into consideration while the logic controlled gearshifts does not. This could be one aspect of why the fuel consumption is lower for the developed EMS. Another aspect can be how the gearshift logic is tuned. For example, if it is tuned in order to maximise the ICE efficiency or the efficiency of the entire drive line.

8.2.4 Pure ECMS

The result for the ECMS solver shows the effect of utilising a DP based algorithm. The fuel consumption for the pure ECMS is considerably higher than the developed EMS. Mode changes, ICE starts, ERAD engagements and gearshifts, occurs much more frequently which can be explained by the fact that transition costs are not taken into account when choosing the operating mode.

The reason for the high fuel consumption and frequent state shifts of the actuators is the fact that the mode with the lowest cost is chosen in each time step. In some cases, for example during a period with constant velocity in the later part of the NEDC, the cost between two modes are so similar that the ECMS oscillates between them. The reason for this could, for example, be the small influence of the SoC.

8.2.5 Extended Hamiltonian

The strategy with the extended Hamiltonian computations should perform as good or slightly better than the developed EMS. The reason for this is that the power split is calculated in the same way and with the same information in both cases. However, the more accurate calculation of the transition costs used when making the decision of active operating mode should make the algorithm with extended Hamiltonian computations more precise. In Table 7.6 the fuel consumption shows a very similar behaviour between the two algorithms which indicates that the extra complexity of the extended Hamiltonian does not improve the control to any greater extent.

8.3 Comparison with Vsim

Comparing the f strategy with the $f_{Vsim,mod}$ strategy yields a measurement of the quality of the simplified models used in the DDP/ECMS strategy.

From all the studied scenarios it is seen that the results from the two strategies are very similar. The SoC trajectories have almost an identical appearance, the small deviations are partly due to the initial and final offset. The fuel consumptions vary due to the difference in used electric energy but also due to facts such as different models for the fuel required to start the engine. In summary, from the results we see an overall similar behaviour between the two strategies why the simplified models seems to catch the fundamental behaviour of the system.

With the model agreement, the $f_{Vsim,mod}$ strategy can be compared with the f_{Vsim} strategy to get a quality measurement of the developed control strategy. For all the studied scenarios the $f_{Vsim,mod}$ strategy yields a lower fuel consumption than the f_{Vsim} strategy. In the studied scenarios the reduction spans from 3.6% to 5.6% while using the same amount of electric energy. In Scenario 3 where the $f_{Vsim,mod}$ strategy depletes the battery, the reduction is 24.4%.

All this suggests that the developed ECMS/DDP strategy performs better than the current EMS implemented in Vsim in the aspect of fuel economy.

8.4 Reoccuring behaviour of the DDP/ECMS strategy

8.4.1 Choice of operating mode

As mentioned in Section 7.3.1, occasionally the ERAD and ISG are charging the battery while the ICE is propelling the vehicle. The reason for this behaviour is investigated by looking at the efficiency if only one of the electric actuator is used compared to the total efficiency in the optimal control strategy. The efficiency is calculated by a comparison of the available power for charging with the actual charging power. Equation (8.1) shows how the requested power can be distributed between the actuators.

$$P_{req,wh} = P_{ICE} + P_{ISG} + P_{ERAD} - P_{GB,loss}\left(P_{ICE}, P_{ISG}\right) - P_{EGB,loss}\left(P_{ERAD}\right) \tag{8.1}$$

$$P_{GB,loss}(P_{ICE}, P_{ISG}) = f(P_{ICE} + P_{ISG}) \tag{8.2}$$

$$P_{EGB,loss} = f(P_{ERAD}) \tag{8.3}$$

If only the ICE is active the required ICE power is calculated according to Equation (8.4)

$$P^{*}_{ICE} = P_{req,wh} + P_{GB,loss} \tag{8.4}$$

where $P_{req,wh}$ is the required power at the wheels for the active power distribution.

If the vector $P_{ICE,vec} = [P_{ICE,min}, P_{ICE,max}]$ represents all the possible ICE powers and only one of the ISG or ERAD is active, the ISG and ERAD powers can be described according to Equation (8.5) and (8.6).

$$P_{ISG} = P_{req,wh} + P_{GB,loss} - P_{ICE,vec} \tag{8.5}$$

$$P_{ERAD} = P_{req,wh} + P_{GB,loss} + P_{EGB,loss} - P_{ICE,vec} \tag{8.6}$$

The efficiencies are calculated according to Equation (8.7) and (8.8).

$$\eta_{ISG} = \left(\frac{P_{ISG}}{P_{ICE,vec} - P^*_{ICE}} \right)^{sign(P_{ISG})} \tag{8.7}$$

$$\eta_{ERAD} = \left(\frac{P_{ERAD}}{P_{ICE,vec} - P^*_{ICE}} \right)^{sign(P_{ERAD})} \tag{8.8}$$

The results from these calculations is shown in figure 8.2. The circle represents the combined efficiency of the EMS and for the current ICE power the simultaneous use of the ISG and the ERAD is higher than if only one of the electric actuators had been used.

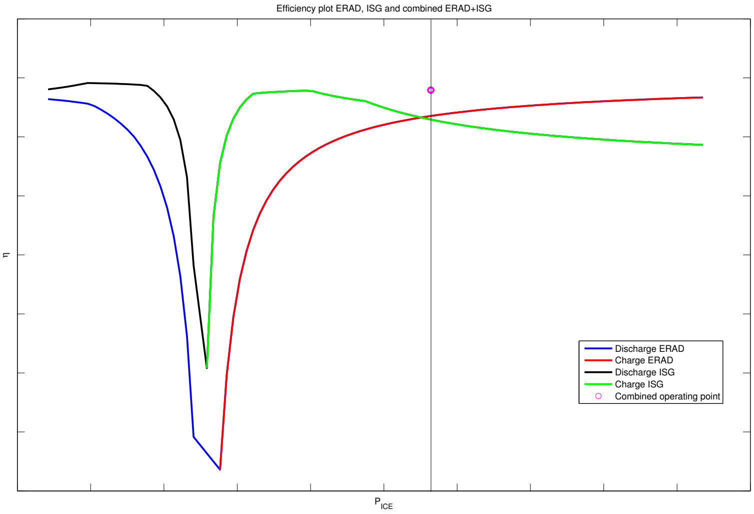


Figure 8.2: *The efficiencies of the ERAD and ISG as a function of ICE power during a time sample when all actuators are active simultaneously. The graph makes a difference if the actuator is charging or discharging the battery. The marked value on the x-axis is the current ICE power and the magenta ring marks the operating point chosen by the EMS. The curves represent the operating efficiency of a electric actuator when the power of the other electric actuator is set to zero.*

8.4.2 ERAD synchronisation

As mentioned in Section 7.3.2, the algorithm disengages the ERAD and enables it again short thereafter. To investigate the optimality of this behaviour the cost of shutting down and on the ERAD is compared to if the ERAD had been enabled the entire time. An illustration of this can be seen in Figure 8.3. Equation (8.9) represents the cost of the algorithm behaviour and Equation (8.10) represents the cost if the ERAD is enabled during the entire time.

$$\gamma_1 = \lambda \left(\int_{t_1}^{t_2} (P_{ERAD,ech})\, dt + E_{ERAD,off} - E_{ERAD,on} \right) \tag{8.9}$$

$$\gamma_2 = \lambda \int_{t_1}^{t_2} P_{ERAD,ech}\, dt \tag{8.10}$$

γ_X is the electric cost of the drive mission, $P_{ERAD,ech}$ is the ERAD contribution to the electrochemical power, $E_{ERAD,on}$ and $E_{ERAD,off}$ are the energy loss and gain from synchronisation of the ERAD.

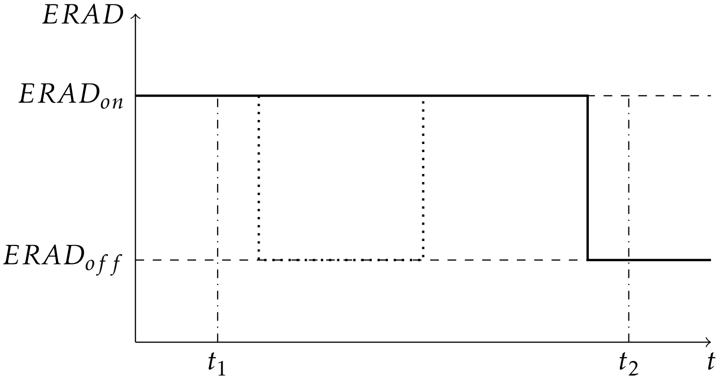


Figure 8.3: *The dotted line represent the ERAD behaviour according to the EMS during low velocities in decelerations and the filled line represents the ERAD in regenerative braking.*

The cost to turn the ERAD on and off is in the performed calculations, according to Equation (8.9) and (8.10), lower than the cost to keep the ERAD on. A reason for this could be that the low efficiency of the ERAD at low angular velocities makes the kinetic energy gain from the ERAD synchronisation, with its constant and assumed efficiency, more beneficial than the regenerative braking for that specific time period. Another aspect to it can be the time discretisation that makes the velocity, and thus the required torque, vary between the time steps.

8.5 Solution split

The result in Table 7.10 indicates that the fuel consumption after the solution split is higher than before. This could be expected since the problem optimised in the solution split is constrained by the SoC limits and therefore the feasible solution candidates will be limited compared to when no SoC constraints are affecting the solution.

Figure 7.6 indicates that the new SoC trajectory is within the SoC constraints and therefore the solution split has successfully fulfilled its purpose.

9 Conclusions and future work

9.1 Conclusions

From the analysis of the results, some conclusions can be drawn. The most significant are summarised in the list below.

- The results from the simulations with the developed EMS indicates that it yields a lower fuel consumption than the logic based strategies. It can also be seen that it changes modes less frequently than the logic based strategies which can implicate a better driveability.
- The results indicate that it, in some cases, can be beneficial to charge the battery with both the ISG and the ERAD while simultaneously propelling the vehicle with the ICE. This is an interesting result that is validated by the increased total efficiency when using both electric actuators, see Figure 8.2. However, the model used during the simulations is simplified and does not, for example, take any slipping in the wheels or driveline wind-up into consideration. Therefore the behaviour should be investigated with a more complex vehicle model in order to analyse it further.
- When comparing the results from the developed EMS and the EMS with extended Hamiltonian computations the difference is very small. From this the conclusion can be made that the extra computations does not give any substantial difference to the results. The reason for this could be that the more accurately calculated losses have little effect on the result. It can

also be assumed that the bisection algorithm compensates for the difference, leading to that the two strategies end up in very similar operational mode and power split trajectories.

- The results from the comparison with Vsim shows a good agreement between the models used in the developed EMS and the models used in Vsim. The developed control strategy yields a lower fuel consumption for all the studied scenarios compared to the currently implemented Vsim controller.

9.2 Future work

The developed product in its current state can be used as a benchmark against other control strategies. To improve the product further, additional analysis and development has to be made. This could be including additional physical effects that as of now are simplified or neglected, this to get a model that better describes a real vehicle.

An example of this could be examining the simplification of the transition between operating modes. In the current implementation it is assumed that the time it takes to start the engine, start the ERAD and perform a gearshift is lower than the time interval. For small time intervals this might not be a valid assumption. Additionally, the transition between the operating modes is assumed to be instantaneous. This means that effects such as torque loss during gear shifts are neglected. In reality, the transition is probably performed gradually.

Additionally, the following subjects are potential areas of further development:

- It this thesis, only fuel consumption analysis has been made. Aspects such as emissions and driveability could also be analysed. For example, as of now there is no minimum time limit of how long the engine has to be running. Setting such a limit would remove frequent engine starts and shutdowns which otherwise could have a negative effect on driveability. A similar solution could be investigated for gear shifts as frequent gear shifts may have a negative effect on driveability.

- Examine the effect of replacing the constant engine start costs with a more advanced model. This might adjust the optimal solution as there might exist more beneficial times when the engine should be started.

- To gain more insight about the accuracy of the developed control strategy it should be compared to a global optimisation strategy, for example a pure dynamic programming strategy. The developed EMS is not guaranteed to yield global optima due to the simplifications within the algorithm. One of those simplification is the use of a constant costate value. When minimising the Hamiltonian function, Equation (3.2), both the optimal trajectory of discrete states and the optimal costate value needs to be found. Equation (9.1) represents the derivative of the optimal costate value

$$\dot{\lambda}^* = \frac{-\partial H}{\partial \xi} \approx 0 \tag{9.1}$$

which is assumed to be zero. However, this is not entirely true as the battery internal resistance and open-circuit voltage is affected by the SoC. The dependency can, according to [7], be neglected when the open voltage of the battery is assumed to be constant, which is usually done when using the ECMS. But the simplification could still potentially lead to non global yielding solutions.

- The utilisation of a constant costate is relying on the assumption that the internal resistance and open circuit voltage are independent of the SoC. For a PHEV this can be considered to be a rough estimation as the battery is operating in a wide SoC range. An alternative would be using a time varying costate. Such an approach has been investigated in [6] and [14], where the costate is updated depending on the difference between the SoC and a discharge reference.

- If Volvo aims to implement the developed EMS in Vsim or to improve their currently implemented EMS, further analysis should be done on the individual effects of the DDP and ECMS. The thesis does not investigate this aspect mostly due to the difficulties of obtaining the reference depletion when implementing the strategies separately. Furthermore, some modifications in the implementation can be done. As of now, the DDP requires a pre-generated, problem-specific lookup table of the discrete control variables which might not be desirable, instead a Vsim integrated alternative should be investigated.

Appendix

A

Drive cycles

A.1 SHC

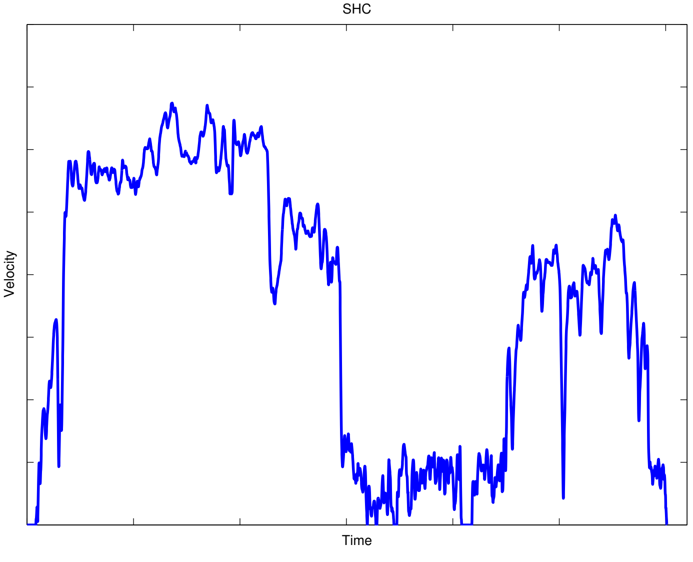


Figure A.1: *Velocity profile of the SHC.*

A.2 NEDC

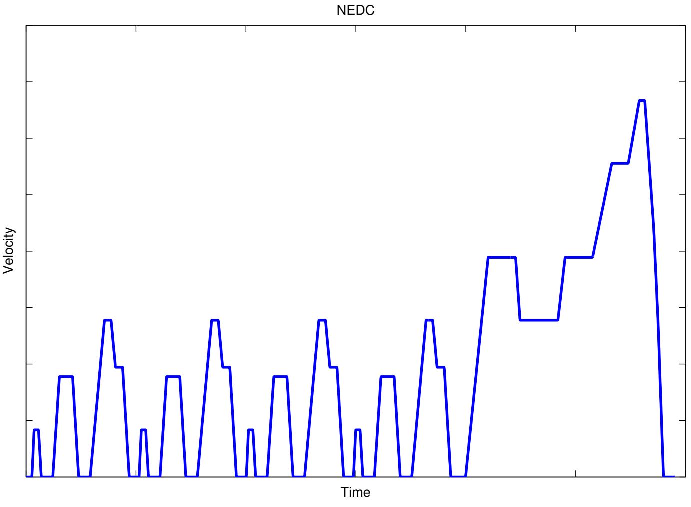


Figure A.2: *Velocity profile of the NEDC.*

Bibliography

[1] Dimitri P. Bertsekas. Dynamic Programming and Optimal Control, volume 1. Athena Scientific, 3 edition, 1995. Cited on pages 15 and 16.

[2] Wisdom Enang and Chris Bannister. Robust proportional ecms control of a parallel hybrid electric vehicle. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 231(1):99–119, 2017. Cited on pages 3 and 4.

[3] Lino Guzzella, Antonio Sciarretta, et al. Vehicle propulsion systems, volume 1. Springer, 3 edition, 2013. Cited on pages 7, 8, 9, 10, 15, 16, 25, and 30.

[4] Cong Hou, Minggao Ouyang, Liangfei Xu, and Hewu Wang. Approximate pontryagin's minimum principle applied to the energy management of plug-in hybrid electric vehicles. Applied Energy, 115:174–189, 2014. Cited on page 4.

[5] S Kutter and B Bäker. An iterative algorithm for the global optimal predictive control of hybrid electric vehicles. In Vehicle Power and Propulsion Conference (VPPC), 2011 IEEE, pages 1–6. IEEE, 2011. Cited on page 34.

[6] Federica Lacandia, Laura Tribioli, Simona Onori, and Giorgio Rizzoni. Adaptive energy management strategy calibration in phevs based on a sensitivity study. SAE International Journal of Alternative Powertrains, 2(2013-24-0074):443–455, 2013. Cited on page 71.

[7] Viktor Larsson. Route optimized energy management of plug-in hybrid electric vehicles. Chalmers University of Technology, 2014. Cited on pages 16, 19, 20, and 71.

[8] Jan Lundgren, Mikael Rönnqvist, and Peter Värbrand. Optimization. Studentlitteratur, 1.1 edition, 2003. Cited on pages 15 and 16.

[9] Viet Ngo, Theo Hofman, Maarten Steinbuch, and Alex Serrarens. Optimal control of the gearshift command for hybrid electric vehicles. IEEE Transactions on Vehicular Technology, 61(8):3531–3543, 2012. Cited on pages 3, 23, 35, and 61.

[10] Simona Onori and Laura Tribioli. Adaptive pontryagin's minimum principle supervisory controller design for the plug-in hybrid gm chevrolet volt. Applied Energy, 147:224–234, 2015. Cited on page 14.

[11] Gino Paganelli, Makoto Tateno, Avra Brahma, Giorgio Rizzoni, and Yann Guezennec. Control development for a hybrid-electric sport-utility vehicle: strategy, implementation and field test results. In American Control Conference, 2001. Proceedings of the 2001, volume 6, pages 5064–5069. IEEE, 2001. Cited on page 16.

[12] Aishwarya Panday and Hari Om Bansal. A review of optimal energy management strategies for hybrid electric vehicle. International Journal of Vehicular Technology, 2014, 2014. Cited on pages 2 and 3.

[13] Martin Sivertsson. Optimal Control of Electrified Powertrains. PhD thesis, Linköping University Electronic Press, 2015. Cited on pages 16, 18, and 34.

[14] Martin Sivertsson and Lars Eriksson. Design and evaluation of energy management using map-based ecms for the phev benchmark. Oil & Gas Science and Technology–Revue d'IFP Energies nouvelles, 70(1):195–211, 2015. Cited on page 71.

[15] Conny Tempelhahn, Stephan Uebel, and Bernard Bäker. 5b. 4 state of the art in optimal control of series hybrid electric vehicles taking account of driveability. In Conference: Electric & Electronic Systems in Hybrid and Electrical Vehicles and Electrical Energy Management, volume ISBN: 978-3-8169-3346-5, 2016. Cited on pages 3, 23, 35, and 62.

[16] Thijs van Keulen, Jan Gillot, Bram de Jager, and Maarten Steinbuch. Solution for state constrained optimal control problems applied to power split control for hybrid vehicles. Automatica, 50(1):187–192, 2014. Cited on page 35.

[17] Ximing Wang, Hongwen He, Fengchun Sun, and Jieli Zhang. Application study on the dynamic programming algorithm for energy management of plug-in hybrid electric vehicles. Energies, 8(4):3225–3244, 2015. Cited on page 16.

[18] S.G. Wirasingha and A. Emadi. Classification and review of control strategies for plug-in hybrid electric vehicles. IEEE Transactions on Vehicular Technology, 60(1):111–122, 2011. Cited on pages 2 and 3.

[19] Pei Zhang, Fuwu Yan, and Changqing Du. A comprehensive analysis of energy management strategies for hybrid electric vehicles based on bibliometrics. Renewable and Sustainable Energy Reviews, 48:88–104, 2015. Cited on page 2.