

Institutionen för systemteknik

Department of Electrical Engineering

Examensarbete

Verification of hybrid operation points

Examensarbete utfört i Fordonssystem
vid Tekniska högskolan i Linköping
av

Otto Dunbäck och Simon Gidlöf

LITH-ISY-EX--09/4226--SE

Linköping 2009



Linköpings universitet
TEKNISKA HÖGSKOLAN

Verification of hybrid operation points

Examensarbete utfört i Fordonssystem
vid Tekniska högskolan i Linköping
av

Otto Dunbäck och Simon Gidlöf

LITH-ISY-EX--09/4226--SE

Handledare: **Erik Hellström**
ISY, Linköpings universitet
Johan Dufberg
General Motors Powertrain Sweden AB
Magnus Källvik
General Motors Powertrain Sweden AB

Examinator: **Jan Åslund**
ISY, Linköpings universitet

Linköping, 29 February, 2009

	Avdelning, Institution Division, Department Division of Automatic Control Department of Electrical Engineering Linköpings universitet SE-581 83 Linköping, Sweden		Datum Date 2009-02-29
	Språk Language <input type="checkbox"/> Svenska/Swedish <input checked="" type="checkbox"/> Engelska/English <input type="checkbox"/> _____	Rapporttyp Report category <input type="checkbox"/> Licentiatavhandling <input checked="" type="checkbox"/> Examensarbete <input type="checkbox"/> C-uppsats <input type="checkbox"/> D-uppsats <input type="checkbox"/> Övrig rapport <input type="checkbox"/> _____	ISBN _____ ISRN LITH-ISY-EX--09/4226--SE Serietitel och serienummer ISSN Title of series, numbering _____
URL för elektronisk version http://www.control.isy.liu.se http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-ZZZZ			
Titel Title Verification of hybrid operation points Författare Otto Dunbäck och Simon Gidlöf Author			
Sammanfattning Abstract <p>This thesis is an approach to improve a two-mode hybrid electric vehicle, which is currently under development by GM, with respect to fuel consumption. The study is not only restricted to the specific two-mode HEV but also presents results regarding parallel as well as serial HEV's.</p> <p>GM wishes to verify if the online-based controller in the prototype vehicle utilizes the most of the HEV ability and if there is more potential to lower the fuel consumption. The purpose is that the results and conclusions from this work are to be implemented in the controller to further improve the vehicle's performance.</p> <p>To analyze the behavior of the two-mode HEV and to see where improvements can be made, models of its driveline and components are developed with a focus on losses and efficiency. The models are implemented in MATLAB together with an optimization algorithm based on Dynamic Programming. The models are validated against data retrieved from the prototype vehicle and various cases with different inputs is set up and optimized over the NEDC cycle. Compensation for cold starts and NOx emissions are also implemented in the final model.</p> <p>Deliberate simplifications are made regarding the modeling of the power split's functionality due to the limited amount of time available for this thesis.</p> <p>The optimizations show that there is potential to lower the fuel consumption for the two-mode HEV. The results are further analyzed and the behavior of the engine, motors/generators and battery are compared with recorded data from a prototype vehicle and summarized to a list of suggestions to improve fuel economy.</p>			
Nyckelord Keywords one-mode, two-mode, deterministic dynamic programming, optimal fuel consumption			

Abstract

This thesis is an approach to improve a two-mode hybrid electric vehicle, which is currently under development by GM, with respect to fuel consumption. The study is not only restricted to the specific two-mode HEV but also presents results regarding parallel as well as serial HEV's.

GM wishes to verify if the online-based controller in the prototype vehicle utilizes the most of the HEV ability and if there is more potential to lower the fuel consumption. The purpose is that the results and conclusions from this work are to be implemented in the controller to further improve the vehicle's performance.

To analyze the behavior of the two-mode HEV and to see where improvements can be made, models of its driveline and components are developed with a focus on losses and efficiency. The models are implemented in MATLAB together with an optimization algorithm based on Dynamic Programming. The models are validated against data retrieved from the prototype vehicle and various cases with different inputs is set up and optimized over the NEDC cycle. Compensation for cold starts and NOx emissions are also implemented in the final model.

Deliberate simplifications are made regarding the modeling of the power split's functionality due to the limited amount of time available for this thesis.

The optimizations show that there is potential to lower the fuel consumption for the two-mode HEV. The results are further analyzed and the behavior of the engine, motors/generators and battery are compared with recorded data from a prototype vehicle and summarized to a list of suggestions to improve fuel economy.

Sammanfattning

Detta arbete är en ansats att förbättra en two-mode HEV med avseende på dess bränsleförbrukning. HEV:en är för närvarande under utveckling av GM. I arbetet presenteras även generella resultat för parallell- och seriellhybrider.

GM önskar verifiera om den online-baserade kontrollenheten i prototypfordonet nyttjar dess egenskaper till fullo och om det finns potential att minska dess förbrukning. Syftet är att resultaten och slutsatserna från detta arbete skall implementeras i kontrollenheten för att ytterligare förbättra fordonets prestanda.

För att analysera beteendet hos two-mode HEV:en och för att ta reda på var förbättringar skall adresseras är modeller av drivlinan och ingående komponenter utvecklade med fokus på förluster och effektivitet. Modellerna är implementerade i MATLAB tillsammans med en optimeringsalgoritm kallad Dynamisk Programmering. Modellerna är validerade mot data erhållen från prototypen och ett antal fall med olika indata har ställts upp och optimerats över NEDC-cykeln. Kompensering för kallstartar och NO_x-emissioner är också implementerade i den slutgiltiga modellen.

Medvetna förenklingar gällande modelleringen av power-splittens funktionalitet är gjorda med anledning av arbetets begränsade tidsram.

Optimeringarna visar att finns potential att minska förbrukningen för two-mode HEV:en. Resultaten är analyserade och beteendet hos förbränningsmotor, motor/generator samt batteri är jämförda med data från prototypen vilket resulterat i en lista med förslag för att reducera förbrukningen.

Acknowledgments

First, we would like to thank our supervisors Magnus Källvik and Johan Dufberg at GM Powertrain in Trollhättan for all their valuable help and ideas - without them this thesis would not have been possible to carry through. We would also like to show our appreciation to Lars Johansson and Leif Hermansson for their guidance, encouragement and feedback during this work. Our supervisors at ISY at Linköping University, Erik Hellström and Jan Åslund also deserve a big thank for their support. In addition, we would like to thank all engineers at GM Powertrain and everyone else who have contributed to this work.

Örebro, May 2009

Otto Dunbäck

Simon Gidlöf

Contents

1	Introduction	5
1.1	Company description	5
1.2	Background	5
1.3	Thesis purpose and goal	6
1.4	Problem framing	6
1.5	Method	7
1.6	Limitations	7
1.7	Outline of the thesis	8
2	Hybrid electric vehicles	9
2.1	Historic overview	10
2.2	Architecture	11
2.2.1	Series HEV	11
2.2.2	Parallel HEV	12
2.2.3	Combined HEV	12
2.2.4	Complex HEV	13
2.3	Classifications	14
2.3.1	Micro	15
2.3.2	Mild	15
2.3.3	Full	15
2.4	Power split hybrid powertrain	15
2.4.1	Input power split	16
2.4.2	Compound power split	17
2.4.3	Combined power split	18
3	Dynamic programming	19
3.1	Theory and mathematical problem formulation	19
3.2	Implementation	22
4	Modeling	23
4.1	Vehicle	23
4.1.1	Vehicle validation	24
4.2	Battery	25
4.2.1	Battery validation	26
4.3	Motor and generator	28

4.4	PSD	28
4.4.1	PSD validation	32
4.4.2	Losses	34
4.4.3	Gear oil pump	35
4.5	Internal combustion engine	35
4.5.1	ICE efficiency	36
4.5.2	Engine inertia	36
4.5.3	NOx limitations	37
4.5.4	ICE validation	37
4.6	Cold start compensation	38
4.7	Auxiliary load	38
4.8	Parallel HEV	39
4.9	Serial HEV	39
4.10	Two-mode HEV	40
4.10.1	Two-mode HEV validation	42
5	Case studies	43
5.1	Parallel HEV	43
5.1.1	Input	43
5.1.2	Results	44
5.2	Serial HEV	47
5.2.1	Case 1	47
5.2.2	Case 2	51
5.3	Two-mode HEV	54
5.3.1	Input	54
5.3.2	Results	55
5.3.3	Engine	55
5.3.4	Battery	60
5.3.5	Motors/generators	62
5.4	Comparison	65
5.4.1	Engine	66
5.4.2	Battery	70
5.4.3	Motor/generator	70
5.4.4	Fuel consumption	72
5.4.5	Summary	73
6	Speed improvements	75
6.1	Vectorization	75
6.2	Interpolation	77
6.2.1	Adjusting the fuel map	77
6.2.2	Implementing C-code	78
6.2.3	Polynomial function	78
6.3	Improving the MATLAB code	78
6.4	Summary	79

7	Conclusions and future work	81
7.1	Summary of results	81
7.1.1	Suggested modifications	81
7.1.2	Additional results	82
7.2	Future Works	82
	Bibliography	83
A	Abbreviations	85
B	MATLAB code parallel	86
C	MATLAB code serial	89
D	Driving cycle	92

List of Tables

4.1	Measured and calculated fuel consumption.	38
5.1	Inputs for the parallel.	43
5.2	Inputs for the Serial HEV during case 1.	47
5.3	Inputs for the Serial HEV during case 2.	51
5.4	Inputs for the two-mode HEV during case 1.	54
5.5	Inputs for the two-mode HEV during case 2.	55
5.6	Optimized fuel consumption for the two-mode HEV.	65
5.7	Measured and optimized fuel consumption for the two-mode HEV.	72
6.1	Running time for a scalar- contra a vectorized system.	77

List of Figures

2.1	Possible power flow paths for the serial HEV.	11
2.2	Possible power flow paths for the parallel HEV.	12
2.3	Possible power flow paths for the combined HEV.	13
2.4	Possible power flow paths for the complex HEV.	14
2.5	HEV's classified according to their degree of hybridization.	14
2.6	Power split device. Reference: http://www.carbibles.com/transmission_bible.html	16
2.7	Possible power flow paths for the input power split.	17
2.8	Possible power flow paths for the compound power split.	17
3.1	Bellmans principle of optimality illustrated, if a trajectory is the optimal policy from x_0 to x_N , then the subpath from x_i to x_{i+1} , and all other sub paths's are optimal.	21
4.1	Vehicle validation, comparison of measured and calculated data.	24
4.2	Estimated- and calculated battery power.	26
4.3	Estimated- and calculated battery loss.	27
4.4	Effective battery power for the calculated and the estimated case.	27
4.5	Stick-lever diagram for the clutch configuration in mode 1	29
4.6	Speed validation of m/g_A and m/g_B for the two-mode HEV.	32
4.7	Validation of m/g_A torque for the two-mode HEV.	33
4.8	Validation of m/g_B torque for the two-mode HEV.	33
4.9	Pump losses for a 2-mode during NEDC.	35
4.10	Specific fuel consumption.	36
4.11	Specific fuel consumption with the NOx limitation curve.	37
4.12	Validation of engine torque for the two-mode HEV.	42
5.1	SoC level for the parallel HEV.	44
5.2	Engine Torque during UDC for the parallel HEV.	45
5.3	Engine Torque during EUDC for the parallel HEV.	45
5.4	Engine, vehicle and motor/generator power for the parallel HEV during UDC.	46
5.5	Engine, vehicle and motor/generator power for the parallel HEV during EUDC.	46
5.6	SoC for the serial HEV during case 1.	48
5.7	Battery and vehicle power for the serial HEV during case 1.	49
5.8	Engine and vehicle power for the serial HEV during case 1.	50
5.9	Engine torque for the serial HEV during case 1.	50
5.10	Engine speed for the serial HEV during case 1.	51
5.11	SoC for the serial HEV during case 2.	52
5.12	Engine speed for the serial HEV during case 2.	53
5.13	Engine torque for the serial HEV during case 2.	53
5.14	Engine and vehicle power for power-split HEV during the UDC for case 1 & 2.	56
5.15	Engine and vehicle power for power-split HEV during the EUDC for case 1 & 2.	56

5.16	Engine torque for power-split HEV during UDC for case 1 & 2. . .	57
5.17	Engine torque for power-split HEV during EUDC for case 1 & 2. .	57
5.18	Engine speed for power-split HEV during UDC for case 1 & 2. . .	58
5.19	Engine speed for power-split HEV during EUDC for case 1 & 2. . .	58
5.20	Specific fuel consumption with operation points for Case 1.	59
5.21	Specific fuel consumption with operation points for Case 2.	60
5.22	SoC for Power-split HEV during case 1 & 2.	60
5.23	Battery and vehicle power for power-split HEV during UDC for case 1 & 2.	62
5.24	Battery and vehicle power for power-split HEV during EUDC for case 1 & 2.	62
5.25	Power for motor/generator _A and vehicle during UDC for case 1 & 2.	63
5.26	Power for motor/generator _A and vehicle during EUDC for case 1 & 2.	63
5.27	Power for motor/generator _B and vehicle during UDC for case 1 & 2.	64
5.28	Power for motor/generator _B and vehicle during EUDC for case 1 & 2.	64
5.29	Vehicle speed before processing the data set.	66
5.30	Vehicle speed after processing the data set.	66
5.31	Measured engine power and vehicle power for the two-mode.	67
5.32	Measured engine torque for the two-mode.	68
5.33	Measured engine speed for the two-mode.	68
5.34	Specific fuel consumption with operation points for the prototype.	69
5.35	Measured SoC for the two-mode.	70
5.36	Vehicle power and measured power from motor/generator _A	71
5.37	Vehicle power and measured power from motor/generator _B	72
6.1	Analysis of speed performance with MATLAB's Profiler.	76
6.2	Torque matrix. Relevant elements within the torque limit are marked with green.	79
D.1	Speed- and gear profile for the NEDC.	92

Chapter 1

Introduction

1.1 Company description

General Motors is one of the world's largest automaker with manufacturing in 34 countries, employing about 252,000 people around the world. Nearly 8.4 million GM cars and trucks were sold in 2008 under brands such as Cadillac, Chevrolet, Daewoo, Opel and Saab. GM Powertrain, the division responsible for engines, transmissions, castings and components for both General Motors and other OEM manufacturers' vehicles, has manufacturing plants and engineering centers in North and South America, the Asia-Pacific region and Europe, and there among others sites in Trollhättan in Sweden. Global headquarters though, is located in Pontiac Michigan in United States.

1.2 Background

The recent years of escalating oil prices and a growing global awareness among the public for environmental issues have increased the demand of less pollutant and more fuel efficient transports. The automotive industry has probably felt this quite sudden change in the consumer's behavior more than other industries. People are demanding smaller cars with lower fuel consumption or cars fueled by alternative energy sources. Even markets such as the US, where of tradition big trucks with large fuel guzzling engines have been the norm, are now demanding more fuel efficient cars. Governments around the world are also redefining the rules and laws concerning emissions for cars. For instance has the EU commission suggested a law that, if put into practice, will be costly for automakers whose cars on an average exceeds 130 g CO₂ per kilometer. Hybrid electric vehicles (HEV from now on) have experienced an increasing interest ever since the world's first mass produced hybrid vehicle was released in Japan. It has showed that it possible to reduce a vehicle's fuel consumption and thus its emission of CO₂ without reducing its drivability. Several automakers now have either a HEV in their model program or one on its way to be launched.

The individual component in a HEV, if it might be the engine, the generator or the electric motor is not extremely complicated in itself, but when put together to function as a unit the complexity increases. Not only must the actuators be controlled in such way that the fuel consumption is kept at a minimum in all driving situations, but also emissions must be kept at an acceptable level and, not to mention, should the driver of the vehicle not notice when all this happens. The vehicle should behave to what the driver would describe as a familiar way, which means accelerate when the accelerator is pressed down as well as decelerate when the brake is applied, and all this without delays, yanks or jerks. With this background given, combined with the insight of the large costs involved in the development of HEVs, it is clear that maximizing the performance of the HEV, i.e. minimize the fuel consumption for a given HEV driveline, is of most importance. An optimal solution can function as a yardstick to see what improvements, at least in theory, can be made for this specific HEV and where focus to achieve these improvements should be placed.

1.3 Thesis purpose and goal

The goal for this thesis is to analyze the differences in fuel consumption and behavior of the actuators for a two-mode HEV. The comparison is to be made between data measured from a prototype in a test cell and the theoretical, optimal fuel consumption obtained from optimizations performed on a model of the same vehicle for a given driving cycle. The purpose is that result from this analysis can help to improve the current controller of the HEV and thereby obtain a better fuel economy.

1.4 Problem framing

The controller used in the HEV prototype is online-based, i.e. it calculates the optimal instantaneous (according the known data; the speed of the vehicle, the power demanded by the driver etc. within given constraints; emissions, peak power of electric motor(EM) etc.) power flow through the transmission (i.e. planetary gears, EM) at every instant. However, the system does not look ahead and does not, for instance, know the amount of torque required 5 seconds ahead. This information is of course impossible to get hold on since the system can never know how the driver will react in the future.

Given that the controller optimizes the powertrain at every instant, based on the data available at that instant, seen over a whole driving cycle the operating points and gears selected might not be (and probably is not) the globally optimal. By studying how the controller chooses to supply the power demanded (when is the ICE turned on, how much torque is supplied by EM's etc.) and compare it to the optimal power distribution obtained from the optimization it is possible to tweak the controller parameters to get a lower fuel consumption. The problem can be separated into the following parts.

- Build/put together a MATLAB model of the two-mode powertrain. The complexity of the model should be well-matched for this work, i.e. losses in the system and the efficiency of the different parts are of most importance. The model needs to be validated.
- Find and adapt a suitable optimization algorithm. Since the optimization problem will contain many variables it is important to find an algorithm that can handle the complexity and find a global minimum in a reasonable amount of time.
- Perform simulations and optimize the fuel consumption for the model for a given driving cycle.
- Analyze the results and compare them with the measurements for the prototype. This is the goal with this thesis and also where the main effort should be put. The previous steps must be done thoroughly in order to achieve an equitable analysis.

1.5 Method

This project has been carried out as follows

- A literature study, which had its main focus on optimization algorithms, was done. This study also served a purpose of gather knowledge of HEV's in general and the two-mode HEV in specific.
- Models of a parallel and a serial HEV were developed and implemented in MATLAB using the chosen algorithm. Optimizations were conducted and the results analyzed to gain credibility for further work.
- Models of the two-mode HEV's components were developed using known equations and relationships from the actual controller.
- The model was validated against real data and expanded and refined to better match the actual vehicle.
- Finally, several optimizations with different inputs were performed. The results were analyzed and compared with data obtained from a test cell. Conclusions from the work was drawn.

1.6 Limitations

- The optimization was done with regard to fuel efficiency. During tests with the 2-mode, restrictions for NO_x-emissions was also included, however wear etc. was not taken into consideration.
- The car model and the optimization algorithm should be implemented using software available at GM on a standard PC.

- Due to the complexity and the limited time available for this thesis, there was no optimization done regarding the switch between mode-1 and mode-2, instead a predefined switching-time was used. Also no consideration was taken to gear change in the second mode.

1.7 Outline of the thesis

This thesis is structured as follows

Chapter 2 gives a general introduction about hybrid electric vehicles, furthermore it gives a brief introduction to different existing architectures, and an explanation about HEV classifications. Chapter 2 also includes a description about how the planetary gear works and a presentation on different sorts of configurations.

Chapter 3 presents the basic theory behind dynamic programming, and how theory has been implemented in this thesis.

Chapter 4 presents how different parts of the hybrid electric vehicle were implemented, such as the battery, motor/generator, internal combustion engine and the power split device. Implementation of the whole powertrain for the parallel, serial and two-mode is also included in this chapter, complemented with validation.

Chapter 5 presents inputs that have been used during simulations, whereupon results from simulations are studied and compared for the different HEV configurations.

Chapter 6 presents different approaches on how to improve optimization time - both those which were actually implemented and how well they performed as well as a brief discussion regarding further improvements which can be made in future work.

The closing chapter, Chapter 7 presents the conclusions of this thesis, and recommendations for future work, this is followed by bibliography. Appendix A includes abbreviations, while Appendix B, C and D contain source code for parallel, serial and the two-mode HEV. Finally Appendix E presents information about The New European driving cycle, or NEDC.

Chapter 2

Hybrid electric vehicles

Hybrid vehicles are characterized by two or more prime movers and power sources, but only one energy source (except for plug-in hybrids). A hybrid electric vehicle includes an engine as fuel converter or irreversible prime mover. Different types of motors are used as electric prime movers, e.g. standard DC, brushless DC, induction AC etc. Many configurations include a secondary electric energy converter, primarily used as a generator. Both batteries and supercapacitors can be used as electric energy storage, whereas the latter is more likely to find in prototypes. With respect to normal, ICE-based vehicles, HEV's benefit of several possibilities for improving fuel economy, for example:

- Reduce engine size and still fulfill the power requirements of the vehicle
- Recuperate energy during deceleration instead of dissipate it as heat when braking
- Optimize the energy distribution between the prime movers
- Turn off the engine during standstill and thus eliminate idle fuel consumption
- Eliminate clutch losses by engaging the engine only when the speeds match
- Optimize engine load point

The possibilities above are typically not used simultaneously and their potential is system dependent [7]. The architecture of HEV's can loosely be defined as the connection between components and the energy flow between them. Until 2000 were the architecture divided into two categories; parallel and series, but the introduction of new HEV's created two new categories: series-parallel or combined and complex. The four categories are described in detail in Section 2.2 [6].

2.1 Historic overview

In the contrary to what many people might think, hybrid vehicles are not a new phenomenon. In 1898, just over 10 years after Karl Benz invented what is acknowledged as the first modern automobile [1], Justus B. Entz, chief engineer of the Electric Storage Battery Company of Philadelphia tested his creation - a electric and gasoline-powered automobile, almost surely the world's first. Unfortunately his creation caught fire during its maiden voyage and no more were built [9].

In the early years of automobiles electric vehicles, steam cars and internal combustion powered automobiles were highly competitive. The reason for combining electric and internal combustion propulsion was to increase the poor range battery powered automobiles provided. The beginning of the 20th century was not only the beginning for the modern automotive industry but also a period where many experiments of concepts and layouts of hybrid vehicles were conducted. The Lohner-Porsche, Auto-Mixte, Mercedes-Mixte and the Krieger are examples of petro-electric cars that were built but never became popular.

Baker and Woods, two battery-only car manufacturers developed in 1917, independently, petro-electric automobiles. Woods', called "Woods' Dual" had a power plant consisting of a 14-hp, four-cylinder engine mounted in the front with a motor-generator placed right behind it. A magnetic clutch was mounted in between and a 24-cell, 48-volt battery was carried in the back of the frame which was only half of the usual number of cells thanks to the combustion engine. The arrangement of the units made it possible to drive the car solely by the gasoline engine, entirely by electricity or both simultaneously. The vehicle could reach a maximum speed of 56 km/h and the battery alone had the capacity to propel the vehicle 48 km. Another feature which is also seen in today's hybrids was the ability to use the electric machine as a generator, thereby recuperate energy when descending a hill and store electricity in the battery until needed. After experimenting, testing and operation both Baker and Woods abandoned their dual-drive systems after drawing the conclusions that the hybrid electric approach added complexity, increased maintenance, introduced gasoline and oil, multiplied the weight and amplified the cost. In 1917, Woods' car sold for \$2.950, which was much more than an equivalent gasoline car [9].

The literature does not mention further hybrid concepts until the 1960's - 1970's, when engineers and innovator once again wanted to extend the range of battery propelled vehicles by adding an additional power source. Many interesting concepts were shown in the 70's and the following years. To mention one, the McKee Engineering Company's concept, the Range Extender, consisted basically of an electric powered car. When driving in urban areas the car was driven solely on electricity and could be recharged from a standard 230 volt outlet. When further range was needed, a mobile gasoline power plant housed in a trailer was connected [9].

Until 1997, when Toyota's Prius went on sale in Japan, all hybrid cars manufactured so far were either built as prototypes or built in small series. The first generation of the Prius became the world's first mass-produced hybrid vehicle and

Toyota has now, during Prius' third generation, sold over 1 million vehicles [2].

2.2 Architecture

The following sections describe the four most common hybrid architectures in both text and illustrations. In the belonging figures are mechanical power denoted as a solid arrow while electrical power is represented by a dotted arrow. The arrowheads show the possible directions of the power flow.

2.2.1 Series HEV

The architecture of the series HEV is easiest described as electric vehicle combined with an auxiliary power source acting as a range extender. A generator converts mechanical energy into electric which can either be fed directly to the motor or stored in the battery, see Figure 2.1. The traction motor acts as a generator during deceleration thus reducing speed while charging the battery. Since the engine is decoupled from the drive shaft its power output is not directly related to the current power requirement and hence can the engine operation points be chosen freely to optimize fuel consumption and emissions. Another benefit is the absence of clutch which eliminates friction losses. A series hybrid needs three machines: one engine, one motor and one generator, where at least the motor's maximum power output has to match the vehicle's requirement, thus increasing the weight of the vehicle compared to a standard ICE vehicle. Furthermore is the tank-to-wheel efficiency relatively low because of the two-step energy conversion (mechanical to electrical in the generator and electrical to mechanical in the motor) [7].

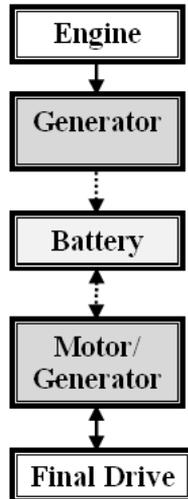


Figure 2.1. Possible power flow paths for the serial HEV.

2.2.2 Parallel HEV

While a series hybrid can be considered as an electric vehicle with an additional ICE-based energy path, a parallel HEV is rather a conventional ICE-powered vehicle assisted by an electric motor coupled to the transmission, see Figure 2.2. This configuration enables the vehicle to be driven solely by the engine, the motor only or the two together, which gives an additional degree of freedom to fulfill the power requirement of the vehicle. Typically, the engine can be turned off at idle and the electric motor can assist at high-power demands, i.e. acceleration and high-speed cruising. This gives the advantage that the motor and engine can be designed for only a fraction of the required maximum power, which in its turn requires smaller, lighter machines. This together with the fact that only two machines are needed is a benefit compared to a series hybrid. A disadvantage is the need for a clutch since the engine is mechanically coupled to the drive shaft. All in all, the system efficiency of the parallel hybrid is in principle higher in comparison to the ICE-based vehicle [7].

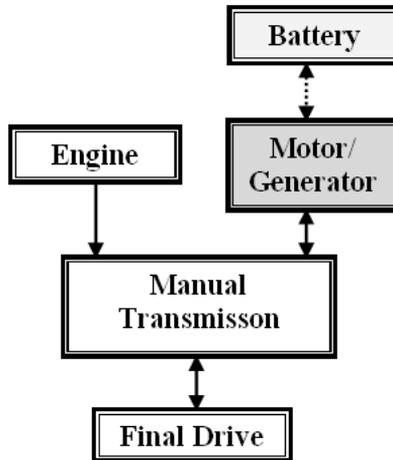


Figure 2.2. Possible power flow paths for the parallel HEV.

2.2.3 Combined HEV

As the title reveals can a combined HEV be described as a combination between a series and parallel hybrid but has more in common with the latter one, see Figure 2.3 . A combined HEV features both mechanical and electrical link together with two electrical machines, one acting as a motor for traction and for generative braking while the other as generator for charging the battery and for stop-and-start function. The most common way to achieve a combined HEV is to link the engine and motors with a planetary gear set (PGS), but other means to achieve

this function has been demonstrated. The Toyota Prius, mentioned earlier, is an example of a combined hybrid (in this case a "one-mode"), as well as the hybrid analyzed in this thesis (which is a "two-mode") [7].

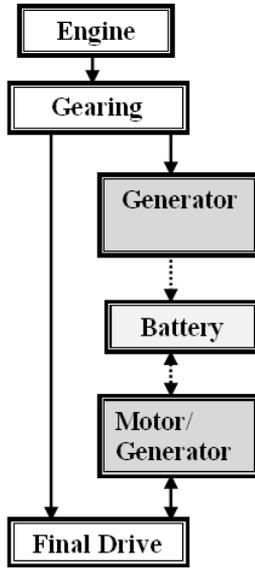


Figure 2.3. Possible power flow paths for the combined HEV.

2.2.4 Complex HEV

As the name reveals, this configuration is more complex than the other three stated above. Figure 2.4 shows that the complex configuration shares similarities with combined hybrids. However, a complex hybrid is equipped with an additional motor/generator. Typically, this motor is acting on the rear axle while the engine and the first motor/generator provide power to the front axle. In other words enables this configuration three propulsion devices to simultaneously propel the vehicle. During deceleration both electric machines act as generators charging the battery. An extra feature this system provides is an advanced four-wheel drive system. In case the front wheels slip, the front electric machine works as a generator to absorb the change of engine output power. This power difference is used, trough the battery, to drive the rear wheels to achieve axle balancing [5].

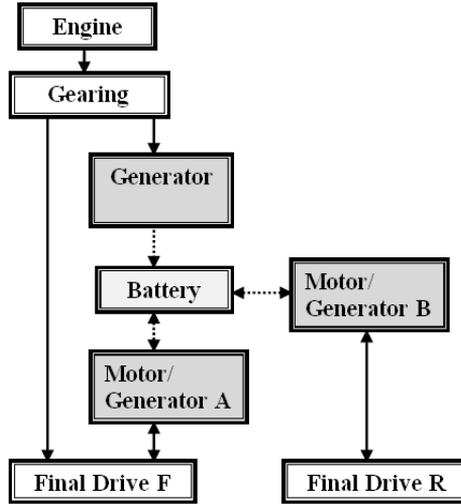


Figure 2.4. Possible power flow paths for the complex HEV.

2.3 Classifications

The previous section explains the four different architectures of hybrids, how they are principally built and how the energy is distributed. Another way of describing hybrids without focusing on their fundamental design is to classify HEV’s according their degree of hybridization. This classification is divided into micro, mild and full hybrids. Figure 2.5 shows how increased functionality affects fuel economy and costs.

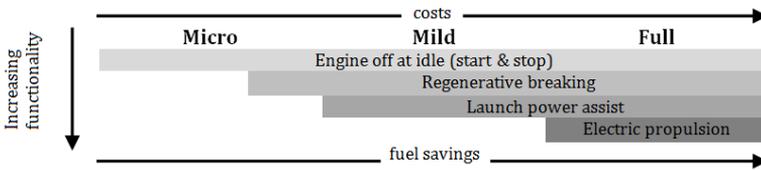


Figure 2.5. HEV’s classified according to their degree of hybridization.

2.3.1 Micro

The lowest degree of hybridization, called micro hybrid, covers standard ICE based vehicles equipped with a small electric motor which enables start-and-stop functionality. The motor is typically connected to the crankshaft via a belt or directly connected to the crankshaft and does not require any high battery power or complex power electronics since the system voltage is below 42 volt. A micro hybrid configuration does not generally involve generative braking capability but it is possible to achieve to some extent. The total electrical power for a micro hybrid is limited to around 5 kW.

2.3.2 Mild

A mild hybrid shares almost the same characteristics as the micro hybrid. However, the electric motor is larger as well as the overall electric system voltage and power, which has a span between 5-20 kW. Generative braking is fully implemented and the motor provides enough torque to assist the ICE during acceleration.

2.3.3 Full

In a full hybrid all features associated with a hybrid is implemented, i.e. start-and-stop function, regenerative braking and boost function. The system voltage and maximum electric power output is higher, from 20 kW and upwards, than for the mild hybrid which enables the vehicle to be driven solely on electricity - if just for short distances and moderate speeds. The vehicle addressed in this thesis is classified as a full hybrid.

2.4 Power split hybrid powertrain

The power split hybrid uses a power split device, to split power into two paths: All-mechanical, which has high efficiency (more than 90 %) and electro-mechanical, which has low efficiency (around 75 %).

Power split devices, or PSD:s are often found in combined and complex hybrid vehicles, combining mechanical power from various power sources to various mechanical loads. Typically, a PSD consists of a planetary gear set (also referred to as an epicyclic gearing) which connects an engine, a motor, a generator and the drive train. The power split device in this thesis consists of two combined planetary gear sets connected to an engine and two electric motors/generators, which from now on will be indexed as motor/generator_A, or m/g_A respectively motor/generator_B, or m/g_B. This configuration enables so called "two-mode" operation, which will be discussed later.

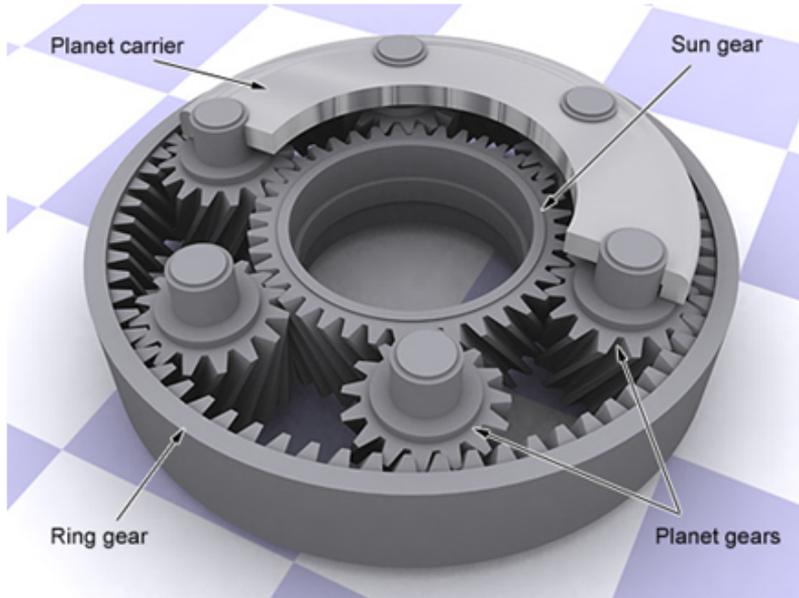


Figure 2.6. Power split device. Reference:http://www.carbibles.com/transmission_bible.html

A basic planetary gear set has three main rotating parts, which can be seen in Figure 2.6. The most outer part is the ring, the inner is the sun and the intermediate part is the carrier, which carries rotating gears called planets. Each of the three parts can be connected to either input or output shaft or can be held stationary. More complex configurations, as the compound planetary gear set, exist and are widely used in automatic transmissions together with a hydraulic torque converter.

2.4.1 Input power split

An input power split splits the power at the input, so that one electric motor/generator is geared to the engine while the other turns with the output, see Figure 2.7. This configuration gives zero electric power at zero speed and one mechanical point. Toyota Prius uses this configuration for the whole driving range, while this is for low power, low speed (first mode) for the two-mode HEV analyzed in this thesis.

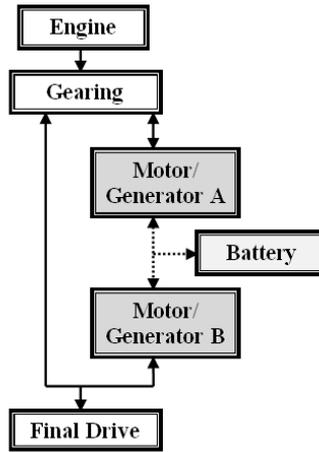


Figure 2.7. Possible power flow paths for the input power split.

2.4.2 Compound power split

Compared to the input power split, compound split has two mechanical points (zero electric power). Both motors are geared, one at the input and the other at the output, as seen in Figure 2.8. GM uses this split configuration for high speeds (second mode).

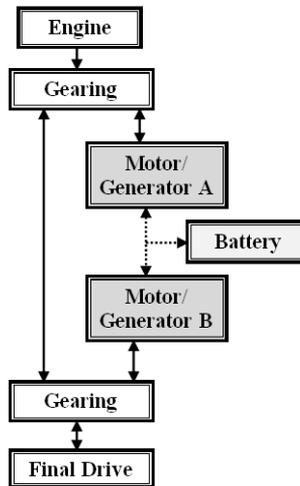


Figure 2.8. Possible power flow paths for the compound power split.

2.4.3 Combined power split

Combined power split combines input and compound power split to achieve a two-mode EVT with three mechanical points. This configuration is more expensive than the input power split but the benefit of keeping the electric power low (remember, the electro-mechanical path has lower efficiency than the all-mechanical) results in better fuel economy for a wider driving range. The shift between the modes and gears is achieved by four clutches.

Chapter 3

Dynamic programming

Dynamic programming or DP was developed in the late 50's by R.E. Bellman and is a mathematical method based on Bellman's principle of optimality, and was created for mathematical problems arising from studies of various multi-stage decision processes [3]. The theory has been successfully applied to a wide area of disciplines such as economics, artificial intelligence and control system.

The most important advantage of using Dynamic programming is that an optimal trajectory or path for a certain problem is always guaranteed to be found. However, there are some disadvantages with using DP, a major disadvantage is that the computational time grows exponentially with the number of states and control inputs, also called the curse of dimensionality [4], the consequence is when an extra state are implemented, or when the size of the grid is being enlarged or refined, the simulation time grows explosively fast. Another drawback is that a high memory storage capacity is needed.

In this chapter the theory of dynamic programming and the principle of optimality, the fundamental core in DP, are described. A section describing the implementation is also included.

3.1 Theory and mathematical problem formulation

The dynamic programming technique is suited for problem involving multi-stage decisions, and can therefore be used to compute the optimal control actions during an in advanced known driving cycle. In this scenario the optimal control actions corresponds to the optimal power split between combustion engine, motor/generator and the battery in the parallel case, in the serial and two-mode case the split between combustion engine, motor/generator_A, motor/generator_B and the battery.

The approach here is to start at the end of the driving cycle, and work backwards to the start, also known as backward dynamic programming. A model of the vehicle for a discrete-time system can then be expressed as

$$x_{k+1} = f(x_k, u_k, k) \quad k = 0, 1, \dots, N - 1, \quad (3.1)$$

where $x_k \in X_k \subset \mathfrak{R}^n$ is the system state vector, in this context it consists of the State of Charge (SoC) in the parallel case, in the serial and two-mode case it consists of SoC and the engine speed (Ne). $u_k \in U_k \subset \mathfrak{R}^m$ is the control inputs such as the output torque of the ICE, k is the present stage, in this case it represent a time instant.

A specific policy can be denoted as $\pi = \{\mu_0, \mu_1, \dots, \mu_{N-1}\}$ and the cost of using that specific policy on Equation (3.1) with the initial condition x_0 , in the two-mode case the initial condition is SoC_0 and Ne_0 , is defined by

$$J_\pi(x_0) = g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, \mu_k(x_k), k) \quad (3.2)$$

With the stated equation above (Equation (3.1)) the optimal path, denoted here as $\{\pi = \mu_0^0, \mu_1^0, \dots, \mu_{N-1}^0\}$, is the path that minimizes J_π in (3.2) and can be expressed as

$$J^0(x_0) = \min_{\pi \in \Pi} J_\pi(x_0) \quad (3.3)$$

The optimization problem stated above can be solved by the use of Bellmans principle of optimality, the principle states that:

Theorem 3.1 (Bellmans principle of optimality) *"An optimal policy has the property that whatever the initial decision are, the remaining decisions must constitute an optimal policy with the regard to the state resulting from the first decision."* [1]

In other words the theorem states that if the trajectory, or path is the optimal policy from x_0 to x_N , then the subpath from x_k to x_{k+1} , and all other sub trajectory's using the same policy will be optimal, this is illustrated in Figure 3.1.

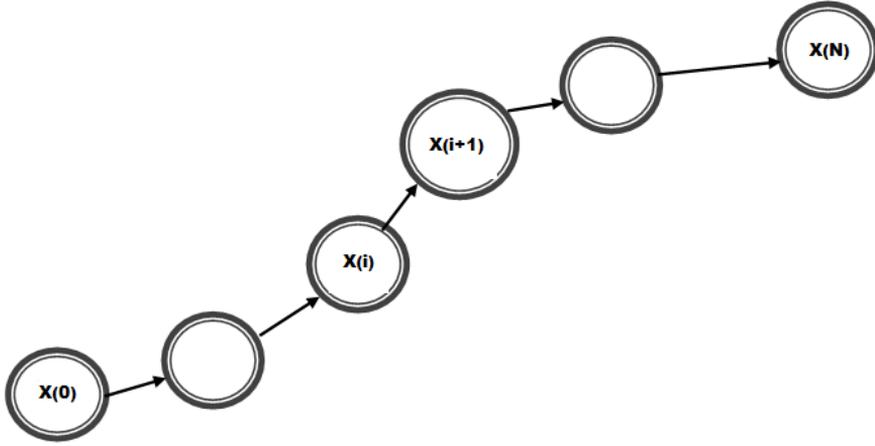


Figure 3.1. Bellmans principle of optimality illustrated, if a trajectory is the optimal policy from x_0 to x_N , then the subpath from x_i to x_{i+1} , and all other sub paths are optimal.

When applying the principle of optimality to (3.2)-(3.3), it gives that if $\{\pi^0 = \mu_0^0, \mu_1^0, \dots, \mu_{N-1}^0\}$ is a optimal policy, and when using this policy π^0 a given state x_k is reached at the time instant i , then the cost-to-go (J_π) from i to N will be defined as

$$J_\pi(x_i) = g_N(x_N) + \sum_{k=i}^{N-1} g_k(x_k, \mu_k(x_k), k), \tag{3.4}$$

where the policy $\{\pi^0(x_i) = \mu_i^0, \mu_{i+1}^0, \dots, \mu_{N-1}^0\}$ is optimal.

The previous stated optimization problem can now be calculated with the following algorithm, which proceeds backwards in time and is normally referred to as deterministic dynamic programming.

- 1.

$$J_N(x_N) = g_N(x_N) \tag{3.5}$$

- 2.

$$J_k(x_k) = \min_{u_k \in U_k(x_k)} \{g_k(x_k, u_k, k) + J_{k+1}(f(x_k, u_k, k))\}, \tag{3.6}$$

where Equation (3.5) is the cost calculation of the end step, (3.6) is the cost calculation of the intermediate step, and can be seen as the current cost (arc cost) plus the cost to go. The optimal solution is now the policy $\{\pi^0 = \mu_0^0, \mu_1^0, \dots, \mu_{N-1}^0\}$ that minimizes the right side of (3.6) for each x_k and k in $u_k^0 = \mu_k^0(x_k)$.

3.2 Implementation

The above described algorithm can be implemented in MATLAB using the following routine

- 1 Initialize (Final costs, $N_E(i)$, SoC(j) and time(k))
- 2 Outer loop over time(k)
 - 3 loop over SoC(j)
 - 4 loop over $N_E(i)$
 - 5 A point in the grid is reached $x(i,j,k)$, calculate the costs for all arcs from this point.
 - 6 Find the arc with the minimum sum of cost to go + + running cost (arc cost).
 - 7 Store that arc.
 - 8 Store the associated cost.
 - 10 Next $N_E(i)$
 - 11 Next SoC(j)
 - 12 Next time(k)

Chapter 4

Modeling

The vehicle model used in this thesis is based on general, widely used equations as well as more specific found in literature concerning hybrids. As for all modeling one has to compromise between model accuracy versus model complexity, where a more detailed model tend to lead to better accuracy but also tend to increase simulation and optimization time. An important aspect when creating the model is to keep in mind the expected results from the model. In this specific case, the goal is to optimize the power distribution in the powersplit (the most efficient path and combination) and hence losses and efficiencies are crucial to model as accuracy as feasible. Other aspects are not as important and therefore deliberate simplifications have been made. For instance does the model not consider certain dynamic changes, such as transients.

This chapter describes how different components are modeled and how the components are implemented and related to each other.

Constants, variables and look-up tables in this chapter are, if nothing else is said, provided by GM.

4.1 Vehicle

The driving cycle is specified so that the conditions represent driving on a straight line on a flat road, hence are not lateral forces or forces associated with driving up a hill taken into consideration. Three effects mainly build up the force which must be overcome by the vehicle's propulsion units when driving a predefined driving pattern:

- Aerodynamic friction
- Rolling friction
- Vehicle inertia

The first two are summed up in the polynomial equation, Equation (4.1) below, consisting of three constants which vary between different vehicle types, wheels etc.

$$F_{friction} = \begin{cases} a_1 + a_2 \cdot v(t) + a_3 \cdot v(t)^2 & , v > 0 \\ 0 & , v = 0 \end{cases} \quad (4.1)$$

The vehicle inertia is described by Newton's second law, which in this specific case yields

$$F_{mass} = m_{vehicle} \cdot \dot{v} \quad (4.2)$$

When multiplying Equation (4.1) and (4.2) with the vehicle speed one obtain the power required to propel the vehicle, according to

$$P_{vehicle} = (F_{friction} + F_{mass}) \cdot v \quad (4.3)$$

4.1.1 Vehicle validation

The vehicle validation has been carried out by comparing the vehicle power, calculated in the same way as described in the previous chapter, with data retrieved from a test run with the actual prototype. The two resulting $P_{vehicle}$ are not completely comparable for two reasons, but here is behavior and order of magnitude of most interest. Firstly is the data retrieved from the test run not the power acting on the vehicle but rather the power demanded by the driver, while the vehicle speed, which is the input in the calculation, is the actual speed. Secondly contains the data with the speed variable many transients, which leads to a very spiky behavior when derivated. Figure 4.1 shows the result of the validation. The dashed curve is the measured data and the solid is the calculated.

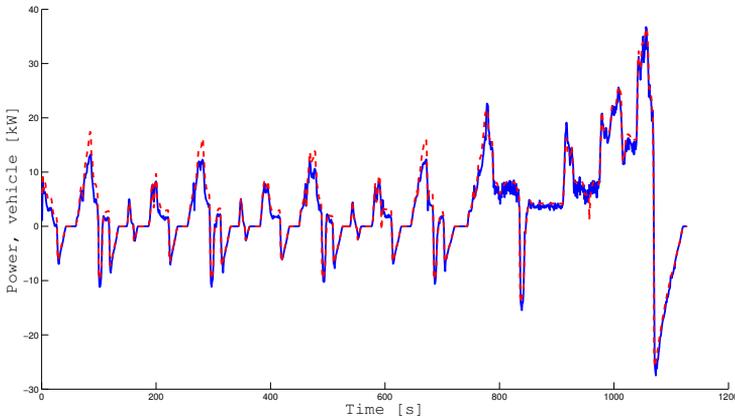


Figure 4.1. Vehicle validation, comparison of measured and calculated data.

Despite the somewhat poor prerequisites the result is satisfying. The measured data shows higher power peaks than the calculated during the low speed parts,

but shows on the other hand lower dips during the decelerations. During the high speed, EUDC part, are they very much alike.

4.2 Battery

The power flow from a battery can be expressed as the total power flow from a battery without considering its losses minus the actually losses.

$$P(t)_{batt} = P(t)_{batt,tot} - P(t)_{batt,loss} \quad (4.4)$$

The total battery power is then obtained from the trivial relationship between $U_{oc}(t)$ and $I(t)$ that yields

$$P(t)_{batt,tot} = U(t)_{oc} \cdot I(t), \quad (4.5)$$

where the $U(t)_{oc}$ is the open-circuit voltage representing the fully charged voltage obtained from a constant-current discharge test. Battery current $I(t)$ is obtained by multiplying the derived state of charge \dot{SoC} , with the nominal battery capacity, Q_0 .

$$I = \dot{SoC} \cdot Q_0 \quad (4.6)$$

With a determined $P_{batt,tot(t)}$, obtained by inserting Equation (4.6) into (4.5) and a look-up table with estimated values for the battery losses for different $P_{batt,tot(t)}$ values, a resulting P_{batt} can be calculated. To improve calculation time during simulations, a second degree polynomial is used instead of the look-up table, according to

$$P_{batt} = (A_{batt} \cdot (P_{batt,tot})^2 + B_{batt} \cdot (P_{batt,tot})), \quad (4.7)$$

where A_{batt} and B_{batt} are estimated constants derived from the look-up table.

4.2.1 Battery validation

It is important to make sure that the alternative to the look-up table, the polynomial, is accurate enough. Figure 4.2 shows the battery power for the two different cases. The one calculated from the polynomial is the dashed dotted line, the interpolated power is the solid line and the total battery power ($U_{oc} \cdot I$) is represented by the dashed line. As seen, the difference between these two methods is negligible.

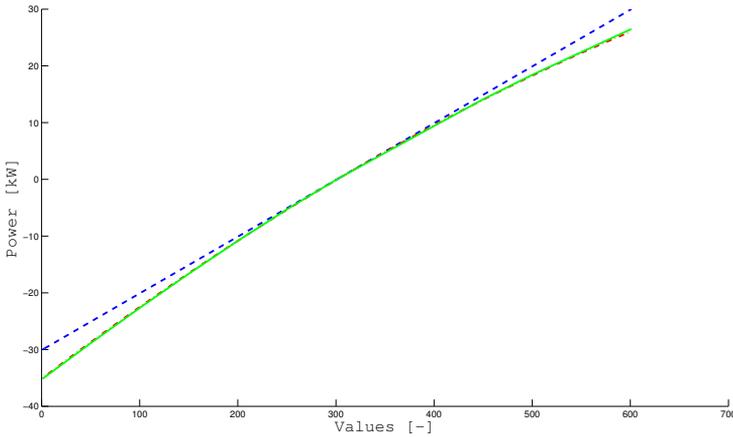


Figure 4.2. Estimated- and calculated battery power.

Secondly, the model must be validated against battery in the actual vehicle. Unfortunately no figures of the actual battery efficiency during a cycle were available which meant that the losses had to be estimated from other, accessible data. When power is either taken from or transferred to a battery the battery current deviate from the open circuit voltage. A higher output results in a larger difference. By multiplying this ΔU with the current one obtain an estimation of the power loss, according to

$$P(t)_{loss,estimated} = (U_{oc} - U(t)) \cdot I(t), \quad (4.8)$$

Figure 4.3 shows this estimated battery loss together with the one obtained from Equations (4.4), (4.5) and (4.7) The estimated loss is the solid line while the calculated ditto is illustrated by the dashed line.

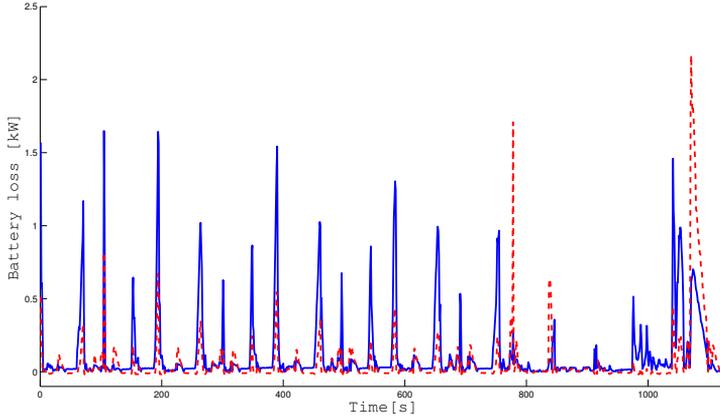


Figure 4.3. Estimated- and calculated battery loss.

The two curves are by no means identical, but considered that the comparison is made with an estimated value the results are acceptable. Figure 4.4 gives another perspective on this issue. This figure shows the effective battery power for the estimated and the calculated case, and as in the previous figure the calculated case is represented by a dashed line and the estimated ditto by a solid line. The overall appearances for the two powers are, as one can see, quite similar.

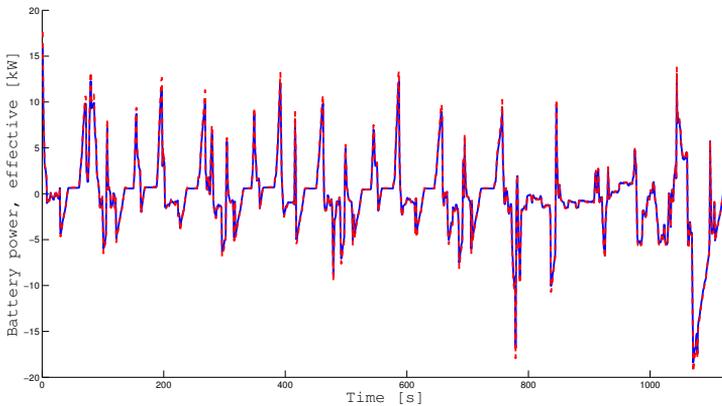


Figure 4.4. Effective battery power for the calculated and the estimated case.

4.3 Motor and generator

The relevant information wanted from the motor/generator model is, as mentioned before, the efficiency of these electric machines. This can be described with a simple equation as follows

$$P_{electric} = P_{mechanic} + P_{loss}, \quad (4.9)$$

were $P_{electric}$ is the electric power, either produced when the machine is acting as generator, or consumed when acting as a motor. $P_{mechanic}$ is the mechanical power, either produced when the machine is acting as a motor, or consumed when acting as a generator. P_{loss} summarizes all losses occurring within electrical machines, i.e. mechanical losses and iron losses. The losses resulted by the power electronics is treated separately. Due to the electric machine's ability to act either as a generator or motor one must keep in mind that $P_{electric} > P_{mechanic}$, for a motor and $P_{electric} < P_{mechanic}$, for a generator.

There are two common methods to describe P_{loss} . The first uses an efficiency map, similar to the one shown in Figure 4.10, where one interpolates the motor's efficiency using torque and angular velocity as inputs. In the second method, P_{loss} is calculated from a polynomial equation using torque and angular velocity as variables. During this work both methods were used but the latter one showed to be much faster when optimizing. The equation describing P_{loss} is as follows

$$P_{loss} = a_1 \cdot T_{m/g}^2 + a_2 \cdot T_{m/g} + a_3, \quad (4.10)$$

where the coefficients a_1 , a_2 and a_3 are variables dependet on the angular velocity.

4.4 PSD

The PSD modeled in this thesis and used in the two-mode hybrid is a power split device consisting of two combined planetary gear sets connected to an engine and two electric motors/generators, more information about the device can be found in Chapter 2 under Section 2.4. In Figure 4.5 a stick-lever diagram for the clutch configuration in mode one is shown.

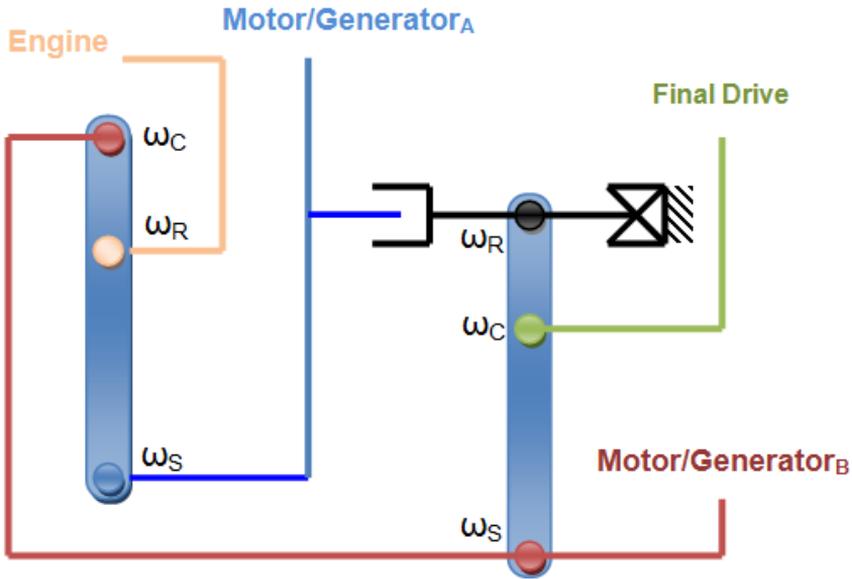


Figure 4.5. Stick-lever diagram for the clutch configuration in mode 1

From the power split device configuration in Figure 4.5, a relationship between the rotational velocities of the involved parts can be stated, this relationship yields in mode one for the first planetary gear set

$$\left. \begin{aligned} \omega_S &= \omega_a \\ \omega_C &= \omega_b \\ \omega_R &= \omega_e \\ N_S \omega_S - (N_S - N_R) \cdot \omega_C - N_R \cdot \omega_R &= 0 \end{aligned} \right\} \Rightarrow \omega_e = \frac{N_S}{N_R} \cdot \omega_a (1 - \frac{N_S}{N_R}) \omega_b, \quad (4.11)$$

where N_S and N_R are the number of teeth of the sun-/ring gear in the first planetary gear set. $\omega_S / \omega_C / \omega_R$ is the angular velocity of the sun-/ carrier-/ ring-gear. Introducing the nomenclature X_{ω_x, ω_y} , representing the speed contribution from ω_y to ω_x , gives that ω_e can be represented as

$$\begin{aligned} \omega_e &= \frac{N_S}{N_R} \cdot \omega_a (1 - \frac{N_S}{N_R}) \omega_b \\ &= X_{\omega_e, \omega_a} \cdot \omega_a + X_{\omega_e, \omega_b} \cdot \omega_b \end{aligned} \quad (4.12)$$

In the secondary planetary gear set, the ring is held by a clutch to zero speed ($\omega_R = 0$), and the sun-/carrier speed ratio is now given by

$$\left. \begin{aligned} \omega_S &= \omega_b \\ \omega_C &= \omega_o \\ \omega_R &= 0 \\ N_S \omega_S + N_R \omega_R - (N_S + N_R) \cdot \omega_C &= 0 \end{aligned} \right\} \Rightarrow \omega_o = \frac{N_S}{N_S + N_R} \cdot \omega_b \quad (4.13)$$

ω_o can with the introduced nomenclature be rewritten as

$$\begin{aligned} \omega_o &= \frac{N_S}{N_S + N_R} \cdot \omega_b \\ &= X_{\omega_o, \omega_b} \cdot \omega_b \end{aligned} \quad (4.14)$$

Combining these equations, gives that (ω_A) and (ω_B) can be expressed in mode one as

$$\begin{aligned} \omega_a &= X_{\omega_a, \omega_{e1}} \cdot \omega_e + X_{\omega_a, \omega_{o1}} \cdot \omega_o \\ \omega_b &= X_{\omega_b, \omega_{o1}} \cdot \omega_o \end{aligned} \quad (4.15)$$

The configuration for mode two differs from mode one, and can be stated as

$$\begin{aligned} \omega_e &= \frac{N_{S1} \cdot \omega_a - \omega_b \cdot N_{S1} + \omega_b \cdot N_{r1}}{N_{r1}} \\ \omega_o &= \frac{N_{S2} \cdot \omega_b + N_{r2} \cdot \omega_a}{N_{S2} + N_{r2}}, \end{aligned} \quad (4.16)$$

where number 1 and 2 represent the first, respectively the second gear set. Using the same nomenclature as previous, the (ω_A) and (ω_B) can be expressed as

$$\begin{aligned} \omega_a &= X_{\omega_a, \omega_e} \cdot \omega_e + X_{\omega_a, \omega_{o2}} \cdot \omega_o \\ \omega_b &= X_{\omega_b, \omega_{e2}} \cdot \omega_e + X_{\omega_b, \omega_{o2}} \cdot \omega_o \end{aligned} \quad (4.17)$$

The two Equations, (4.15) and (4.17) can be rewritten to a more general equation

$$\begin{aligned} \omega_a &= X_{\omega_a, \omega_e} \cdot \omega_e + X_{\omega_a, \omega_o} \cdot \omega_o \\ \omega_b &= X_{\omega_b, \omega_e} \cdot \omega_e + X_{\omega_b, \omega_o} \cdot \omega_o, \end{aligned} \quad (4.18)$$

where X_{ω_a, ω_e} , X_{ω_a, ω_o} , X_{ω_b, ω_o} and X_{ω_b, ω_e} are variables, dependent on the current mode.

The Torque equations for (T_A) and (T_B) , representing the combined power split used in the two-mode, can with a similar nomenclature as above, be stated as

$$\begin{aligned} T_A &= X_{T_A, T_O} \cdot T_O + X_{T_A, T_E} \cdot T_E \\ T_B &= X_{T_B, T_O} \cdot T_O + X_{T_B, T_E} \cdot T_E. \end{aligned} \quad (4.19)$$

X_{T_A, T_O} , X_{T_A, T_E} , X_{T_B, T_O} and X_{T_B, T_E} are variables dependent on the mode. For a more realistic model, losses were added to the equation, representing losses for the whole powertrain. For more information about the losses see Section 4.4.2

$$\begin{aligned} T_A &= X_{T_A, T_O} \cdot T_O + X_{T_A, T_E} \cdot T_E + losses, m/g_A \\ T_B &= X_{T_B, T_O} \cdot T_O + X_{T_B, T_E} \cdot T_E + losses, m/g_B \end{aligned} \quad (4.20)$$

4.4.1 PSD validation

To verify that the speed- and torque equations for the PSD in the previous section are valid, and can be used for modeling the PSD during simulations, test comparing the measured data with the calculated has been carried out. The Equations validated are (4.18) and (4.19), these equations consist of N_E , N_O , T_E and T_O , and are during the test represented by measured data from test cell. The result is presented in Figure 4.6, 4.7, 4.8 and shows the validation for the fourth UDC and the following EUDC part in the NEDC cycle.

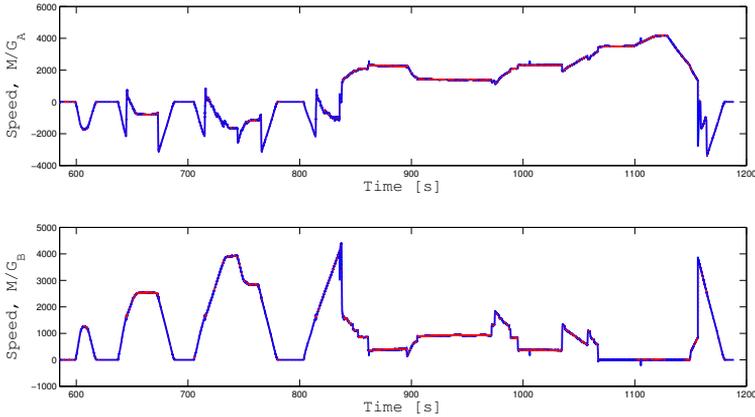


Figure 4.6. Speed validation of m/g_A and m/g_B for the two-mode HEV.

In the figure above, consisting of two subplots, the first subplot shows the speed for the m/g_A , where the measured curve is represented by a solid line and the speed obtained from Equation (4.18) is represented by a dashed line. In subplot two the m/g_B is shown, and as in subplot one the measured speed curve is a solid line and the calculated one, obtain from the second line in Equation (4.18) is dashed. The measured and the calculated ditto in subplot one, as well as in subplot two are almost identical, thus making it hard to distinguish them apart in Figure 4.6. The only difference between the curves are a transient occurring during the deceleration at the end of the EUDC for m/g_A . This transient depends on a temporary fault in the measured engine speed used in the equations during the validation.

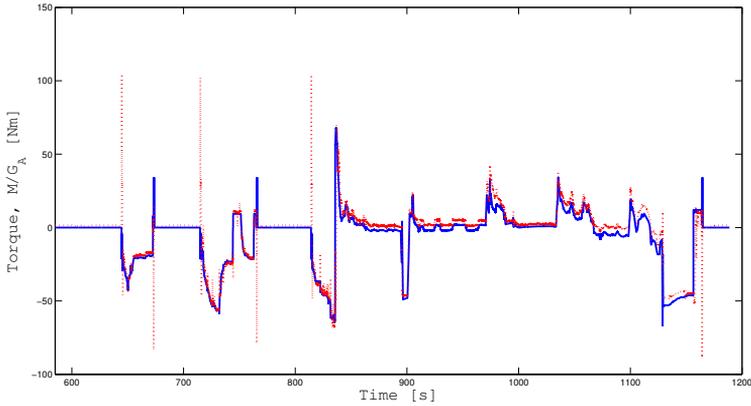


Figure 4.7. Validation of m/g_A torque for the two-mode HEV.

Figure 4.7 shows the m/g_A torque during the NEDC, the figure shows a similar behavior between the two curves. The dashed curve shows the calculated torque, based on Equation 4.19, and has a very similar behavior with the measured torque, represented by a solid line. There are some differences between the two curves, first there are several transients during the NEDC cycle, these occurs when m/g_A is turned on or off, and is seen as instant change in value during a short period of time, second, there is a small difference in value between the two curves during the whole cycle, which probably depends on bias fault in the measured data.

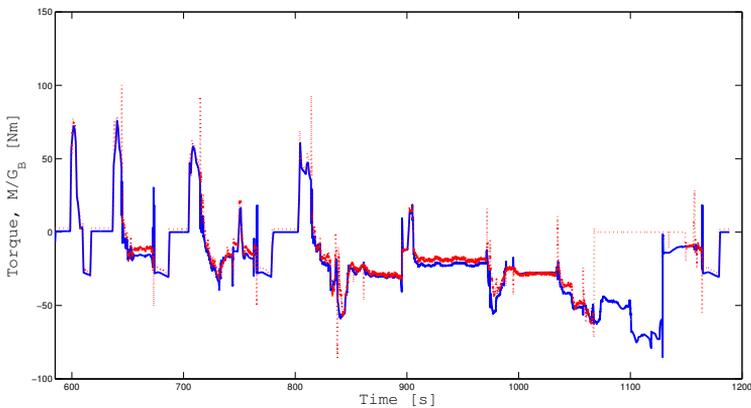


Figure 4.8. Validation of m/g_B torque for the two-mode HEV.

m/g_B is shown in Figure 4.8, where the dashed curve is the measured torque,

and the solid curve represent the calculated curve. As in the previous figure, transients are found during the cycle, especially when the motor /generator is turned off or on. A small difference between the two curves is seen, and during the end of the NEDC cycle there is a major difference between the values of the two curves. This difference depends on that the real two-mode has the ability to change gear in second mode, which has not been implemented in this thesis, making the two curves in the end of the EUDC part to differ.

4.4.2 Losses

The losses related to the rotating and accelerating elements in the PSD are summed up in the variable $losses, m/g$, seen in Equation (4.21), which has a separate value for each motor/generator. The two types are described in the following two sections.

$$\begin{aligned} losses, m/g_A &= losses, spin_A + losses, inertia_A \\ losses, m/g_B &= losses, spin_B + losses, inertia_B \end{aligned} \quad (4.21)$$

Spin losses

Spin losses represents the friction losses due to the interaction between the rotating parts in the PSD and in motors/generators. This can for instance be friction between the cogs and friction in bearings etc. and is dependent on the input and output speed (i.e. N_e and N_o) and how the PSD is configured (i.e. first or second mode). The relationship is shown in Equation (4.22), where $X_{T_{a/b}, N_e}$ and $X_{T_{a/b}, N_o}$ are mode-dependant constants.

$$\begin{aligned} losses, spin_A &= X_{T_a, N_e} \cdot N_e + X_{T_a, N_o} \cdot N_o \\ losses, spin_B &= X_{T_b, N_e} \cdot N_e + X_{T_b, N_o} \cdot N_o \end{aligned} \quad (4.22)$$

The temperature of the gearbox does also impact these losses and is further discussed in Section 4.4.3.

Inertia losses

Apart from the angular velocity, the angular acceleration acting on the gears and carriers in the PSD contributes to the total losses in the form of inertia losses. The relationship is shown in Equation (4.23), where X_{T_a, \dot{N}_o} and X_{T_b, \dot{N}_o} are mode-dependant constants. The engine's contribution to the inertia losses is dealt with in Section 4.5.2

$$\begin{aligned} losses, inertia_A &= X_{T_a, \dot{N}_o} \cdot \dot{N}_o \\ losses, inertia_B &= X_{T_b, \dot{N}_o} \cdot \dot{N}_o \end{aligned} \quad (4.23)$$

4.4.3 Gear oil pump

Gear oil pump losses can be estimated as addition of an offset value and its slope multiplied with the pump speed. Where slope and offset value is both dependent on pressure and oil temperature.

$$losses, pump = (Offset(Rpm, Celsius) + Slope(Rpm, Celsius)) \cdot Pressure \quad (4.24)$$

During simulations a 500 bar pressure and an oil temperature at 30° degrees Celsius is considered, which correspond to normal working condition for the gear oil pump. Equation (4.24) can now be simplified to

$$losses, pump = (Offset(Rpm) + Slope(Rpm)) \cdot 500, \quad (4.25)$$

whereupon one-dimensional look-up tables are used to find corresponding Offset and Slope value for a specific pump speed. Figure 4.9 shows the pump losses for a two-mode during the NEDC-cycle.

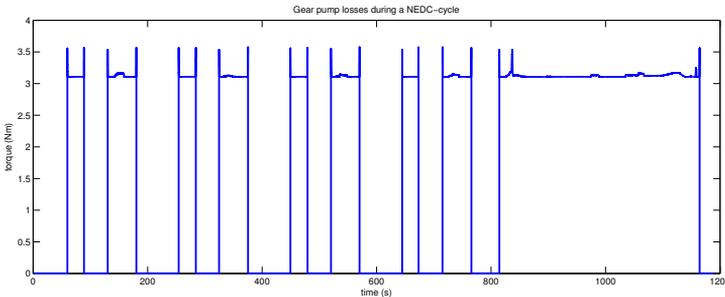


Figure 4.9. Pump losses for a 2-mode during NEDC.

4.5 Internal combustion engine

This section is divided into four subsections; the first describing the engine efficiency, the second the engine inertia, the third the modeling of the NOx limitation and the last shows the validation of the engine model.

4.5.1 ICE efficiency

As with the motor/generator the efficiency of the ICE can be described using either look-up table (with subsequent interpolation) or by means of an equation. In this case, the polynomial approach is not accurate enough, thus a look-up table is used instead. The map uses engine torque and angular velocity as inputs and grams consumed fuel per second as output (e.g. fuel consumption). To better see how the engine's efficiency depends on the operating points a mussel diagram is often used. Figure 4.10 below shows the specific fuel consumption as a function of torque and speed. Lower specific consumption equals higher efficiency, which means that the area defined by the circle in the middle of the diagram is where the engine's efficiency is at its best, and hence here one can expect the engine often to be if the optimization functions as expected.

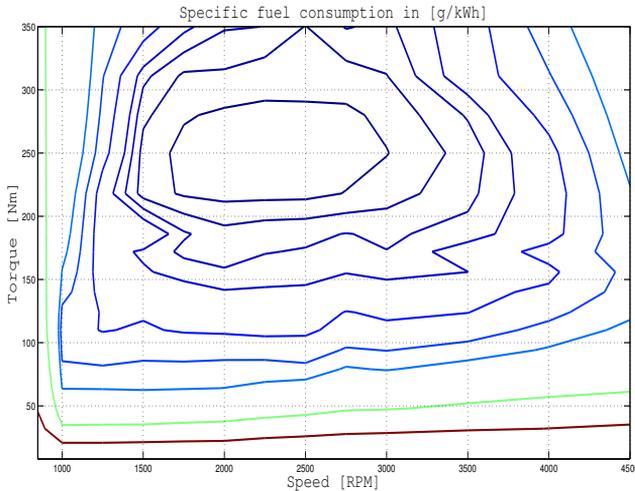


Figure 4.10. Specific fuel consumption.

4.5.2 Engine inertia

Losses due to inertial forces are described by

$$T_{J_e} = J_e \cdot \dot{w}_e, \quad (4.26)$$

where J_e is the inertia of the engine and the engine side of the transmission and w_e is the angular acceleration of the engine.

4.5.3 NOx limitations

It is not only the fuel consumption that is measured during a certification. NOx, particles, hydrocarbons and carbon oxide are examples of exhaust ingredients which are kept under supervision and need to be kept below certain levels decided by legislations. Modern diesel engines are with few exceptions equipped with diesel particulate filters which reduce soot dramatically and hence are particles not a problem. NOx, which is an umbrella term for nitrogen oxides, are on the other hand something that has to be taken in consideration, especially for diesel engines. Unfortunately is the origin of nitrogen oxides linked with high torque outputs at low engine speeds, e.g. the most efficient area of the engine. The approach chosen to avoid too high levels of NOx in this thesis is hence to limit the engine torque. The limitation is dependent of the engine speed. This method is perhaps a bit rough and not as detailed as for the actual engine, but it is best to be on the safe side in this case. Figure 4.11 shows the same efficiency mussel seen in Figure 4.10 but now together with the NOx limitation curve. Values above this dashed line are not valid.

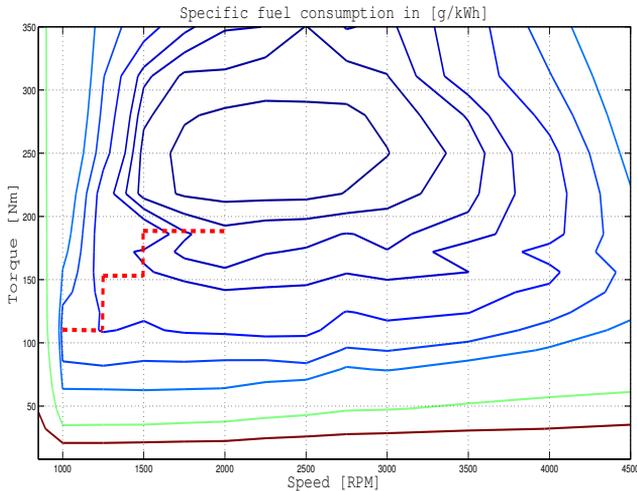


Figure 4.11. Specific fuel consumption with the NOx limitation curve.

4.5.4 ICE validation

The engine is validated by comparing the actual, measured fuel consumption from five GM measurements performed in a test cell on the NEDC cycle with calculated ditto. The data containing the engine torque and speed from these five tests are used to interpolate the fuel consumption, using the engine map described in Section 4.5.1 ICE efficiency. The results from the validation can be seen in Table

4.1. About 3 % difference was observed in these five tests, which must be considered as acceptable. There are various reasons for this disparity which can be derived from, amongst other, poor engine speed and torque data (i.e. transients) and due to the discretization of the look-up table.

Table 4.1. Measured and calculated fuel consumption.

Test	A	B	C	D	E	Unit
Fuel consumption NEDC (from test cell)	4.98	5.29	5.04	5.44	5.25	l/100 km
Fuel consumption NEDC (calculated from fuel map)	4.87	5.11	5.22	5.32	5.39	l/100 km
Difference (test cell/fuel map)	2.257	3.483	-3.467	2.256	-2.688	%

4.6 Cold start compensation

In the models of the vehicle and its components described in previous sections no consideration is taken to changes in temperatures. They are rather formulated for, and hence most accurate, when compared to the prototype operating at working temperatures. The efficiencies of the transmission, engine and, to some extent, the battery are worse at lower temperatures due to the fluids higher viscosity etc. To compensate for this when optimizing, the fuel consumption achieved during the low speed, UDC-part, of the cycle is multiplied by a factor. The components are assumed to have reached their normal operating temperature when reaching the high speed, EUDC part. The factor is driveline dependent and the value is 1.22 for the power-split HEV. In the parallel and the serial case are no compensation made for cold starts.

4.7 Auxiliary load

Auxiliary losses are losses that come from the use of electrical equipment in the car and other functions that do not take part in the actual propulsion of the vehicle, but rather maintain other functions. This can for example be the stereo, headlights, power steering, air condition, fuel pumps, oil pumps etc. Modern cars are often equipped with various electronic apparatus that can have, when turned on, a significant impact on the fuel economy. When performing fuel consumption tests there are rules according to standards regarding the equipment that needs to be activated, which is principally the brake lights. The auxiliary losses in this model are represented by simplest means by a DC load which is kept constant

during the cycle, see equation 4.27. This constant covers brake light, fuel and oil pumps and electronic control units.

$$P_{DC,load} = Constant \quad (4.27)$$

4.8 Parallel HEV

The reason for implementing and optimizing a parallel HEV is that the model is rather simple, which makes it possible, at least to some extent, to validate the DP-algorithm. It also makes a good foundation for the further work with the serial HEV and the power-split, since the different types of HEV's share both same and similar components. Additionally, since GM is also developing a parallel hybrid, a fully operating implementation of a parallel HEV is useful. For further reading about parallel HEV's in general, see Section 2.2.2.

The model for the vehicle, engine are described in their respective section. Since a parallel HEV only contains one electric machine for handling electric propulsion and generative braking, the relationship between battery and motor/generator can be described as follows

$$\begin{aligned} P_{battery} &= P_{electric,m/g} \\ &= P_{mechanical,m/g} + P_{losses} \\ &= (A_1 \cdot T_a + A_2)^2 + C a + P_{DC,load}. \end{aligned} \quad (4.28)$$

Both the ICE and the motor/generator are connected to the drive shaft yielding

$$\omega_{final} = \frac{\omega_e}{i_g} = \frac{\omega_{m/g}}{i_g}, \quad (4.29)$$

where i_g is the gear ratio. Since the ICE and the motor/generator together are responsible for meeting the propulsion power demanded, the equation for power balance equals

$$P_{vehicle} = P_{ice} + P_{mechanical,m/g}. \quad (4.30)$$

The entire MATLAB code for the parallel HEV is shown in Appendix B

4.9 Serial HEV

The serial HEV shares many similarities with the two-mode HEV, both when it comes to the architectural layout and the optimization procedure. Both need for instance two state variables. For further reading about serial HEV's in general, see Section 2.2.1.

The model for the vehicle, engine are described in their respective section. The following equation describes the power relationship between the battery, generator_A and motor/generator_B

$$\begin{aligned}
P_{batt} &= P_{el,m/g_A} + P_{el,m/g_B} + P_{DC,load} \\
&= P_{mech,m/g_A} + P_{loss,m/g_A} + P_{mech,m/g_B} + P_{loss,m/g_B} + P_{DC,load}
\end{aligned} \tag{4.31}$$

Since the motor/generator_B is solely responsible for the vehicles' propulsion,

$$P_{mech,m/g_B} = P_{vehicle} \tag{4.32}$$

The engine is coupled directly to the generator_A, yielding

$$P_{engine} = -P_{mech,m/g_A} \tag{4.33}$$

By inserting Equation (4.32) and (4.33) into (4.30), all variables are known or given by the states. The entire MATLAB code for the serial HEV is shown in Appendix C.

4.10 Two-mode HEV

The two-mode HEV, as said earlier, is when it comes to the architectural layout like a combination of the parallel and the serial HEV. Optimization wise is the procedure similar to the serial HEV in the meaning that they both require two state variables, which in this case is chosen to be the SoC and the engine speed. More information about the components involved and the power flow between them can be found under Section 2.4.3. For further reading about the two-mode HEV in general, see Section 2.4. The model for the vehicle, engine and PSD are described in their respective section. The relationship between the battery, auxiliary load, motor/generator_A and motor/generator_B can be described as follows

$$P_{batt} = P_{A,elec} + P_{B,elec} + P_{DC,load} \tag{4.34}$$

Equation (4.34) can be rewritten using the fact that the electric power equals the mechanical power plus losses for motor/generator_A and the motor/generator_B. The losses can then be estimated as second degrees polynomial, see Equation (4.35).

$$\begin{aligned}
P_{batt} &= (P_{A,mech} + P_{A,loss}) + (P_{B,mech} + P_{B,loss}) + P_{DC,load} \\
&= (N_A \cdot T_A + (a_1 \cdot T_A + a_2)^2) + (N_B \cdot T_B + (b_1 \cdot T_B + b_2)^2) + P_{DC,load} \\
&= (A_1 \cdot T_A + A_2)^2 + A_3 + (B_1 \cdot T_B + B_2)^2 + B_3 + P_{DC,load},
\end{aligned} \tag{4.35}$$

where $A_1, A_2, A_3, B_1, B_2, B_3$ depend on N_A respectively N_B , which are known from Equation (4.18) and the relation $N = \frac{\omega \cdot 30}{\pi}$. $P_{DC,load}$ is a known constant, see section 4.7. Combining the last expression for P_{batt} in the above equation with Equation (4.20) derived in section (4.4.2), gives that P_{batt} can be rewritten

as an equation consisting of T_E and T_O , where T_O is known. The calculated T_E can then be used to calculate the cost matrix, further information about the cost matrix can be found in chapter 3.

4.10.1 Two-mode HEV validation

The validation of the two-mode HEV is done by comparing data for engine torque measured in a test cell during a NEDC cycle, with the calculated engine torque, derived in the previous section. The derived engine torque is an entity, dependent on P_{batt} , T_{out} , N_E and N_O . These are during the validation obtained from measured data.

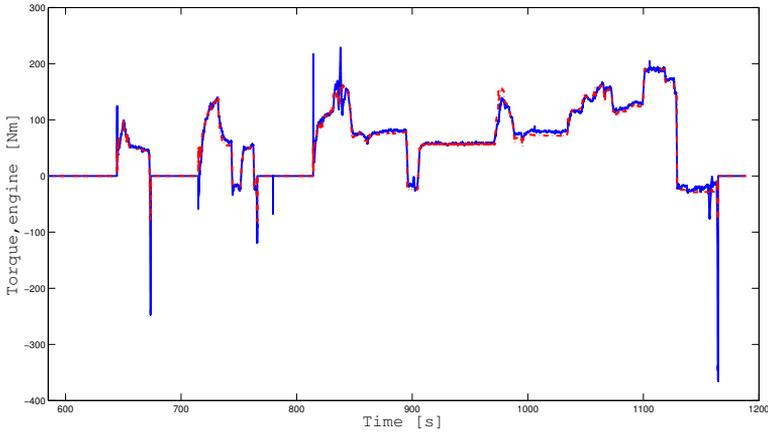


Figure 4.12. Validation of engine torque for the two-mode HEV.

Figure 4.12 shows the engine torque obtain from measured data (dashed line) and the calculated ditto (solid line) for the fourth UDC and the following EUDC in the NEDC cycle. The curves show no major difference in the appearance between them, except some transients in the calculated torque, occurring when the internal combustion engine it's turned on respectively turned off. These transients originate from transients in the measured P_{batt} .

Chapter 5

Case studies

In this chapter are the results of the optimizations of the three different types of HEV's analyzed. Section 5.1 and 5.2 treat two entirely fictional HEV's, a parallel and a serial, where the aim is to gain knowledge of the DP algorithm as well as the components involved. Section 5.3 presents the results, divided into two cases, of the optimization of the two-mode HEV. In Section 5.4 are data retrieved from the prototype compared to the optimal solution.

5.1 Parallel HEV

The model of the parallel HEV on which the optimization is performed can be seen in Section 4.8. For the complete MATLAB code, see Appendix B.

5.1.1 Input

Although many optimizations on different driving cycles and with different grid sizes etc were performed just one will be presented here. The inputs used can be seen in Table 5.1.

Table 5.1. Inputs for the parallel.

Input	Value
Cycle	NEDC
SoC, min and max	50 % and 61 %
SoC, start-end	58 %
SoC, spacing	1/35000, which equals 0.2 kW battery power
DC load	0 kW
Engine	Diesel, no NOx limitation
1st→5th gear ratio	9.97, 5.86, 3.84, 2.68, 2.14

5.1.2 Results

Figure 5.1 shows the SoC during the cycle (solid line) as well as the speed profile for the NEDC (dashed line). It is noticeable that the entire SoC range is used, and especially that the battery recovers 8 % SoC during the last few seconds of deceleration. Since 1 % equals roughly 70 kW this is a great amount of energy being transferred to the battery and not very realistic in real life. A lot of time was spent on figuring out the reason for this behavior, and the most probable explanations are that the battery losses are too small or that the equation describing the vehicle power is somewhat inaccurate. But on the other hand did not the validation of the battery or the vehicle, presented in Chapter 4, show on any major weaknesses. More expected behavior is the consistency during the four first UDC cycles (the first 800 seconds) and that the SoC is steady during standstill due to no DC losses.

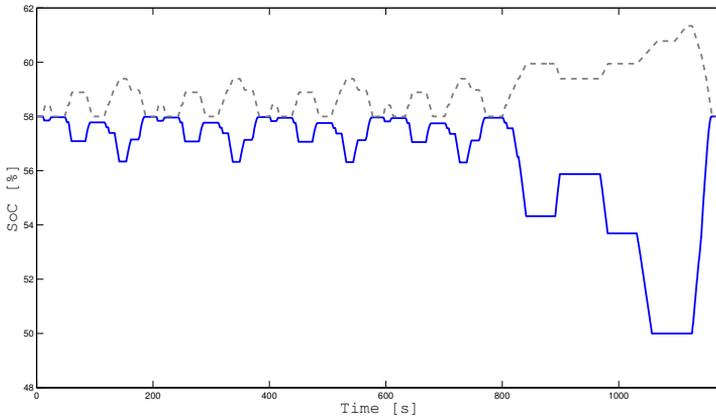


Figure 5.1. SoC level for the parallel HEV.

Similar to the SoC, the engine torque shows a repetitive behavior and therefore is only the first of the four UDC parts shown together with the EUDC part. Since the model lacks a clutch, the engine is always connected to the drive shaft and hence rotates with the wheel. This together with the engine's fuel cutoff, which occurs when the engine is dragging, is the reason why the torque (solid line) seen in Figures 5.2 and 5.3 is negative during deceleration. A finer grid will probably show more of this behavior.

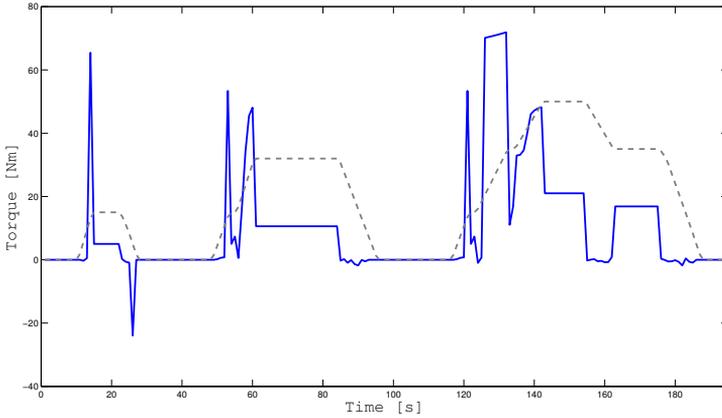


Figure 5.2. Engine Torque during UDC for the parallel HEV.

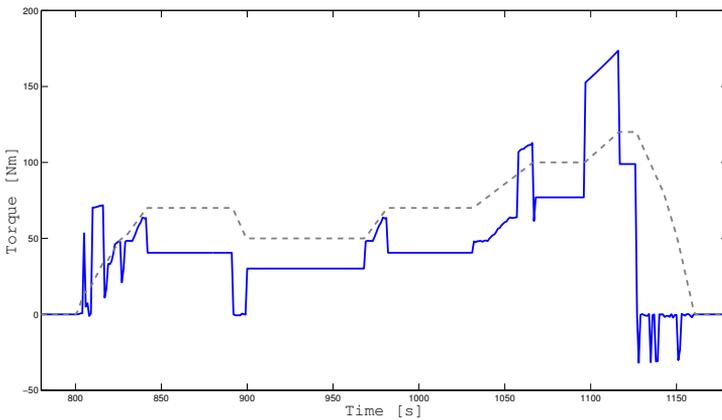


Figure 5.3. Engine Torque during EUDC for the parallel HEV.

Figures 5.4 and 5.5 show the power for the engine (dashed dotted line) and m/g (solid line) during the cycle. The total tractive power is represented by the dashed line. Figure 5.4 shows one of the four UDC parts and Figure 5.5 shows the EUDC part. One can see that the engine is solely responsible for the propulsion during steady state. The engine is also taking part in accelerating the vehicle, except for the first couple of seconds of each acceleration phase, which is handled by the motor/generator.

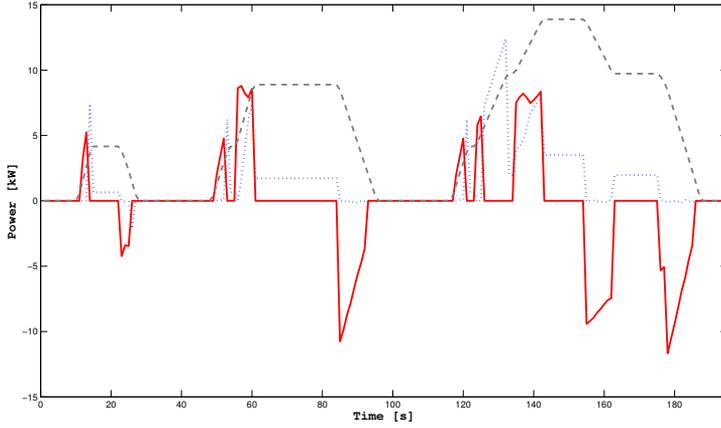


Figure 5.4. Engine, vehicle and motor/generator power for the parallel HEV during UDC.

What is expected and seen in Figure 5.4 and 5.5 is that the motor/generator absorbs most of the power throughout the deceleration and delivers roughly half of the power needed during acceleration in the EUDC part. The electric machine does not deliver nor receive power during the constant speed phases.

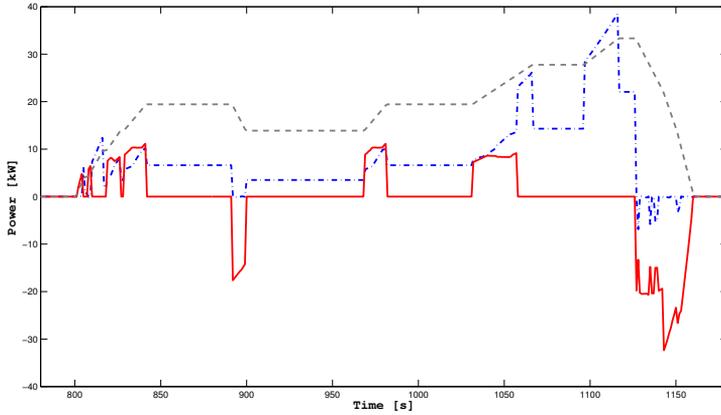


Figure 5.5. Engine, vehicle and motor/generator power for the parallel HEV during EUDC.

The optimal fuel consumption for the parallel HEV given the mentioned inputs is 3.94 l/100 km.

5.2 Serial HEV

This section will treat the implementation and optimization of the serial HEV.

5.2.1 Case 1

Although many optimizations on different driving cycles and with different grid sizes etc were performed just two different cases will be presented here. The inputs for the first case are shown in Table 5.2.

Table 5.2. Inputs for the Serial HEV during case 1.

Input	Value
Cycle	NEDC
SoC, max	62 %
SoC, min	48 %
SoC, start-end	58 %
SoC, spacing	1/7000, which equals 1 kW battery power
ICE speed grid	0 1000 1150 1300 1450 1600 1750 2000 rpm
DC load	0.6 kW
Engine	Diesel, no NOx limitation

Results

Figure 5.6 shows the SoC during the cycle (solid line). The entire SoC range is not used in this case. One can see that the lower bound, 48 %, is reached during the acceleration in the EUDC part (the velocity profile is seen as the gray dashed line), while SoC is never even close to reach the upper bound. In fact, the only point where the SoC is above the initial value of 58 % is at the very end of the cycle.

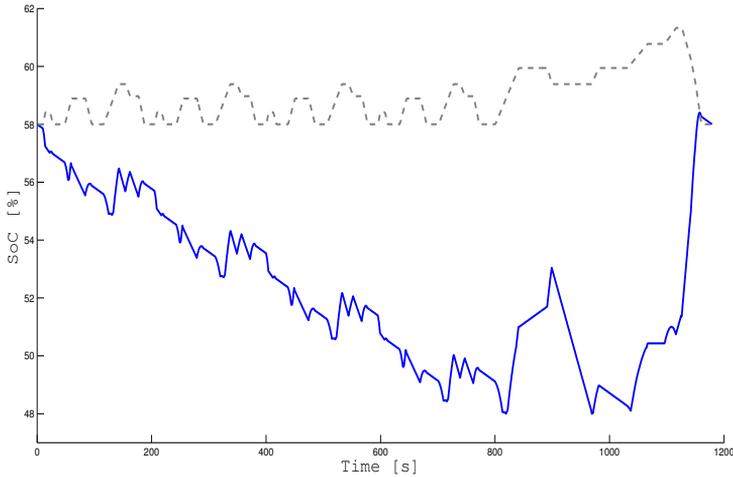


Figure 5.6. SoC for the serial HEV during case 1.

Furthermore does it seem like the vehicle drains a lot of power from the battery, even during standstill, and closer look at Figure 5.7, reveals that roughly 1 kW is consumed during a second of standstill. This makes sense for two reasons. First, as seen in table 5.2, is the DC load in this case 0.6 kW, which must be provided for even when the vehicle has come to a halt. Secondly, equals the smallest jump possible in the SoC grid approximately 1 kW. This means that this 0.6 kW load (plus losses) is powered with 1 kW battery power. A finer grid will therefore probably lead to a slight better use of the battery and perhaps lower fuel consumption.

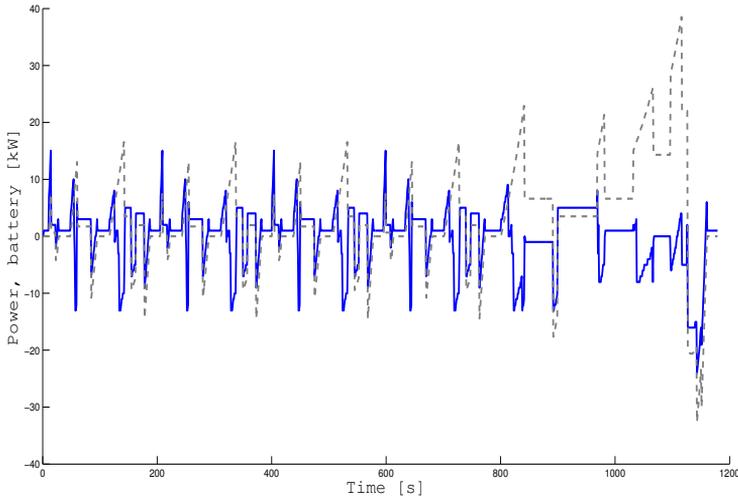


Figure 5.7. Battery and vehicle power for the serial HEV during case 1.

One should also notice the high positive outputs in Figure 5.7 (solid line), especially those which occur at $t = 10$ seconds, $t = 210$ seconds and so on. These peaks are about twice as high as the required vehicle power, represented by the dashed line, which verifies the poor efficiency of the serial HEV stated in Section 2.2.1. This is also seen in the fuel consumption, 4.93 l/100 km, which is substantially higher than for the parallel HEV.

The power output from the engine during the cycle can be seen in Figure 5.8 as the solid line (total tractive power is represented by the dashed line). Although the engine power is not directly related, at least in theory, to the vehicle power it is easy to see a relationship in this optimal solution. By synchronizing the engine output with the high power output required during acceleration, the loss due to energy conversion in the battery is eliminated. During the low speed sections in the UDC parts is energy solely taken from the battery.

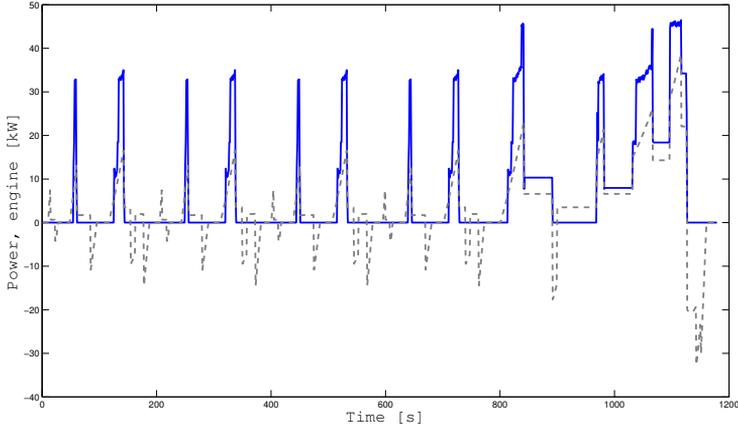


Figure 5.8. Engine and vehicle power for the serial HEV during case 1.

The engine torque, which can be seen in Figure 5.9 as the solid line (the vehicle speed curve is illustrated as the dashed grey line), seldom go below 90 Nm during the cycle. This must be considered as an expected behavior since the engine is most efficient at high torque, low speed outputs.

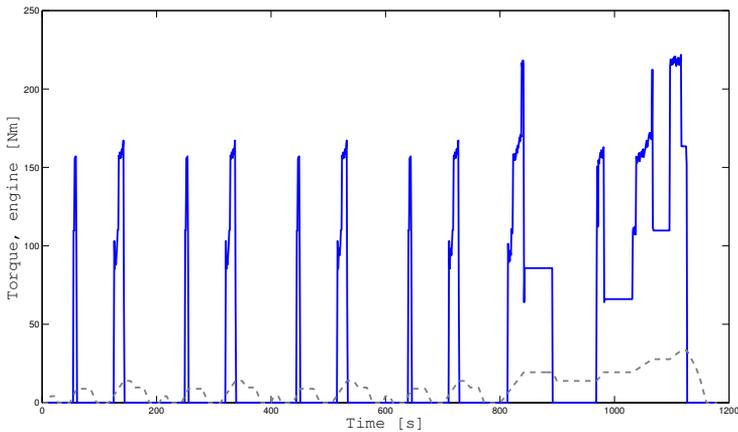


Figure 5.9. Engine torque for the serial HEV during case 1.

What is more remarkable is the engine speed seen in Figure 5.10. Even though the upper limit for the engine speed is set relatively low (2000 rpm) in this case, it is interesting to see that the engine often chooses to keep itself at this maximum. Figures describing the electric motor and the generator are deliberately left out

of this work. Since the motor’s output is equal to the power required to propel the vehicle and the generator is directly coupled to the engine these Figures are redundant.

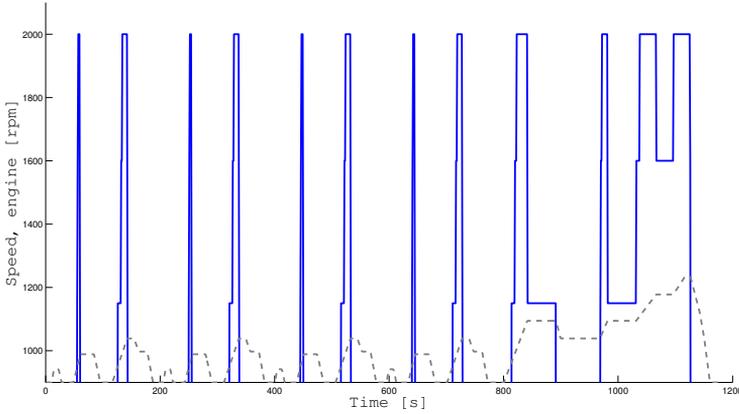


Figure 5.10. Engine speed for the serial HEV during case 1.

5.2.2 Case 2

The main difference between this case and the first case is that the ICE speed grid is changed. The upper limit is increased and is now reaching 3000 rpm. The speed steps had to be increased to keep the optimization time within reasonable limits. The SoC upper limit was lowered to 60 % , see Table 5.3

Table 5.3. Inputs for the Serial HEV during case 2.

nput	Value
Cycle	NEDC
SoC, max	60 %
SoC, min	48 %
SoC, start-end	58 %
SoC, spacing	1/7000, which equals 1 kW battery power
ICE speed grid	0 1000 1250 1500 1750 2000 2250 2500 2750 3000 rpm
DC load	0.6 kW
Engine	Diesel, no NOx limitation

Results

This optimization took 11 hours to perform and resulted in a fuel consumption of 4.95 l/100 km, which is 0.02 liters more than the previous run. The slightly higher consumption is probably a result from the increased spacing in the grid for the engine speed. The SoC does not differ much from the previous run when it comes to the overall look. But when Figure 5.11 is closer compared to Figure 5.6 in Section 5.2.1 one can see that the SoC in this run reaches the lower bound much later in the cycle. The upper bound is not even close to be reached, which was expected. The velocity profile is represented by a dashed line all in figures in this section.

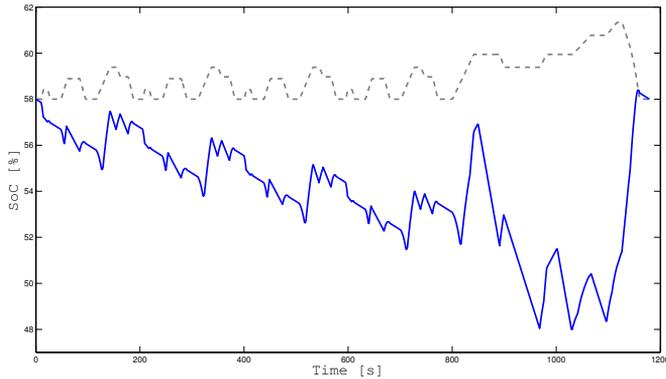


Figure 5.11. SoC for the serial HEV during case 2.

Now to the engine speed, seen in Figure 5.12, which was the main focus for this run. In contrary to the prior test the engine is now allowed to rev up to 3000 rpm. It is satisfying to see that the engine seldom exceeds 2000 rpm. It is in fact just during the EUDC part of the cycle where speeds over 2000 rpm are used, and even then never over 2500 rpm. The conclusion can be made that rpm's exceeding 2500 are unnecessary for an optimal solution.

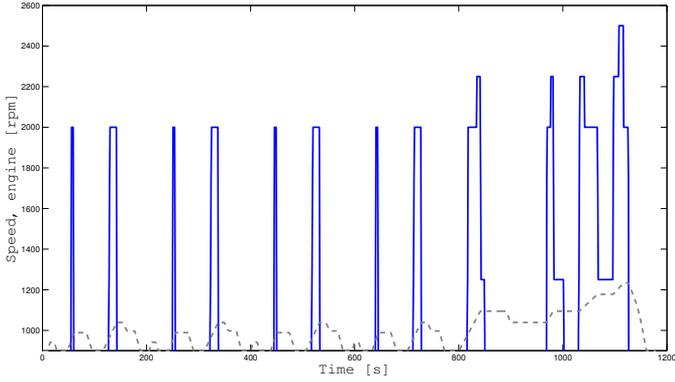


Figure 5.12. Engine speed for the serial HEV during case 2.

The engine torque, seen in Figure 5.13 is at the same level as before, but is distributed a bit different. Now the engine is only on during the acceleration phases, even during the EUDC part, which was not the case in the past case.

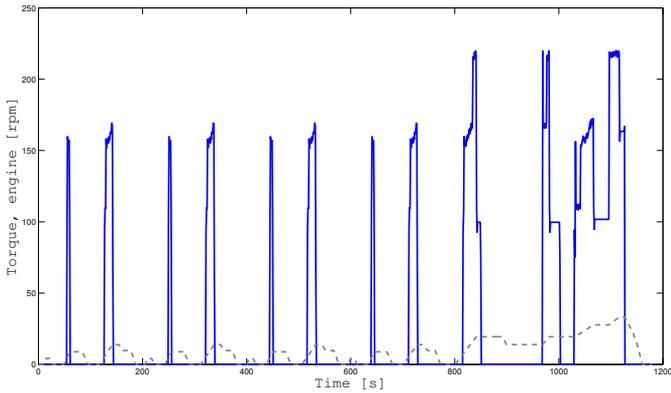


Figure 5.13. Engine torque for the serial HEV during case 2.

5.3 Two-mode HEV

This section will handle the optimization of GM's two-mode HEV (generally known as combined power split hybrid, see Section 2.2.3) and is separated into two different cases. The first case will treat the optimization of the two-mode HEV without considering NOx emissions, while the second case takes the NOx into consideration.

5.3.1 Input

Two different input configurations have been chosen. Each will be further described under respective section.

Case 1

This case will in one sense represents the most optimal solution for fuel consumption. All losses are represented but the engine is just constrained to its physical limitations (that is speed and torque) and no limitations due to NOx emissions are taken into consideration. To reduce optimization time the maximum engine speed is limited to 2000 rpm. This should not affect the results since the actual engine does not rev beyond 2000 rpm during the NEDC cycle. The inputs for case 1 can be seen in Table 5.4

Table 5.4. Inputs for the two-mode HEV during case 1.

Input	Value
Cycle	NEDC
SoC, max	62 %
SoC, min	48 %
SoC, start-end	58 %
SoC, spacing	1/7000, which equals 1 kW battery power
ICE speed grid	0 1000 1150 1300 1450 1600 1750 2000 rpm
DC load	0.6 kW
Engine	Diesel, no NOx limitation

Case 2

The inputs in this case will better represent the real vehicle, since the NOx limitation is activated (see Section 4.5.3 for details). This is also the only different from Case 1, see Table 5.5.

Table 5.5. Inputs for the two-mode HEV during case 2.

Input	Value
Cycle	NEDC
SoC, max	62 %
SoC, min	48 %
SoC, start-end	58 %
SoC, spacing	1/7000, which equals 1 kW battery power
ICE speed grid	0 1000 1150 1300 1450 1600 1750 2000 rpm
DC load	0.6 kW
Engine	Diesel, NOx limitation

5.3.2 Results

The results from the two optimization cases are presented here. If nothing else is mentioned in the text is Case 1 represented by a solid curve in the figures and Case 2 by a dashed-dotted curve. In some figures, where powers are shown, is $P_{vehicle}$, which is the power required to propel the vehicle is illustrated by a dashed curve. Otherwise does the dashed curve represent the velocity profile. For better readability are some of the figures of the whole NEDC cycle divided into two plots, one showing the first two UDC parts and the other the EUDC part.

5.3.3 Engine

In Figures 5.14 and 5.15 below can the engine power output be seen together with the total tractive power. The most remarkable with the appearance is that Case 2 has a power peak at around $t = 60$ seconds, seen in Figure 5.14, while Case 1 has not. The explanation for this can be seen in the subsequent power peaks where they are considerably lower for the second case due to the torque limitation. This is however compensated with wider peaks. One can also see that the power output during the accelerations is much higher, for both cases, than the one required to accelerate the vehicle, which means that the excess power is stored in the battery.

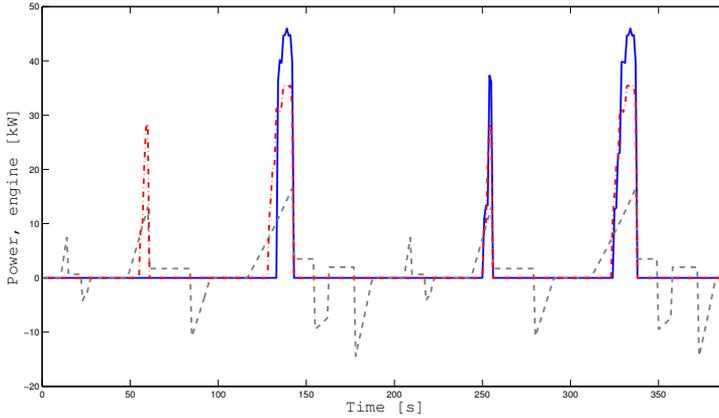


Figure 5.14. Engine and vehicle power for power-split HEV during the UDC for case 1 & 2.

The behavior during the EUDC part, seen in Figure 5.15, is quite similar between the two cases except for the second case's somewhat lower power output during the accelerations, which also can be related to the NOx limitation.

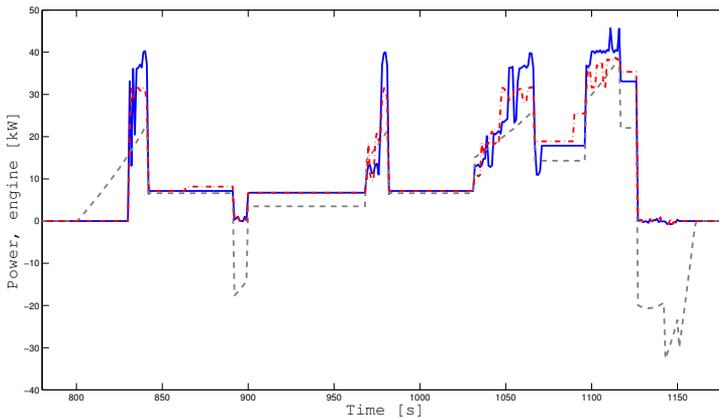


Figure 5.15. Engine and vehicle power for power-split HEV during the EUDC for case 1 & 2.

Figures 5.16 and 5.17 show the engine torques during the cycle. As seen, the engine tries to keep the torques as high as possible, often around 220 Nm for the first case, while the torque for case 2 is effectively limited to around 170 Nm to reduce NOx emissions. One noticeable effect of the transition from the first to the second mode (which occur at $t=830$ seconds), seen in Figure 5.17, is that the engine no longer can be turned off, which results in negative torques during

decelerations, similar to the engine torques for the parallel HEV seen in Section 5.1.

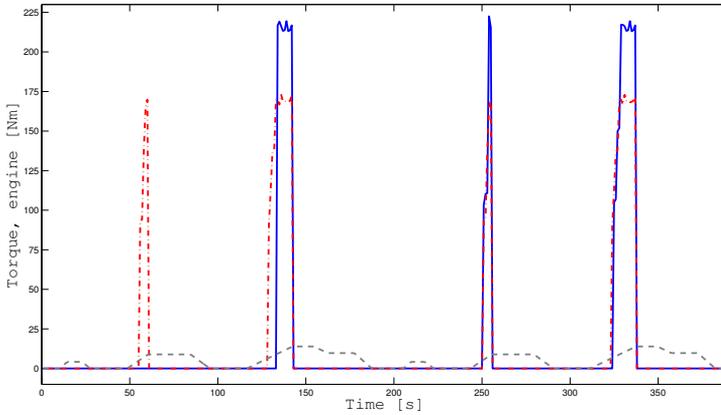


Figure 5.16. Engine torque for power-split HEV during UDC for case 1 & 2.

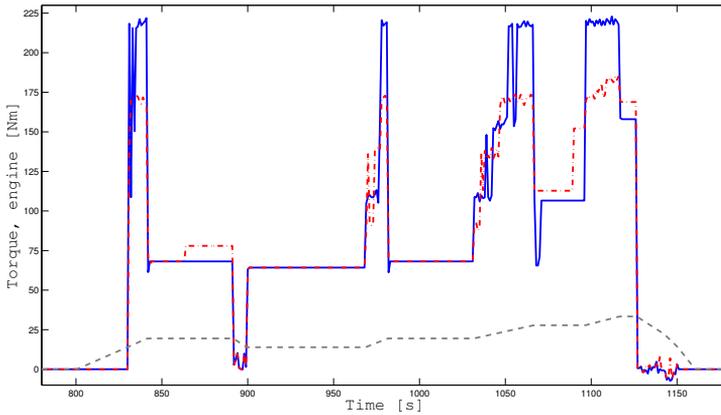


Figure 5.17. Engine torque for power-split HEV during EUDC for case 1 & 2.

The engine often remains at low speeds, which can be seen in Figure 5.18, and does only rev up to the 2000 rpm limit during short periods and only during the high acceleration parts. Since the engine is not allowed to be turned off in second mode, seen in Figure 5.19, it is kept at low speeds during the coasting and deceleration parts. No significant difference can be seen between the two cases.

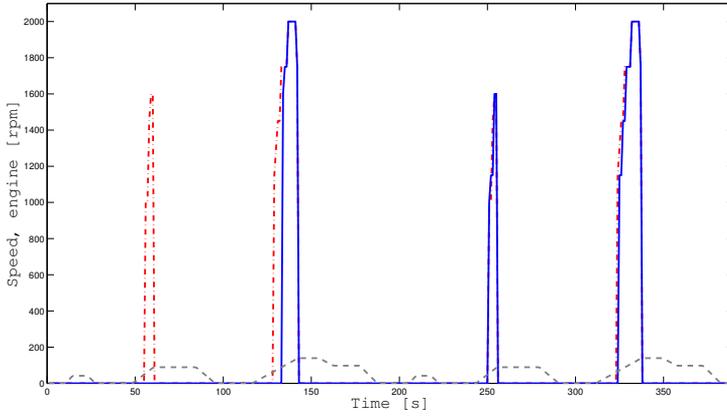


Figure 5.18. Engine speed for power-split HEV during UDC for case 1 & 2.

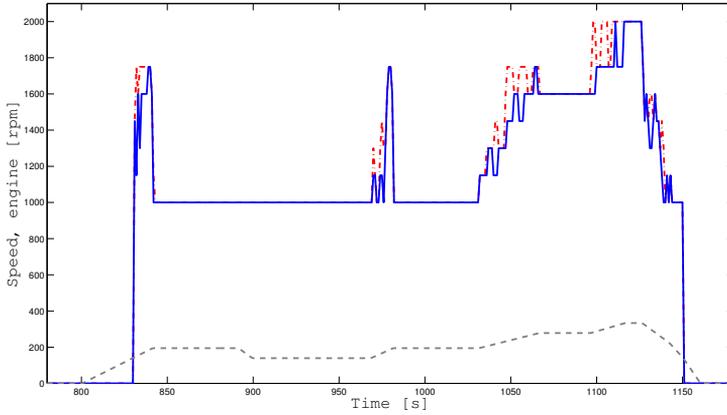


Figure 5.19. Engine speed for power-split HEV during EUDC for case 1 & 2.

Figures 5.16 - 5.19 are summarized in Figures 5.20 and 5.21 which show the musells of the specific fuel consumption (as seen in Section 4.5.1) along with the operation points for the two cases, each dot representing one second in the cycle. When comparing the two figures one sees that the points for Case 1 in Figure 5.20 are spread further apart than the ones for Case 2 seen in Figure 5.21. The points present in the most efficient area in Figure 5.20 are forced down to the less efficient area, seen in Figure 5.21 due to the NO_x limitation. A glance at Figure 4.11 verifies that the operation points for Case 2 are kept below the dashed line as intended. Apart from a few points in the lower right regions most point are situated according a pattern following an imaginary line reaching in a direction from the lower left to the upper left, for both cases. The descritization of the engine speed is also easy to see in the figures, since the points are grouped according to the Ne grid seen in Tables 5.4 and 5.5. One could almost distinguish a torque descritization since most of the points are also grouped according to the y-axis. However, this is not the case but rather a result of the DP algorithm which seems to choose a few points which are the most efficient.

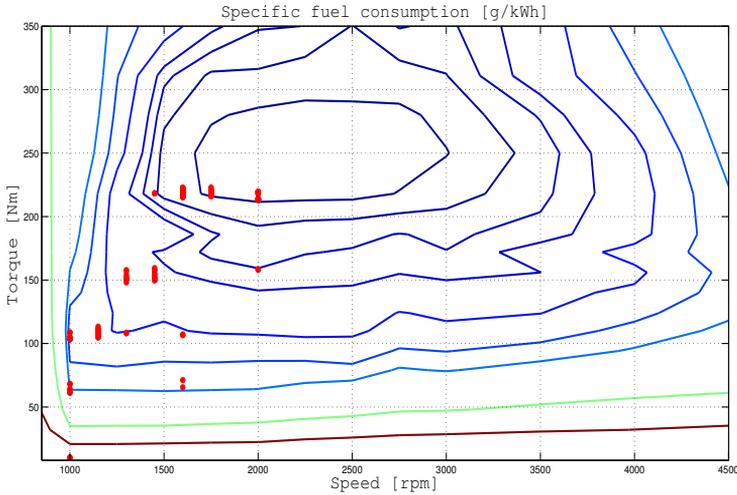


Figure 5.20. Specific fuel consumption with operation points for Case 1.

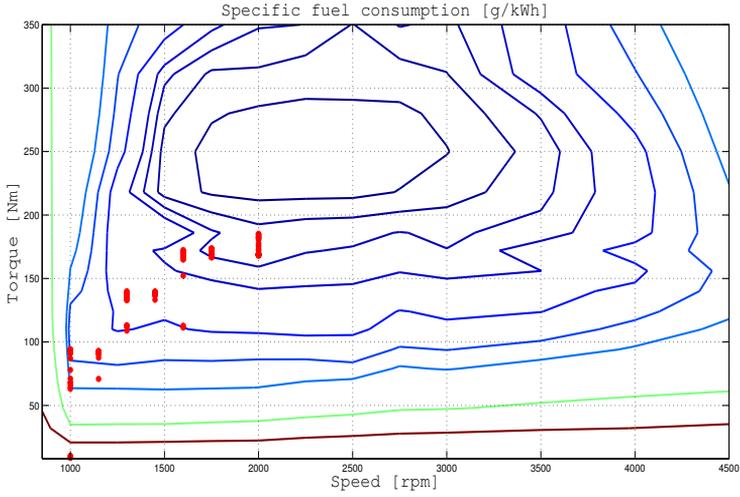


Figure 5.21. Specific fuel consumption with operation points for Case 2.

5.3.4 Battery

Figure 5.22 shows the SoC for the two different cases. There is no major difference in the appearance between them, except that the SoC for the first case drops significantly more than the second at around $t = 60$. Both reach the lower bound simultaneously.

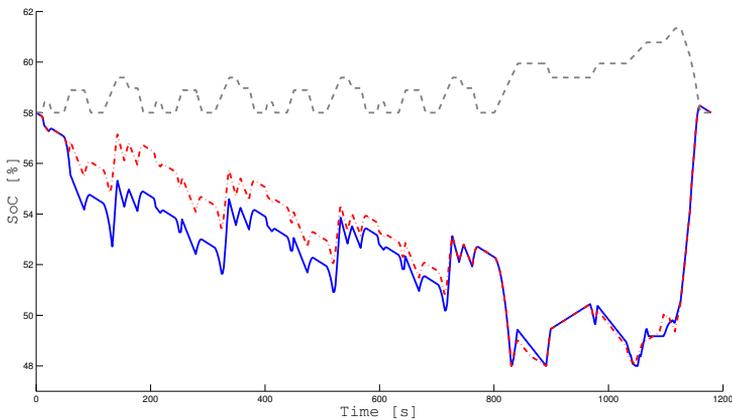


Figure 5.22. SoC for Power-split HEV during case 1 & 2.

Where difference can be seen in the SoC can also, logically, a difference be seen in the battery power. The two cases differ most in the negative powers (seen in Figure 5.23), because of the NO_x limitation, which keeps the engine in case 2 from load the battery as much during the acceleration peaks. One can also see the battery output, at around 1 kW, running the DC load during standstills.

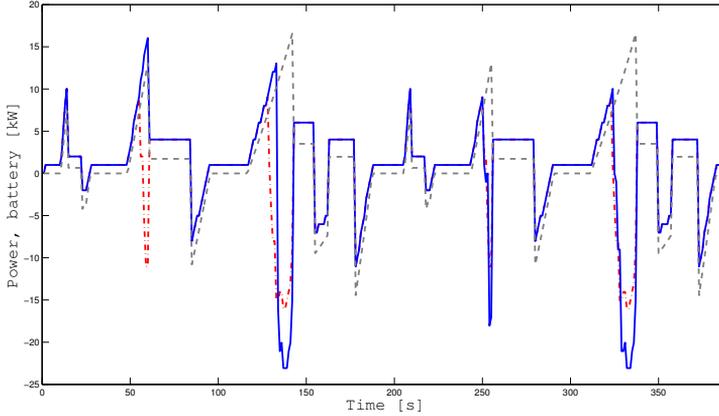


Figure 5.23. Battery and vehicle power for power-split HEV during UDC for case 1 & 2.

One can also notice the transition from first to second mode at $t = 830$ seconds in Figure 5.24, which shows the battery power for the EUDC part. During mode two is the positive power output much lower than the first. There is no major difference between the two cases during this part.

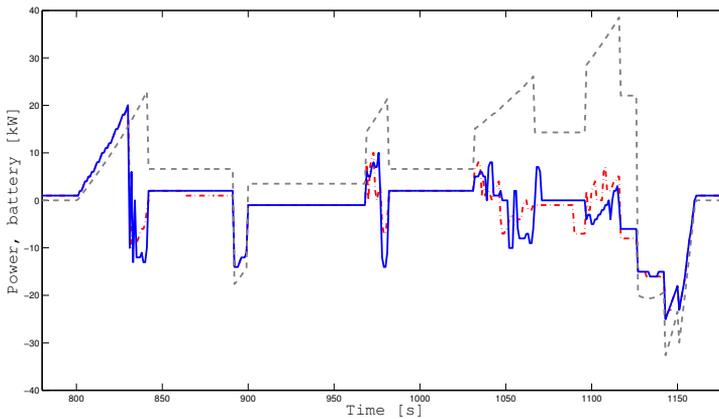


Figure 5.24. Battery and vehicle power for power-split HEV during EUDC for case 1 & 2.

5.3.5 Motors/generators

Figures 5.25 and 5.26 illustrate the power for motor/generator_A. As seen does not m/g_A take part much in neither the electrical propulsion nor the generative

braking while in first mode (seen in Figure 5.25). However, in second mode, seen in Figure 5.26, is m/g_A very much responsible for the regenerative braking. It is hard to tell the two cases apart, especially in the first mode. Here one notice the dip at $t = 60$ seconds, which originates from the fact that the engine is turned on at that point in the first case but not during the second.

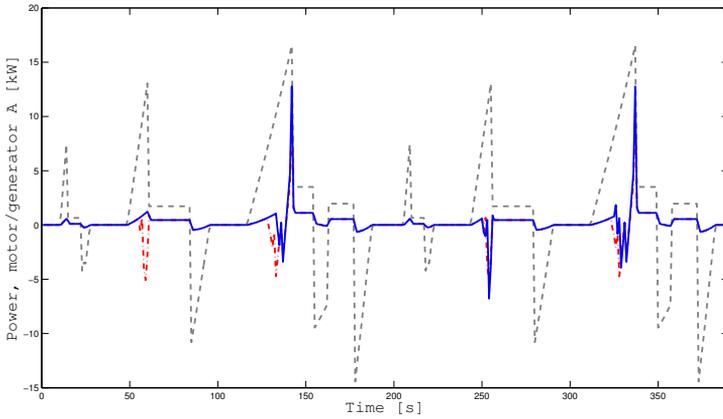


Figure 5.25. Power for motor/generator_A and vehicle during UDC for case 1 & 2.

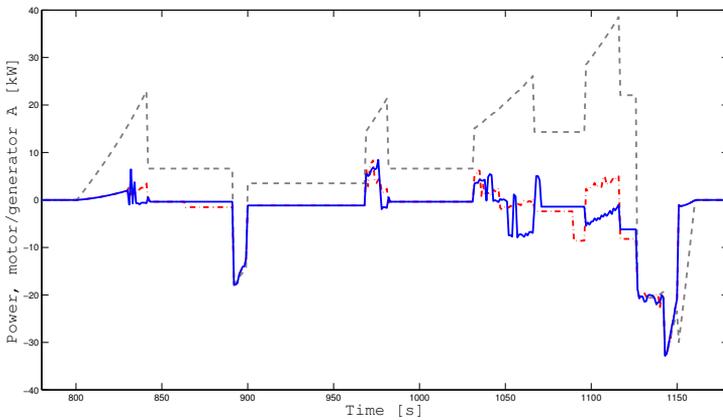


Figure 5.26. Power for motor/generator_A and vehicle during EUDC for case 1 & 2.

As seen in Figure 5.27, the motor/generator_B is much more present in the first mode than motor/generator_A (Figure 5.25). In fact, it would not be an exaggeration to say that m/g_B is responsible for almost all electrical propulsion

and deceleration in the first mode. It is also noticeable to see the quick transition m/g_B undergoes between acting as a motor and a generator in the acceleration parts in the first mode, as well as at the shift between mode 1 and mode 2 seen in Figure 5.28. The impact of the NOx limitation, active in the second case, can also be clearly seen in Figure 5.27. Due to the lower power output from the engine during the acceleration parts in this case, the negative power dips are shallower, but at the same time wider.

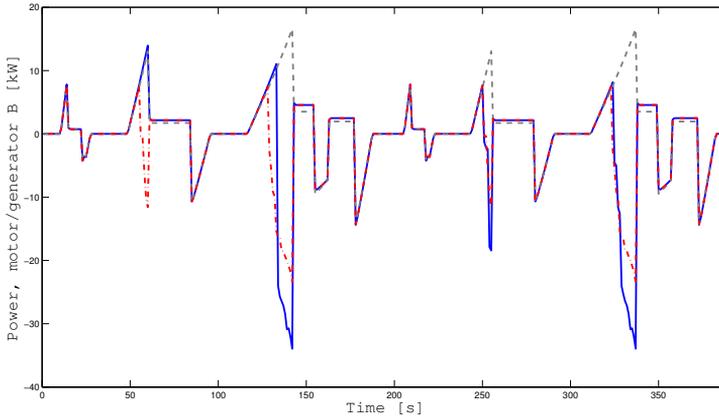


Figure 5.27. Power for motor/generator_B and vehicle during UDC for case 1 & 2.

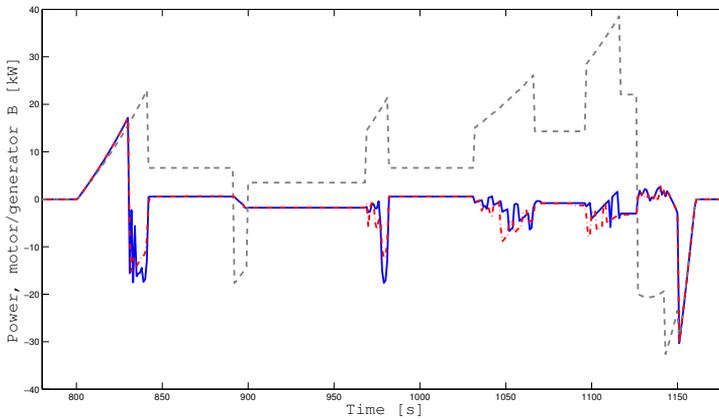


Figure 5.28. Power for motor/generator_B and vehicle during EUDC for case 1 & 2.

Fuel consumption

The optimization resulted in a fuel consumption of 4.736 l/100 km for case 2 with NOx limitations and cold start compensation, without the cold start compensation the fuel consumption ends at 4.439 l/100 km. Case 1 results in a consumption of 4.695 l/100 km with no NOx limitations but with compensation for cold start, without the compensation the consumption ends at 4.401 l/100 km. The result is summarized in Table 5.6, a comparison of the two cases gives a 0.0410 l/100 km, or 0.87 % lower consumption for case 1 with compensation for cold start and 0.0387 l/100 km, or 0.88 % without.

Table 5.6. Optimized fuel consumption for the two-mode HEV.

	Case 1, without NOx limitations	Case 2, with NOx limitations	
Fuel consumption without cold start compensation	4.401	4.439	l/100 km
Fuel consumption with cold start compensation	4.695	4.736	l/100 km

The difference in fuel consumption between the two cases is surprisingly small. The torque limitation was expected to result in a much higher increase. It seems like the DP algorithm succeeds to find a solution quite near the optimum even when limitations are applied.

5.4 Comparison

In following sections are the results from the optimization of the two-mode HEV (Case 2 in Section 5.3) compared with data obtained from the prototype in a test cell. These data were provided by GM. Since the raw data is sampled with, at least compared to the optimized time resolution, relatively short sample time (12.5 ms) the data set was resampled to 1 s to make the comparison easier. Additional post-processing was made on the raw data to smoothen out the original's rather transient and spiky curves. The two Figures, 5.29 and 5.30 below show an example of data before and after it has been processed. One might notice that the data set has been shortened compared to its original. The reason for this is to improve the readability, since the first couple of seconds of the cycle is missing in the original recording. In the following figures in this chapter, if nothing else is mentioned, is the total vehicle power, $P_{vehicle}$, represented by a grey dashed curve.

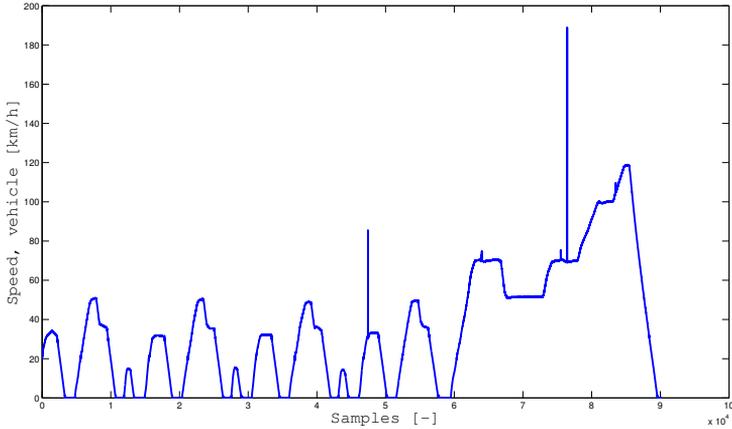


Figure 5.29. Vehicle speed before processing the data set.

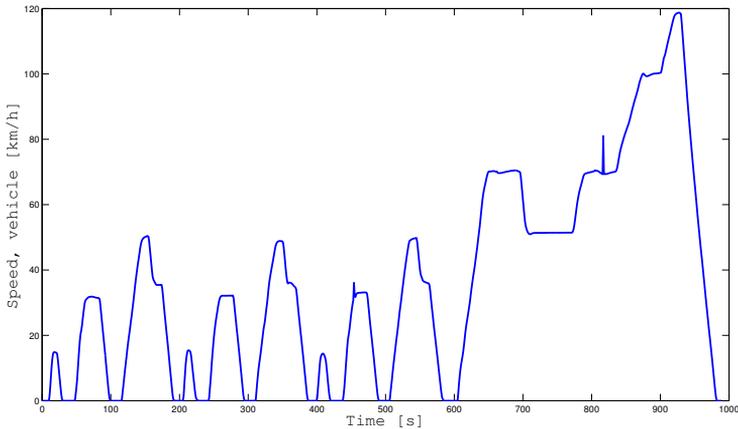


Figure 5.30. Vehicle speed after processing the data set.

5.4.1 Engine

First out is a figure of the measured engine power during the cycle. The general appearance looks very much alike to the one seen in Figures 5.14 and 5.15 in Section 5.3. The engine is turned on at the same parts of the cycle in both cases, i.e. during all speed sections except the smallest. However, both the duration and the engine output when the engine in fact is turned on differs from the optimal solution. While the controller in the real vehicle chooses to keep the engine on

during both the last parts of the acceleration phases and the steady state phases (see for instance $t = 130-155$ s in Figure 5.31), the optimal solution is to keep the engine on only during the end of the acceleration phases and then with a significant higher output. The output at $t = 710 - 770$ in Figure 5.31 also differs from the output seen for the optimal solution (at $t = 900 - 970$, seen in Figure 5.15) in the sense that the latter chooses a higher output than needed to propel the vehicle, and hence chooses to charge the battery with the excess power.

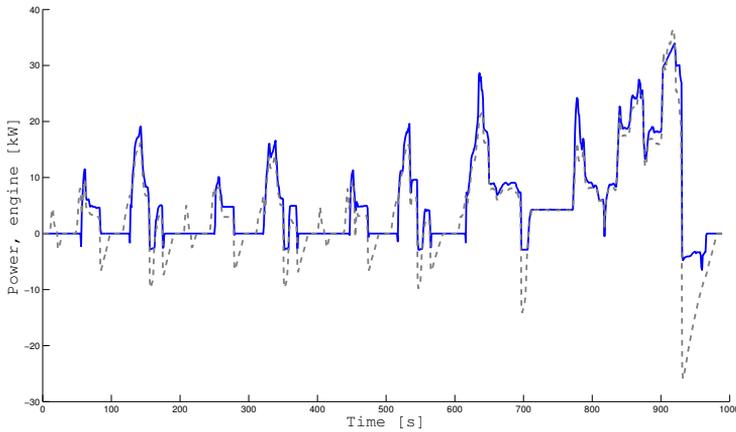


Figure 5.31. Measured engine power and vehicle power for the two-mode.

Figure 5.32 shows the engine torque during the cycle, where the difference compared to the optimal solution, seen in Figures 5.16 and 5.17, is quite clear. The torque peaks during the UDC parts are significantly lower and wider than the optimal ditto. This difference is not only expected but also very satisfying to see since the optimal solution should, according to Figure 4.10, keep the torque as high as possible. The EUDC part on the other hand is quite similar to the optimal solution, except that the torque during the steady state section (around $t = 710 - 770$) is below 40 Nm while the optimal path is to keep the torque above 60 Nm. This is the reason for the higher output mentioned in the previous section. There are also differences during the deceleration phases, where the engine in the prototype is set to fuel cut-off, which occurs at torques around -30 Nm, while the optimal solution keeps the torque at around or just below 0 Nm. However, this behavior seen in the optimal solution is most likely a result of the low resolution of the SoC grid.

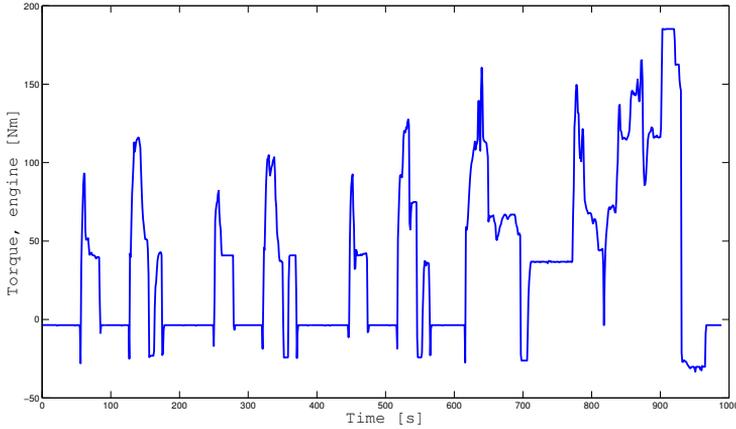


Figure 5.32. Measured engine torque for the two-mode.

Even though the engine speeds are not fully comparable because of the somewhat rough discretization of the N_e grid (see Table 5.4), it is still interesting to compare the general behavior. The engine speed in Figure 5.33 below, which is the prototype's, does not rev up as much as the optimal, at least not during the UDC parts, shown in Figure 5.18. As with the torques, the engine speeds show quite similar behavior when compared during the EUDC part.

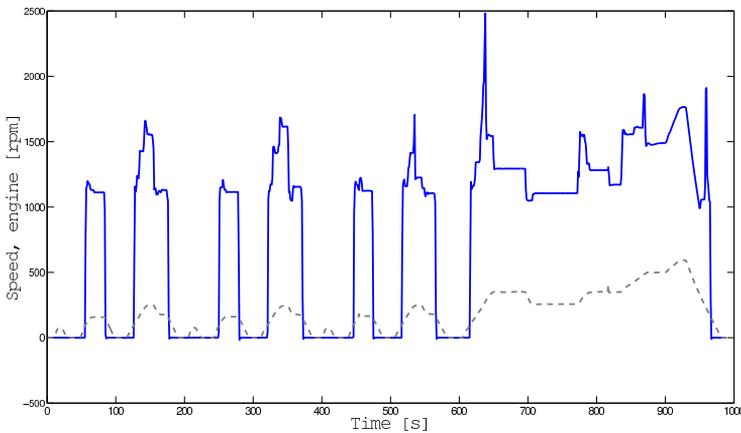


Figure 5.33. Measured engine speed for the two-mode.

The operating points for the prototype, seen in Figure 5.34, are much more scattered than the ones representing the optimal solution, seen in Figures 5.20 and

5.21. This is expected since the values for torque and speed are not descritized as for the optimized solution. Figure 5.34 shows resemblance with Figure 5.21 in the sense that no points are seen above 190 Nm, which is a result of the NOx limitation. What is more interesting is the cluster of dots present in the low efficiency region below the level at 60 Nm in figure 5.34. This cannot be seen in either Figure 5.20 or 5.21 and gives further proof in the matter discussed in the previous sections.

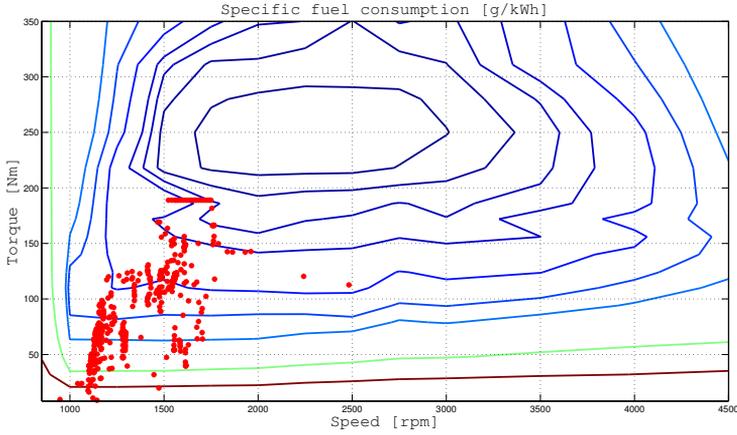


Figure 5.34. Specific fuel consumption with operation points for the prototype.

5.4.2 Battery

The SoC, seen in Figure 5.35, has a very digital behavior and does not share many similarities with the optimal shown in Figure 5.22. There is however a big difference in how the SoC is calculated for the two cases which has a significant impact on the results. Since the SoC is a state variable for the optimization, a change in SoC has a direct impact on the battery power and hence a direct impact on the motors/generators which in its turn, if the grid is tight enough, results in a smooth curve. On the other hand, since SoC in real life is not a variable that is easy to measure, but is instead estimated from the battery current and voltage, the accuracy is poor. It is therefore no need to have such a high resolution on the SoC value, which clearly can be seen in the figure below. Even so, the general behavior can be further investigated. The most remarkable, compared to the optimal solution, is the short SoC span and the small SoC recuperation during the last deceleration. A brief discussion in this matter is made in Section 5.1.2.

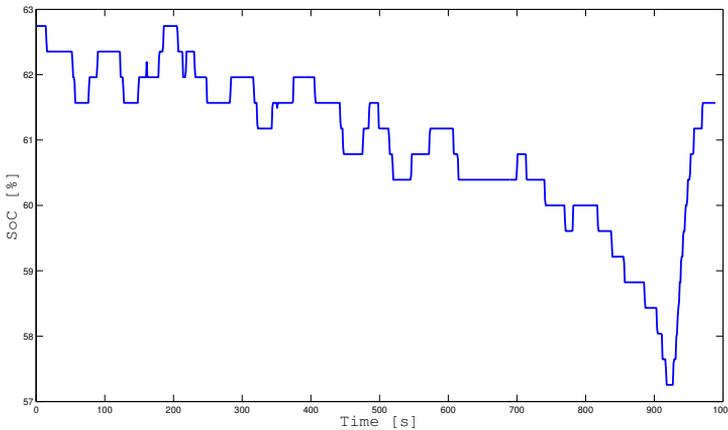


Figure 5.35. Measured SoC for the two-mode.

5.4.3 Motor/generator

As with the other figures shown in this chapter, shares the motor/generator_A power in Figure 5.36 below more similarities with the optimal power output, in the EUDC part, seen in Figure 5.26 than the UDC parts, shown in Figure 5.25. In fact, the behavior during the last two sections in the UDC parts is quite the opposite. While the optimized power output peaks in the last few seconds of the acceleration (seen for instance at $t = 337$ s) is the duration of the measured ditto longer as well as lower. In the EUDC part however is the behavior much the same. But it is noticeable that the deceleration is managed all together by motor/generator_A when optimized (seen for instance at $t = 895$ s) while some of

the braking is performed by the engine in the prototype (seen at $t = 698$ s). This is in line with what is stated in Section 5.4.1.

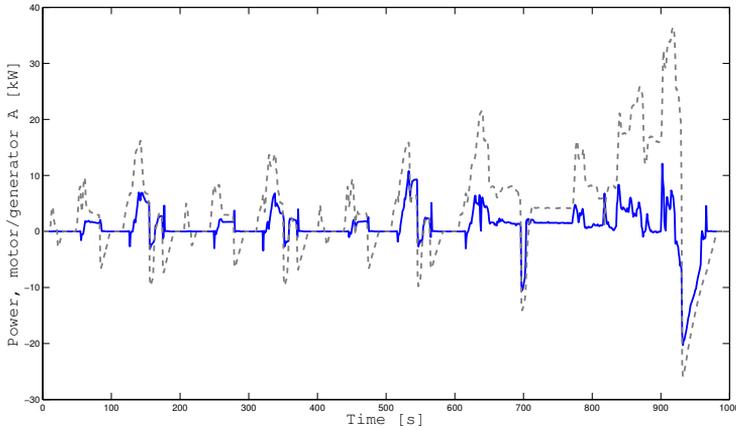


Figure 5.36. Vehicle power and measured power from motor/generator_A.

The vehicle is propelled entirely by motor/generator_B during the lowest speed section in the UDC parts, seen for instance at $t = 10 - 30$ s in Figure 5.37. This can also be seen in Figure 5.27, which shows the optimal power output. The two following, higher speed sections, are treated a bit different for the two cases. The first seconds of acceleration, until the vehicle reaches around 20 km/h, are handled entirely by motor/generator_B until the engine kicks in and handles the rest of the acceleration as well as charging the battery (the power for motor/generator_B turns negative, seen at $t = 127$ s). This behavior is also seen for the optimal solution at $t = 128$. The engine is then continuing the propulsion during the following low speed coasting for the prototype until turned off during the deceleration. In the optimal solution however, is the engine turned off directly after the acceleration phase is completed, and the following coasting is again powered by motor/generator_B. There is also a noteworthy different when it comes to the magnitude of the output. The negative values retrieved from the optimization are about twice as large as the corresponding values for the prototype. This is mostly due to the higher engine output discussed in Section 5.4.1.

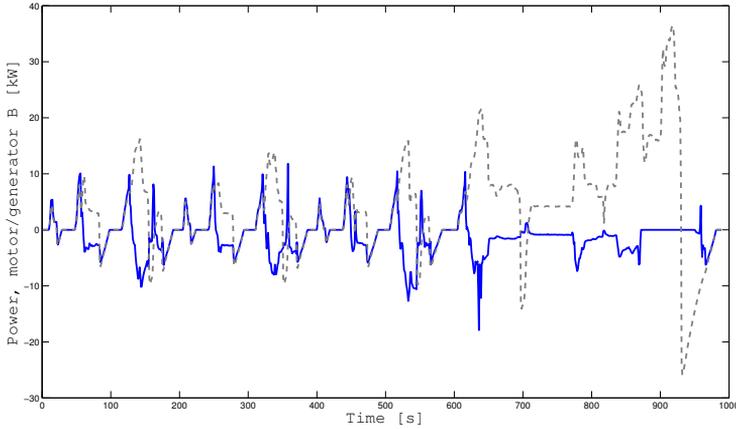


Figure 5.37. Vehicle power and measured power from motor/generator_B.

5.4.4 Fuel consumption

Table 5.7 shows the fuel consumption for the two cases as well as the five measured consumptions from the test cell. Comparing Case 1 with the average measured consumption shows a 9.7 % reduction of fuel utilization. The consumption for Case 2, which in this context is the most reasonable to compare with because of the NO_x limitation, is 8.9 % lower than the average measured consumption. This is an immense improvement, especially since the model does not take the gears present in the actual vehicle into account. The shift between first and second mode, which in this thesis is chosen at an arbitrary point in the cycle, is also plays a major role in the results from the optimization.

Table 5.7. Measured and optimized fuel consumption for the two-mode HEV.

Origin of data	GM test cell						Optimized data	
Data set	A	B	C	D	E	Average	Case 1	Case 2
Fuel consumption (l/100 km)	4.98	5.28	5.04	5.44	5.25	5.20	4.69	4.74

5.4.5 Summary

In the previous sections is the behavior of the two-mode HEV analyzed and compared to the optimal solution. The most significant differences seen in the optimal solution, compared to the prototype vehicle, are summed up in the following list

1. The duration when the engine is turned on is shorter and only during acceleration phases.
2. The power output when the engine is on is of greater magnitude.
3. The higher engine output is mainly achieved by higher torque.
4. Motor/generator_B is used to propel the vehicle during all coasting sections in the UDC part.
5. Motor/generator_A is responsible for all deceleration during the EUDC part, which is otherwise handled by the engine and motor/generator_A together.
6. A greater SoC span is being used and a larger percentage of the SoC is recuperated at the last deceleration of the cycle.

The first four statements are reasonable measures to improve fuel economy. Statement five is in the grey zone of either being a result of shortcomings in the discretization of the grids, or a fuel saving measure. Statement six is most likely a result of inaccurate modeling of the battery.

Chapter 6

Speed improvements

As mentioned earlier in this work, optimization in general, and especially optimizing using dynamic programming is rather time consuming. The computational burden using DP algorithms scales linear with the problem time, which in this case means linear with the length of the driving cycle. Unfortunately, DP algorithms have a complexity which is exponential increasing with the number of states and controls. This makes this algorithm more suitable for low-order systems (i.e. parallel hybrid). Even when optimizing a series hybrid, which is a two-state problem, the computational time using a standard desktop computer can become unpractical when high resolution is required. Guzzella addresses this problem and outlines some tricks to reduce the optimization time. The first, and the one that Guzzella claims has the biggest potential, is to rewrite the algorithm from low-order language using for-loops, to high order code implementing vectorized functions. MATLAB is known for its poor performance using for-loops, but is on the other hand built for, and supporting vectorization. The average computation time ratio, shown in the literature, between scalar and set implementation obtained with different state and input grids is about 1000 times. A larger grid size causes a better computational time ratio, according to Guzzella. This approach is analyzed in the next section. The second hint, which is more general, regards the general structure of the code. Regardless of the programming language used, different functions fulfilling the same task can be more or less computational efficient depending of the use of the sub-functions, while-loops, the already mentioned for-loops etc. An example of this, which will be discussed further in the following chapter, is the use of interpolation functions.

6.1 Vectorization

To gain confidence in using vectorized programming and to see whether vectorization results in the time reduction expected the first case which received a "vector makeover" was the one-dimensional DP problem which is used for the parallel hybrid. Vectorization, while shortening the optimization time, also tends to lead to a more abstract code. While an experienced programmer might find the code

cleaner and easier to overview, an inexperienced ditto might find it, which was confirmed during this work, more illogical and more difficult to trouble-shoot.

To help analyzing speed tests and the performance of MATLAB-code, MATLAB provides a built-in graphical user interface called Profiler. This tool showed to be very useful when comparing different methods and approaches. When comparing vectorized programming versus scalar two initial tests were performed using Profiler. Both tests were conducted on a 195-second long, one-dimensional DP problem with a 3000 wide SoC-grid. The result did not come out as expected. As seen in 6.1 below, the scalar test was actually faster, even though one can see that the sub-function `parallelHybrid` was called more than 582,000 times compared to only 194 (one per second) for the vectorized ditto.

<u>Function Name</u>	<u>Calls</u>	<u>Total Time</u>	<u>Self Time*</u>	Total Time Plot (dark band = self time)
<u>Hybrids</u>	1	597.386 s	0.124 s	
<u>dynProg1D_vec_v2</u>	1	596.753 s	231.310 s	
<u>parallelHybrid_vec</u>	194	336.812 s	336.812 s	
<u>repmat</u>	198	28.635 s	28.635 s	

<u>Function Name</u>	<u>Calls</u>	<u>Total Time</u>	<u>Self Time*</u>	Total Time Plot (dark band = self time)
<u>Hybrids</u>	1	490.714 s	0.102 s	
<u>dynProg1D</u>	1	490.090 s	218.302 s	
<u>parallelHybrid</u>	582194	271.788 s	271.788 s	

Figure 6.1. Analysis of speed performance with MATLAB's Profiler.

An additional two sets of tests were conducted, now with reduced grid sizes. A 1500 wide grid, that is half the size of the grid compared to the previous test, resulted in a 7 % speed improvement for the vectorized version. Yet another reduction, from a 1500 to a 750 wide grid, which is a quarter of the size of the initial test grid, resulted in a 13 % speed improvement. The results from these three tests are summarized in Table 6.1 below.

Table 6.1. Running time for a scalar- contra a vectorized system.

Grid size	Running time, scalar	Running time, vectorized	Time ratio
3000	490.714 s	597.386 s	- 18 %
1500	183.412 s	171.646 s	7 %
750	48.137 s	42.583 s	13 %

In comparison to Guzzella's tests, these tests show a behavior that is quite the opposite. That is, smaller grid sizes lead to a smaller computational time reduction when using vectorized implementation. This is not only unexpected but also unwanted, since a high resolution optimization will inevitably lead to large grid sizes. A plausible explanation for this behavior is as follows: Guzzella compares computational speed between "pure" scalar implementation and vectorized implementation. In plain talk this means that (remember, this is an assumption) the comparison is made between a DP-algorithm that sends one, single scalar value to the subfunction where the hybrid model is, and one that sends a set of values (a vector or a matrix) to the model function. In the comparison shown in Table 6.1 the initial, "scalar" case is actually vectorized, since the DP-algorithm sends a $1 \times n$ -matrix (a vector) to the model function while in the "vectorized" case the DP-algorithm send an $m \times n$ -matrix. Unfortunately, further implementation of vectorized programming, that is for two dimensional problems had to be aborted for two reasons. The first is that MATLAB, even though especially designed to effectively handle matrices, does not seem to cope with the large matrices that vectorizing of a two dimensional problem requires (not to mention a three dimensional). The second reason is the one somewhat feared and mentioned earlier in this section, which is the increasing complexity and abstractness when dealing with multidimensional matrices. Some of MATLAB's standard functions do not apply for these matrices which creates a need for specially designed dittos. Troubleshooting, as mentioned in one previous paragraph, also becomes impractical.

6.2 Interpolation

Since the MATLAB built in function Profiler showed that the main part of the optimization time was spent on interpolating the torques and speeds to achieve the correct efficiencies, a logical step was to investigate how the interpolation scripts works and whether or not "tricks" could be made to speed them up.

6.2.1 Adjusting the fuel map

According to MATLAB's help manual, large speed improvements can be made by forcing the `interp2`-function to use linear interpolation (other options are "nearest", "cubic" and "spline" interpolation). This on the other hand requires that the

vectors used are monotonic and equally spaced. The first requirement was already fulfilled, but the second one needed to be addressed. When this was done, by simply manually interpolate the vectors to get an equally spaced map, a simple speed test was conducted by running a for-loop 7000 times interpolating a single scalar. This test was successful and reduced the time from 3.36 seconds (normal interp2) to 1.36 seconds (forced linear method). To verify this possible improvement another set of Profiler was run. The outcome was unfortunately a disappointment, reducing the total optimization time by only 5 percent. It seemed like the large speed improvement when using scalars was lost when implemented and hence using matrices. A short test by running a for-loop 1000 times verified that (from 39.55 seconds to 39.37 seconds).

6.2.2 Implementing C-code

MATLAB also has a built-in function called mex, which compiles C/C++ or Fortran source code to binary code which then can be used and treated as any other M-file. In this case the MATLAB's interp2 was exchanged for an equivalent mex function based on C-code. Except for some minor implementation difficulties the result was positive with a reduced optimization time.

Apart from rewriting some sub-functions to C-code there is also, logically, a possibility to rewrite the entire MATLAB code used in this thesis to C-code. Such a change in programming language will result in a much faster optimization time, while the simplicity and the straightforward trouble-shooting associated with the MATLAB language will, to some extent, be lost. Unfortunately, the time available for this thesis limited the possibilities to implement C-code to only involve the mentioned mex-functions.

6.2.3 Polynomial function

To further reduce optimization time, the one-dimensional interpolation which handles the battery losses was replaced by a second degree polynomial function. The coefficients were approximated using MATLAB's polyfit. This has been implemented in battery model, see 4.3

6.3 Improving the MATLAB code

To shorten the optimization time by improving the code two different approaches were addressed, on top of just cleaning up unnecessary code. One was to reduce the number of functions called during an optimization. Since much data is known beforehand, a large amount of the code can be transferred out of the function which is called several thousand times per second and put in the initiation code, which is just called one time. The second approach was to avoid performing calculations on data that is irrelevant. An example showing the engine torque before it is interpolated is seen in Figure 6.2, where the relevant data is marked with green color. Given that the maximum torque this engine can deliver is 350 Nm all values above that are irrelevant. The most left column represents the engine torque when

the engine is turned off, and because the engine does not deliver any power at zero rpm there is no need to interpolate these data. And since the figure in this example only illustrates a small part of the matrix, which in its whole contains 981 rows and where all rows above 47 are irrelevant, it is easy to understand that eliminating these excessive calculations can save a great amount of time.

	1	2	3	4	5	6	7	8
9	Inf	104.1215	90.4289	80.0332	72.5450	66.6445	61.7682	55.2239
10	Inf	115.9691	100.4232	88.7101	80.2388	73.5691	68.0751	60.7255
11	Inf	128.0949	110.6152	97.5406	88.0593	80.6022	74.4781	66.3085
12	Inf	140.5115	121.0116	106.5287	96.0094	87.7462	80.9789	71.9742
13	Inf	153.2325	131.6192	115.6787	104.0921	96.0033	87.5794	77.7243
14	Inf	166.2724	142.4455	124.9952	112.3105	102.3759	94.2816	83.5602
15	Inf	179.6475	153.4985	134.4831	120.6681	109.8666	101.0876	89.4836
16	Inf	193.3754	164.7866	144.1473	129.1682	117.4781	107.9995	95.4961
17	Inf	207.4755	176.3190	153.9932	137.8145	125.2131	115.0196	101.5995
18	Inf	221.9694	188.1057	164.0267	146.6110	133.0743	122.1501	107.7955
19	Inf	236.8808	200.1570	174.2536	155.5617	141.0651	129.3936	114.0862
20	Inf	252.2365	212.4847	184.6805	164.6708	149.1884	136.7526	120.4734
21	Inf	268.0662	225.1011	195.3143	173.9431	157.4475	144.2297	126.9593
22	Inf	284.4034	238.0198	206.1623	183.3831	165.8461	151.8278	133.5460
23	Inf	301.2864	251.2553	217.2324	192.9961	174.3879	159.5499	140.2366
24	Inf	318.7585	264.8240	228.5329	202.7874	183.0767	167.3989	147.0307
25	Inf	336.8698	278.7433	240.0731	212.7627	191.9166	175.3783	153.9337
26	Inf	355.6782	293.0329	251.8624	222.9282	200.9119	183.4913	160.9470
27	Inf	375.2517	307.7141	263.9115	233.2903	210.0674	191.7416	168.0737
28	Inf	395.6706	322.8107	276.2318	243.8560	219.3878	200.1331	175.3163
29	Inf	417.0314	338.3494	288.8353	254.6326	228.8782	208.6696	182.6780
30	Inf	439.4511	354.3599	301.7356	265.6279	238.5442	217.3554	190.1619
31	Inf	463.0747	370.8757	314.9472	276.8506	248.3915	226.1951	197.7712
32	Inf	488.0850	387.9350	328.4860	288.3099	258.4264	235.1932	205.5096
33	Inf	514.7188	405.5809	342.3694	300.0153	268.6554	244.3550	213.3808
34	Inf	543.2911	423.8636	356.6166	311.9777	279.0856	253.6856	221.3886
35	Inf	574.2388	442.8408	371.2489	324.2084	289.7245	263.1909	229.5370
36	Inf	608.1981	462.5801	386.2897	336.7200	300.5801	272.8767	237.8307
37	Inf	646.1627	483.1619	401.7653	349.5259	311.6613	282.7497	246.2742
38	Inf	689.8506	504.6819	417.7051	362.6409	322.9774	292.8167	254.8724
39	Inf	742.7756	527.2570	434.1423	376.0813	334.5385	303.0850	263.6305
40	Inf	815.0894	551.0313	451.1144	389.8647	346.3553	313.5627	272.5543

Figure 6.2. Torque matrix. Relevant elements within the torque limit are marked with green.

6.4 Summary

Unfortunately, not all ideas of how to speed up the optimization proved to be successful - some even resulted in a longer optimization time and lead to a more complex and abstract code. Of the ones discussed in the preceding sections only a few were actually implemented and used in the final code. All one-dimensional interpolation were replaced by polynomial functions were applicable. The two-dimensional interpolation was substituted with faster C-code and the additional improvements of the MATLAB code, mentioned in Section 6.3, were performed on all scripts.

Chapter 7

Conclusions and future work

This thesis presents suggestions where attention should be focused to improve the control unit in the two-mode HEV with the aim to reduce its fuel utilization. The proposed measures are based on detailed modeling of the vehicle, its propulsion units and remaining involved components followed by quasi static optimization over the whole NEDC cycle. Initial work with the parallel and serial HEV gave valuable insight in the subsequent implementing procedure of the power-split, as well as gained confidence in the results of the Dynamic Programming algorithm and in the models used.

7.1 Summary of results

The main conclusion is that the controller used in the prototype is not optimal when it comes to minimize the two-mode HEV's fuel consumption, but rather that there is potential to improve utilization. The result from the optimization on the NEDC cycle showed on an average a 8.9 percent improvement compared to the prototype vehicle.

7.1.1 Suggested modifications

The outcome of the two-mode optimization were analyzed and compared to data obtained from the prototype in Chapter 5 and summed up in a list in Section 5.4.5 Based on these results the following conclusions, with the aim to trim down the fuel consumption, are drawn:

- The engine should be kept turned off as often as possible, when possible. That is while in first mode.
- The power output from the engine, when turned on, should be as high as possible. The high output should be achieved by keeping the engine in its most efficient operating area, by primarily keeping the torque high. The excessive power not used to propel the vehicle should be handled by the motor/generator_B to charge the battery.

- Low speed acceleration and low speed cruising should be powered solely electro-mechanically.
- The high speed acceleration and high speed cruising should be powered allmechanically.

Whether the statements above are fully implementable in the real vehicle or if such an implementation will result in the anticipated fuel savings seen in theory or not, is not realistic to investigate during the time frame of this thesis, but rather the natural next step for further work. For instance must trade-offs for noise and vibration be done for a final optimization (apart for NOx balancing). Such trade-offs can be the reason that in the two-mode vehicle, the ICE is not run at so high loads and short durations as the optimal solution suggests.

7.1.2 Additional results

Apart from the main purpose of this thesis, the preparatory work have resulted in fully working MATLAB scripts, which by exchanging the values for weight, gear ratios, engine maps and so on can be used to optimize the fuel consumption for any arbitrary serial or parallel HEV. Much work was also put into speeding up the optimization algorithm, seen in Section 6.3, which indeed gave result - not only beneficial to this thesis but as well to the future work.

7.2 Future Works

Several actions can be done to improve the model, making it more accurate and realistic. However, two main topics are more important than the others: the adding of a third state and improvement of the deterministic dynamic programming algorithm. To fully represent the real two-mode and its combined power split device, an additional state would be necessary. This third state would realize the optimal change between the two modes as well as the four gears. This can be compared to the implementation in this thesis, where a speed limit in the EUDC part of the cycle was used to switch between the two modes, without any involvements of the gears in mode two. To realize this third state and implement it to the model, an improvement of the deterministic dynamic programming algorithm would be necessary. Otherwise the simulation time would become unrealistic, and a reasonable resolution would not be achievable. While analyzing the optimization results and the behavior of the components in the HEVs, phenomenon which were hard to explain could be seen for the SoC and the battery power. Hence a more profound analysis and further improvement of the battery model would be of interest to gain a more realistic model. Other parts where focus should be put is how the temperature affects the modeled parts, as well as the time duration for switching between modes and gears. Another interesting aspect to study is the two-mode HEVs drivability while using the optimized solution. No consideration has been made to this matter during the thesis, but is a topic that surely will have an impact on the final result.

Bibliography

- [1] The history of the automobile. <http://inventors.about.com/library/weekly/aacarsgasa.htm>, viewed 2009-04-20.
- [2] Toyota news release. <http://www.toyota.co.jp/en/news/08/0515.html>, viewed 2009-04-19, 2008-05-15.
- [3] R.E Bellman. *Dynamic programming*. Princeton University Press, Princeton, NJ, 1957.
- [4] R.E Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, Princeton, NJ, 1961.
- [5] K.T. CHAU. *MODERN ELECTRIC VEHICLE TECHNOLOGY*. Oxford Science Publications, 2001. ISBN 0198504160.
- [6] Mehrdad Ehsani, Yimin Gao, Sebastien E. Gay T, and Ali Emadi. *Modern Electric, Hybrid Electric, and Fuel Cell Vehicles*. CRC Press LLC, 2005. ISBN 0-8493-3154-4.
- [7] Lino Guzzella and Antonio Sciarretta. *Vehicle Propulsion Systems*. Springer, 2 edition, 2007. ISBN 978-3-540-74691-1.
- [8] Ecopoint Inc. Dieselnet. http://www.dieselnet.com/standards/cycles/ece_eudc.html, viewed 2009-04-19.
- [9] Ernest Henry Wakefield. *History of the Electric Automobile: Hybrid Electric Vehicles*. SAE, 1998. ISBN 0768001250, 9780768001259.

Appendix A

Abbreviations

DP - Dynamic Programming

EM - Electric Motor

ICE - Internal Combustion Engine

HEV - Hybrid Electric Vehicle

PSD - Power Split Device

SoC - State of Charge

RPM - Revolutions Per Minute

NEDC - The New European Driving Cycle

EUDC - Extra Urban Driving Cycle

UDC - Urban Driving Cycle

SFC - Specific Fuel Consumption

EVT - Electrically Variable Transmission

Appendix B

MATLAB code parallel

```
function [T_motor_mech,T_e,costVector]=parallelHybrid(t_vec,SoC_start,SoC_final)
% Function for calculating all the arc-costs during one time interval,
% from one node to all other possible nodes.
%
%
% Inputs:
% t_vec      - 1x2 matrix, with the start and stop time for the interval.
% SoC_start  - A single start value for SoC during the interval.
% SoC_final  - Vector with all possible final values for the interval.
%
% Output:
% T_motor_mech  - Matrix with calculated mechanical motor torque
% T_e           - Matrix with calculated engine torque
% costVector    - Matrix with the costs for all arcs from SoC_start
%
%
%% Global variables
global P_vehicle;
global w_e;
global g_per_s
global rpm
global torque
global gamma
global A1
global A2
global C
global N_e
global w_m
```

```

%% Calculation of cost matrix, engine torque and mechanical motor torque,
% under the constrain v = 0 (vehicle velocity equals zero).
if gamma(t_vec(1)) == 0
    costVector = inf*ones(size(SoC_final));
    costVector(SoC_final==SoC_start) = 0;
    T_e = zeros(size(SoC_final));
    T_motor_mech = zeros(size(SoC_final));
else

%% Constants
% Nominal battery capacity
Q_0 = 6.5*3600; %[As]
%Open circuit voltage, used in calculation of the battery's
%electrical power flow.
U_oc = 300;      %[V]
%Estimated constants, used in the second degree polynomial,
%that determine P_batt.
A_batt = -0.004927141444030;
B_batt = 1.018372550083824;
%Auxiliary load;
P_dc = 0.0;     %[kW];

%% Battery
%Calculation of battery current [A]
I = (SoC_start-SoC_final)*Q_0;
% Electric power flow from battery [kW]
P_batt_el = I*U_oc/1000;
%Mechanical power flow from battery [kW]
P_batt = (A_batt*(P_batt_el).^2 +B_batt*(P_batt_el));

%% Calculation of the engine torque matrix
%Calculation of mechanical motor torque
T_motor_mech = (sqrt((P_batt - C(t_vec(1)) + P_dc)) - ...
A2(t_vec(1)))./A1(t_vec(1));

%Finding complex numbers
Complex = imag(T_motor_mech);

T_motor_mech(Complex > 0) = inf;
T_motor_mech(P_batt == 0) = 0;

%Calculation of the electric power
P_motor_mech = T_motor_mech.*w_m(t_vec(1));

%Calculation of required engine power
P_engine = P_vehicle(t_vec(1))-P_motor_mech;

%Calculation of engine torque
T_e = P_engine./w_e(t_vec(1));

```

```
%% Calculation of cost matrix
T_e = double(T_e);
N_e_spec=ones(size(P_batt))*N_e(t_vec(1));

%Initiating of the cost matrix
cost = ones(size(P_batt))*inf;

%Linear interpolation of torque- and speed matrix, with boundaries.
cost(T_e ≥ -50 & T_e ≤ 350) = ...
lininterp2f_uneq(rpm,torque,g_per_s, ...
N_e_spec(T_e ≥ -50 & T_e ≤ 350),T_e(T_e ≥ -50 & T_e ≤ 350));

%% Matrix with the costs for all arcs from SoC_start
costVector = cost;

end
    end
```

Appendix C

MATLAB code serial

```
function [T_ice,costVector]=seriesHybrid(t_vec,SoC_start,SoC_final,Ne_start,Ne_final)
% Function for calculating all the arc-costs during one time interval,
% from one node to all other possible nodes.
%
%
% Inputs:
% t_vec      - 1x2 matrix, with the start and stop time for the interval.
% SoC_start  - A single start value for SoC during the interval.
% SoC_final  - Vector with all possible final values for the interval.
% Ne_start   - A single start value for engine speed during the interval.
% Ne_final   - Vector with all possible final values for the interval.
%
% Output:
% T_ice      - Matrix with calculated engine torque
% costVector - Matrix with the costs for all arcs from SoC_start
%
%
%% Global variables
global g_per_s;
global torque;
global rpm;
global A1
global A2
global Ca
global B1
global B2
global Cb
global T_b_mech
global N_e
```

```

%% Constants
% Constant used to estimate the engines inertia
J_e = 0.1323; % [kgm2]
% Nominal battery capacity
Q_0 = 6.5*3600; %[As]
%Open circuit voltage, used in calculation of the battery's
%electrical power flow.
U_oc = 300; %[V]
%Estimated constants, used in the second degree polynomial,
%that determine P_batt.
A_batt = -0.004927141444030;
B_batt = 1.018372550083824;
%Auxiliary load;
P_dc = 0.6; %[kW];

%% Battery
%Calculation of battery current [A]
I = (SoC_start-SoC_final)*Q_0;
% Electric power flow from battery [kW]
P_batt_el = I*U_oc/1000;
%Mechanical power flow from battery [kW]
P_batt = (A_batt*(P_batt_el).^2 +B_batt*(P_batt_el));
%Resizing of matrix
P_batt = P_batt *ones(size(Ne_final));

%% Motor/generator
%Initiating and Resizing of T_b_mech_spec used in calculation of T_a_mech
T_b_mech_spec = T_b_mech(t_vec(1,1),:);
T_b_mech_spec = ones(length(P_batt),1)*T_b_mech_spec;

%Losses, used in calculation of T_a_mech
C_sum = (Cb + Ca);

temp = sqrt(-((ones(length(P_batt),1)*B1(t_vec(1,1),:)).*T_b_mech_spec+...
ones(length(P_batt),1)*B2(t_vec(1,1),:)).^2 - P_batt + ...
ones(length(P_batt),1)*C_sum(t_vec(1,1),: ) + P_dc));

%Calculation of the mechanical torque required from motor A
T_a_mech = (temp-ones(length(P_batt),1)*A2(t_vec(1,1),:))./ ...
(ones(length(P_batt),1)*A1(t_vec(1,1),:));

%Finding complex numbers and erase them from the solution by
%giving them the value of inf.
T_a_mech(imag(T_a_mech) ≠ 0) = inf;

```

```

%% ICE
%Calculation of angular acceleration for the engine
w_e_dot = ones(size(SoC_final)) * (Ne_final-Ne_start)*2*pi/60;

%Engine inertia
T_Je = (J_e*w_e_dot);

%Calculation of engine torque
T_ice = -T_a_mech + T_Je;
T_ice = double(T_ice);

%% Calculation of cost matrix
N_e_spec = N_e(t_vec(1,1),:);
N_e_spec = ones(length(P_batt),1)*N_e_spec;
T_ice_spec = T_ice(1:end);
N_e_spec = N_e_spec(1:end);

%Initiating of the cost matrix
cost = ones(size(P_batt))*inf;

%Linear interpolation of torque- and speed matrix, with boundaries.
cost(T_ice_spec >= -52.3 & T_ice_spec<=350 & N_e_spec <= 0) = ...
lininterp2f_uneq(rpm,torque,g_per_s, ...
N_e_spec(T_ice_spec >= -52.3 & T_ice_spec<=350 & N_e_spec <= 0), ...
T_ice_spec(T_ice_spec >= -52.3 & T_ice_spec<=350 & N_e_spec <= 0));

%% Calculation of SOC_final when w_e = 0.
constant = (B1(t_vec(1,1),1)*T_b_mech(t_vec(1,1),1) + ...
B2(t_vec(1,1),1)).^2 + Cb(t_vec(1,1),1) + P_dc;

I_0 = -sqrt(constant/(A_batt*U_oc^2) + ...
(B_batt*U_oc)/(2*A_batt*U_oc^2))^2 - (B_batt*U_oc)/(2*A_batt*U_oc^2);

%I_0 in kW
I_0 = I_0*1000;
SoC_final_0 = SoC_start - (I_0/Q_0) ;

%Creation of cost matrix and introducing of constraints when w_e = 0
if SoC_final_0>SoC_final(1,1) && SoC_final_0<SoC_final(1,end)
    temporary = SoC_final(1:end)-SoC_final_0;
    temporary(temporary > 0) = -10;
    [value,index]=max(temporary);
    cost(index,1) = 0;
    T_ice(index,1) = 0;
end

%% Matrix with the costs for all arcs from SoC_start
costVector = cost;

end

```

Appendix D

Driving cycle

The New European driving cycle or NEDC is a cycle that represents the typical usage of cars in Europe. It is used by car manufactures to measure fuel consumption and as an emission certification for light duty vehicles in Europe [8].

The NEDC is a combined cycle, consisting of four UDC, or ECE-15 cycles followed by an EUDC, also called Extra Urban driving cycle. The ECE-15 parts represent urban driving, which is characterized by low engine load and low driving speed. While the EUDC represent sub-urban driving with a maximum speed at 120 km/h, compared to ECE-15 that has a maximum speed at 50 km/h. Figure D.1 shows both speed- and gear profile for the whole NEDC (four UDC followed by an EUDC). The gear profile is used when testing the Parallel HEV, where not only the speed profile must be kept but also the gear profile.

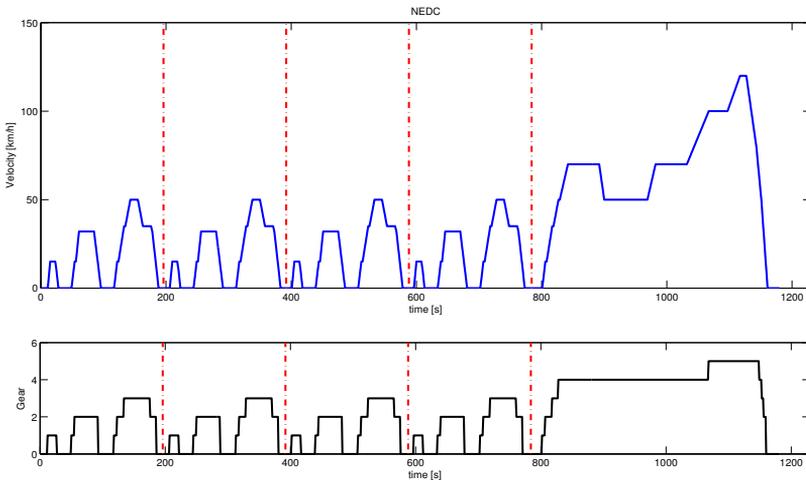


Figure D.1. Speed- and gear profile for the NEDC.