Master of Science Thesis in Mechanical Engineering
Department of Electrical Engineering, Linköping University, 2021

Predictive Energy Management of Long-Haul Hybrid Trucks

Using Quadratic Programming and Branch-and-Bound

Erik Jonsson Holm

LINKÖPING UNIVERSITY

Master of Science Thesis in Mechanical Engineering

Predictive Energy Management of Long-Haul Hybrid Trucks: Using Quadratic Programming and Branch-and-Bound

Erik Jonsson Holm

LiTH-ISY-EX--21/5424--SE

Supervisor: **Iman Shafikhani**
ISY, Linköpings universitet
Esteban Gelso
Volvo Group

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

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

Abstract

This thesis presents a predictive energy management controller for long-haul hybrid trucks. In a receding horizon control framework, the vehicle speed reference, battery energy reference, and engine on/off decision are optimized over a prediction horizon. A mixed-integer quadratic program (MIQP) is formulated by performing modelling approximations and by including the binary engine on/off decision in the optimal control problem. The branch-and-bound algorithm is applied to solve this problem. Simulation results show fuel consumption reductions between 10-15%, depending on driving cycle, compared to a conventional truck. The hybrid truck without the predictive control saves significantly less. Fuel consumption is reduced by 3-8% in this case. A sensitivity analysis studies the effects on branch-and-bound iterations and fuel consumption when varying parameters related to the binary engine on/off decision. In addition, it is shown that the control strategy can maintain a safe time gap to a leading vehicle. Also, the introduction of the battery temperature state makes it possible to approximately model the dynamic battery power limitations over the prediction horizon. The main contributions of the thesis are the MIQP control problem formulation, the strategy to solve this with the branch-and-bound method, and the sensitivity analysis.

Acknowledgments

I would like to express gratitude to my supervisor at Volvo Group, Esteban Gelso, and at Linköping University, Iman Shafikhani, and examiner Associate Professor Jan Åslund for all their feedback and insightful remarks in our meetings. This really improved the end result of this thesis and made the work more interesting.

Erik Jonsson Holm

Contents

Notation

Nomenclature

Notation	Meaning
F_p	Propulsion force
F_d	Drag force
F_E	Wheel force delivered from the engine
$F_{E,1}$	Wheel force delivered from the engine at top gear
$F_{E,2}$	Wheel force delivered from the engine at lower gears
F_M	Wheel force delivered from the electric machine force
F_{brk}	Brake force
P_{Emax}	Maximum engine power
P_{Mmax}	Maximum electric machine power
P_{Mmin}	Minimum electric machine power
F_g	Gravitational force
F_r	Rolling resistance force
F_a	Aerodynamic drag force
m	Vehicle mass
a	Vehicle acceleration
v	Vehicle speed
t	Time
C_{rr}	Rolling resistance coefficient
C_D	Aerodynamic drag coefficient
A_f	Vehicle frontal area
r_w	Wheel radius
ρ	Air density
α	Road slope
g	Gravitational acceleration
E_K	Vehicle kinetic energy
E_B	Battery energy
λ_B	Battery energy costate
η_D	Driveline efficiency
T_E	Engine torque
ω_E	Engine speed
T_M	Electric machine torque
ω_M	Electric machine speed
i	Current
R_0	Ohmic resistance
v_{oc}	Open circuit voltage
v_{bat}	Battery voltage

Nomenclature cont.

Notation	Meaning
γ	Fuel equivalent that converts a change in kinetic energy to an equivalent fuel mass
μ	Fuel flow
β	Fuel equivalent for time
N	Number of stages in optimization problem
Δs	Sampling distance
$P_{B,el}$	Battery electrical power
P_B	Battery chemical (internal) power
$P_{B,loss}$	Battery power losses
$P_{M,el}$	Electric machine electrical power
$P_{M,mech}$	Electric machine mechanical power
$P_{M,loss}$	Electric machine power losses
$P_{aux,el}$	Electrical auxiliary power
T_B	Battery average cell temperature
$T_{coolant}$	Battery coolant temperature
C_B	Battery heat capacity
H_B	Battery heat transfer coefficient
$F_{B,el}$	Battery electrical force
F_B	Battery chemical (internal) force
$F_{B,loss}$	Battery losses force
$F_{M,el}$	Electric machine electrical force
$F_{M,loss}$	Electric machine losses force
U	Upper bound
L	Lower bound

Abbreviations

Abbreviation	Meaning
DP	Dynamic programming
MPC	Model predictive control
RHC	Receding horizon control
QP	Quadratic programming
MIQP	Mixed-integer quadratic programming
MILP	Mixed-integer linear programming
BNB	Branch and bound
HEV	Hybrid electric vehicle
SOC	State of charge
SOP	State of power
GPS	Global positioning system

1

Introduction

To avoid the serious consequences of global warming, the Paris agreement aims at limiting the temperature increase to well below 2° C [3]. Heavy-duty vehicles, such as trucks and buses, are today responsible for about 25% of the CO_2 emissions from road transport, and for about 6% of the total CO_2 emissions in the EU [2]. Therefore, a recent EU regulation imposes target CO_2 reductions of 15% for 2025, and 30% for 2030 [2]. Moreover, since fuel cost for a diesel-fuelled long-haul vehicle as a share of the total cost of ownership amounts to approximately 30%, as presented in [21], there is not only an environmental incentive but also a cost incentive to reduce the fuel consumption.

Electrified propulsion is a key technology to reduce fuel consumption. The reasons for this include both the high efficiency of the electrical driveline and the possibility to regenerate energy with the electric machine. The hybrid electric vehicle (HEV) is a vehicle concept that includes both an electric propulsion system and a conventional internal combustion engine. To exploit the best vehicle efficiency and performance, control systems play a crucial role. However, due to its multiple energy sources, the control task of an HEV is more complex than that of conventional or fully electric powertrains. The control problem includes both continuous states, e.g., battery energy and vehicle speed, and a mix of continuous and discrete control signals, e.g., torque split, engine on/off decision and gear selection. With the on-board availability of road topography data and GPS technology, a prediction horizon can be utilized by the controller. While rule-based approaches may often lead to suboptimal performance and extensive calibration, model-based methods using optimal control have the potential of providing superior fuel efficiency with a limited calibration effort.

1.1 Purpose and problem formulation

The purpose of this thesis is to develop a predictive energy management controller for a long-haul truck with a hybrid electric powertrain. The problem is formulated as follows:

- Develop an algorithm that minimizes the fuel consumption of the vehicle by optimizing the vehicle speed reference, battery energy reference, and engine on/off decision over a prediction horizon.

1.2 Delimitations

Results are only produced using a simulation tool in a PC environment. No implementation in a real truck or in a production control unit is performed. Furthermore, model parameters such as vehicle mass and aerodynamic drag are assumed to be known or estimated with high accuracy. Only highway driving in the cruise control mode is considered, meaning that, e.g., low-speed driving is not studied.

1.3 Thesis outline

Chapter 2 presents related research in the field of predictive energy management and engine on/off control. Chapter 3 describes the selected control strategy and gives a theoretical background. Chapter 4 derives the models, which are used in the control problem formulation, presented in Chapter 5. Chapter 6 presents simulation results and performs a sensitivity analysis. Chapter 7 contains extensions to the control formulation, including a traffic scenario and the modelling of the battery temperature in order to capture the battery dynamic power limitations. Chapter 8 discusses the results and methods. Finally, Chapter 9 presents conclusions and suggestions for future work.

2

Related research

Predictive energy management for heavy-duty vehicles is a well-researched topic. In [9], a receding horizon controller based on dynamic programming (DP) is developed to control vehicle speed and gear for a conventional long-haul truck. By the use of fuel equivalents, a compromise between fuel consumption and trip time is achieved. Furthermore, in [13], a predictive control scheme is developed for hybrid long-haul trucks. Through a layered approach, the continuous variables, vehicle speed and battery energy, are optimized in a top layer, using quadratic programming (QP). In the next layer, the discrete variables for gear selection and engine on/off decision are optimized using DP, taking the optimal vehicle speed and battery energy references as input. The purpose of the scheme is to avoid solving a mixed-integer program. In [14], vehicle speed of the surrounding vehicles is also modelled and included in the predictive controller. In [8], a similar approach as in [13] is taken for a conventional truck. However, the engine on/off decision is included in the QP, leading to mixed-integer QP (MIQP). This is in turn solved with the use of a commercial solver with an inbuilt MIQP solver.

Moreover, optimal gear shifting for a passenger car is explored in [7] by solving a mixed-integer nonlinear problem using the branch-and-bound method. Using a heuristic branching strategy, the required nonlinear programming (NLP) evaluations are substantially reduced. In [24], a mixed-integer linear program (MILP) is formulated for a heavy-duty truck to optimize vehicle speed, gear selection and eco-roll (neutral gear) decision. Due to the solver performance, only a few binary variables could be achieved.

Based on convex optimization and Pontryagin's minimum principle, an engine on/off strategy is developed in [4]. Performing optimal sizing and energy management for a hybrid city bus, the engine on/off decision is made based on the computation of a threshold value, based on analytically derived conditions,

for the requested power. A power threshold, defined by a heuristic, is also used in [17], which also studies a hybrid city bus. The engine on/off profile is defined in a first loop. Then another loop selects the gear profile based on another heuristic and then optimizes the battery size by solving a semidefinite convex program until convergence is reached.

3

Control strategy and theoretical background

This chapter introduces the concept of receding horizon control and describes the functional architecture in which the developed algorithm resides. Also, it presents the theoretical background on convex optimization and mixed-integer programming, where the latter part focuses on the branch-and-bound algorithm.

3.1 Receding horizon control

Receding horizon control (RHC), also known as model predictive control (MPC), is a control strategy that solves an optimization problem repeatedly over a prediction horizon. The procedure, in the context of predictive energy management for long-haul trucks, is illustrated in Figure 3.1. The controller looks ahead a certain prediction horizon, in the order of 3-10 km, and solves the problem, discretized with a certain sample step at approximately 50-100 m. The solution to the optimization problem is used as set-points and requests for lower level controllers. An event, such as time or distance reaching a certain level, triggers an update of the controller and the procedure is repeated.

3.2 Hierarchical control strategy

Since solving the complete energy management problem in one optimization problem, meeting real-time requirements, would be too computationally expensive, a hierarchical control strategy is applied, see Figure 3.2. This is similar to the control strategy presented in [13] and primarily differs regarding the engine on/off modelling. Referring to the functional architecture in Figure 3.2, the top layer performs mission management over the complete transport mission and may provide the vehicle with information such as vehicle speed and battery state

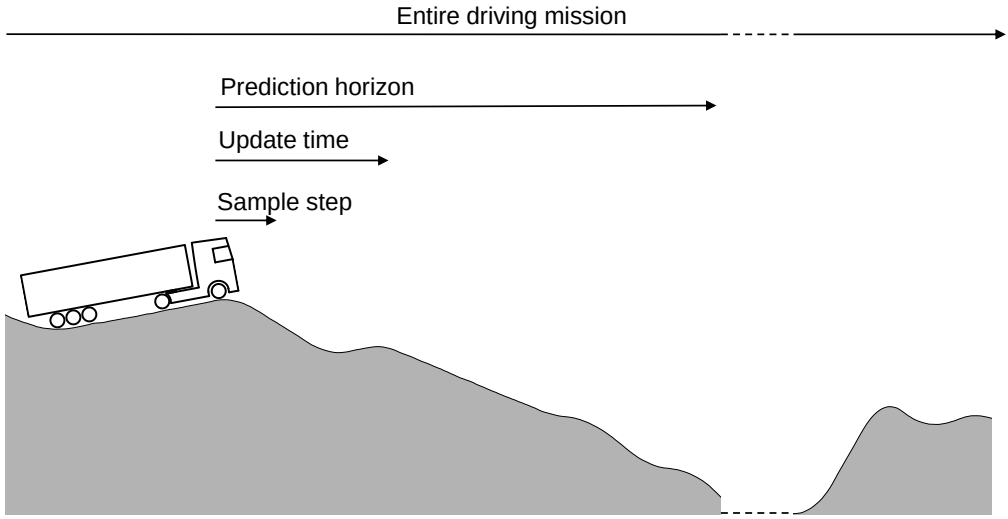


Figure 3.1: *Receding horizon control in the context of predictive energy management for long haul trucks.*

of charge (SOC) targets. The next layer, which is the focus of this thesis and is highlighted with a red box, performs the energy optimization over a prediction horizon. The third layer is responsible for the predictive gear selection, given the optimal trajectories from the upper layer. The horizon is around 1 kilometer. At the bottom, the real-time decisions are taken. The torque split algorithm is based on the well-known ECMS algorithm, [19], which receives the optimal equivalence factor from the energy management layer in the red box. The optimal equivalence factor is calculated based on the battery energy costate, denoted λ_B.

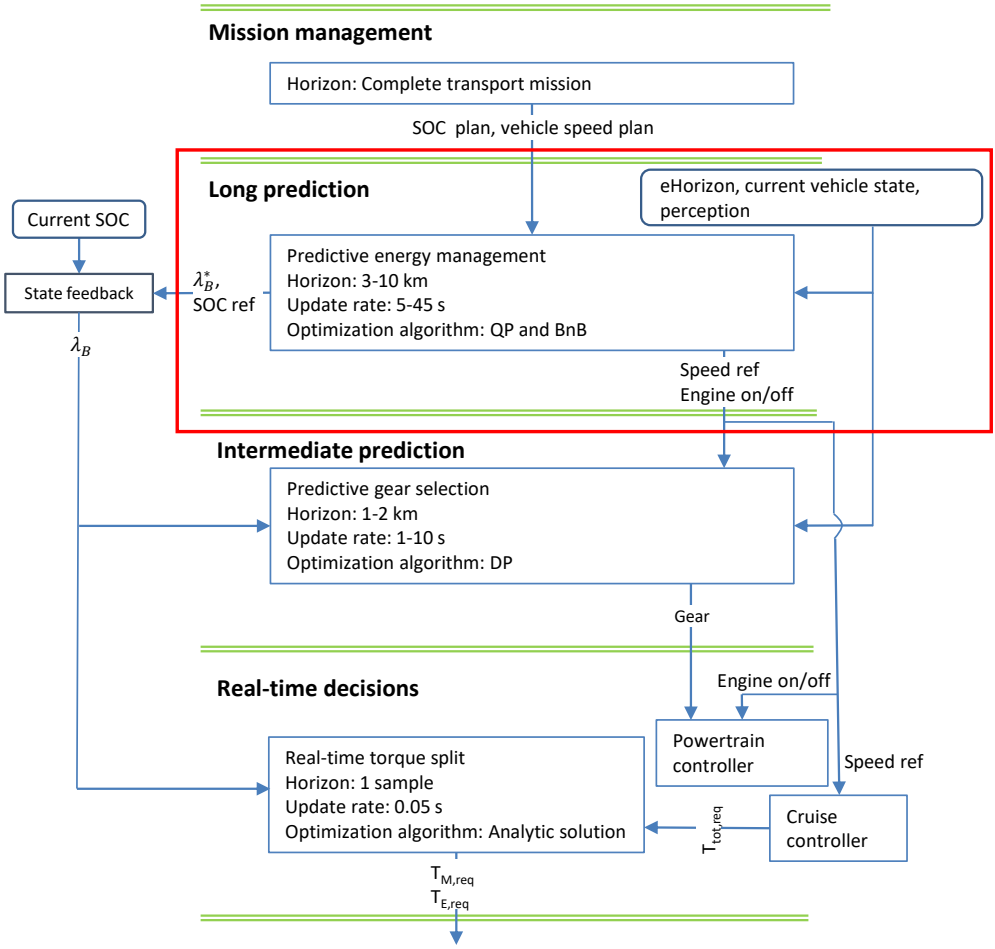


Figure 3.2: *Functional architecture of the hierarchical control strategy. The red box highlights the focus of this thesis. The mission management layer is not evaluated in this thesis but is added for completeness.*

3.3 Convex optimization and quadratic programming

Convex optimization is a class of mathematical optimization problems that fulfils some important criteria that make it possible to solve these problems reliably and efficiently [1]. This is especially the case for the subclass called quadratic programming (QP), which involves a quadratic cost function and affine constraints. This type of problem can be formulated as a multistage problem, [5], over a finite horizon according to

$$\begin{aligned} \underset{x_0 \dots x_N, u_0 \dots u_{N-1}}{\text{minimize:}} \quad & \sum_{k=0}^{N} \frac{1}{2} \begin{bmatrix} x_k \\ u_k \end{bmatrix}^T H_k \begin{bmatrix} x_k \\ u_k \end{bmatrix} + g_k^T \begin{bmatrix} x_k \\ u_k \end{bmatrix} \\ \text{subject to:} \quad & x_{k+1} = A_k x_k + B_k u_k \\ & \underline{x}_k \leq x_k \leq \overline{x}_k \\ & \underline{u}_k \leq u_k \leq \overline{u}_k \\ & D_k^x x_k + D_k^u u_k \leq d_k \\ & D_N^x x_N \leq d_N \\ & x_0 = \overline{x}_0 \end{aligned} \tag{3.1}$$

where x_k is the state vector, $\overline{x}_0$ is the initial value for the state, u_k is the vector of control variables, H_k is the hessian matrix, g_k is the gradient vector, N is the number of stages in the horizon, D_k^x and D_k^u represent inequality constraint matrices for states and controls, respectively, $\underline{x}_k$, $\underline{u}_k$, $\overline{x}_k$, $\overline{u}_k$ represent lower and upper bounds on states and controls respectively, and $k = 0 \dots N$. This formulation can easily be formulated as a quadratic program, with a special structure, for which there are efficient solvers developed

$$\begin{aligned} \underset{\bar{z}}{\text{minimize:}} \quad & \frac{1}{2} \bar{z}^T H \bar{z} + g^T \bar{z} \\ \text{subject to:} \quad & A\bar{z} = b \\ & F\bar{z} \leq g \end{aligned} \tag{3.2}$$

In Equation 3.2, $z_k = [x_k, u_k]$, $\bar{z} = [z_0, \dots, z_N]$, A is a matrix composed by stacking the A_k and B_k matrices by evolving the state dynamics from the initial state, and F is also a matrix composed by stacking the matrices representing the stage-wise inequalities, forming a block-diagonal matrix. Similarly, H and g are matrices composed by stacking of H_k and g_k into block-diagonal matrices with H_k and g_k placed along the diagonal.

3.3.1 Optimality conditions

An optimization problem can a bit more generally be written as

$$\begin{aligned} \underset{x}{\text{minimize:}} \quad & f_0(x) \\ \text{subject to:} \quad & f_i(x) \leq 0 \quad i = 1 \dots m \\ & h_i(x) = 0 \quad i = 1 \dots p \end{aligned} \tag{3.3}$$

where $f_0(x)$ is the cost function, f_i and h_i are inequality and equality constraint functions, respectively, m is the number of inequality constraints and p is the number of equality constraints. Using duality theory, as described in [1], it is possible to form the Lagrangian

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x) \tag{3.4}$$

This function represents the cost function augmented with a weighted sum of the constraint functions. The so-called Karush-Kuhn-Tucker (KKT) conditions state the optimality conditions, which for convex f_i and affine h_i are sufficient conditions, and are stated as

$$f_i(x^\star) \leq 0 \quad i = 1 \dots m \tag{3.5}$$

$$h_i(x^\star) = 0 \quad i = 1 \dots p \tag{3.6}$$

$$\lambda_i^\star \geq 0 \quad i = 1 \dots m \tag{3.7}$$

$$\lambda_i^\star f_i(x^\star) = 0 \quad i = 1 \dots m \tag{3.8}$$

$$\nabla f_0(x^\star) + \sum_{i=1}^{m} \lambda_i^\star \nabla f_i(x^\star) + \sum_{i=1}^{p} \nu_i^\star \nabla h_i(x^\star) = 0 \tag{3.9}$$

where $x^\star$ and $(\lambda^\star, \nu^\star)$ are primal and dual optimal points respectively. Equation 3.5 and 3.6 state that the primal problem is feasible; Equation 3.7 states that the dual variable related to the inequality constraints is non-negative; Equation 3.8 represents the complementary slackness, which means that either $f_i = 0$ or $\lambda_i = 0$ at the optimum; and Equation 3.9 states that the gradient of the Lagrangian vanishes at the optimum and can be interpreted as a force balance between the gradient of the cost function and the gradient of the constraints, scaled with the dual variables [1]. These conditions are at the base for many QP solvers. Furthermore, these solvers return the optimal dual variables with their solution. Similar to [13], the optimal dual variable related to battery energy is used as the equivalence factor (or costate) in the ECMS algorithm in the bottom layers in the control hierarchy, see λ_B in Figure 3.2. This can be interpreted as a weighting factor, which converts battery power into fuel power [19].

3.4 Mixed-integer programming

The optimality conditions and methods that are used to solve convex optimization problems such as the QP are unfortunately not applicable for a mixed-integer problem, where some of the variables are only allowed to take on integer values. This is because the set of allowed solutions is a non-convex set, since the points between two adjacent points are not part of the feasible set [16]. The most commonly used methods for these types of problems include cutting plane methods, relaxation and decomposition methods, heuristics, and branch and bound methods. When it comes to the mixed-integer quadratic program (MIQP), [6] shows

that branch-and-bound is superior to other approaches, performing an order of magnitude faster.

3.4.1 Branch-and-bound

For small problems, a feasible method is the brute-force method, which for a mixed-integer problem solves the problem with only real-valued variables for all different combinations of the integer variables. For a mixed-boolean problem, the running-time scales at 2^n, where n is the number of binary variables. Branch-and-bound is a method for global optimization and the basic idea is to successively partition the feasible set and solve simplified (relaxed) subproblems [16]. When relaxing the subproblems, integer constraints are replaced with interval constraints. This yields a lower bound to the problem. When an integer solution is found, this can form an upper bound to the original problem. The best upper bound found so far is called the incumbent. Often, an integer solution is guessed based on, e.g., a heuristic forming the initial incumbent, as presented in Section 5.3.1. Using this information, it is possible to only search in the regions where the optimal objective value is located and eliminate, or prune, partitions where the optimal objective value cannot be. Branch-and-bound is a tree search algorithm and can thus be illustrated as Figure 3.3 shows. The circles are called nodes, the top node is called root node, and the nodes at the last level of the tree are called leaves. The figure also shows that pruning can be based on any of these conditions:

- Problem is infeasible
- Dominated by upper bound
- Integer feasible

This means that if an infeasible solution is found in any of the nodes, no further branching is needed on that node since all subsequent nodes will also be infeasible. Further, if integer solutions are found somewhere in the tree, this solution will be the best in that particular branch, thus no more branching on that node is needed. Also, when the lower bound at a certain node is higher than the best upper bound found so far, the node can be pruned since this node's children cannot give better solutions than the lower bound of the parent node. When entire subtrees can be pruned in theses manners, the computational demand can be reduced compared to the brute-force method.

Two important decisions for the branch-and-bound algorithm are the node selection and the selection of branching variable. The most common search strategies for the node selection are best-first, depth-first, and breadth-first [16]. In best-first, the node with best lower bound is selected. The advantages are that the number of subproblems to solve is minimized. The disadvantages are that warm-starting is not straightforward and that it may require a lot of memory to store open problems. Also, an integer feasible solution is usually not found until the end of the search. In depth-first, the deepest node in the tree is selected. The

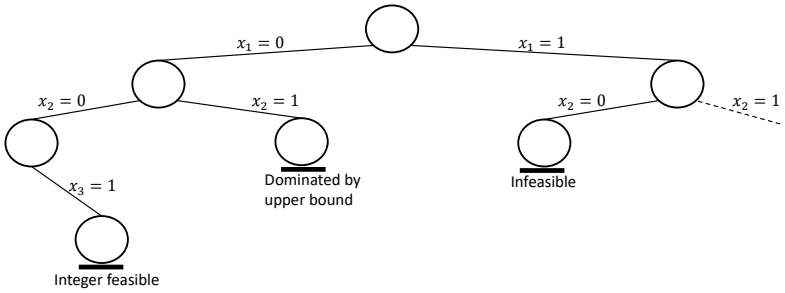


Figure 3.3: *Branch-and-bound tree and pruning conditions.*

advantage is that the search downwards in the tree is fast, increasing the probability of finding integer solutions. Further, it makes warm-starting, or using a good starting point, possible because only a single bound is changed. The disadvantage is that it might be necessary to explore many nodes before optimality can be proven, especially if no upper bound is found. In breadth-first, all nodes on a certain level in the tree are first solved before a new level can be considered. It works well if an early expansion of the tree is beneficial or if adding many constraints makes it hard to solve the problem. The selection of branching variable is important in order to minimize the size of the tree that needs to be searched. The branching can be done in different ways, and the selection can be up to the user decide on. One way is to select the variable closest to an integer. Another option is to select the variable with the largest fractional part. Further, it is possible to assign a certain order in advance, for example if the variables are time-dependent. Randomizing the selection is also a possibility.

Moreover, to avoid searching exactly all the created subproblems, a stopping criterion is typically defined. In [16], the following is presented

$$\frac{U - L}{L} < \epsilon \tag{3.10}$$

where U, and L are the best upper and lower bounds found so far, and ϵ is a threshold value, typically selected as 0.01.

4

Modelling

In this chapter, convex models for the vehicle and powertrain, engine, electric machine and battery are derived. The modelling follows in large the derivations in [13] and [9].

4.1 Vehicle and powertrain

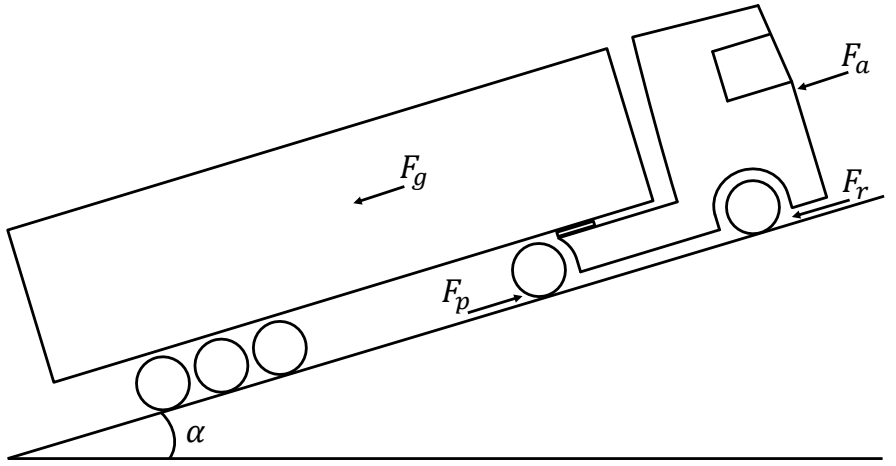


Figure 4.1: *Forces acting on the vehicle in the longitudinal dimension.*

Analysing Figure 4.1, the vehicle longitudinal dynamics can be modelled as

$$ma = m\frac{dv}{dt} = F_p - F_d \tag{4.1}$$

$$F_p = F_E + F_M + F_{brk} \tag{4.2}$$

$$F_d = F_g + F_r + F_a = mg\sin(\alpha) + mg\cos(\alpha)C_{rr} + \frac{C_D A_f \rho v^2}{2} \tag{4.3}$$

where m is the vehicle mass, a is the vehicle acceleration, v is the vehicle speed, F_p is the propulsion force, F_d is the drag force, F_g is the gravity force due to road slope, F_r is the rolling resistance, F_a is the aerodynamic drag force, F_E is the force delivered from the engine, F_M is the force delivered from the electric machine, F_{brk} is the brake force, α is the road slope, g is the gravitational acceleration, C_{rr} is the rolling resistance coefficient, C_D is the aerodynamic drag coefficient, A_f is the vehicle frontal area, and ρ is the air density. Combining Equations 4.1-4.3 yields

$$m\frac{dv}{dt} = F_E + F_M + F_{brk} - mg\sin(\alpha) - mg\cos(\alpha)C_{rr} - \frac{C_D A_f \rho v^2}{2} \tag{4.4}$$

Since road slope is naturally position-dependent and since Equation 4.4 is quadratic in vehicle speed, a reformulation in space domain with kinetic energy is performed

$$ma = m\frac{dv}{dt} = m\frac{dvds}{dsdt} = mv\frac{dv}{ds} = \frac{d}{ds}\left(\frac{mv^2}{2}\right) = \frac{dE_K}{ds} \tag{4.5}$$

where the chain rule was used, E_K is the vehicle kinetic energy and it was used that

$$E_K = \frac{mv^2}{2} \tag{4.6}$$

The longitudinal dynamics then becomes a linear expression in E_K

$$E_K'(s) = \frac{dE_K}{ds} = F_E + F_M + F_{brk} - mg\sin(\alpha) - mg\cos(\alpha)C_{rr} - \frac{C_D A_f \rho E_K}{m} \tag{4.7}$$

A schematic diagram of the hybrid powertrain is presented in Figure 4.2. The powertrain includes an internal combustion engine, transmission, electric machine and a battery. The electrical power balance is given by

$$P_{B,el} = P_{M,el} + P_{aux,el} \tag{4.8}$$

where $P_{B,el}$ is the battery electrical power, $P_{M,el}$ is the electric machine electrical power and $P_{aux,el}$ is the electrical auxiliary load. Since the aim is to reach a position-dependent model, a division with vehicle speed is performed, which converts power, expressed as energy per time unit, into a force, expressed as energy per distance. This yields

$$F_{B,el} = F_{M,el} + \frac{P_{aux,el}}{v} \tag{4.9}$$

where $F_{B,el}$ and $F_{M,el}$ represent battery and electric machine electrical force, respectively.

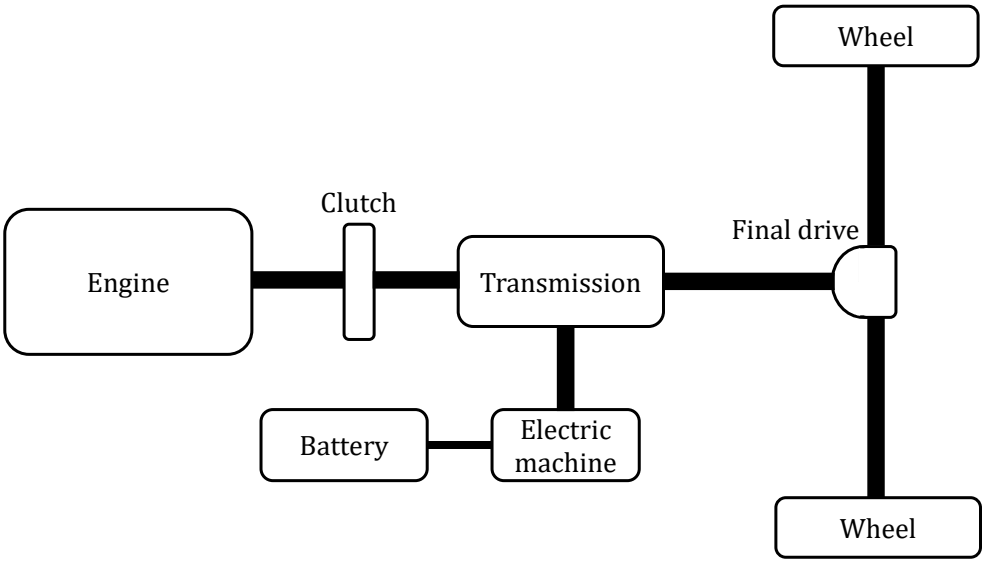


Figure 4.2: *Schematic diagram of the hybrid powertrain. The engine and electric machine deliver torque through the transmission and final drive to the wheels. It is possible to shut off and disconnect the engine from the driveline by opening the clutch and then drive in full electric mode.*

4.2 Engine

This section first derives a model for the engine fuel flow followed by the derivation of the engine force limitations.

4.2.1 Engine fuel flow

The measured fuel consumption map for the diesel engine is translated to a distance-based fuel consumption model. The distance-based fuel consumption is calculated by noting that in a four-stroke diesel engine, one cycle corresponds to two engine revolutions, denoted n_r. One revolution of the engine, in turn, correspond to $1/r$ revolutions of the wheel, where r is the driveline total gear ratio. The distance travelled after one wheel revolution is $2\pi r_w$, where r_w is the wheel radius. This yields

$$\mu = n_{cyl} \frac{r}{n_r 2\pi r_w} u_f \tag{4.10}$$

where μ is fuel flow (kg/m), n_{cyl} is the number of cylinders, and u_f is engine fueling $(kg/cycle/cylinder)$, which can be described as a function of engine torque and speed. Furthermore, based on the powertrain topology in Figure 4.2, engine

force and vehicle speed are given by

$$F_E = \frac{\eta_D T_E r}{r_w} = \frac{\eta_D T_E}{R_E} \tag{4.11}$$

$$v = \omega_w r_w = \frac{\omega_E}{r} r_w = \omega_E R_E \tag{4.12}$$

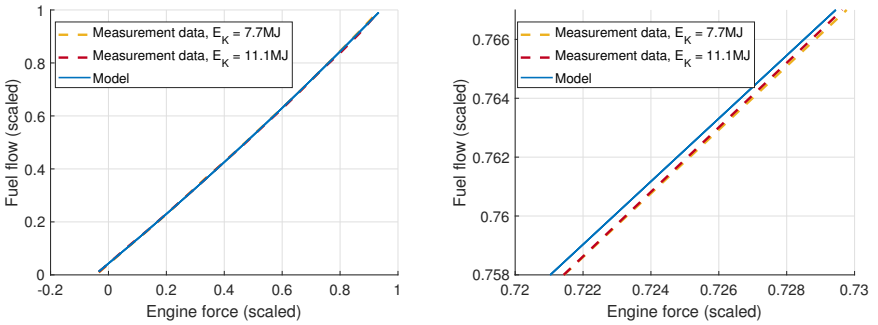
where F_E is the propulsion force delivered from the engine, η_D is the driveline efficiency, T_E is the engine torque, r is the driveline total gear ratio, ω_w is the wheel speed, r_w is the wheel radius, ω_E is the engine speed, and R_E is the total ratio between the road and the engine. The rotating inertia effect is here not included since top gear modelling is assumed and the inertia in this gear is considered small. Combining Equation 4.6 with Equation 4.12 yields

$$E_K = \frac{m(\omega_E R_E)^2}{2} \tag{4.13}$$

By using Equations 4.10, 4.11 and 4.13, the fuel flow μ is approximated with a second order polynomial in engine force

$$\mu = a_0 + a_1 F_E + a_2 F_E^2 \tag{4.14}$$

where a_i are coefficients of the polynomial. This yields the model fit in Figure 4.3.



(a) *Engine fuel flow model fit.* ***(b)*** *Engine fuel flow fit zoomed in.*

Figure 4.3: *Engine fuel flow model fit. The kinetic energies correspond to 75 km/h and 90 km/h.*

To avoid modelling the engine force in each gear, a simplification is made according to Figure 4.4. In this figure, $F_{E,1}$ is the force delivered from the cruise gear, which is the highest gear and also the gear with the lowest losses, and $F_{E,2}$ is the force contribution from any gear lower than the cruise gear. The fuel flow caused by $F_{E,1}$ is modelled with Equation 4.14 according to

$$\mu_1 = a_0 + a_1 F_{E,1} + a_2 F_{E,1}^2 \tag{4.15}$$

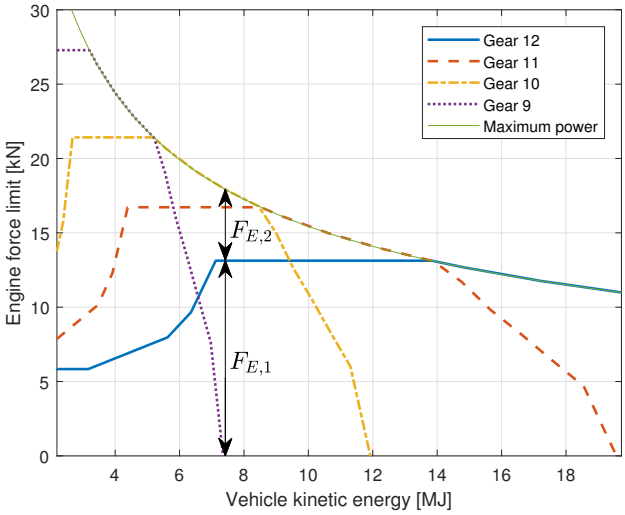


Figure 4.4: *Engine force limits for different gears. $F_{E,1}$ is the force delivered from the cruise gear and $F_{E,2}$ is the force contribution from any gear lower than the cruise gear. The range of kinetic energies corresponds to 40-120 km/h for a vehicle with a total mass of 35.5 tonnes.*

The fuel flow caused by $F_{E,2}$ is modelled with a lower efficiency, similarly as in [18], promoting usage of the highest gear

$$\mu_2 = ca_1 F_{E,2} + ca_2 F_{E,2}^2 = \tilde{a}_1 F_{E,2} + \tilde{a}_2 F_{E,2}^2 \tag{4.16}$$

where μ_2 is the fuel flow caused by $F_{E,2}$, and c is a constant coefficient > 1. Summing up the fuel flow components, the fuel flow can finally be modelled as

$$\mu = \mu_1 + \mu_2 = a_0 + a_1 F_{E,1} + a_2 F_{E,1}^2 + \tilde{a}_1 F_{E,2} + \tilde{a}_2 F_{E,2}^2 \tag{4.17}$$

4.2.2 Engine limitations

The maximum force that can be delivered by the engine can be approximated by three segments. The first two segments represent a speed-dependent limitation at low speed and the maximum force limitation at the cruise gear, as Figure 4.5 shows. The third segment represent the maximum power limit, illustrated by Figure 4.4 as the combined maximum torque curve for each gear. The limitations are given by the following inequalities

$$F_{E,1} \leq b_1 + b_2 E_K \tag{4.18}$$

$$F_{Emin,1} \leq F_{E,1} \leq F_{Emax,1} \tag{4.19}$$

$$F_{E,1} + F_{E,2} \leq \frac{P_{Emax}}{v} = \frac{P_{Emax}}{\sqrt{2E_K/m}} \tag{4.20}$$

where b_1 and b_2 are model coefficients, $F_{Emax,1}$ is the constant maximum engine force, $F_{Emin,1}$ is the constant friction force, derived below, and P_{Emax} is the engine maximum power. Equation 4.20 contains the nonlinear part $1/v = 1/\sqrt{(2E_K/m)}$.

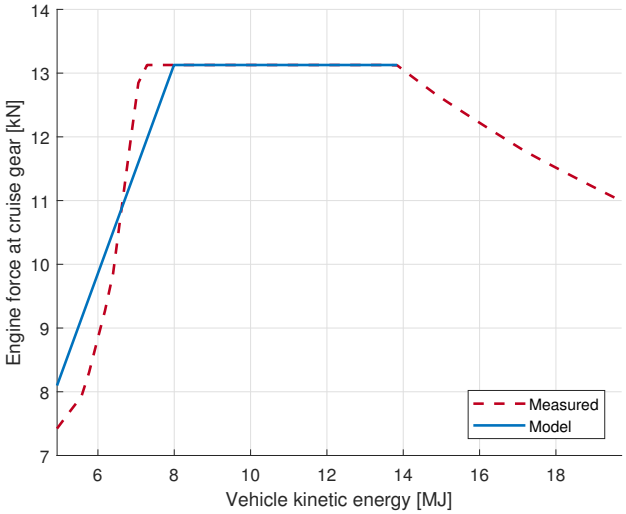


Figure 4.5: *Engine force limits for gear 12. The range of kinetic energies corresponds to 60-120 km/h for a vehicle with a total mass of 35.5 tonnes.*

This expression will be linearised in Equation 5.8. Using this result, it is possible to approximate Equation 4.20 as

$$F_{E,1} + F_{E,2} \leq P_{Emax}(h_0 + h_1 E_K) \tag{4.21}$$

Furthermore, $F_{Emin,1}$ is modelled by setting Equation 4.14 to zero and solving for the, negative, engine force when no fuel is injected

$$F_{Emin,1} = -\frac{a_1}{2a_2} + \sqrt{\left(\frac{a_1}{2a_2}\right)^2 - \frac{a_0}{2a_2}} \tag{4.22}$$

4.3 Electric machine

This section first derives a model of the electric machine losses followed by the derivation of the electric machine force limitations.

4.3.1 Electric machine losses

The electrical power of the electric machine can be modelled as

$$P_{M,el} = P_{M,mech} + P_{M,loss} \tag{4.23}$$
$$P_{M,mech} = \omega_M T_M \tag{4.24}$$

where $P_{M,el}$, $P_{M,mech}$ and $P_{M,loss}$ are electric machine electrical power, mechanical power and losses power, respectively, and ω_M and T_M are electric machine speed and torque, respectively. Furthermore, based on the powertrain topology in Figure 4.2, relations between vehicle and electric machine properties are given by

$$F_M = \begin{cases} \frac{\eta_D T_M \tilde{r}}{r_w} = \frac{\eta_D T_M}{R_M} & \text{if } F_M \geq 0 \\ \frac{T_M \tilde{r}}{\eta_D r_w} = \frac{T_M}{\eta_D R_M} & \text{if } F_M < 0 \end{cases} \tag{4.25}$$

$$v = \frac{\omega_M r_w}{\tilde{r}} = \omega_M R_M \tag{4.26}$$

$$P_M = F_M v \tag{4.27}$$

where η_D is the driveline efficiency, $\tilde{r}$ is the driveline ratio between the electric machine and the wheels, r_w is the wheel radius, R_M is the total ratio between the electric machine and the road, v is the vehicle speed, and F_M and P_M represent force and power at the wheels delivered by the electric machine. Introducing a driveline power losses term due to propulsion power delivered from the electric machine, $P_{M,D,loss}$, it holds that

$$P_{M,mech} = P_M + P_{M,D,loss} \tag{4.28}$$

Using this in Equation 4.23 yields

$$P_{M,el} = P_M + P_{M,D,loss} + P_{M,loss} \tag{4.29}$$

Lumping the losses terms and dividing by vehicle speed gives

$$F_{M,el} = F_M + F_{M,loss} \tag{4.30}$$

The energy losses related to the electric machine will then be given by the integration of the electric machine force losses, $F_{M,loss}$, over a distance. Figure 4.6 plots $F_{M,loss}$ as a function of F_M for different kinetic energies. Approximating the losses as piecewise affine in electric machine force and affine in vehicle kinetic energy yields

$$F_{M,loss} = \begin{cases} d_0 + d_{1,p} F_M + d_2 E_K & \text{if } F_M \geq 0 \\ d_0 + d_{1,n} F_M + d_2 E_K & \text{if } F_M < 0 \end{cases} \tag{4.31}$$

where d_i, $i = \{0, 1, 2\}$, are model coefficients, and where $d_{1,p}$ and $d_{1,n}$ are coefficients for positive and negative electric machine force, respectively. Figure 4.6 shows the model fit when varying the electric machine force for two different kinetic energies. The approximated piecewise affine model is then relaxed with inequalities to obtain a linear model. The optimal solution will always hold with equality, however, since otherwise energy would be wasted.

$$F_{M,loss} \geq d_0 + d_{1,p} F_M + d_2 E_K \tag{4.32}$$

$$F_{M,loss} \geq d_0 + d_{1,n} F_M + d_2 E_K \tag{4.33}$$

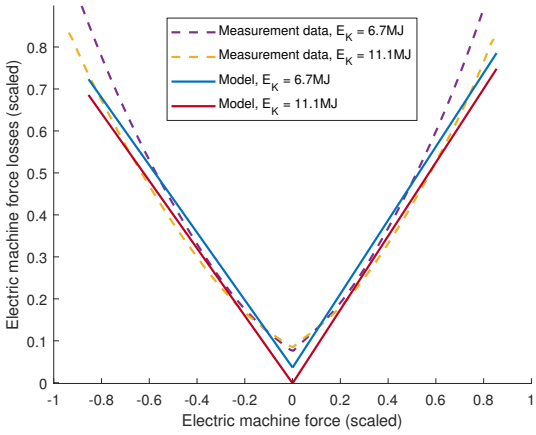


Figure 4.6: *Electric machine losses, measurement data vs model. The vehicle kinetic energies correspond to a vehicle speed of 70 and 90 km/h.*

4.3.2 Electric machine limitations

The electric machine torque limits generally consist of one part with a constant maximum torque and another part where the maximum torque is limited by the maximum power of the electric machine. The derivation of the electric machine limitations model is inspired by the modelling in [12]. The electric machine force limitation due to the constant maximum torque limit can be expressed as

$$F_M \leq F_{Mmax} = \frac{\eta_D T_{Mmax}}{R_M} \tag{4.34}$$

where T_{Mmax} is the electric machine maximum torque. For the part of the maximum torque curve where, approximately, the maximum power is limiting, a model is fitted to the measurement data according to

$$F_{MPmax} = \frac{c_{1,p}}{v} = \frac{c_{1,p}}{\sqrt{2E_K/m}} \tag{4.35}$$

where F_{MPmax} is the curve associated with the maximum electric machine power and $c_{1,p}$ is a model coefficient. Figure 4.7 shows both the original force limitation and the combination of the two expressions for the maximum electric machine force. Further, in Equation 4.35, it is also possible to include the battery power limitation leading to

$$F_{MPmax} = \frac{\min\left(c_{1,p}, \eta_e \eta_D P_{Bmax}\right)}{\sqrt{2E_K/m}} \tag{4.36}$$

where P_{Bmax} is the maximum battery power and η_e is the efficiency of the electrical path, i.e., the combined efficiency of the battery and the electric machine. As

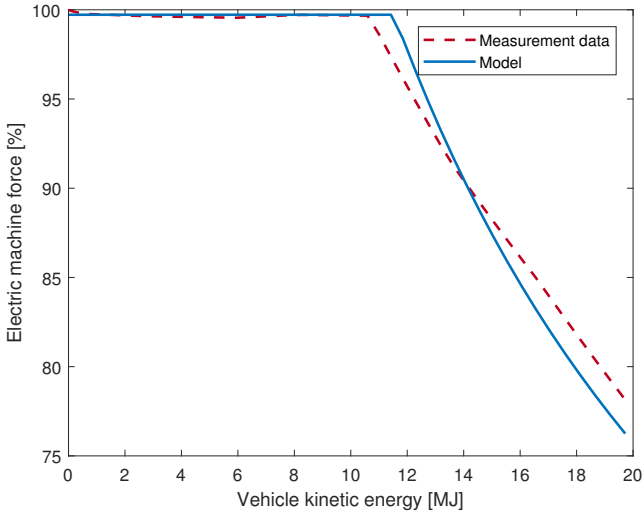


Figure 4.7: *Electric machine torque limit. The vehicle kinetic energy correspond to 0-120 km/h.*

performed in Section 4.2.2, see Equation 4.21, a linearisation is done leading to

$$F_M \leq \min(c_{1,p}, \eta_e \eta_D P_{Bmax})(h_0 + h_1 E_K) \Rightarrow \tag{4.37}$$

$$F_M \leq \bar{P}_{Mmax}(h_0 + h_1 E_K) \tag{4.38}$$

where $\bar{P}_{Mmax} = \min(c_{1,p}, \eta_e \eta_D P_{Bmax})$. Similarly, for the electric machine minimum force limits

$$F_M \geq F_{Mmin} = \frac{T_{Mmin}}{\eta_D R_M} \tag{4.39}$$

$$F_M \geq \min(c_{1,n}, \eta_e P_{Bmax}/\eta_D)(h_0 + h_1 E_K) \Rightarrow \tag{4.40}$$

$$F_M \geq \bar{P}_{Mmin}(h_0 + h_1 E_K) \tag{4.41}$$

where $\bar{P}_{Mmin} = \min(c_{1,n}, \eta_e P_{Bmax}/\eta_D)$, T_{Mmin} is the electric machine minimum torque, and $c_{1,n}$ is a model coefficient, which is fitted to measurement data similarly as $c_{1,p}$.

4.4 Battery

The battery energy dynamics can be expressed as

$$\frac{dE_B}{dt} = -P_B \tag{4.42}$$

where E_B is the battery energy and P_B is the battery chemical (internal) power. Reformulating to the space domain yields

$$\frac{dE_B}{ds}\frac{ds}{dt} = v\frac{dE_B}{ds} \tag{4.43}$$

Inserting this in Equation 4.42 gives

$$\frac{dE_B}{ds} = -\frac{P_B}{v} = -F_B \tag{4.44}$$

where F_B is the battery force and could be interpreted as the traction force delivered by the battery.

The battery losses are modelled based on an equivalent circuit model according to Figure 4.8. Using Kirchoff's voltage law and Ohm's law over the resistor

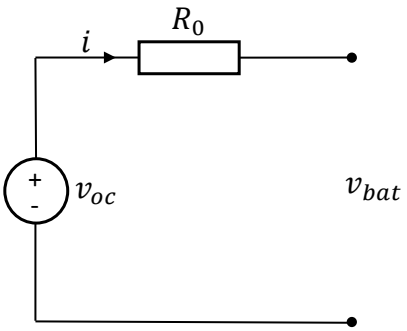


Figure 4.8: *Equivalent circuit model of the battery.*

yields

$$v_{bat} = v_{oc} - R_0 i \tag{4.45}$$

where v_{bat} is the terminal voltage, v_{oc} is the open circuit voltage, R_0 is the internal ohmic resistance of the battery and i is the battery current. Multiplying both sides with the current gives

$$v_{bat} i = v_{oc} i - R_0 i^2 \Rightarrow \tag{4.46}$$

$$P_{B,el} = P_B - P_{B,loss} \tag{4.47}$$

where $P_{B,el}$ is the battery electrical power, P_B is the battery chemical (internal) power, and $P_{B,loss}$ are the battery power losses. Since $v_{bat} i = P_{B,el}$, it is possible to express the current as

$$i = \frac{P_{B,el}}{v_{bat}} \tag{4.48}$$

Then the losses of the battery can be expressed as

$$P_{B,loss} = R_0 i^2 = R_0 \frac{P_{B,el}^2}{v_{bat}^2} \tag{4.49}$$

If assuming a constant voltage v_{bat}, a piecewise affine approximation, relaxed with inequalities, is given by

$$P_{B,loss} \geq r_{0,i} + r_{1,i} P_{B,el} \qquad i = 1 \dots N_p \tag{4.50}$$

where $r_{0,i}$ and $r_{1,i}$ are model parameters and N_p is the number of pieces selected. This model is illustrated in Figure 4.9, using four pieces. Dividing with vehicle speed gives

$$F_{B,loss} \geq r_{0,i}/v + r_{1,i} F_{B,el} \qquad i = 1 \dots N_p \tag{4.51}$$

where $F_{B,loss}$ is the battery losses force. Using the electrical force balance in Equation 4.9 yields

$$F_{B,loss} \geq r_{0,i}/v + r_{1,i}(F_{M,el} + P_{aux,el}/v) \qquad i = 1 \dots N_p \tag{4.52}$$

Further, using Equation 4.30 gives

$$F_{B,loss} \geq r_{0,i}/v + r_{1,i}(F_M + F_{M,loss} + P_{aux,el}/v) \qquad i = 1 \dots N_p \tag{4.53}$$

The division with vehicle speed on the electrical auxiliary power and $r_{0,i}$ is handled with a linearisation using Equation 5.8

$$F_{B,loss} \geq r_{0,i}(h_0 + h_1 E_K) + r_{1,i}(F_M + F_{M,loss} + P_{aux,el}(h_0 + h_1 E_K) \qquad i = 1 \dots N_p \tag{4.54}$$

This is now a linear model, which fits into the QP framework. The introduced inequalities make it possible for the losses to be larger than the original piecewise affine function. However, for the optimal solution, equality will always hold, since otherwise energy would be wasted.

Further, dividing Equation 4.47 with vehicle speed and rearranging gives

$$F_B = F_{B,el} + F_{B,loss} \tag{4.55}$$

The battery energy dynamics is then finally given by

$$E_B'(s) = \frac{dE_B}{ds} = -F_B = -(F_M + F_{M,loss} + F_{B,loss} + P_{aux,el}(h_0 + h_1 E_K)) \tag{4.56}$$

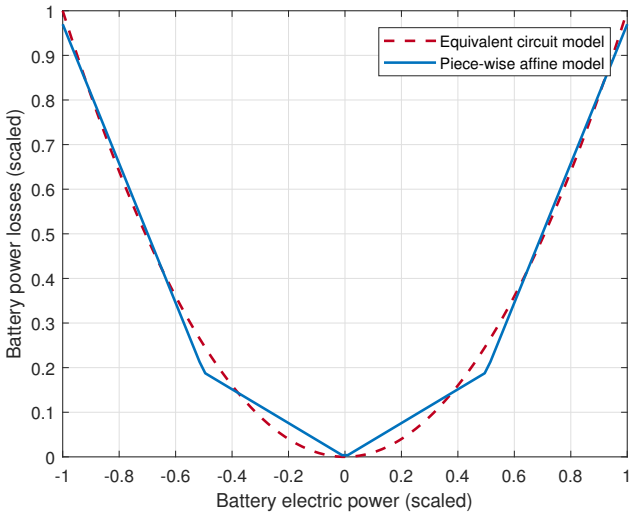


Figure 4.9: *Battery losses power and model approximation with a four pieces piecewise affine model.*

5

Control problem formulation

This chapter compiles the models from previous chapter and formulates the optimal control problem. Also, it describes the residual cost and penalties, and the design and implementation of the mixed-integer program. Finally, it briefly presents how the optimal control problem is solved.

The main control problem is to minimize the fuel consumption under the constraint of fulfilling a certain trip time:

$$\begin{aligned} &\text{minimize: } \int_{s_s}^{s_f} \mu ds \\ &\text{subject to: } t(s_f) \leq T_{max} \end{aligned} \tag{5.1}$$

where s_s is the start position, s_f is the end position, $t(s_f)$ is the total trip time, T_{max} is the maximum trip time and μ is the fuel flow. The total trip time is defined as

$$t(s_f) = \int_{s_s}^{s_f} \frac{1}{v} ds \tag{5.2}$$

where v is the vehicle speed. Expressing this using kinetic energy, see Equation 4.6, yields

$$t(s_f) = \int_{s_s}^{s_f} \frac{1}{\sqrt{(2E_K/m)}} ds \tag{5.3}$$

Further, the time derivative with respect to distance is

$$t'(s) = \frac{dt}{ds} = \frac{1}{\sqrt{(2E_K/m)}} \tag{5.4}$$

This is a nonlinear expression in the optimization variable E_K. A first order Taylor expansion of a function $f(x)$ around a point $x = a$ is generally given by

$$f(x) = f(a) + f'(a)(x - a) \tag{5.5}$$

The derivative with respect to kinetic energy of the time derivative is given by

$$\frac{t'(s)}{dE_K} = \frac{-1}{2}\left(\frac{2E_K}{m}\right)^{-3/2}\frac{2}{m} = -\frac{1}{(2E_K/m)^{3/2}m} \tag{5.6}$$

The time derivative is then approximately

$$t'(s) = 1/\sqrt{(2\hat{E}_K/m)} - \frac{1}{(2\hat{E}_K/m)^{3/2}m}(E_K - \hat{E}_K) \Rightarrow \tag{5.7}$$

$$t'(s) = h_0 + h_1 E_K \tag{5.8}$$

where $\hat{E}_K$ is the point for the Taylor expansion. This point is chosen as the kinetic energy from the previous optimal solution. Since it is defined on the old prediction horizon it will be a bit too short on the new horizon. The reference kinetic energy, $E_{K,ref}$, is then used to fill the remaining part of the horizon. $E_{K,ref}$ is further presented at the bottom of this section. The complete problem formulation is stated on the next page.

minimize:

$$\int_{s_s}^{s_f} (a_0 e_{on/off} + a_1 F_{E,1} + a_2 F_{E,1}^2 + \tilde{a}_1 F_{E,2} + \tilde{a}_2 F_{E,2}^2) ds + W_R + W_C \tag{5.9}$$

subject to:

$$t'(s) = h_0 + h_1 E_K \tag{5.10}$$

$$E_K'(s) = F_{E,1} + F_{E,2} + F_M + F_{brk} \tag{5.11}$$
$$- mg\sin(\alpha) - mg\cos(\alpha) C_{rr} - C_D A_f \rho E_K / m$$

$$E_B'(s) = -(F_M + F_{M,loss} + F_{B,loss} + P_{aux,el}(h_0 + h_1 E_K)) \tag{5.12}$$

$$e_{on/off} F_{Emin,1} \leq F_{E,1} \leq e_{on/off} F_{Emax,1} \tag{5.13}$$

$$F_{E,1} \leq b_1 + b_2 E_K \tag{5.14}$$

$$F_{E,1} + F_{E,2} \leq P_{Emax}(h_0 + h_1 E_K) \tag{5.15}$$

$$0 \leq F_{E,2} \leq e_{on/off} \kappa \tag{5.16}$$

$$F_{brk} \leq 0 \tag{5.17}$$

$$F_{Mmin} \leq F_M \leq F_{Mmax} \tag{5.18}$$

$$\bar{P}_{Mmin}(h_0 + h_1 E_K) \leq F_M \leq \bar{P}_{Mmax}(h_0 + h_1 E_K) \tag{5.19}$$

$$F_{M,loss} \geq d_0 + d_{1,p} F_M + d_2 E_K \tag{5.20}$$

$$F_{M,loss} \geq d_0 + d_{1,n} F_M + d_2 E_K \tag{5.21}$$

$$F_{B,loss} \geq r_{0,i}(h_0 + h_1 E_K) + r_{1,i}(F_M + F_{M,loss} + P_{aux,el}(h_0 + h_1 E_K)) \tag{5.22}$$

$$t(s_f) \leq T_{max} \tag{5.23}$$

$$E_{Kmin} \leq E_K(s) \leq E_{Kmax} \tag{5.24}$$

$$E_{Bmin} \leq E_B(s) \leq E_{Bmax} \tag{5.25}$$

$$E_B(s_f) \geq E_{Bf} \tag{5.26}$$

where E_{Bf} is the desired battery energy at the end of the horizon, $e_{on/off}$ represents the binary engine on/off variable, which is discussed in Section 5.3, W_R is the residual, or terminal, cost and W_C is the comfort and wear penalty. W_R and W_C are further detailed in Section 5.1 and Section 5.2. In Equation 5.22 $i = 1 \ldots N_p$, where N_p is the number of pieces in the piecewise affine approximation. Also, κ is a high number included to not put any additional constraint on $F_{E,2}$ when the engine is on, but when the engine is off, $F_{E,2}$ will still be limited to zero since $e_{on/off} = 0$ then.

Thus, the state vector is

$$x = \begin{bmatrix} t \\ E_K \\ E_B \end{bmatrix} \tag{5.27}$$

Moreover, the vector of control variables is

$$u = \begin{bmatrix} F_{E,1} \\ F_{E,2} \\ F_{brk} \\ F_M \\ F_{M,loss} \\ F_{B,loss} \\ e_{on/off} \end{bmatrix} \tag{5.28}$$

The reference kinetic energy, $E_{K,ref}$, the bounds on kinetic energy, E_{Kmin} and E_{Kmax}, and the maximum trip time T_{max} are calculated based on the pre-filter presented in [18]. Based on the vehicle longitudinal dynamics, slope profile, selected cruise speed, powertrain physical limitations and allowed speed variation, the function calculates a feasible reference speed trajectory over the horizon, and minimum and maximum bounds on vehicle speed. These are then converted to kinetic energy, and the reference speed trajectory is used to calculate a maximum trip time over the horizon.

5.1 Residual cost and fuel equivalents

Introducing a residual cost, or terminal cost, makes it possible to solve the optimization problem over a truncated horizon, which is the case in receding horizon control. The idea is to include a term, W_R, which reflects that energy at the end of the horizon can be used to save fuel in the future [11]. The different degrees of freedom in the optimization problem formulation are the states, i.e. time, kinetic energy and battery energy. These can be translated to an equivalent fuel mass by multiplying with the corresponding fuel equivalents according to the equations below.

Using an affine approximation of the engine fuel flow, instead of quadratic as in Equation 4.14, gives the following relation

$$\Delta\mu = \gamma \Delta F_E \tag{5.29}$$

where the fuel equivalent γ is the slope of the affine approximation and have the unit kg/J. For a unit distance, ΔF_E can be expressed as work in energy and $\gamma \Delta F_E$ then describes the fuel mass, $\Delta\mu$, that this work corresponds to. Kinetic energy at the terminal stage in the horizon can thus be assigned an equivalent fuel energy

$$\Delta\mu_K = \gamma \Delta E_K \tag{5.30}$$

where ΔE_K is the change in kinetic energy relative to the reference kinetic energy, $E_{K,ref}$, at the final stage. Similarly, an increase in battery energy above the target battery energy can be modelled as an equivalent fuel by

$$\Delta\mu_B = \gamma \eta_e \Delta E_B \tag{5.31}$$

where ΔE_B is the increase in battery energy above the target battery energy at the final stage, E_{Bf}, and η_e is the efficiency of the electrical path of the driveline. The fuel equivalent for time was derived analytically in [10] and is given by

$$\beta = 2\gamma\hat{v}F_a = 2\gamma\hat{v}\frac{C_D A_f \rho \hat{v}^2}{2} \tag{5.32}$$

where $\hat{v}$ is a certain constant vehicle speed, here chosen as the vehicle set speed. The corresponding residual cost is then

$$\Delta\mu_t = \beta\Delta t \tag{5.33}$$

In the simulation study, β was scaled down by a factor 10 in order to avoid a too large influence on the solution from $\Delta\mu_t$. Further analysis is needed to understand the correct setting of this parameter. Summing up the residual costs gives

$$W_R = \mu_K + \mu_B + \mu_t \tag{5.34}$$
$$= \gamma\Delta E_K + \gamma\eta_e\Delta E_B + \beta\Delta t \tag{5.35}$$

where W_R is the total residual cost, introduced in Equation 5.9. Moreover, it is worth noting that the dual variables of the optimization problem can also be interpreted as fuel equivalents that describe how a change in each related state affects the cost function. The reason that these are not used in this context of fuel equivalents is because they are calculated for the look-ahead horizon and thus do not specify future fuel equivalents beyond this horizon.

5.2 Penalties in the cost function

In order to obtain comfortable driving and to reduce too frequent shifting of control signals, which might also result in wear of components, penalization is an important concept. The term W_C reflecting this was introduced in Equation 5.9. The constituent parts of this term, $W_{C,i}$, are described in the equations below. In the equations, c_i are penalty parameters that are tuned for desired performance. Since the optimal control problem will be discretised, see Section 5.4, discrete penalization terms are presented in this section. In these expressions, k is the sample index, N is the number of stages in the horizon and Δs is the sample distance.

Kinetic energy deviation from reference kinetic energy
A comfort penalty is added to the cost function and is defined as the squared deviation of kinetic energy with respect to the reference kinetic energy

$$W_{C,1} = \sum_{k=1}^{N} c_1 \left(E_K(k) - E_{K,ref}(k)\right)^2 \Delta s \tag{5.36}$$

This penalty reduces speed variations when these do not significantly reduce the fuel consumption.

Kinetic energy trajectory difference to previous solution of kinetic energy
This penalty is defined as

$$W_{C,2} = \sum_{k=1}^{N_s} c_2 (E_K(k) - E_K^{i-1}(k))^2 \Delta s \tag{5.37}$$

where N_s is the index in the current prediction horizon where the horizon of the previous solution ended, and $i-1$ refer to the previous solution. This penalty makes sure that the solution in the current horizon is close to the previous solution, unless fuel savings from altering the speed profile dominates.

On/off rate penalty
The purpose of this penalty is to avoid frequent on/off behaviour within the horizon and it is given by

$$W_{C,3} = \sum_{k=1}^{N} c_3 (e_{on/off}(k) - e_{on/off}(k-1))^2 \Delta s \tag{5.38}$$

where $e_{on/off}(0)$, at $k = 1$, refers to the current state of the engine (on or off). Rate penalties, in a multi-stage formulation, may need an additional state to act as a memory, which enables the penalization of the previous on/off decision. Differences between optimization variables could also be penalized by adding non-zero elements in the Hessian matrix in the QP formulation, see Equation 3.3, outside the diagonal, creating cross terms between two successive $e_{on/off}$ variables.

Propulsion force rate penalty
The purpose of this penalty is to avoid rapid change of the propulsion force and it is given by

$$W_{C,4} = \sum_{k=1}^{N} c_4 (F_p(k) - F_p(k-1))^2 \Delta s \tag{5.39}$$

where $F_p = F_{E,1} + F_{E,2} + F_M + F_{brk}$. At $k = 1$, $F_p(0)$ refers to the actual propulsion force.

5.3 Mixed-integer problem design and implementation

The binary engine on/off variable is a computationally expensive variable. Knowing that the engine should typically not be turned on and off very often, a possibility is to set the binary on/off decision variable more sparse, as Figure 5.1 exemplifies, and hold the binary on/off variable constant between the binary grid

points, e.g. between stage 1 and 5, according to the figure. Further, since with receding horizon control, only the first part of the solution is actually used until a new solution is calculated, binary variables are only used in the first part of the horizon, and relaxed, continuous 0-1 variables are used in the remaining part of the horizon.

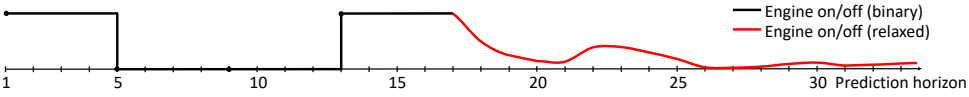


Figure 5.1: *Binary on/off variables over a prediction horizon.*

Figure 5.2 illustrates how the branch-and-bound tree would look like (x_i denotes the engine on/off decision variable) if the full tree is expanded in case of four binary variables. Forbidden nodes are also included in this figure. These nodes are set based on avoiding switching between engine on and off too often. In addition, the actual engine mode (x_0) is also considered in the selection of forbidden nodes. For example, if the engine is on, then node number 5, and all of its children, are also forbidden. Similarly, if the engine is currently on, node 6 and all its children are forbidden nodes.

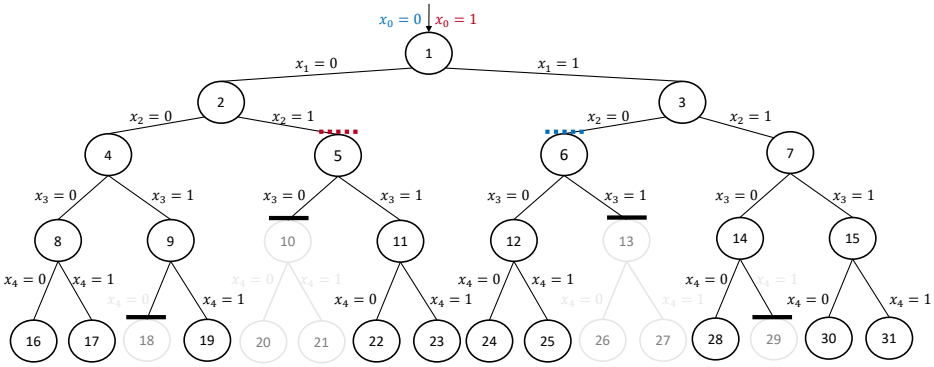
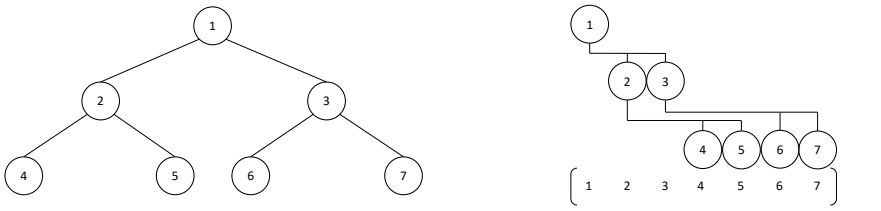


Figure 5.2: *Branch-and-bound tree with forbidden nodes in grey and marked out with a thick line over these nodes. The dotted lines are active depending on if the engine is currently on or off.*

Although there are advanced commercial mixed-integer program solvers available, and simpler open source versions as well, there are benefits of implementing the algorithm. These include the possibility of tailor-making the algorithm for the hybrid vehicle application and the possibility of debugging the branch-and-bound process. The implementation of the branch and bound algorithm and its tree representation could be done by using a heap data structure, [22], see Figure 5.3a and Figure 5.3b. The tree can thus be represented with an array. There are some important index relations:

- The root node is the first index
- The children of the node at position n will be at position $2n$ and $2n + 1$
- The index of the parent node of node n is located at position $n/2$

These relations allow for moving up and down in the tree through index computations.



(a) *Tree representation with heap data structure*

(b) *Representation with an array.*

Figure 5.3: *Branch-and-bound tree representation.*

5.3.1 Initial incumbent and simple heuristic

After solving the root node in Figure 5.2 with all binary variables relaxed between 0-1, the initial incumbent is calculated in order to have an upper bound to the problem. By defining a threshold value for the on/off variable, and by averaging the relaxed on/off variables over the number of segments it is held constant, an on/off profile is determined. Then, the QP is solved again with the engine on/off profile fixed for the complete part with binary variables. After these two QP problems are solved, and unless the stopping criteria in Equation 3.10 is met, the branch-and-bound algorithm goes either left or right in the tree, i.e., going to either node 2 or 3 in Figure 5.2.

Moreover, to reduce the number of iterations in the branch-and-bound procedure, a simple heuristic is developed. As the branch-and-bound algorithm advances down the tree it takes the remaining relaxed engine on/off solution and adds a cost representing engine friction when it is likely that the engine will be on. For all remaining binary variables in the horizon that are currently relaxed and larger than a threshold value the following cost is added

$$a_0(1 - e_{on/off}(k)), \qquad k = N_1 \dots N_2 \tag{5.40}$$

where a_0 is the engine friction coefficient, and $N_1 - N_2$ represent the relaxed remaining binary variables.

5.4 Solving the optimal control problem

To numerically solve the optimal control problem, the problem needs to be discretized. In this thesis, the Euler Method is chosen

$$x(k+1) = x(k) + hf(x_k, u_k) \tag{5.41}$$

where h refer to the sample step in distance. The optimal control problem is solved using the quadprog solver in Matlab. The solver is interfaced by the use of the Matlab toolbox for modelling and optimization called YALMIP [15].

6
Results

This chapter presents the results of the simulation study. It describes a solution on a single prediction horizon, presents simulation results for different driving cycles, comparing the hybrid truck with predictive control against a hybrid and a conventional truck without predictive control. In addition, results related to the branch-and-bound algorithm are given, as well as a sensitivity analysis. The results in this chapter are based on simulations with a complete vehicle simulation model, which has a vehicle configuration specified in the section below.

6.1 Vehicle configuration

The vehicle is configured with a 13L 500 hp diesel engine, and an electric machine with a maximum power of 200 kW. The battery has a slightly higher maximum power and a usable energy content of approximately 8 kWh. For comparisons, a conventional truck is also simulated. This has the same powertrain (except for the electrical driveline). The vehicle mass of the hybrid truck is selected to be 35.5 tonnes, and the vehicle mass of the conventional truck is set to 35 tonnes. This means that an assumed weight increase of 500 kg is added to the hybrid truck to account for the weight of the battery, electric machine, etc.

6.2 Optimization result for a single prediction horizon

Figure 6.1 shows an example solution for one prediction horizon within a driving cycle. The top subplot shows that the vehicle slightly increases the speed the first hundreds of meters before the crest and then, as the third subplot shows, the energy manager decides to shut off the engine, letting the vehicle roll over the crest and at the same time reducing the vehicle speed, however, still within

the allowed speed range. In the following downhill the vehicle regains its speed. From the middle and towards the end of the horizon, the third subplot shows that the engine on/off decision is relaxed since it is neither on nor off. The fourth subplot shows the optimal battery energy, scaled to the dimensionless state of charge (SOC) value, and its related dual variable (costate), scaled to a dimensionless equivalence factor using the diesel specific energy. For this particular horizon, the battery chemical power is, according to the figure, worth approximately 2.3 times fuel power, meaning that the electrical path of the driveline is slightly more than twice as efficient as the fuel path. As Figure 3.2 showed, this equivalence factor is sent to the sublayers in the functional architecture.

Moreover, the dotted lines show the previous solutions for the vehicle speed, engine force, electric machine force and SOC. These are all fairly close to the current solution. This is beneficial for both driver comfort and to avoid component wear.

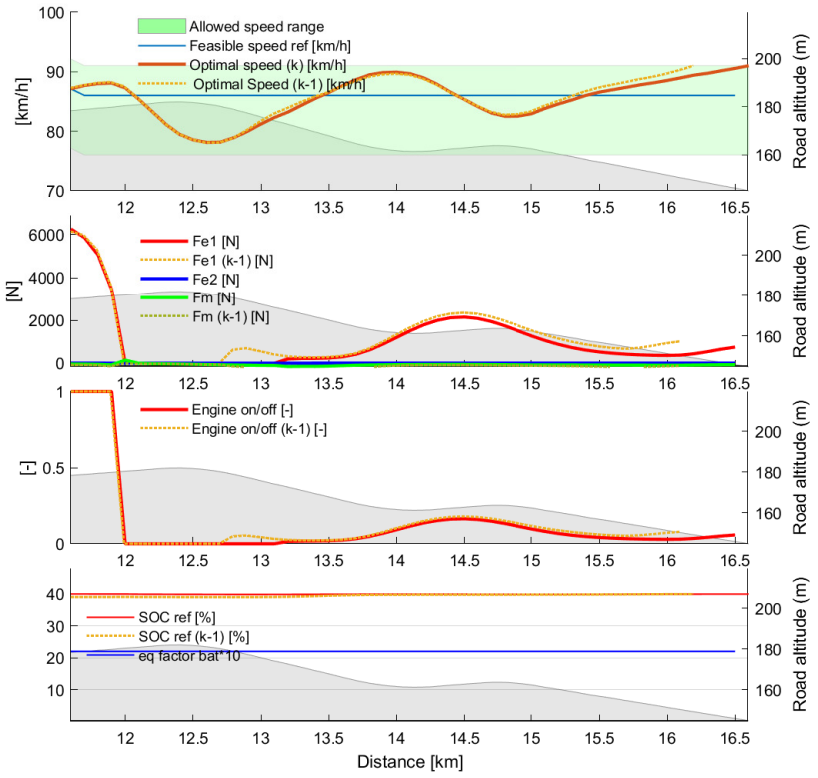


Figure 6.1: *Optimization result for a single prediction horizon. The engine is shut off at 12 km, and the vehicle rolls over the crest losing some speed but regaining this in the downhill. The first third of the horizon consists of a binary engine on/off control signal and the latter two thirds have a relaxed control signal between 0-1.*

6.3 Complete driving cycles

This section presents simulation results for two different driving cycles. These represent the road between Borås and Landvetter, back and forth, in Sweden, and the road between Frankfurt and Koblenz in Germany. The first cycle could be classified as somewhere between flat and hilly and the latter cycle is hilly. Fuel consumption results for a hybrid truck with and without the predictive energy management controller are compared against a conventional truck without predictive energy management functionality.

Figure 6.2 shows distance-plots for the first cycle with the predictive controller active. The truck is primarily driving in full electric mode, with the engine shut off, in the downhill parts of the cycle, either regenerating with the electric machine or providing minor propulsion power. In the steepest downhill around 60 km, the engine is engaged to obtain engine braking since the brake power demand is higher than what the electric machine and battery can manage. Also, the gear shift to gear 10 at the same point in the cycle is performed to increase engine braking power and to put the electric machine in a more efficient operating point. In the steepest uphills the electric machine is supporting the diesel engine, making it operate in more efficient operating points and reducing the need of downshifting.

6.3.1 Fuel consumption

Figure 6.3 shows that, on the Borås-Landvetter-Borås cycle, the hybrid truck with predictive control saves almost 10% fuel relative to the conventional truck. Moreover, the hybrid truck without predictive control saves a bit more than 3%. On the Frankfurt-Koblenz cycle, the hybrid truck with predictive control saves around 15.5%, whereas the hybrid truck without predictive control saves almost 8%.

The reported numbers in this section include a fuel consumption compensation for the SOC difference between start and end of cycle. This is done with the method presented in [19]

$$M_0 = M - \sigma \Delta SOC \tag{6.1}$$

where M_0 is the SOC-compensated fuel consumption, M is the measured fuel consumption, ΔSOC is the SOC difference between start and end of cycle and σ is a curve fitting coefficient, which was fitted to a series of $(\Delta SOC, M)$ measurements using the simulation model. Furthermore, the set speed in the simulations is slightly adjusted in order to make average vehicle speed fairly close to that of the conventional truck. To fine-tune the fuel consumption results even more, a similar compensation approach as for the SOC difference is performed for the vehicle speed difference as well. This is assumed needed since the fuel consumption differences in the Section 6.4 are fairly small.

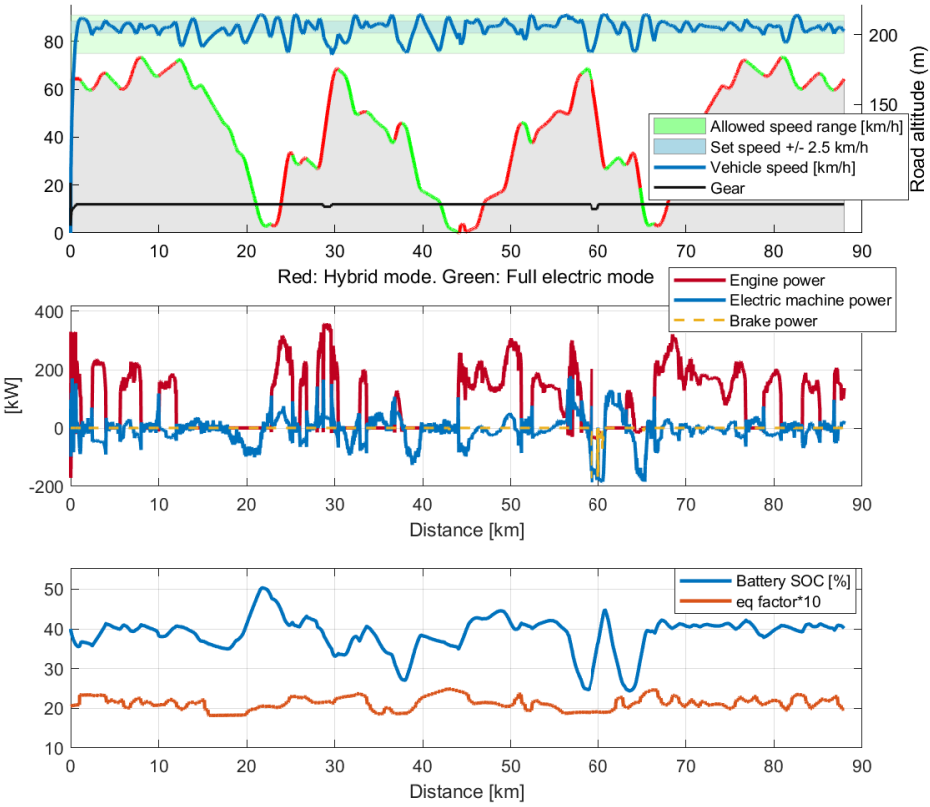


Figure 6.2: *Simulation results for the Borås-Landvetter-Borås driving cycle. The engine is turned off 48% of the time. The light green speed range in the top subplot shows the window of allowed speed variation. The light blue speed range, plotted in the same subplot, is added for illustrative purposes to indicate a more narrow speed band where the vehicle speed often fluctuates between when the topography is fairly flat.*

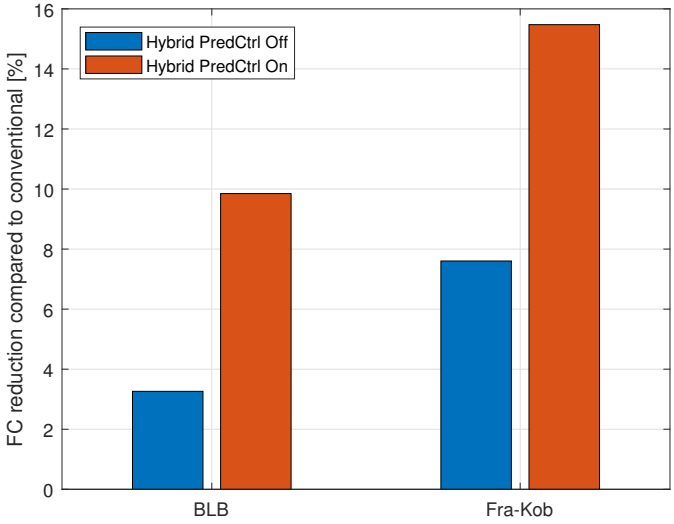


Figure 6.3: *Fuel consumption (FC) results for the hybrid trucks with and without predictive control, compared to a conventional truck without predictive control. Approximately 10-15.5% fuel is saved with the predictive control and 3-8% fuel is saved without predictive control. BLB refers to the Borås-Landvetter-Borås cycle and Fra-Kob refers to the Frankfurt-Koblenz cycle.*

6.3.2 Branch-and-bound results and solver time

Figure 6.4 shows the best indices from the branch-and-bound algorithm, refer to Figure 5.2 for an interpretation of the indices. The figure shows that the initial incumbent, based on defining a threshold value of the engine on/off variable and then rounding, see Section 5.3.1, is often the best index, in 81% of the cases. Specifically, the initial incumbent is often the best index for the more simpler sections with, e.g., very long downhills at a suitable slope, for example between 15-20 km in the cycle. Furthermore, the bottom subplot shows how the number of iterations vary over the cycle. The average number of iterations for this cycle was 8.0, meaning that this number of QP solutions, on average, need to be performed at each update of the receding horizon controller.

The average solver time for each QP solution was approximately 50 ms, using the Matlab quadprog function through YALMIP on a PC with an Intel i7-9850H 2.60 GHz processor. If the branch-and-bound algorithm requires 8 QP solutions on average, each taking 50 ms to complete, this corresponds to 0.4 s. Considering that the update time, based on defining the event trigger as travelling the distance 400 m, is 18 s when the vehicle speed is 80 km/h, real-time requirements are met by a considerable margin.

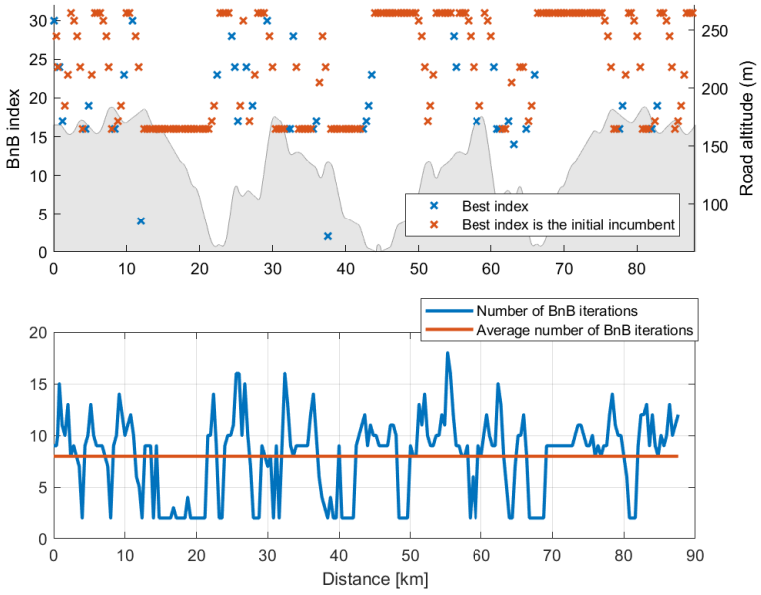
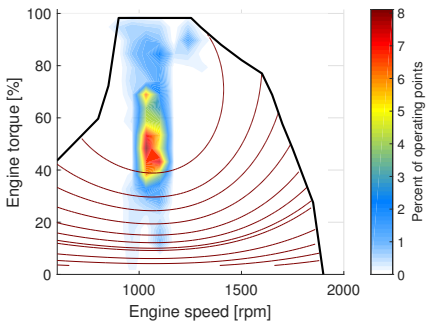


Figure 6.4: *Number of branch-and-bound (BnB) iterations and best indices for the BLB cycle.*

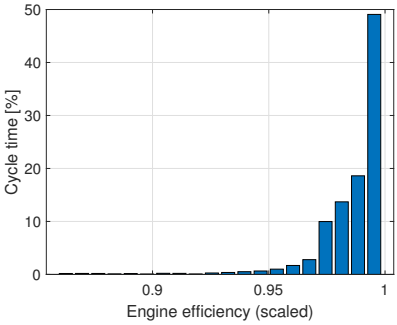
6.3.3 Efficiency plots of engine and electric machine

Comparing Figure 6.5 and Figure 6.6 shows that with the predictive controller active, better engine efficiency points can be attained. This is mainly since several low-load operating points are taken by the electric machine when driving in full electric mode, instead of using the engine in points where it has a lower efficiency. The operating points are also wider in the speed range of the engine because the vehicle speed is varied more with the predictive controller.

Figure 6.7a shows a small light blue coloured area at the minimum torque level. These operating points correspond to the steepest downhills in the cycle. Since these operating points are relatively few they are almost not visible. However, the energy from these points is significant since these are close to peak power of the electric machine. Moreover, Figure 6.7a and Figure 6.8a look fairly similar but the operating points with the predictive controller active seems to be a bit wider in both the speed, due to the predictive speed control, and the positive torque, due to full electric propulsion, dimensions.

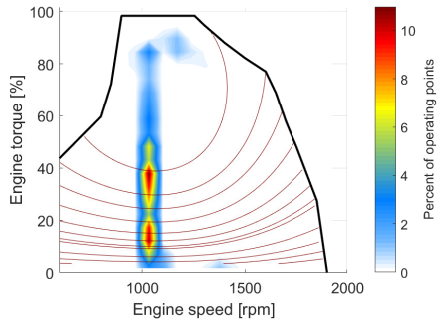


(a) Engine operating points. The level curves indicate approximate engine efficiency and are added to illustrate where the highest efficiency region is located.

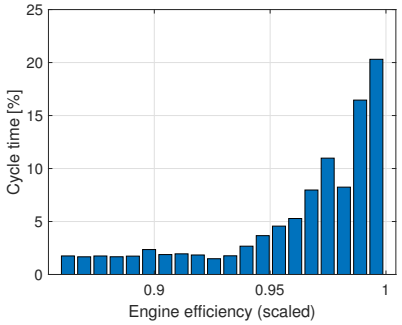


(b) Engine efficiency distribution.

Figure 6.5: *Engine efficiency and operating points – Predictive energy management controller.*

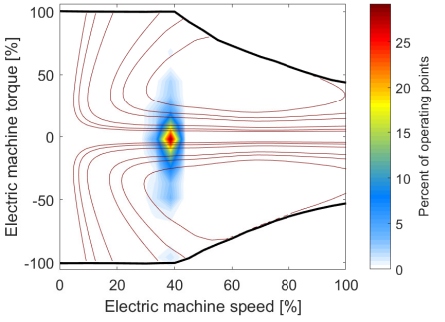


(a) Engine operating points.

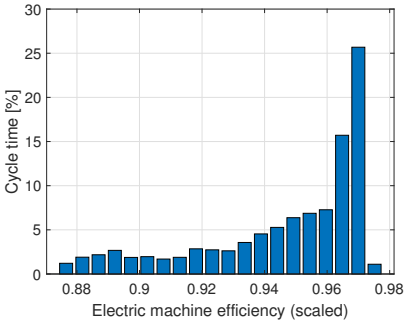


(b) Engine efficiency distribution.

Figure 6.6: *Engine efficiency and operating points – Non-predictive energy management controller.*

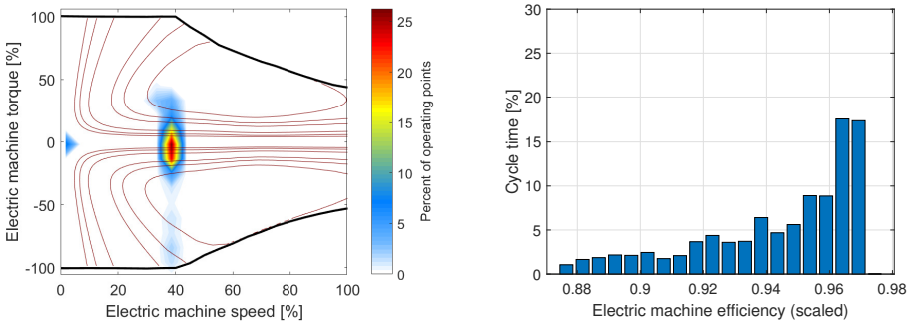


(a) *Electric machine operating points. The level curves indicate approximate electric machine efficiency and are added to illustrate where the highest efficiency region is located.*



(b) *Electric machine efficiency distribution.*

Figure 6.7: *Electric machine efficiency and operating points – Predictive energy management controller.*



(a) *Electric machine operating points.*

(b) *Electric machine efficiency distribution.*

Figure 6.8: *Electric machine efficiency and operating points – Non-predictive energy management controller.*

6.3.4 Engine power vs electric machine power

Figure 6.9 shows a distribution of engine and electric machine operating points plotted together. Points along the y-axis are points where only the engine is used and the electric machine is providing zero power. Points along the x-axis are points where the engine is off and full electric mode is activated. The first quadrant shows that at mid to high engine power, the engine is sometimes assisted by the electric machine. The second quadrant shows that the engine sometimes charges the battery where the electric machine power is negative. Most points are however concentrated closely to the y-axis, indicating that the engine mainly propels the vehicle. The area around the origin are points where the engine is turned off and the electric machine needs to provide zero or low positive or negative power. Going more to the right or left on the x-axis indicate points where the electric machine is either propelling in full electric mode or performing regenerative braking. To the very far left on the x-axis, somewhat negative engine power is visible since the engine needs to be engaged to perform engine braking there (only the engine friction power is shown here, and not the additional engine brake power from the engine brake system since this was not modelled in the vehicle simulation model).

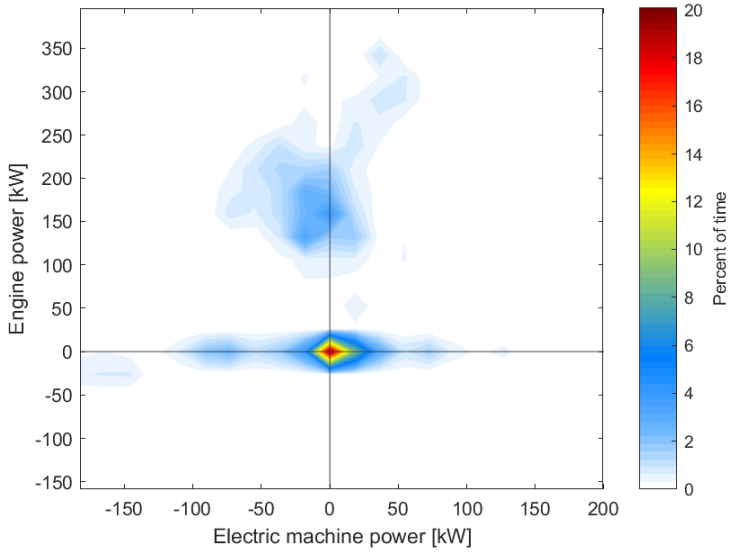


Figure 6.9: *Engine vs Electric machine, XY-plot, for the hybrid truck with the predictive energy management controller active.*

6.4 Sensitivity analysis of horizon parameters and search method

There are several parameters in the description of the horizon that are possible to study the sensitivity for. These include:

- Number of binary variables
- Number of segments the binary variable is held constant
- Resolution of horizon
- Horizon length

A more complete study of all these parameters is out of scope in this thesis. Instead, the sensitivity for certain parameters is analysed. The horizon length and resolution are kept constant at five kilometers and 100 m, respectively. The varied parameters are the number of binary variables and the number of segments where the binary variable is held constant. As an example, if the number of binary variables is 4, the number of segments where the binary variable is held constant is 4 and the sample step in the horizon is 100 m, the total distance covered with binary variables is 1600 m. As a consequence, the remaining 3600 m is relaxed with continuous variables. This description is defined as the base case in the analyses below.

In the results below, the branch-and-bound method, with and without the simple heuristic presented in Section 5.3.1 is used and compared to the brute-force method. In addition to varying the horizon parameters, the depth-first search method of the branch-and-bound algorithm, see Section 3.4.1, and a suboptimal solution using the initial incumbent directly as the solution are also analysed.

6.4.1 Results of sensitivity analysis

Figure 6.10 shows how the number of segments where the binary engine on/off variable is held constant affects the fuel consumption, for different numbers of binary variables. It seems like having the binary variable constant for only one segment increases the fuel consumption somewhat, whereas five segments held constant seems to increase fuel consumption slightly in the case of 5 binary variables. Around 2-4 seems to be the most promising amount of segments held constant. Having only two segments constant gives a fairly short horizon of binary engine on/off control. However, thanks to the receding horizon control methodology the solution is frequently updated, which may decrease the need for a long horizon of binary variables. It is also interesting to note that having fewer segments held constant seems to reduce the needed amount of QP iterations in the branch-and-bound algorithm.

The top subplot in Figure 6.11 shows that the average number of iterations of the branch-and-bound algorithm increases at a fairly slow rate as the number of binary variables increases, as compared to the brute-force method. Also, applying the simple heuristic in the branch-and-bound procedure gives a significant

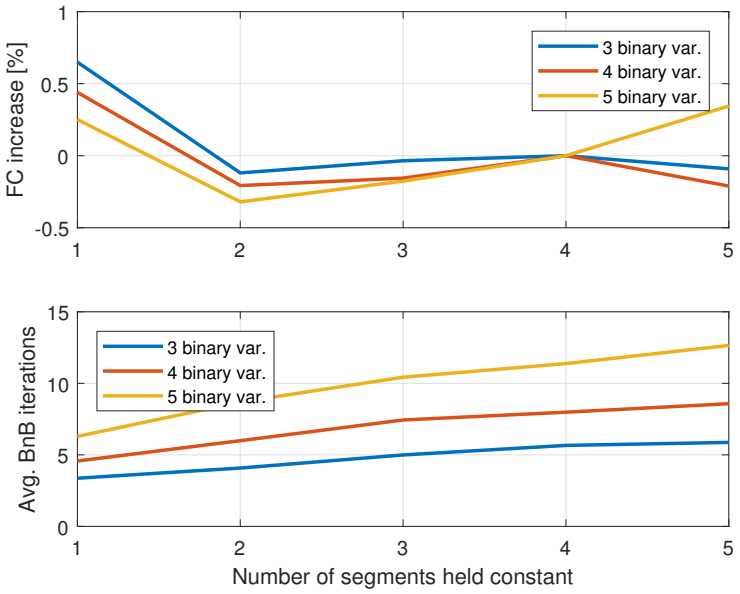


Figure 6.10: *Effect of the number of segments the binary variables are held constant on fuel consumption (FC) and average number of branch-and-bound iterations. Holding the binary variable constant between 2-4 segments seems to give better fuel consumption. Also, the fewer the segments, the fewer iterations are needed on average.*

decrease in the number of iterations. However, fuel consumption is slightly deteriorated. The simple heuristic was compared to the original branch-and-bound method, giving the exact same solution in 94% of the cases the energy management function was called during a simulation. The middle subplot also shows that as the number of binary variables increases, the branch-and-bound algorithm performs better relative to the brute-force method. For example, at 7 binary variables, the branch-and-bound algorithm needs on average approximately 50% less iterations than the brute-force method. The bottom subplot shows that increasing the number of binary variables improves fuel consumption but the gains seem to flatten out after 5 binary variables.

When applying the depth-first search method, the fuel consumption is approximately the same (0.1% improvement using depth-first) compared to the base case, which uses the best-first method. The similar results are expected since the control problem itself is not changed and the branch-and-bound algorithm is a global optimization method. However, the number of iterations is slightly higher, 8.9 iterations on average instead of 8.0 for the best-first search method.

For the suboptimal case of directly using the initial incumbent, defined in Section 5.3.1, as the solution to the mixed-integer program, the fuel consumption increased by 0.4%, with a constant of 2 QP solutions per receding horizon update.

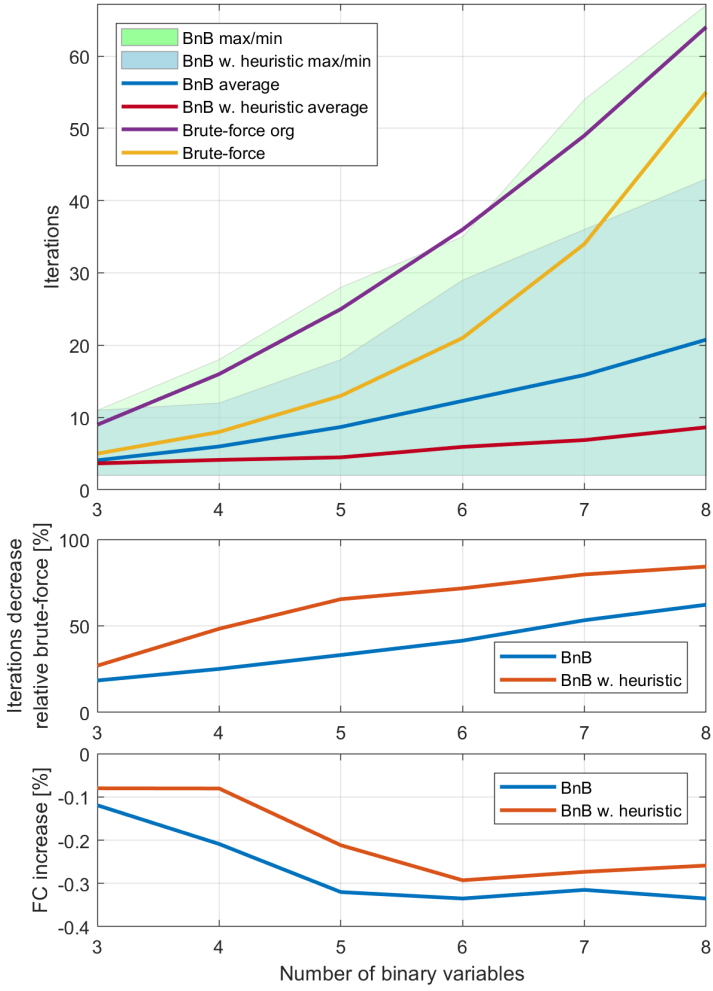


Figure 6.11: *Branch-and-bound (BnB) vs brute-force iterations and fuel consumption (FC). Both the original BnB algorithm and the BnB algorithm with the simple heuristic are plotted. The number of brute-force iterations have been compensated by subtracting the number of forbidden nodes. Brute-force org represents the original brute-force method with 2^n iterations, where n is the number of binary variables. The fuel consumption increase relative to the base case is plotted in the bottom subplot. All cases have a lower fuel consumption, hence the negative increase. Fuel consumption and iterations for a certain number of binary variables represent the best fuel consumption as the number of segments held constant is varied between 1-5.*

7

Extensions to control formulation

In this chapter, a traffic scenario with a slow-moving leading vehicle is studied. Also, dynamic battery power limitations are incorporated into the control problem by including a battery temperature state and associated constraints on this.

7.1 Traffic scenario – Slow-moving leading vehicle

Since time is a state in the problem formulation, including a constraint on minimum time, representing the movement of the leading vehicle, is straightforward. In this study, only the concept of safe and fuel optimal predictive energy management with the utilized problem formulation is analysed. Thus, any fuel comparisons to a baseline controller, e.g., a cruise controller with gap control functionality, is out of scope. Also, it is assumed here that the speed of the leading vehicle is known. A leading vehicle observer that predicts the future speed of the leading vehicle is presented in [23]. Utilizing such techniques would be beneficial for testing real-world driving scenarios. Moreover, any aerodynamic drag reduction due to proximity to the leading vehicle, as presented in [23], is not included.

Similarly to [14], the following constraint is added to the problem formulation given in Chapter 5

$$t(s) \geq t_{min}(s) \tag{7.1}$$

where $t_{min}(s)$ is calculated for the prediction horizon based on the predicted movement of the leading vehicle, plus a safety margin. The constraint on maximum travel time over the horizon is kept, but it is now adjusted to take the vehicle speed of the leading vehicle into consideration, in order to set a travel time limit that is feasible, given the traffic situation.

Figure 7.1 shows an example solution where there is a slow-moving vehicle at a position 150 m in front of our vehicle, driving at 70 km/h. The time margin to

the leading vehicle is set to 2 s. As the figure shows, this limit is always respected. In the beginning of the horizon, the vehicle starts slowing down early. Over the crest, the vehicle then drops to a speed lower than the speed of the leading vehicle, and in the downhill it accelerates, and charges the battery, to a speed faster than the speed of the leading vehicle. Towards the end of the horizon the vehicle is at the time margin of 2 s.

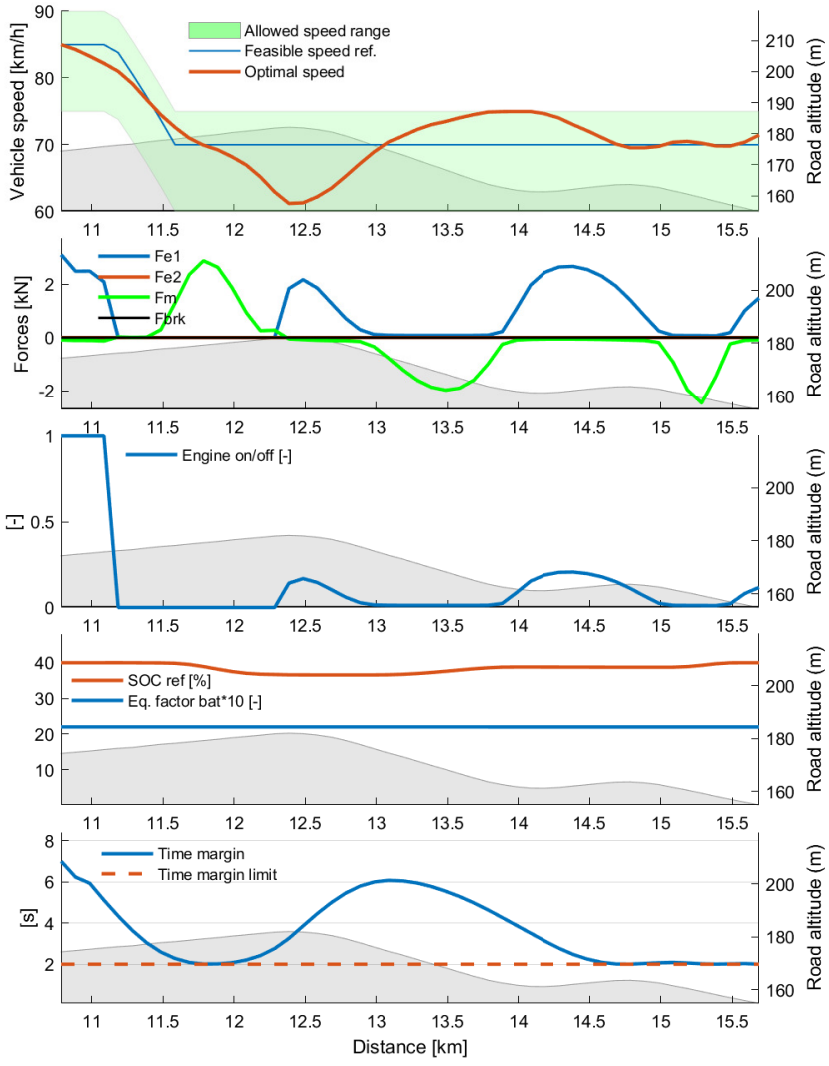


Figure 7.1: *Traffic scenario with a slow-moving leading vehicle. The top subplot shows that the vehicle speed is reduced below the speed of the leading vehicle at 70 km/h over the crest and then accelerates in the downhill to a speed faster than the speed of the leading vehicle. The bottom subplot shows that the time margin limit is always respected.*

7.2 Battery dynamic power limitations

The battery power limitations typically vary depending on the time duration of the charge or discharge pulse. For example, a higher power take out is allowed if it lasts for 2 s, compared to 10 or 30 s. The instantaneous power limitation is referred to as state of power, SOP. Battery management systems can sometimes also provide predictions of SOP for different time steps. These SOP predictions represent the maximum constant power allowed over a certain time window. For the predictive energy management problem, horizons over several minutes are often of interest. However, SOP predictions for these time spans might not always be available. Furthermore, the SOP predictions are, as mentioned above, often given as constant power levels for a defined time. Road topography typically implies irregular charge and discharge patterns, which might be difficult to map to the SOP predictions given by the battery management system. SOP estimation is a complex problem, see, e.g., [25], and including a fully accurate model in the energy management problem is not practical. However, a simplified model, which puts a limit on the thermal load is more reasonable to include. For these reasons, including the battery temperature state in the optimal control problem is investigated in this section. Moreover, the battery thermal management system can be a significant power consumer, and the derived model can also be a step on the way to an optimization scheme that considers both the propulsion and the thermal management, as in, e.g., [20].

7.2.1 Battery temperature state modelling

A simple dynamic model to describe the battery cell temperature is

$$C_B \frac{dT_B}{dt} = P_{B,loss} + H_B(T_{coolant} - T_B) \tag{7.2}$$

where T_B is the battery average cell temperature, $T_{coolant}$ is the battery coolant temperature, C_B is the battery heat capacity, H_B is the battery heat transfer coefficient, and $P_{B,loss}$ is the battery losses. Heat transfer to ambient air through the battery case is not included here since the battery is assumed to be well insulated. Reformulation to space domain yields

$$\begin{aligned} C_B \frac{dT_B}{ds}\frac{ds}{dt} &= P_{B,loss} + H_B(T_{coolant} - T_B) \Rightarrow \\ C_B \frac{dT_B}{ds} &= \frac{P_{B,loss}}{v} + \frac{H_B(T_{coolant} - T_B)}{v} \Rightarrow \\ C_B \frac{dT_B}{ds} &= F_{B,loss} + \frac{H_B(T_{coolant} - T_B)}{\sqrt{(2E_K/m)}} \end{aligned} \tag{7.3}$$

The time derivative as a function of distance, i.e., $t'(s) = 1/v = 1/\sqrt{(2E_K/m)}$ is approximated in Equation 5.8. Inserting this approximation in Equation 7.3 yields

$$\begin{aligned} C_B \frac{dT_B}{ds} &= F_{B,loss} + H_B(h_0 + h_1 E_K)(T_{coolant} - T_B) \Rightarrow \\ C_B \frac{dT_B}{ds} &= F_{B,loss} + H_B(h_0(T_{coolant} - T_B) + h_1(E_K T_{coolant} - E_K T_B)) \end{aligned} \tag{7.4}$$

Due to the multiplication of E_K and T_B, this is a nonlinear expression. Linearisation of a multivariable function $f(x)$ at a point a is given by

$$f(x) \approx f(a) + (\nabla f)_a \cdot (x - a) \tag{7.5}$$

The function $f(x)$ and the point for linearisation are then defined as

$$f(x) = f(E_K, T_B) = E_K T_B \tag{7.6}$$

$$a = \left(\hat{E}_K, \hat{T}_B\right) \tag{7.7}$$

This yields

$$\nabla f = \begin{bmatrix} T_B \\ E_K \end{bmatrix} \tag{7.8}$$

$$(\nabla f)_a = \begin{bmatrix} \hat{T}_B \\ \hat{E}_K \end{bmatrix} \tag{7.9}$$

Inserting Equations 7.6, 7.7 and 7.9 in Equation 7.5 yields

$$\begin{aligned} E_K T_B &\approx \hat{E}_K \hat{T}_B + \begin{bmatrix} \hat{T}_B \\ \hat{E}_K \end{bmatrix} \cdot \begin{bmatrix} E_K - \hat{E}_K \\ T_B - \hat{T}_B \end{bmatrix} \\ &= \hat{E}_K \hat{T}_B + \hat{T}_B(E_K - \hat{E}_K) + \hat{E}_K(T_B - \hat{T}_B) \\ &= -\hat{E}_K \hat{T}_B + \hat{T}_B E_K + \hat{E}_K T_B \end{aligned} \tag{7.10}$$

where the final expression above is linear. Inserting this in Equation 7.4 yields

$$\begin{aligned} C_B \frac{dT_B}{ds} =& F_{B,loss} + H_B(h_0(T_{coolant} - T_B) \\ &+ h_1(E_K T_{coolant} + \hat{E}_K \hat{T}_B - \hat{T}_B E_K - \hat{E}_K T_B) \end{aligned} \tag{7.11}$$

Equation 7.11 is added to the control formulation presented in Chapter 5, see Equations 5.9-5.26. In addition, a constraint on maximum temperature is added

$$T_B \leq T_{B,max} \tag{7.12}$$

where $T_{B,max}$ is the maximally allowed battery average cell temperature.

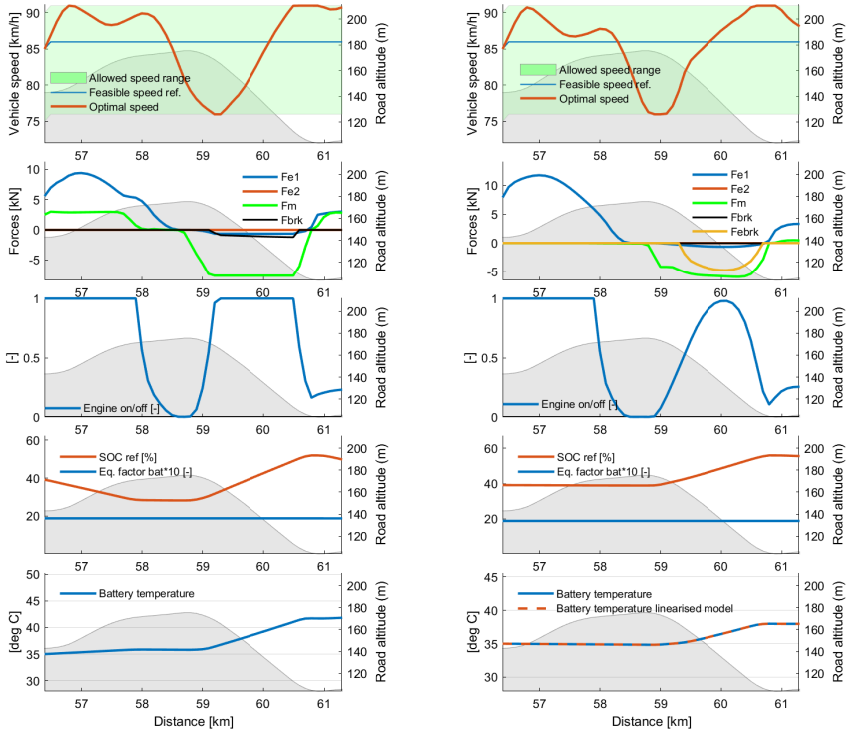
7.2.2 Results

Figure 7.2 shows the optimization results for one prediction horizon, including a fairly steep downhill. With the battery temperature modelled, the predictive energy management controller chooses to use the electric machine more conservatively, i.e., avoiding any propulsion and mainly uses it to recuperate the free energy in the downhill as much as possible, while respecting the imposed temperature limit, which here was set to 3 degrees above the initial temperature. The fourth subplot in the figure to the left, without the battery temperature model, shows that the SOC is decreasing in the first part of the horizon, while the figure to the right shows a constant SOC in the same part of the horizon. This is because the controller in this case chooses to wait to use the electric machine until the downhill. It is interesting to note that the equivalence factors are almost identical between the two cases. An expected result might otherwise be that the right figure should have a bit higher equivalence factor, to make lower layers in the functional architecture use less battery energy for propulsion. However, the lower layers also consider the optimal SOC reference, hence a similar electric machine power pattern as the figure shows will finally be actuated by the low-level controllers.

To avoid large relaxation errors on the electric machine losses, $F_{M,loss}$, a small penalty on squared electric machine force, F_M, and losses, $F_{M,loss}$, was imposed. Also, an engine brake force variable was introduced, without penalization in the cost function, having a constant minimum force limit

$$c_0 e_{on/off} \leq F_{Ebrk} \leq 0 \tag{7.13}$$

where c_0 is a constant parameter fitted to the engine brake minimum force curve. It is not fully clear why these modifications are needed but a possible reason could be that since brake force is penalized in the cost function, and the battery losses are limited through the battery temperature constraint, excess brake force end up in the F_M and $F_{M,loss}$ variables, if not introducing the above mentioned changes.



(a) *Without battery temperature state inclusion.*

(b) *With battery temperature state inclusion.*

Figure 7.2: *Optimization results over one prediction horizon, representing a fairly steep downhill, with and without the battery temperature state included. With the battery temperature modelled, the electric machine is used more conservatively for propulsion in this horizon, in order to make sure that as much energy as possible can be regenerated in the downhill.*

8 Discussion

8.1 Results

Existing literature splits the energy management problem in different ways, compromising computational load vs degree of suboptimality. This thesis chose to include only one integer decision variable in the receding horizon controller, leading to a MIQP, as an attempt to decrease the suboptimality but still not increase the computational load substantially. Whether the load is too high or not is hard to argue about at this point. The section below discusses the chosen solver and what a customized solver could bring about. Moreover, exploring early termination and improving the heuristic of the branch-and-bound algorithm might reduce the computational load further. However, with the increasing computational capabilities of on-board control units, and the fact that the update rate of the predictive energy management controller may be fairly slow, pursuing optimality is reasonable approach.

Existing literature also include different energy buffers and states in the problem formulation, representing different systems and physical domains, which might not only include the powertrain but also auxiliary systems. QP scales well with increasing continuous states as compared to, e.g., DP. Powertrain technology is evolving rapidly, and the electrification improves controllability, hence the scalable features of the QP are interesting.

Regarding the generalizability of the results it should be noted that there are several calibration parameters in the cost function that have been calibrated to achieve reasonable results for the base case with four binary variables and holding the binary variables constant over four segments. Under a different calibration it is possible that the results would be a bit different and that the sensitivity analysis could possibly point in other directions to some degree. Also, the results are generated with a complete vehicle simulation model with several controllers

in the layers below the predictive energy management controller, e.g., the predictive gear selection controller. It might be that these controllers are somewhat sensitive to the input from the predictive energy management controller, and thereby slightly affecting the final fuel consumption results.

8.2 Method

Since this thesis focuses on the potential of the selected control strategy, it did not evaluate commercial, efficient and tailored QP solvers for embedded targets and receding horizon control problems. Such work needs to be performed in order to understand what the actual load on a vehicle control unit would be.

The controller was evaluated in a complete vehicle simulation model, and not in real-world testing in a truck. One could argue that the former is more suitable to draw conclusions on the fuel saving potential due the possibility of repeatability in a virtual environment. However, actually testing in a real vehicle would of course have provided further insights into the controller behaviour and a better representation of the actual limitations and dynamics of hardware components such as the battery and the engine. Furthermore, to draw more certain conclusions, testing on more driving cycles and with different vehicle mass would be needed.

8.3 The work in a wider context

With an increasing concern on the environmental impact of the transportation sector and the desire to reduce cost, fuel efficient hybrid electric vehicles have the possibility to be a part of the solution to these aspects. The control, in particular, can reduce the fuel consumption without hardware modifications. This thesis points to the potential of optimally and predictively controlling the vehicle.

9

Conclusions and future work

The purpose of this thesis is to develop a predictive energy management controller, which minimizes the fuel consumption of a long-haul hybrid truck by optimizing the vehicle speed reference, battery energy reference, and engine on/off decision. To this end, a receding horizon controller using a mixed-integer quadratic programming (MIQP) formulation was developed. The binary engine on/off decision was included in the control formulation as an optimization variable. The main contributions of the thesis are the MIQP control problem formulation, the strategy to solve this with the branch-and-bound method, and a sensitivity analysis.

With the developed predictive energy management controller applied on a complete vehicle simulation model, fuel savings between 10-15%, depending on the driving cycle, is achieved compared to a conventional truck. The hybrid truck without the predictive control saves significantly less. Fuel consumption is reduced by 3-8% in this case.

The sensitivity analysis showed that holding the binary engine on/off control signal constant for at least two segments in the horizon seems most promising from a fuel consumption perspective. The average number of branch-and-bound QP iterations seems, however, to moderately increase with the number of segments the binary variable is held constant. Adding more binary variables increases the required number of iterations. However, compared with the brute-force method, the branch-and-bound algorithm scales at a significantly lower rate on the studied problem. Introducing a simple heuristic improves the branch-and-bound performance but gives a slightly higher fuel consumption. Moreover, the report indicates that, given that the controller does not need a fast update rate, the algorithm is faster than real-time requirements. This means that future online implementation in a vehicle control unit is not unreasonable.

Furthermore, it is shown that a suboptimal solution, using the initial incum-

bent as the solution, gives fuel consumption results 0.4% higher than the results with the branch-and-bound method. The initial incumbent is calculated based on a simple heuristic that require two QP iterations constantly.

The thesis also demonstrates that the problem formulation handles well the inclusion of traffic and the controller can keep a safe time margin to the leading vehicle. Moreover, adding the battery temperature as a state is shown to be a possibility to approximately model the dynamic battery power limitations over the prediction horizon.

Future work may include further investigation of branch-and-bound heuristics and early termination to reduce the peak number of iterations. Another interesting work would be to include the gear selection in the problem formulation. Also, generating a tailor-made solver for the control problem formulation is a future work to better understand the computational and memory requirements.

Bibliography

[1] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[2] European Commission. Reducing CO2 emissions from heavy-duty vehicles, 2021. URL `https://ec.europa.eu/clima/policies/transport/vehicles/heavy_en`. Accessed: 2021-06-05.

[3] European Commission. Paris agreement, 2021. URL `https://ec.europa.eu/clima/policies/international/negotiations/paris_en`. Accessed: 2021-06-05.

[4] P. Elbert, T. Nüesch, A. Ritter, N. Murgovski, and L. Guzzella. Engine on/off control for the energy management of a serial hybrid electric bus via convex optimization. *IEEE Transactions on Vehicular Technology*, 63(8):3549–3559, 2014.

[5] H. J. Ferreau, S. Almér, R. Verschueren, M. Diehl, D. Frick, A. Domahidi, and C Jones. Embedded optimization methods for industrial automatic control. *IFAC-PapersOnLine*, 50(1):13194–13209, 2017.

[6] Roger Fletcher and Sven Leyffer. Numerical experience with lower bounds for MIQP branch-and-bound. *SIAM Journal on Optimization*, 8(2):604–616, 1998.

[7] Matthias Gerdts. Solving mixed-integer optimal control problems by branch&bound: a case study from automobile test-driving with gear shift. *Optimal Control Applications and Methods*, 26(1):1–18, 2005.

[8] M. Held, O. Flärdh, F. Roos, and J. Mårtensson. Optimal freewheeling control of a heavy-duty vehicle using mixed integer quadratic programming. *arXiv preprint arXiv:2002.05944*, 2020.

[9] Erik Hellström, Jan Åslund, and Lars Nielsen. Design of an efficient algorithm for fuel-optimal look-ahead control. *Control Engineering Practice*, 18 (11):1318–1327, 2010.

[10] Erik Hellström, Jan Åslund, and Lars Nielsen. Horizon length and fuel equivalents for fuel-optimal look-ahead control. *IFAC Proceedings Volumes*, 43(7):360–365, 2010.

[11] Erik Hellström, Jan Åslund, and Lars Nielsen. Management of kinetic and electric energy in heavy trucks. *SAE International Journal of Engines*, 3(1): 1152–1163, 2010.

[12] M. Hovgard, O. Jonsson, N. Murgovski, M. Sanfridson, and J. Fredriksson. Cooperative energy management of electrified vehicles on hilly roads. *Control Engineering Practice*, 73:66–78, 2018.

[13] L. Johannesson, N. Murgovski, E. Jonasson, J. Hellgren, and B. Egardt. Predictive energy management of hybrid long-haul trucks. *Control Engineering Practice*, 41:83–97, 2015.

[14] Lars Johannesson, Magnus Nilsson, and Nikolce Murgovski. Look-ahead vehicle energy management with traffic predictions. *IFAC-PapersOnLine*, 48(15):244–251, 2015.

[15] Johan Löfberg. Yalmip : A toolbox for modeling and optimization in matlab. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

[16] Jan Lundgren, Mikael Rönnqvist, and Peter Värbrand. *Optimeringslära*. Studentlitteratur, 3 edition, 2003.

[17] N. Murgovski, L. Johannesson, J. Sjöberg, and B. Egardt. Component sizing of a plug-in hybrid electric powertrain via convex optimization. *Mechatronics*, 22(1):106–120, 2012.

[18] Nikolce Murgovski, Bo Egardt, and Magnus Nilsson. Cooperative energy management of automated vehicles. *Control Engineering Practice*, 57:84–98, 2016.

[19] Simona Onori, Lorrenzo Serrao, and Giorgio Rizzoni. *Hybrid electric vehicles: Energy management strategies*. Springer, 2016.

[20] H.T Pham, P.P.J Van Den Bosch, J.T.B.A Kessels, and R.G.M Huisman. Integrated energy and thermal management for hybrid electric heavy duty trucks. *IEEE Vehicle Power and Propulsion Conference*, pages 932–937, 2012.

[21] Michael Schittler. State-of-the-art and emerging truck engine technologies for optimized performance, emissions and life cycle costs. *DaimlerChrysler AG (US).*, 2003.

[22] Robert Sedgewick and Kevin Wayne. *Algorithms*. 4 edition. URL `https://algs4.cs.princeton.edu/home/`.

[23] N. K. Sharma, A. Hamednia, N. Murgovski, E. R. Gelso, and J. Sjöberg. Optimal eco-driving of a heavy-duty vehicle behind a leading heavy-duty vehicle. *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[24] C. Sundström, A. Voronov, O. Lindgärde, and A. Lagerberg. Optimal speed and gear shift control of long-haulage trucks. *IFAC-PapersOnLine*, 52(5): 471–477, 2019.

[25] C. Zou, A. Klintberg, Z. Wei, B. Fridholm, T. Wik, and B. Egardt. Power capability prediction for lithium-ion batteries using economic nonlinear model predictive control. *Journal of Power Sources*, 396:580–589, 2018.