ISSN 0280-5316
ISRN LUTFD2/TFRT--5861--SE

Optimization based Control Strategy for Energy Efficient Decelerations in an Automobile Cruise Controller

Arvid Rudberg

Lund University
Department of Automatic Control
August 2010

Lund University
Department of Automatic Control
Box 118
SE-221 00 Lund Sweden

Document name
MASTER THESIS

Date of issue
August 2010

Document Number
ISRN LUTFD2/TFRT--5861--SE

Author(s)
Arvid Rudberg

Supervisor
Henrik Wigermo at BMW AG München, Germany
Karl-Erik Årzén Automatic Control, Lund (Examiner)

Sponsoring organization

Title and subtitle
Optimization based Control Strategy for Energy Efficient Decelerations in an Automobile Cruise Controller (Optimerarbaserad styrstrategi för energieffektiva deceleratgioner i farthållare).

Abstract

With global warming and increased uncertainty about future vehicle fuel supply, the need for fuel efficient vehicles is growing. In this Master Thesis, design and implementation of a control strategy for fuel efficient deceleration is presented. The resulting deceleration controller, implemented in Embedded Matlab and Simulink, provides autonomous deceleration functionality to a vehicle cruise controller. Deceleration patterns are produced through online optimization of a cost function comprising fuel consumption and time loss. The cost function is calculated based on navigation information for the predicted driving path, which is also used to detect upcoming deceleration situations.
The potential for fuel savings is estimated through simulation, and experiences from tentative driving tests are presented. The optimization approach proves to yield flexibility in balancing fuel savings versus time loss, while being fast enough to be run online as a prototype function in a test vehicle. The deceleration optimizer function could either be used in its current form in future cruise controllers, or as a tool for development of simplified deceleration strategies.

Keywords

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280-5316

ISBN

Language
English

Number of pages
81

Recipient's notes

Security classification

http://www.control.lth.se/publications/

Acknowledgments

The work presented in this thesis was performed at the Energy Management Group at BMW Research and Development in Munich, Germany between October 2009 and April 2010. I would like to thank Mr. Christian Senger and Mr. Geert Schmitz for accepting me as Master Thesis student at their department. Of course, I owe much gratitude to my supervisor Mr. Henrik Wigermo for all advice and support during the project, and to my colleagues at EG-61 for their advice and cooperation. My thanks also to the colleagues at ZT-3, for rewarding teamwork, and to Mrs. Isaure Bouy, for adjusting reality to the needs of my deceleration optimizer.

I would also like to thank Prof. Karl-Erik Årzén at the Department of Control at Lund Institute of Technology for his approval of examining my Master Thesis project at the department and for his assistance along the way. Finally, many thanks to friends and family who helped me out by reading and commenting on the contents of this thesis, making sure numerous mistakes did not make it to the final version of this report.

Arvid Rudberg

Contents

1. Introduction

1.1 Background

Cruise controllers are widely available as standard or optional equipment for new automobiles. A conventional cruise controller regulates speed to a set-point that is manually set by the driver. Varying speed limits, traffic and other disturbances frequently forces the driver to adjust the set speed or to assume manual control of the speed regulation. A development from the regular speed controller is the *Adaptive Cruise Controller* (ACC) by BMW, which incorporates a forward-headed radar unit that allows for detection of slower traffic ahead and automatic speed adjustment to keep a safe distance to the car in front.

Modern cars equipped with integrated navigation systems with GPS and road maps, allow for a more sophisticated control strategy to be undertaken. By using the navigation information, a controller can be designed that adjusts the speed to current speed limits and reacts to upcoming changes in driving conditions in a timely and apprehensive manner. One interesting possibility is to use information about upcoming decreases in speed limit to decrease fuel consumption by early deceleration. The work presented in this thesis explores the possibilities of reducing fuel consumption by energy efficient deceleration, while maintaining a safe and time efficient way of driving. Plans for such fuel efficient deceleration are part of an endeavor by BMW to produce a fuel efficient Adaptive Cruise Controller, titled *GreenACC*.

1.2 Motivation

When driving a car, much energy is wasted while braking to adjust to varying speed limits and surrounding traffic. Braking occurs when fuel has been used to maintain a vehicle speed that is too high for an upcoming traffic situation and thus corresponds directly to excessive fuel usage. The losses can be reduced significantly by driving in an anticipating and apprehensive way. Decoupling and/or releasing the throttle well ahead of an upcoming speed limit decrease will allow the vehicle to decelerate without or with less braking, thus conserving fuel.

It is , however, often hard to anticipate speed limit changes early enough for any decisive action to be taken. Even anticipating driving on a well known stretch of road is hard and requires a lot of attention from the driver. It is hard to estimate the rolling distance of a vehicle and which combination of decoupled rolling and engine braking that will yield low fuel consumption while not being unnecessarily time consuming. A conventional speed controller does nothing to improve this situation. The driver must manually adjust for any changes in speed limits.

This situation can be improved by preemptive deceleration using navigation information as mentioned in the previous section. Calculating the vehicle's possible trajectories for the given geographic situation allows for deceleration profiles to be found that are both time and fuel efficient. A controller computing efficient deceleration trajectories will be denoted a *Deceleration Optimizer*.

1.3 Project Objectives

The objective of the project described in this thesis was composed of several parts:

- To develop a control strategy for fuel efficient deceleration
- To implement this strategy in Matlab and Simulink
- To estimate fuel savings through simulation
- To implement the deceleration controller in a vehicle for street measurements

1.4 Method

With a conventional car, the most fuel efficient way of decelerating from an initial speed v_0 to some target speed v_{target} is generally to decouple the engine at a time so that the correct speed is achieved at the correct position solely through decoupled rolling for the maximum distance possible. This driving scheme does, however, pose some problems. It is time consuming, since deceleration needs to be initiated quite a long distance before the actual decrease in speed limit. The pace of such an early deceleration is also generally slower than that of the surrounding traffic, and might cause uncomfortable or even dangerous situations if traffic is heavy.

Thus, when deceleration for the entire maximum deceleration distance is not desired, the question arises how a fuel efficient deceleration should be performed. With a conventional car there are three ways of decelerating with no throttle:

- Decoupled rolling
- Engine braking
- Braking

The different modes of deceleration have different deceleration rates and different fuel consumptions. The desired scenario is a deceleration profile that saves fuel, but that is also time efficient and does not put the driver in uncomfortable situations due to excessive deviation from the speed of the surrounding traffic. An example of a deceleration sequence on flat ground can be seen in Figure 1.1. By analyzing the equations of motion of a car, analytical functions can be found that describe the behaviour of the car for the different deceleration modes. This allows for the formulation of a cost function that describes the time and fuel costs for a certain deceleration pattern. Optimizing this function over the set of possible deceleration profiles yields solutions that are optimal with respect to the chosen weight coefficient that translates fuel and time costs into comparable quantities. By tuning this weight coefficient, coherent deceleration profiles can be produced for any speed transition and for various geographic conditions. The tuning has been performed first by running test-cases in Matlab and Simulink, then by testing in real traffic situations in a BMW 530i test car. The implementation has been made in Embedded Matlab and Simulink, using Real Time Workshop to compile the model into runnable code. Additionally, simulations for investigation of fuel efficiency have been made using Dymola.

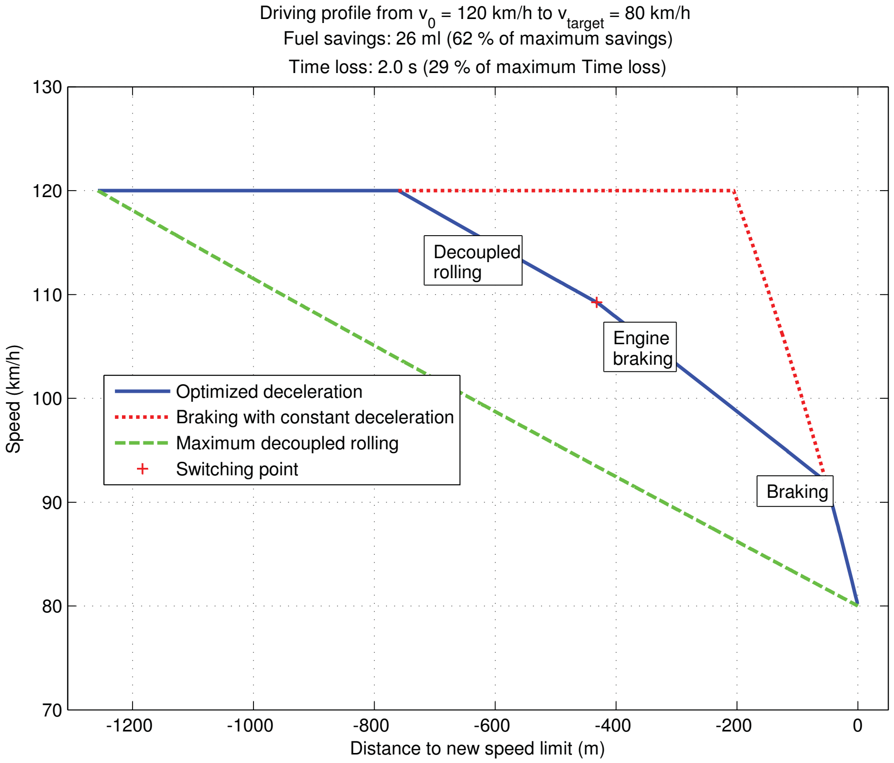


Figure 1.1 Sample Deceleration denoted with deceleration phases.

1.5 Related Work

The field of fuel consumption minimization for cars appeared along with the oil crisis in the 1970s. [Schwarzkopf and Leipnik, 1977] discussed fuel optimal control for close to constant speeds at various road grades, based on a predetermined trip time. A controller design for fuel efficient speed control was proposed, based on current slope and vehicle speed. In [Hellström *et al.*, 2009], look-ahead control of heavy trucks was performed using concurrent optimization of travel time and fuel consumption. The procedure was in many aspects similar to that undertaken in this thesis project. Navigation information was used to predict future driving conditions and the outcome in terms of fuel efficiency versus time loss was governed by a weighting coefficient for the time loss. Substantial fuel savings were obtained. However, the main concern in this study was fuel-efficient control for driving at close to constant speeds over relatively long distances. The question of optimal deceleration was not discussed.

1.6 Thesis Outline

In Chapter 2, a model for longitudinal vehicle dynamics is presented and the equations of motion for a decelerating car are derived. Chapter 3 discusses some fundamental principles of multiobjective optimization and a formulation of the optimization problem for the project at hand is being proposed. Thereafter, in chapters 4 and 5, the implementation of a function for online deceleration optimization is being discussed. First the actual optimization function, written in Embedded Matlab, is being presented. Thereafter the auxiliary Simulink model that couples the optimizer with either the navigation system and speed controller of the vehicle or a simulated vehicle environment is being treated. Chapter 6 summarizes results from fuel efficiency simulations. Finally, Chapter 7 covers the actual implementation in the test vehicle and experimental results. The thesis is rounded off with Chapter 8 in which another application of the Deceleration Optimizer is being described, presenting simulation results from optimized driving patterns for the Nürburgring Green Challenge, an eco-driving competition on the Nürburgring racing track. In the appendix, a list of terms and expressions as well as notations can be found.

2. Deceleration of an Automobile

This chapter describes the dynamics that govern the deceleration of a vehicle. The deceleration modes of a conventional car are described as well as the different components of the forces resisting the movement of an automobile during such a deceleration. In the following, the term *Driving Resistance* will be used to denote the sum of those forces.

2.1 Driveline Topology

Driving Wheels

The driving wheels of a car are the wheels to which the engine is connected and which thereby transmits the engine power from the car to the road. The three main possibilities are rear wheel drive, front wheel drive and all wheel drive. This greatly affects the driving properties of a car, but it has only minor influences on deceleration. Rear wheel drive is standard for all BMW's that do not have four wheel drive and therefore only the case of rear wheel drive has been considered in this thesis. The concept and all results concerning deceleration profiles could, however, be applied to all wheel drive and front wheel drive vehicles as well.

Gearbox type

Although there are many different ways to build automobile gearboxes, the two main types are automatic and manual gearboxes. Whether the car has automatic or manual transmission does not affect the general results concerning deceleration of the car. The available modes of deceleration are the same. However, the selected gear affects the resistance in the engine of a car during deceleration.

Gear Selection With a manual gearbox the driver choses which gear to use, and the switching points can therefore not be predicted with any precision. This means the deceleration pattern will also be hard to predict. In an automatic gearbox, the gear is selected by the control system of the gearbox, based on speed, gas pedal position and other factors. This means that in a situation where the driver does not push the gas pedal, the selected gear and the resulting driving resistance can be accurately predicted.

Automated Mode Switching Another aspect of the Deceleration Optimizer is the desired automation of vehicle deceleration control. Automatic switching between decoupled rolling and engine braking requires a clutch that can be operated by the vehicle's control system, without involvement of the driver.

In the manual gearboxes currently in use by BMW, the clutch is mechanically coupled to the clutch pedal, and decoupling can only be performed by pushing the pedal. This means that for manual gearboxes automatic decoupling is currently not possible. Using a different construction, with a manual clutch implemented in drive-by-wire topology where there is an electric signal controlling the clutch instead of the mechanic connection being used today, automatic decoupling could be made possible. This does, however, require large changes to the current construction that are not likely to be carried out in the near future.

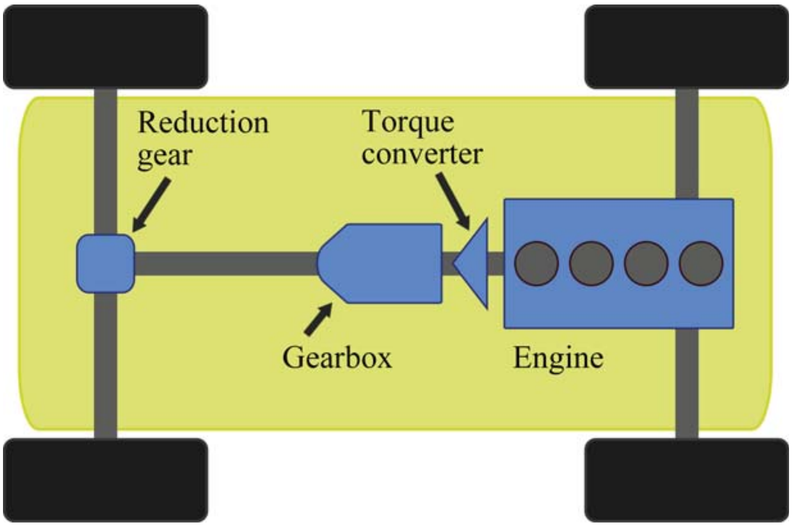


Figure 2.1 Driveline Topology of an Automatic Rear-wheel drive vehicle.

In an automatic gearbox of traditional design, decoupling of the wheels and the engine must be done by moving a mode selection lever from Drive position to Neutral position. In such a layout there is usually a mechanic coupling between lever and gearbox. This means automatic decoupling is not possible. In the gearboxes used in current BMW's the mechanic pathway has been replaced by an electric signal. By intercepting this electric pathway, directions can be given to the gearbox from other sources than the mode selection lever. This allows for other control units, such as a speed controller, to decouple the engine. Therefore, from now on, only vehicles with rear wheel drive and automatic gearboxes with electric mode selection will be considered.

Layout

The driveline layout for a rear wheel drive automatic driveline can be seen in Figure 2.1. Torque from the engine is transmitted by the torque converter to the gearbox. The task of the torque converter is to enable decoupling of the engine and the gearbox as well as acting as a reduction gear at low speeds. That is, it performs the same tasks as a manually operated clutch in a car with manual transmission. Torque is transmitted through a hydraulic coupling, with a pump on the motor side and a turbine on the gearbox side. In between is a stator which redirects oil flow from the turbine back to it. The action of the stator is the source of the torque conversion capabilities [Wallentowitz, 2001]. When engine speed and gearbox speed are reasonably synchronized a lock-up clutch is applied to mechanically couple the shafts to improve efficiency.

The gearbox in the BMW 530iA used in the vehicle experimentation is of planetary gear type with six gears. A gearbox of this type has a series of interconnected planetary gears that can be coupled to the torque converter or held motionless in different configurations in order to produce the gear ratios of different gears. When shifting into neutral position, the planetary gears are decoupled and freewheeling, so that the wheel side of the drivetrain is decoupled from the engine side.

From the gearbox, power is transmitted by the drive shaft to the rear gearbox, which is a fixed differential gear, allowing the wheels to rotate at different speeds and providing another gear reduction.

2.2 Operating Modes

Decoupled Rolling

Decoupled rolling (DR) means that the wheels and the engine are disconnected by the gearbox. The engine is idling at a low engine speed, thus consuming some fuel, but the combined energy losses in form of kinetic and chemical energy are generally smaller than if the engine would be cranked by the wheels at a higher speed. Because the engine is run at a fixed idle speed, fuel losses are proportional to the time spent in DR mode. Kinetic energy is dissipated through friction and tire compression losses to the road, aerodynamic drag and friction in the drivetrain on the wheel-side of the gearbox.

Engine Braking

Engine braking (EB) in a conventional car denotes the process of letting the vehicle roll with engine coupled to the wheels but no throttle. Energy is then dissipated in the same fashion as in the case of decoupled rolling and additionally through friction and compression losses in the engine. This means the maximum rolling distance is significantly lower than for decoupled rolling, but on the other hand no fuel is injected during engine braking. The rate of deceleration is largely dependent on the selected gear. In the following, gear selection will not be used actively by the deceleration optimizer but will be left for the gearbox controller unit to decide. While engine braking with a BMW automatic gearbox, downward gear switching occurs when engine speed approaches the engine idling speed. This means that the selected gear can be determined as a function of vehicle speed during deceleration in EB mode.

Regular Braking

Regular braking (RB) means applying a braking force on the wheels and can be done irrespectively of the clutch position. When braking in coupled position no fuel is used, but energy is lost through friction in the braking system instead of driving resistance, and thus in practice lost for no good cause. Braking in decoupled mode is an outright waste of energy, because fuel is used to keep the engine running in idle while energy is dissipated as heat in the braking system, and will therefore not be considered.

2.3 Longitudinal Vehicle Dynamics

When studying acceleration or deceleration of a car, it is only necessary to take into account the dynamics in the direction of movement of the car. The influence of lateral forces on the deceleration process is negligible during normal driving. Since the primary interest of this study is to predict the car's behavior while performing decoupled rolling and engine braking, a simplified model will be used that is accurate for sequences with moderate acceleration. The models used have been compiled from [Woll, 2005].

The resistive forces acting on a vehicle driving up a slope are shown in Figure 2.2. Positive direction will be taken as the direction of $\hat{(v)}$, which is the direction in which the vehicle moves. In the coming sections the different components of the driving resistance will be discussed.

Rolling Resistance

The rolling resistance F_{roll} of a vehicle is primarily due to losses in the tires. When

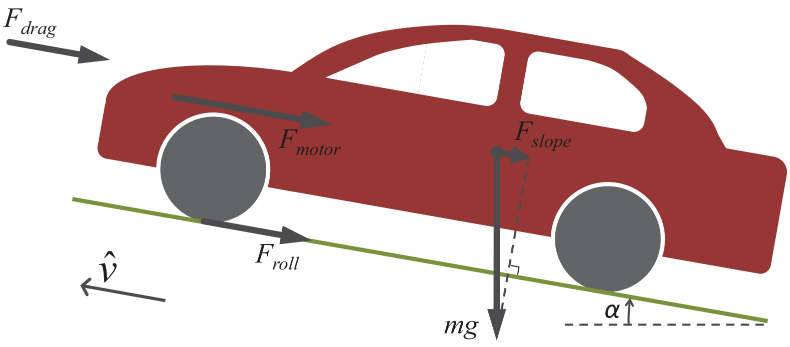


Figure 2.2 Resistive forces acting on an automobile

the sides of the tire and the tire pattern gets loaded, energy losses occur during the compression and decompression of the rubber. Additionally, losses occur in the wheel axle bearings and in the wheel side of the transmission. The rolling resistance is proportional to the normal force on the tire and the resultant force can be modeled as in Equation (2.1).

$$F_{roll} = -\mu_R mg \tag{2.1}$$

where m is the vehicle mass, g is the gravitational acceleration and μ_R is a rolling resistance coefficient. This coefficient is approximately constant up to a certain threshold speed and then increases with increasing velocity. The handling of this variation in the coming calculations will be discussed more thoroughly in Section 2.5.

The mg factor in the expression for rolling resistance should in exact calculations be replaced with the normal force F_N acting on the wheels, which is dependent on the slope α through $F_N = mg\cos(\alpha)$. However, in drivable slopes the deviation in F_{roll} is small compared to the slope resistive force F_{slope} which will be introduced later. Therefore F_{roll} will be approximated as independent from slope.

Aerodynamic Drag

Aerodynamic drag is caused mainly by the pressure difference between front and rear of the car. A smaller fraction is due to friction, both against the exterior of the car and from air flowing through the engine room and the interior of the car. For a body moving at high speeds, the drag force F_{drag} is closely described by the drag equation

$$F_{drag} = -\tfrac{1}{2}c_w A_f \rho v^2 \tag{2.2}$$

where c_w is the drag coefficient, A_f is the frontal area of the car, ρ is the air density and v is the vehicle speed. Current air density is dependent on outside temperature T and air pressure p and can be calculated by [Woll, 2005]

$$\rho = 1.293 \frac{273}{T+273} \frac{p}{1013} \tag{2.3}$$

where T is in °C and p is in mbar

Motor Resistance

Depending on the mode of operation, the force F_{motor} acting on the car originating from the motor varies. During DR $F_{motor} = 0$, since it is decoupled from the driveline, but during EB the increased deceleration because of the motor drag torque must be separately accounted for. The added force resistive to the movement of the vehicle is mainly due to friction in the motor. A small part is also due to compression losses, but most of the compression energy is returned to the crankshaft during a full stroke of the piston. During EB, the resulting force from motor resistance can be expressed as

$$F_{motor} = F_{EB} = -\frac{N_{rag}}{r_w}\tau_f N_i. \tag{2.4}$$

where N_{rag} is the gear ratio of the rear axle reduction gear, r_w is the wheel radius, τ_f is the friction torque of the engine side of the driveline, and N_i is the current gearbox gear ratio. N_i is gear dependent and τ_f varies with engine speed. Variation in τ_f is discussed in Section 2.5. In an exact calculation the wheel radius r_w is also speed dependent and grows larger for high speeds. This effect is, however, negligible at moderate speeds and r_w will therefor be considered as constant.

Slope

The contributing force on the vehicle from moving up or down a slope, F_{slope}, is easily calculated from the mass and the angle of the slope as

$$F_{slope} = -mg\sin(\alpha). \tag{2.5}$$

Obviously the action of this force depends on the direction of the slope and in steep enough slopes F_{slope} will cause acceleration of the vehicle in DR and even in EB mode.

2.4 Equations of Motion

The aim of the project at hand is to calculate optimized deceleration profiles for upcoming deceleration sequences. The optimization must be carried out online, since it is dependent on the current driving situation and geographical information. The optimization procedure implies calculating trajectories for a lot of different possible deceleration patterns. The trajectories could be accurately obtained by numerical integration of the differential equations involved. However, considering the quantity of trajectories that need to be evaluated during the optimization this is not a feasible solution. Therefore analytical functions approximately describing the motion of the vehicle will be sought.

Derivation of Equations of Motion

Adding up equations (2.1) - (2.5) and applying Newton'second law yields:

$$\begin{aligned} ma &= F_{roll} + F_{drag} + F_{slope} + F_{motor} \\ &= -\mu_R mg - mg\sin(\alpha) - \tfrac{1}{2}c_w A\rho v^2 + F_{motor} \qquad (2.6) \\ \Rightarrow\ a &= \dot{v} = A_{mode} + Bv^2 \qquad (2.7) \end{aligned}$$

where

$$B = \frac{c_w A\rho}{2m}$$

and A_{mode} is dependent on the deceleration mode through F_{motor}. The A-parameters for decoupled rolling and engine braking will be denoted A_{DR} and A_{EB} respectively. They are given by

$$A_{DR} = A_{roll} + A_{slope} = -g\left(\mu_R + sin(\alpha)\right) \tag{2.8}$$

and

$$A_{EB} = A_{DR} + A_{motor} = A_{DR} - \frac{N_{rag}\tau_f N}{mr_w} \tag{2.9}$$

The nonlinear differential equation defined in (2.7) has as solutions the equations of motion that describe the trajectory for the vehicle. Several of the variables included in the A and B coefficients vary with driving conditions. Especially influential parameters are slope, motor drag torque and air density.

To emphasize this complexity (2.7) can be written as

$$a = \dot{v} = \ddot{s} = A_{mode}(s) + B(s)\dot{s}^2$$

where s is the distance travelled from a given point. It is generally not possible to find an analytical solution to this equation. However, approximating A and B as constants over an interval, with initial speed v_0, allows for an analytical solution to (2.7) to be found.

$$v(t) = \frac{\tan\left(t\sqrt{AB} + \arctan\left(v_0\sqrt{B/A}\right)\right)}{\sqrt{B/A}} \tag{2.10}$$

Integrating[1] (2.10) over time yields

$$\begin{aligned} s(t) = \int_0^t v(\tau)\,d\tau = \frac{-1}{2B}\Big[2\ln\left(\sqrt{AB} - \tan\left(t\sqrt{AB}\right)Bv_0\right) \\ +\ln\left(1 + \tan^2\left(t\sqrt{AB}\right)\right) + \ln(AB) + 2Bs_0\Big] \end{aligned} \tag{2.11}$$

$s(t)$ is the distance travelled at time t and s_0 is the distance at $t = 0$.

Speed as a function of distance

Equation (2.7) gives us an expression for acceleration. We also have

$$a = \frac{dv}{dt} \text{ and } v = \frac{ds}{dt} \quad\Rightarrow\quad ds = v \cdot dt = v\frac{dv}{a} \quad\Rightarrow\quad \frac{ds}{dv} = \frac{v}{a} = \frac{v}{A + Bv^2}.$$

Integrating over v

$$s(v) = \int \frac{ds}{dv}dv = \int \frac{v}{A + Bv^2}dv = \frac{1}{2B}\left[\ln\left|A + Bv^2\right|\right] + C.$$

Assuming $s(v_0) = 0$ yields $C = -\frac{1}{2B}\ln\left|A + Bv_0^2\right| \Rightarrow$

$$s(v) = \frac{1}{2B}\left[\ln\left|A + Bv^2\right| - \ln\left|A + Bv_0^2\right|\right] \tag{2.12}$$

[1] The actual function was obtained using Matlab's Symbolic Toolbox. Integration was performed by solving $\dot{s} = f(t)$ using the symbolic differential equation solver DSOLVE.

For the problem at hand, the function $v(s)$ is also of interest. It can be found by straighforward inversion of (2.12):

$$
\begin{aligned}
& 2Bs(v) + \ln\left|A + Bv_0^2\right| = \ln\left|A + Bv^2\right| \\
\Rightarrow\ & \exp\left(2Bs(v) + \ln\left|A + Bv_0^2\right|\right) = A + Bv^2 \\
\Rightarrow\ & v(s) = \sqrt{\frac{1}{B}\left[\exp\left(2Bs(v) + \ln\left|A + Bv_0^2\right|\right) - A\right]} \\
\Leftrightarrow\ & v(s) = \sqrt{\frac{A + Bv_0^2}{B}\exp(2Bs) - \frac{A}{B}} \qquad (2.13)
\end{aligned}
$$

The derivation of the equations of motion above leave certain questions unanswered, especially regarding continuity over the desired intervals and complex function values. A deeper mathematical analysis of these functions is not within the scope of this project and the appropriateness of the functions for the task at hand has been verified experimentally through testing in Matlab. When yielding complex results, the real part of the function value has been used.

2.5 Model Parametrization

The vehicle available for real driving experiments with the GreenACC is a BMW 530iA Limousine. This is one of the main models in the BMW product range, and possibly one of the first to be equipped with a series version of the GreenACC. This model was therefore chosen also for use in simulations. The parameters required for calculation of the rolling trajectory of the 530iA were gathered from the simulation models used for vehicle simulation at EG-61, the department where the work was carried out. Table 2.1 shows the relevant drivetrain and air resistance parameters for the 530iA.

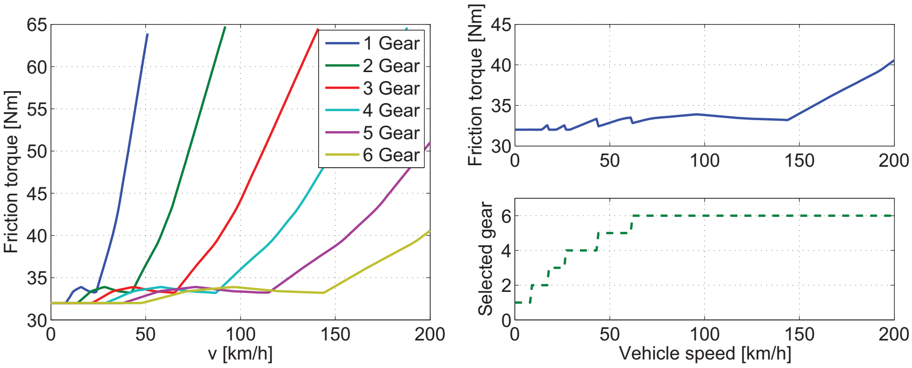
Parameter	Notation	Value	Unit
Unloaded weight	m_{unl}	1540	(kg)
Driving weight	m	1740	(kg)
Rear axle gear ratio	N_{rag}	3.640	(-)
Gearbox ratio, Gear 6-1	N_i	0.69, 0.87, 1.14, 1.52, 2.34, 4.17	(-)
Drag coeff.	c_d	0.264	(-)
Frontal area	A	2.260	(m^2)
Friction torque	τ_f	33	(N m)
Rolling resistance coeff.	μ_r	0.010	(-)

Table 2.1 Vehicle Model Parameters

The first six parameters in Table 2.1 are well defined constants and pose no further questions. The choice of representing friction torque and rolling resistance coefficient as constants does, however, require some justification.

Friction torque

Friction torque τ_f increases with engine speed. Its variance over vehicle speed for different gears can be viewed in Figure 2.3(a). Clearly, high engine speeds leads to high friction losses in the engine. When performing active engine braking to reduce speed during downhill driving, this phenomenon is exploited by using low gears, forcing the engine to run at high speeds and thereby increase the rolling resistance. Contrary, when performing engine braking with an automatic gearbox in drive mode, downward gear changes occur quite late to avoid unnecessary losses in the engine. The gear changing pattern for the 530iA is displayed in Figure 2.3(b), along with the resulting friction torque curve for deceleration with automatic gear downshift. As can be seen from the figure, the combined friction torque is quite constant over a wide range of speeds, and only starts increasing significantly at about 150 km/h. Approximating τ_f as constant at about 33 N m therefore appears feasible, at least for speeds under 150 km/h. In Figure 2.4, the engine braking resistive force is shown calculated for exact and approximated calculations. For speeds up to 150 km/h there is a good match between the two. For higher speeds the deviation starts growing large.



(a) Friction torque per gear   (b) Combined friction torque

Figure 2.3 Friction torque variation for the BMW 530iA

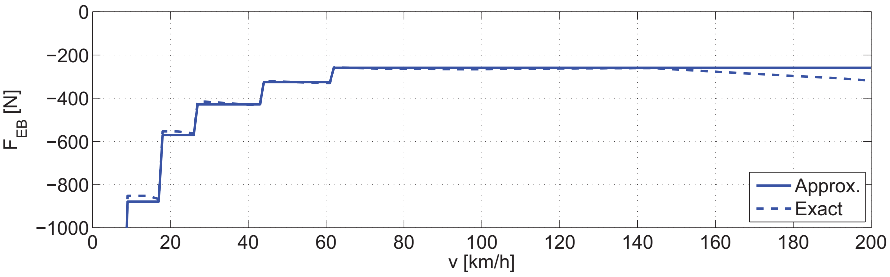


Figure 2.4 Approximate and exact calculations of forces due to engine friction torque

Rolling Resistance Coefficient

The rolling resistance μ_r depends on the tire type and increases with speed. For standard auto tires μ_r is constant at about 0.010 up to 100 km/h and increases to about 0.011 at 150 km/h [Woll, 2005]. For speeds over 150 km/h, the increase in μ_r is quite important. In the vehicle simulations performed at EG-61, a mapping is used to produce a speed dependent μ_r. However, in this project a constant μ_r is desirable. Figure 2.5 displays F_{roll}, computed twice, once approximated with a constant μ_r and once with the speed dependent μ_r from the EG-61 vehicle model. At high speeds, the approximated rolling resistance differs quite a lot from the exact one.

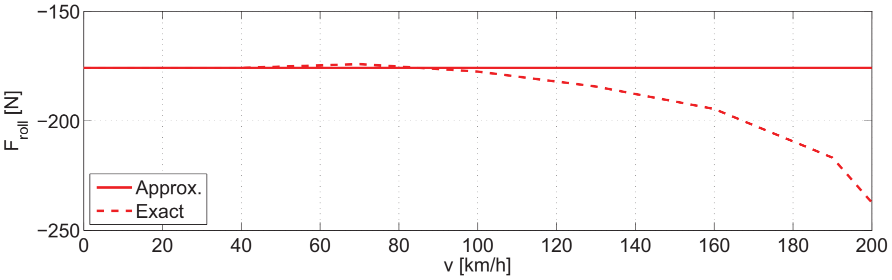


Figure 2.5 Approximate and exact calculations of resistive forces

Synthesis

It seems that approximating τ_f and μ_r as constants would be feasible up to vehicle speeds somewhere around 150 km/h and thereafter yields very poor results. Fortunately, there is a mechanism that counteracts the growing errors in the approximated rolling resistance and engine friction resistance. This mechanism is the aerodynamic drag, modeled as in Equation (2.2), which grows with the square of the vehicle speed. The drag force is displayed in Figure 2.6. It is clear that although being of minor influence at low speeds, the drag is becoming the dominating component among the resistive forces at high speeds.

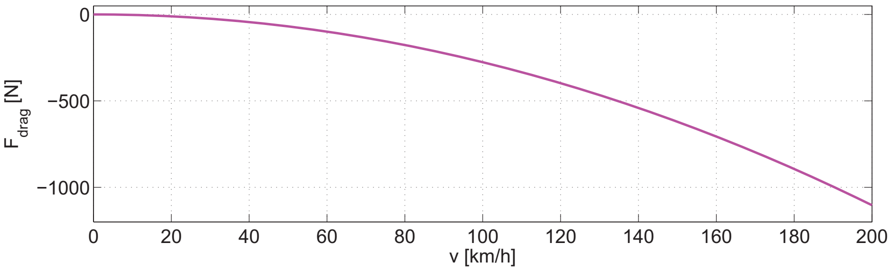
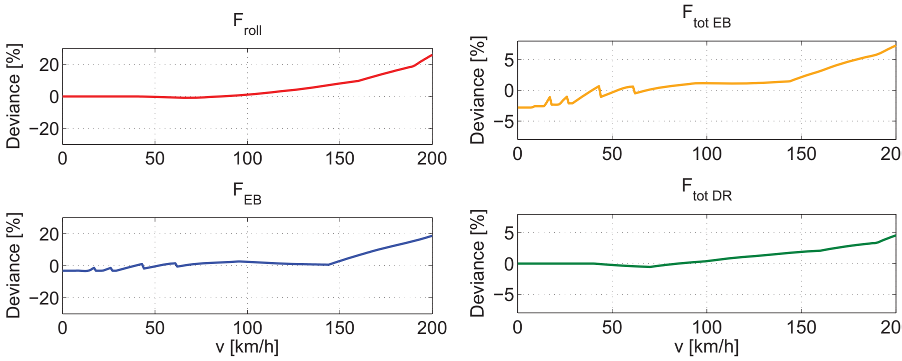


Figure 2.6 Aerodynamic drag force

This means the influence of τ_f and μ_r will diminish, and thus also the influence of errors in the approximation of these parameters. In Figure 2.7(a), the deviation of the approximated forces as a percentage of the exact ones for F_{roll} and F_{EB} are shown. The deviation for the total forces for EB and DR modes are displayed in Figure 2.7(b).



(a) Deviation in F_{roll} and F_{EB} (b) Deviation in $F_{tot_{EB}}$ and $F_{tot_{DR}}$

Figure 2.7 Percental deviation of approximate resistive forces

As can be seen, although deviation in F_{roll} and F_{EB} at 200 km/h are as high as

26 % and 19 % respectively, the deviation in total driving resistance is about 7 % for EB mode and about 5 % for DR mode. This makes the use of constant parameters instead of variables seem feasible, and allows for substantially less complicated calculations of the rolling trajectories. Adding up to the arguments already presented, is that the GreenACC will likely feature a maximum speed limit well under 200 km/h and generally operate at much lower speeds. Another aspect in deceleration at high speeds is that idle fuel consumption in DR mode over a given distance is very insignificant compared to friction torque losses in EB mode. This favors DR over EB at high speeds, giving the advantage of a lesser deviation in approximated total resistive forces, as already discussed.

3. Optimization Problem

For the problem of optimizing a deceleration procedure, it is necessary to define what a good deceleration is and how it should be compared to other solutions. The two quantities that are most important in the case of deceleration is obviously fuel consumption, whose reduction is the motivation for the entire project, but also time losses caused by lower average speed. These costs will be explicitly defined using cost functions, or *objective functions*, subject to optimization. It is easy to realise that concurrently minimizing these two quantities is a problem that has no unique solution. Lowering these costs are two inherently opposed processes, as reducing the travel time requires delaying the deceleration and waste more energy through regular braking. However, much can be done by making the two quantities simultaneously as small as possible. What this means is the main concern for the optimization field known as multiobjective optimization.

3.1 Multiobjective Optimization

For the sake of clarity and ease of discussion, some basic definitions and vocabulary from the field of multiobjective optimization will first be rephrased. The presentation below is mainly due to [Collete and Siarry, 2003] and [Ehrgott, 2005].

Problem Definition

A multiobjective optimization problem can be mathematically formulated as:

$$\left|\begin{array}{lll} \text{minimize} & \overline{f}(\overline{x}) & \text{(objective function)} \\ \text{subject to} & \overline{g}(\overline{x}) \leq 0 & \text{(inequality constraints)} \\ \text{and} & \overline{h}(\overline{x}) = 0 & \text{(equality constraints)} \end{array}\right.$$

where $\overline{f} \in \mathbb{R}_n$, $\overline{x} \in \mathbb{R}_m$, $\overline{g} \in \mathbb{R}_k$ and $\overline{h} \in \mathbb{R}_p$. Here, $\overline{f}$ represents an objective function, whose components are subject to minimization. The term multiobjective goes along with the fact of $\overline{f}$ being vector valued. In the case of a scalar objective function, there is obviously only one objective to optimize. In the present case, $\overline{f}$ will be composed of the fuel consumption and the time loss. $\overline{x}$ is a vector of parameters that are usually referred to as *decision variables*. As this name implies, optimization is performed by trying out different sets of decision variables in a systematic way. In the case of a deceleration profile, $\overline{x}$ represents the parameters used to describe the profile in question. $\overline{g}$ and $\overline{h}$ represents constraints on the search space. Some values of $\overline{x}$ will not represent any valid deceleration profiles, while some $\overline{x}$ will yield deceleration profiles that might be undesirable from a technical point of view or because of comfort issues. Such solutions are excluded through $\overline{g}$ and $\overline{h}$. We will denote vectors $\overline{x}$ that fulfill the constraints *solutions* to the optimization problem. The set of all solutions will be denoted the *feasible set*.

Pareto Optimality

In order to categorize and compare different solutions some definitions are needed.

DEFINITION 3.1—DOMINATION
A solution $\overline{x}_1$ dominates another solution $\overline{x}_2$ if $\overline{x}_1$ is better than $\overline{x}_2$ in at least one objective function while being as good as $\overline{x}_2$ in all the others. □

DEFINITION 3.2—PARETO OPTIMALITY
A solution $\bar{x}$ is said to be Pareto optimal if there exists no solution $\bar{x}'$ that dominates $\bar{x}$. If $\exists\ \delta \in \mathbb{R}$ such that this property holds in a sphere of radius δ around $\bar{x}$, then $\bar{x}$ is locally Pareto optimal. □

In the case of continous objective functions that are contrary in the sense that improvement in one of the objective functions leads to degradation of the other, the Pareto optimal solutions will form a surface referred to as the Pareto surface or trade-off surface. Movement along this surface is only possible by accepting degradation in at least one of the objective functions. Multiobjective optimization in this form thus involves two distinct tasks. First, finding the Pareto surface and thereafter deciding which of the Pareto optimal solutions to pick.

3.2 Deceleration Profile

The aim of the project is to find a method for optimizing deceleration from a certain initial speed v_0, to some target speed v_{target} at a position that will hereby be referred to as the *target point*. A deceleration profile is composed of the three available modes of operation already discussed. The three modes of operation differ in deceleration rate, corresponding to loss of kinetic energy, and in instantaneous fuel consumption. The vehicle speed trajectory over distance will be denote $v(s)$.

Deceleration Scheme

A general deceleration scheme using DR, EB and RB could involve any number of switches between the three modes of operation. Parametrization of such a general deceleration scheme is hard, because of the arbitrary number of switches between the different modes. It also means the number of the decision variables in $\bar{x}$ would become large, which in turn makes the optimization problem harder to solve. The set of every possible deceleration pattern would also include many dominated solutions that are not interesting in the current scenario. Before attempting to find optimal solutions, some restrictions will be imposed on the deceleration profiles allowed, based on the characteristics of each deceleration mode and the preference for fast rather than slow deceleration profiles.

DR before EB DR is the driving mode with the least driving resistance and allows traveling with minimal speed loss. Since fuel consumption in DR is proportional to time, DR at high speeds incurs lower fuel consumption over distance than DR at lower speeds. To minimize the total trip time, it is also desirable to maintain high vehicle speed as long as possible. Performing EB or RB before DR increases traveling time and fuel consumption. Therefore, when DR is to be performed it will be the first deceleration mode used in a deceleration profile.

EB before RB Having determined that deceleration through DR will constitute the first part of a deceleration profile, the next question is how EB and RB should be performed. Because EB and RB both have no consumption and RB allows for faster deceleration, braking only makes sense in the last phase of a deceleration sequence where the vehicle speed needs to be decreased to v_{target}.

Braking curve In deceleration through RB, the rate of deceleration is variable in a wide range. In the case of automatic braking at a speed limit decrease, the controller needs to decide not only at what point to start braking but also how hard to brake. For

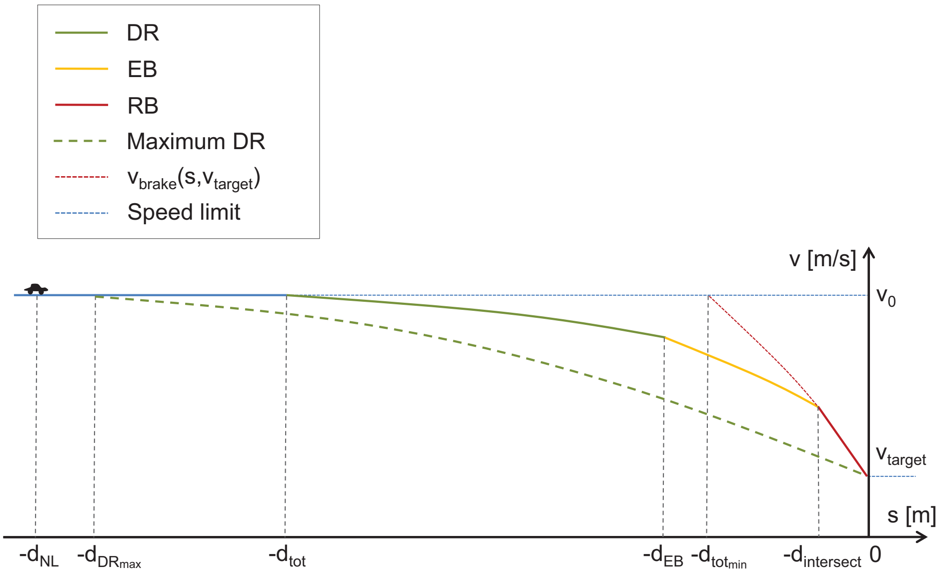


Figure 3.1 Parametrization of a deceleration profile

the Deceleration Optimizer, a braking curve has been constructed that dictates what the deceleration rate should be, based on the current vehicle speed.

$$a_{brake} = f_1(v)$$

For a given v_{target}, the braking curve can be translated as a speed profile over distance

$$v_{brake} = f_2(s, v_{target})$$

and where

$$v_{brake}(0, v_{target}) = v_{target},$$

which means the correct vehicle speed is achieved at the target point.

The braking curve is meant to represent deceleration of a regular customer at a speed limit decrease and should provide a firm but comfortable speed adjustment. Using a predefined braking trajectory has the advantage of removing the decision of when to start braking from the optimization problem. Braking will be initiated when the vehicle speed intersects the braking trajectory at a distance $d_{intersect}$ from the target point. Defining the braking trajectory outside of the optimization also makes it easy to adjust the braking trajectory in response to feedback from driving experiments and customer evaluations. In the vehicle, the braking curve will serve as a reference speed for the speed regulator which effectuates the deceleration profiles calculated by the Deceleration Optimizer.

A deceleration performed according to the description above can be seen in Figure 3.1. The different modes of operation are displayed as well as $v_{brake}(s, v_{target})$. The x-axis displays position and has zero position defined at the change in speed limit. The reason for this is that the target point is the most suitable point of reference since it is defined at a given position and fix over time. All parameters denoted in the form of $d_{subscript}$ are defined as positive to make it easier to keep track of signs when programming.

Finding the Decision Variables

In order to describe a deceleration profile, the switching points between different modes of operation will have to be specified. These will also be the decision variables in $\bar{x}$ of the optimization problem. The switching points in the DR → EB → RB driving scheme are:

- Constant speed driving → DR
- DR → EB
- EB → RB
- RB → Constant speed driving

See Figure 3.1 Potentially, one or several of these switching points might coincide, in the case of deceleration profiles that do not use all of the available deceleration modes. Deceleration profiles will be parametrized by the total deceleration distance, d_{tot} and the engine braking distance, d_{EB}. These distances are marked out in Figure 3.1. The braking distance required for a certain deceleration profile will be included in d_{EB} and is indirectly decided by d_{tot}, d_{EB} and the braking curve. This means there will be two decision variables contained in $\bar{x} = [d_{tot},\ d_{EB}]$. Limiting the number of decision variables is important in order to keep the complexity of the optimization problem down. At the same time it is important that problem reduction is done in such a way that good solutions to the original problem are not excluded from the feasible set of the reduced one. However, the restrictions imposed so far on the optimization problem should not cause any serious limitations on the solutions obtained. The distance in DR mode is not explicitly included in $\bar{x}$, but can be expressed as $d_{DR} = d_{tot} - d_{EB}$. This can be deduced from Figure 3.1.

Carrying Through of a Deceleration

The distance from the vehicle to the next speed limit will be denoted d_{NL}. When performing a deceleration, the car will travel in DR mode while

$$d_{tot} \geq d_{NL} > d_{EB}$$

and in EB mode while

$$d_{EB} \geq d_{NL} > d_{intersect}.$$

However $d_{intersect}$ is not defined in before hand but is defined as the distance to the target point from the point where

$$v(s) \leq v_{brake}(s, v_{target}).$$

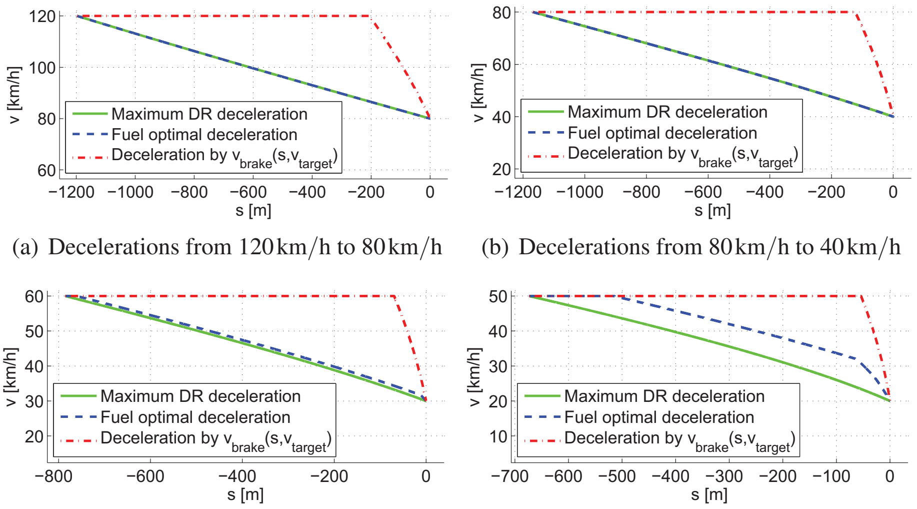
From this point RB will be enabled and the vehicle's speed controller will bring it to the next speed limit following the braking trajectory. Deceleration profiles will generally be described in the form of speed as a function of distance rather than a function of time. This is sometimes cumbersome, but has the advantage of providing a common end point for the deceleration profiles. A change in speed limit is defined for a position along the road, not at a given time. The time needed to arrive to the change in speed limit will obviously depend on what deceleration profile is being used.

3.3 Objective Functions

Fuel Cost

Reduction of fuel consumption is one of the main reasons this project takes place. For speeds above about 30 km/h the most fuel efficient deceleration option is DR for the maximum deceleration distance. At first this might seem contradictory, since no fuel injection occurs during EB, but during DR fuel is injected to maintain the engine's idling speed ω_{idle}. What is important to realise is that during EB, although no injection occurs, energy is still being dissipated through friction between piston and cylinder, just as in the case of DR. The difference is that during DR a low idling speed is being maintained, while during EB engine speed is generally a lot higher. This makes energy dissipation through friction greater in the case of EB than in the case of DR. What this means in terms of deceleration to a given point is that if deceleration through pure EB would be performed, the deceleration would have to be initiated a lot later than if pure DR is being performed. The added fuel consumption from driving at constant speed up to the start of the EB deceleration is generally larger than the idle consumption of the corresponding DR deceleration. Figures 3.2(a) and 3.2(b) displays the fuel optimal deceleration for decelerations from 120 km/h to 80 km/h and 80 km/h to 50 km/h respectively.

However, for deceleration at lower speeds, fuel optimal deceleration is not achieved through pure DR deceleration. Instead, constant speed driving, followed by DR almost up to the braking curve, finished off by a short episode of EB to bring the speed down to v_{target} achieves minimal fuel consumption, as in figures 3.2(c) and 3.2(d). The explanation lies in that fuel consumption is proportional to travel time in DR. The overall higher speed achieved by extra constant speed driving in Figure 3.2(d) reduces the time spent in DR by about 40 %, thereby reducing fuel consumption of the DR phase with the same percentage. Overall, the higher the speed, the higher the advantage of DR over EB for fuel efficiency.



(a) Decelerations from 120 km/h to 80 km/h

(b) Decelerations from 80 km/h to 40 km/h

(c) Decelerations from 60 km/h to 30 km/h

(d) Decelerations from 50 km/h to 20 km/h

Figure 3.2 Fuel optimal deceleration profiles

Fuel Cost Function To construct a fuel cost objective function it is, however, unimportant what the fuel optimal deceleration is. What is needed is a common starting point from which to compute the fuel consumption for the different deceleration profiles. In this project, the fuel cost objective function, Φ_f, has been calculated as the fuel being consumed between the point at distance $d_{DR_{max}}$ from the next speed limit and the speed limit itself. For all options except the one with $d_{tot} = d_{DR_{max}}$, this means driving along by the current speed limit for a while longer, causing a fuel consumption $\Phi_{f_{const}}$. With σ_{v_0} denoting the fuel flow in g/s for driving at constant speed

v_0, $\Phi_{f_{const}}$ can be expressed as a function of σ_{v_0}, v_0 and the distance driven constant speed, $d_{DR_{max}} - d_{tot}$.

$$\Phi_{f_{const}} = \frac{(d_{DR_{max}} - d_{tot})\sigma_{v_0}}{v_0} \tag{3.1}$$

When performing DR fuel is being used to keep the engine running at ω_{idle}. The fuel consumed is a function of the idle consumption σ_{idle} in g/s and the time spent in DR, t_{DR}. Accumulated fuel consumption $\Phi_{f_{DR}}$ during a sequence of DR can be written as

$$\Phi_{f_{DR}} = \sigma_{idle}\, t_{DR} \tag{3.2}$$

With the assumption of $\omega_{EB} > \omega_{idle}$, no fuel is being consumed while performing EB or RB. $\Phi_{f_{EB}} = 0$. For Φ_f, adding up equations (3.1) and (3.2) thus yields

$$\Phi_f = \Phi_{f_{const}} + \Phi_{f_{DR}} = \frac{(d_{DR_{max}} - d_{tot})\sigma_{v_0}}{v_0} + \sigma_{idle}\, t_{DR}. \tag{3.3}$$

Fuel Consumption in Slopes When driving in hilly terrain, slopes obviously have a great impact on fuel consumption. Based on Equation (2.7) it is straightforward to compute the added work required to maintain a constant speed in a uphill slope, or the reduction in work required in the case of a downhill slope. The engine power required to maintain constant speed on flat ground can be expressed as

$$\begin{aligned} a = 0 &= A_{EB_{flat}} + Bv^2 + F_{motor}/m \\ \Leftrightarrow F_{motor} &= -(A_{EB_{flat}} + Bv^2)m \\ \Leftrightarrow P &= -(A_{EB_{flat}} + Bv^2)mv \end{aligned} \tag{3.4}$$

where $A_{EB_{flat}}$ is the EB A-parameter for flat ground.

Let $\sigma_{(v_0,0)}$ denote fuel consumption for constant speed v_0 on flat ground and $\sigma_{(v_0,\alpha)}$ fuel consumption for v_0 at constant angle α. Supposing that aerodynamic drag is not affected by the grade of the road, the work $P_{(v_0,\alpha)}$ required to drive along the mentioned slope can similarly be expressed as

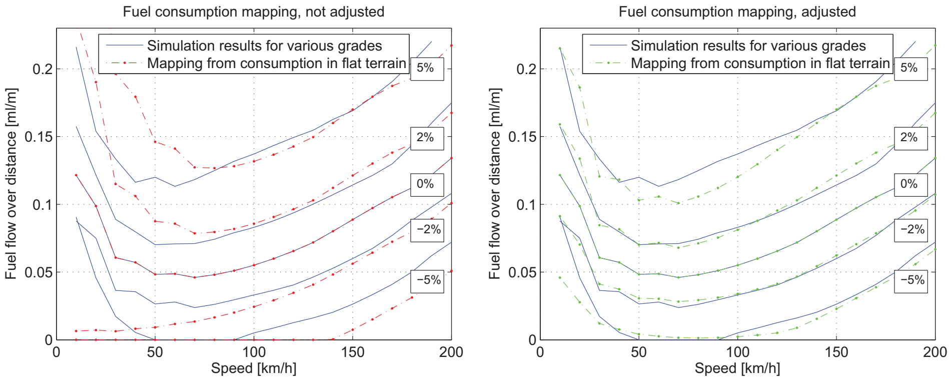
$$P_{(v_0,\alpha)} = -(A_\alpha + A_{EB_{flat}} + Bv^2)mv \tag{3.5}$$

with A_α taking into account the added acceleration from gravity. A ratio ρ_P expressing the change in required engine power output is thus given by

$$\rho_P = \frac{A_\alpha + A_{EB_{flat}} + Bv^2}{A_{EB_{flat}} + Bv^2.} \tag{3.6}$$

If the efficiency of the driveline would be constant, equaling a linear relation between required work and fuel consumption, this would directly translate into $\sigma_{(v_0,\alpha)} = \sigma_{(v_0,0)}\rho_P$. Unfortunately this is not the case. However, to avoid having to deal with an entire engine and gearbox mapping in the calculations of constant fuel consumption, a modified version of the constant work/consumption ratio will be used.

Correction Factor Using ρ_P straight of as given by Equation 3.6 results in a fuel mapping as in Figure 3.3(a). Clearly this is not an acceptable approximation to the actual fuel consumption. The term $A_{EB_{flat}}$ in Equation (3.4) includes both the rolling resistance and the internal friction in the engine, provided that the highest gear is being used. These friction losses correspond to an important non-linearity in the fuel consumption mapping. As can be seen in Figure 3.3(a), the approximation is somewhat acceptable for high speeds and very poor for low speeds. One main reason is that the calculated resulting force from the engine friction torque is incorrect for speeds where lower gears would have been used. One option would be to include gear switching in the calculations, but this requires knowledge of the entire gear switching strategy of the automatic gearbox which in turn would be as involved as an entire fuel consumption mapping. Instead, correction will be made by applying a speed dependent correcting factor $\zeta(v)$ for the A_α-term in Equation (3.6)



(a) Non adjusted mapping

(b) Adjusted mapping

Figure 3.3 Slope fuel consumption

$$\rho_P = \frac{A_\alpha \zeta(v) + A_{EB_{flat}} + Bv^2}{A_{EB_{flat}} + Bv^2}. \tag{3.7}$$

With ζ taking the shape of two bounded linear polynomials, one for uphill slopes and one for downhill slopes, the resulting fuel mapping as shown in Figure 3.3(b) is obtained. The polynomials are functions of speed, and are bounded to one for higher speeds. Adequate parameters for the correction polynomials have been obtained by manual tuning by comparing the output from the mapping with simulated fuel consumption values for various road grades. The result is a reasonably good fit, and a slope fuel consumption mapping that only requires fuel consumption data for constant driving in flat terrain. This allows for quick and straightforward calculation of constant speed driving fuel consumption in slopes. The biggest deficit of this method is that the mapping does not directly build upon the regular vehicle parametrization and will have to be reviewed for every new vehicle. The manual work required to reparametrize the deceleration optimizer for a new vehicle is thus increased, reducing portability of the function.

Time Cost

Time Optimal Deceleration The time optimal deceleration will be defined as the fastest way to travel along at constant speed v_0 up until some point where a firm yet comfortable braking maneuver brings the vehicle speed down to the correct speed v_{target} at the target point. This time optimal braking maneuver will be defined by the same braking curve used for the optimized deceleration profiles. The time optimal deceleration profile will be denoted $v_{t\_opt}(s)$.

Time Cost Function What measure to use as definition for time cost objective function Φ_t is not as obvious as for the fuel cost. One option is to use the extra travel time in seconds for a certain deceleration profile compared to $v_{t\_opt}(s)$.

However, simply using time lost in absolute numbers, although intuitive, might not be the best solution. The problem is that fuel savings are a lot larger for higher speeds. This means that comparing fuel consumption with time will lead to almost no fuel savings for low speeds, since it will be time-wise too expensive. At high speeds, fuel will be cheaper in terms of time and DR and EB will be a lot more profitable. Although this is a rational behaviour, it might not be what would feel natural and intuitive to the driver. The aim of the Deceleration Optimizer is not to save the maximum amount of fuel, but to save a lot of fuel in a way that is comfortable and comprehensible to the driver. This means the function must behave in a way that is intuitively acceptable, but not necessarily strictly rational.

Another option is then to use the integral of the difference in speed between $v_{t\_opt}(s)$ and the predicted vehicle speed. Letting $v(s)$ denote the function of speed over distance for some deceleration profile yields

$$\Phi_t = \int_0^{d_{maxDR}} v_{t\_opt}(s) - v(s)\, ds. \tag{3.8}$$

This formulation punishes not only the time lost, but also deviation from 'normal' driving speed. A graphical representation of Φ_t is shown in Figure 3.4. The deviation of $v(s)$ from $v_{t\_opt}(s)$ is marked out as the colored area between the two plots. Since braking curves are the same for the two driving patterns, the integral can be computed from $d_{intersect}$ instead of 0.

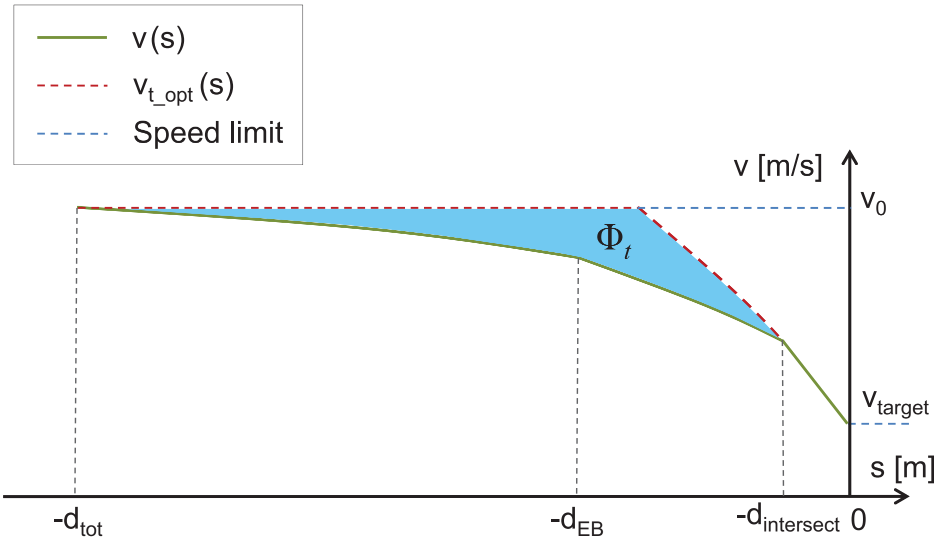


Figure 3.4 Integral defining time cost

The difference in time between two deceleration profiles $v(s)$ and $v_{t\_opt}(s)$ can be computed as

$$\Delta t = \int_0^{d_{maxDR}} \left(\frac{1}{v(s)} - \frac{1}{v_{t\_opt}(s)} \right) ds = \int_0^{d_{maxDR}} \left(\frac{v_{t\_opt}(s) - v(s)}{v(s)v_{t\_opt}(s)} \right) ds. \tag{3.9}$$

As can be seen when comparing equations (3.8) and (3.9), the integral defining (3.8) corresponds to the integral in (3.9) with a factor $v_{t\_opt}(s)v(s)$ multiplying the integrand. This means time cost defined as in Equation (3.8) will be scaled by the speed so that greater absolute time loss will be required at higher speeds to achieve the same time cost. This means that the trade-off between time and fuel will be differently made at different speeds. This might seem irrational at first, but might actually make sense. The cost that we are trying to capture with the time cost function constructed as above is not purely a time loss in seconds. What we want to eliminate is the subjective feeling of going uncomfortably slow or of being a hindrance for surrounding traffic. Deviation from normal driving speed is better captured by Equation (3.8) than the pure time computed in Equation (3.9).

3.4 Deceleration Optimization

Different deceleration patterns can be produced by combining the three available modes of deceleration. Each deceleration pattern is parametrized by its decision variable vector $\bar{x} = [d_{tot}, d_{EB}]$ and is associated with a value of the cost function $\bar{f} = [\Phi_t, \Phi_F]$. As already mentioned, minimizing the time cost Φ_t and the fuel cost Φ_F can be done only to the limit of the Pareto surface, where improvements in one objective is possible only by degradation of the other.

Scalarization

The optimization is meant to be run online in the control system of a car, and therefore has to be reasonably fast. Finding the Pareto surface of a multiobjective optimization problem is quite an extensive task. One way to handle differing objective functions is scalarization of the problem by adding up the objective functions using a weight coefficient. Having two components in $\overline{f}$ requires only one weight coefficient c_t, which will multiply the time cost Φ_t, yielding $\Phi_{tot} = \Phi_F + c_t\Phi_t$. The resulting single-objective optimization problem of minimizing Φ_{tot} then has as solution one of the Pareto optimal points of the multiobjective problem. Optimization for a given c_t is then a regular single objective optimization problem. Choosing c_t means choosing how time cost and fuel cost should be compared. Ultimately it will thus decide the length of the deceleration patterns, and be an important variable for tuning the function.

The steps in the optimization will thus be to first implicitly choose a point on the Pareto surface by choosing c_t, then actually finding the chosen point through single objective optimization of Φ_{tot}

Optimization Area

In Figure 3.5, a contour plot of the scalarized cost function is displayed, for a speed transfer from $120\,\text{km/h}$ to $80\,\text{km/h}$ and a value of c_t of 0.01. For the given speed transfer, the colored triangle represents every possible deceleration pattern. On the x-axis is the total deceleration distance d_{tot} and on the y-axis is the engine braking distance d_{EB}. Each corner of the triangle represents an extremum in the set of possible deceleration patterns, where deceleration is performed using exclusively one of the three available modes of decelerations. The right corner represents deceleration through DR for the maximum DR distance. No EB or RB is involved. The upper corner represents deceleration only through EB, and the left corner represents deceleration by the braking curve.

The shape of the optimization area is determined by the speed transfer in question and is not affected by the choice of time weight c_t. The appearance of the scalar cost function on the other hand, is decided by c_t. In Figure 3.5, optimal solutions for a set of different values of c_t are drawn as red dots. These dots together form the Pareto

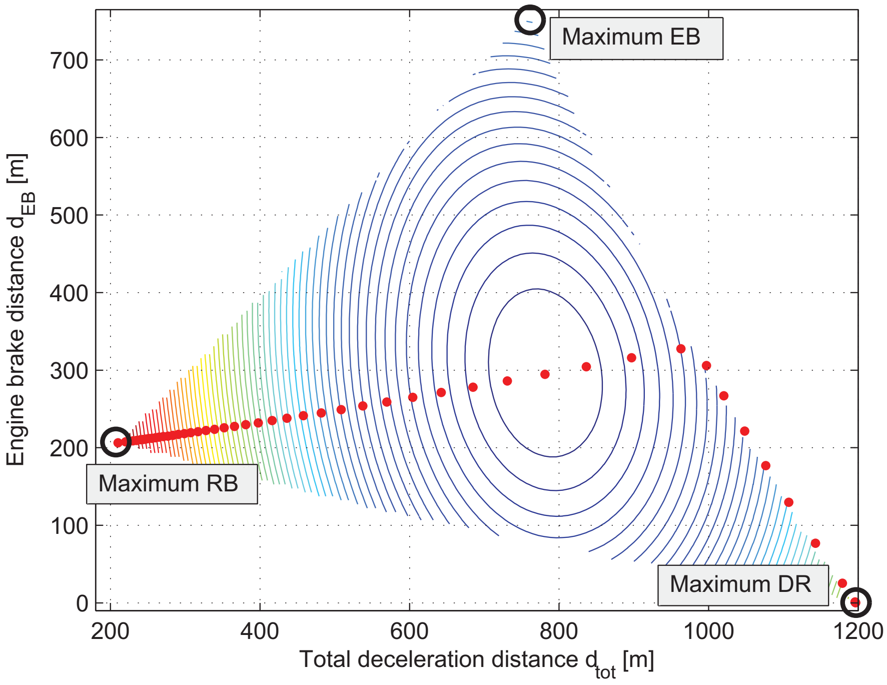


Figure 3.5 Trade-off surface for deceleration from 120 km/h to 80 km/h in flat terrain

Surface for the given speed transfer. It is noteworthy that pure DR deceleration is the optimal solution for low values of c_t, which means time cost has low priority. On the other hand, when c_t is big, the optimal solution tends towards pure RB deceleration. On the contrary, pure EB deceleration is not part of the Pareto surface. This means that when comparing pure EB deceleration to solutions on the Pareto surface, the Pareto optimal solutions will always be better either by lower consumption or by lower time cost, and sometimes better in both parameters. Off course, this this is not necessarily true for every speed transfer, but is generally the case for higher speeds and enough space to perform arbitrarily long decelerations up to $d_{DR_{max}}$.

Optimization Constraints

With the colored triangle in Figure 3.5 representing all feasible solutions for the speed transfer in question, the area outside of the triangle must clearly represent solutions that are impossible or disallowed. In fact, each side of the triangle represents a constraint on the optimization problem. Just as the corners of the triangular area represent solutions that are using only one mode of deceleration, the deceleration profiles along each edge of the optimization area are composed of a combination of the pure deceleration modes at their endpoints. The deceleration mode of the opposite corner is not used along an edge.

Constraints on an Unrestricted Deceleration

- The right edge of the optimization area represents deceleration patterns that arrive at v_{target} with only EB and DR and no RB. Solutions to the right of the edge are excluded because they yield an end speed $v_{end} < v_{target}$. Clearly, this is not desirable. For a deceleration profile to be allowed, it is required that

$$v_{end} = v_{target}. \tag{3.10}$$

It should be noted that this constraint applies to every deceleration pattern in the feasible set. However, for all decelerations that include RB this condition is already fulfilled through the braking curve, which always leads to v_{target} at $s = 0$.

- The lower edge represents deceleration patterns that combine DR and RB. Since the RB part of the deceleration is included in d_{EB}, d_{EB} is not zero except for the right corner, but equals the required braking distance which is dictated by the intersection point with the braking curve. For smaller values of d_{tot}, meaning that DR is initiated closer to the target point, more braking is obviously required to bring the vehicle speed to v_{target}. For each value of d_{tot} there is a maximum braking distance $d_{RB_{max}}(d_{tot})$, during which RB will be performed if only DR is performed up to this point. If switching to EB mode occurs earlier, then the vehicle will decelerate harder and the braking distance will be shorter than $d_{RB_{max}}(d_{tot})$. However, since the braking distance is included in d_{EB}, the minimum allowed value for d_{EB} will be $d_{RB_{max}}(d_{tot})$.

$$d_{EB} \geq d_{RB_{max}}(d_{tot}) \tag{3.11}$$

- The constraints creating the left edge of the optimization area results from the parametrization of the deceleration patterns. Since d_{tot} is the total deceleration distance, obviously d_{EB} has to be smaller that or equal to d_{tot}.

$$d_{EB} \leq d_{tot} \tag{3.12}$$

Excluding Solutions while Preserving Continuity To preserve continuity of the cost function, which in many cases greatly facilitates optimization, the disallowed deceleration patterns are excluded by severely punishing the cost function with a term that is made large but is proportional to the overshoot in the tested variable. For example, Equation (3.12) is excluding solutions for which $d_{EB} > d_{tot}$. Then, when a solution violates (3.12), exclusion is made through a punishment term Φ_{punish} added to the cost function which is defined as $\Phi_{punish} = C_{punish}(d_{EB} - d_{tot})$ where C_{punish} is a large value compared to the regular values of the cost function. Since Φ_{punish} is proportional to the deviation from the feasible set, continuity will be preserved while using this method.

Additional Restrictions In addition to the basic constraints presented above, additional restrictions might be placed on the feasible set. These restrictions are posed by comfort and drivability issues as well as spatial limitations that might arise when the distance to the next speed limit is shorter that $d_{DR_{max}}$ at the time of optimization.

- **Maximum Speed Allowance** In the case of downhill slopes it is possible that the vehicles accelerates down the slope, potentially above the legal speed limit v_{limit}. In the case of constant speed control the constant speed controller would have braked in this situation. Similarly, the Deceleration Optimizer needs to handle these situations. Solutions yielding speeds that exceeds v_{limit} above a tunable tolerance v_{tol} will simply be excluded from the feasible set.

$$v(s) < v_{tol} v_{limit} \tag{3.13}$$

Figure 3.6 shows a contour plot of the area of optimization and the Pareto surface for a deceleration optimization for a 5 % downhill section split up by a 3 % uphill section. Because of the maximum speed constraint two white ridges appear in the optimization area. The white ridges are made up of deceleration

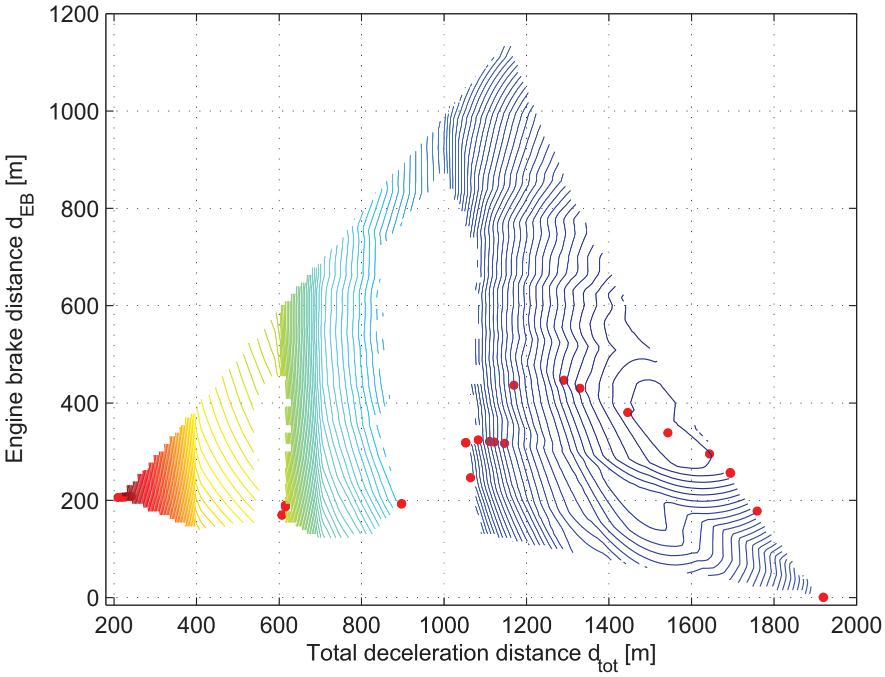


Figure 3.6 Trade-off surface for deceleration from 120 km/h to 80 km/h in hilly terrain

profiles that violate the constraint set by Equation (3.13). The Pareto surface is in this case distorted and separated by the disallowed areas.

When comparing Figure 3.6 to Figure 3.5 it is also clear that the optimization problem in the case with hilly terrain is a lot more demanding than optimization in flat terrain. Locally optimal solutions are plentiful, especially at the edges of the disallowed areas, which requires a global optimization method with the capability of overcoming local optimums.

- **Minimum DR time** Being able to limit the minimum time in DR mode has several advantages. The main concern by BMW is the question of customer acceptance. Short episodes of DR might feel uncomfortable to the driver and give an impression of indecisiveness of the controller. Additionally, there is the question of gearbox durability. Since during both constant speed driving and EB mode the gearbox is in D position, but during DR mode it is in N, switching into DR mode induces more wear to the gearbox than going straight into EB mode. Therefore, if the time spent in DR is short the gains in time and fuel efficiency might be smaller than the cost of the increased wear. This means the DR sequence either has to be long enough, or that no DR should be performed at all.

  Introducing a minimum time in DR, $t_{DR_{min}}$, which translates into an approximated minimum DR distance $d_{DR_{min}} = v_0 t_{DR_{min}}$, and remembering that $d_{DR} = d_{tot} - d_{EB}$, allows for a constraint formulation as below.

$$(d_{tot} - d_{EB} \geq d_{DR_{min}}) \vee (d_{tot} - d_{EB} = 0)$$

$$\Leftrightarrow \tag{3.14}$$

$$(d_{DR} \geq d_{DR_{min}}) \vee d_{DR} = 0 \tag{3.15}$$

  Since the edge where $d_{DR} = 0$ will be separated from the rest of the feasible set by a section of excluded deceleration profiles, it is not easily accessible through the regular optimization. Instead, when the optimal solution has a d_{DR} close to $d_{DR_{min}}$, an additional search will be performed along the pure EB-edge of the unconstrained optimization area.

- **Target point closer than $d_{DR_{max}}$** In real driving situations it is possible that optimization of a deceleration pattern has to be performed when the distance to the new speed limit is smaller than $d_{DR_{max}}$. This might occur for example when the GreenACC is activated close to a speed transfer or when the predicted path in the navigation system changes. In these situations there will be a maximum value of d_{tot}, denoted $d_{tot_{max}}$ and the optimization area in Figure 3.5 will be reduced along a vertical line at $d_{tot_{max}}$.

$$d_{tot} \leq d_{tot_{max}} \tag{3.16}$$

Figure 3.7 displays the contour plot of the cost function for an optimization under additional constraints from a short $d_{tot_{max}}$ of 500 m and from a $t_{DR_{min}}$ of 6 s. The deceleration in question is from 80 km/h to 50 km/h. The $t_{DR_{min}}$ constraint is drawn as a purple line, while the part of the optimization area excluded by $d_{tot_{max}}$ is left out, and is located to the right of the colored area. It can be clearly seen how the Pareto surface first follows the right vertical edge, then makes its way across the allowed area and follows the $t_{DR_{min}}$ edge for a while, until switching over to pure EB decelerations for higher values of c_t.

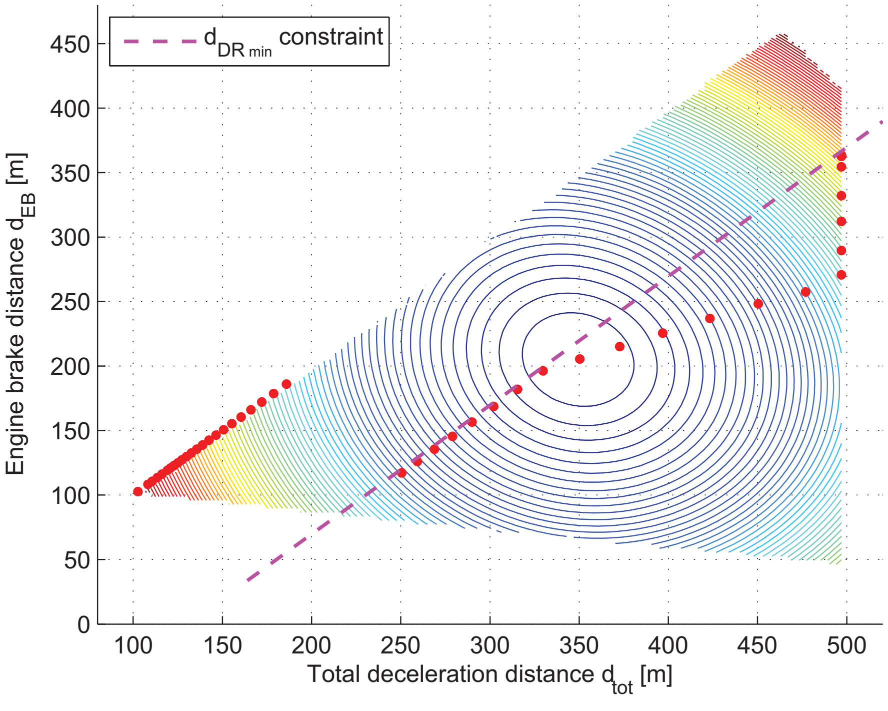


Figure 3.7 Trade-off surface for deceleration from 80 km/h to 50 km/h with restrictions on d_{tot} and on d_{DR}

Optimization Algorithm

The method of making a combined cost function allows for optimization to be performed with a scalar value describing the performance of a certain deceleration profile. This can be done using a number of different optimization methods. Since Embedded Matlab and Real-Time Workshop for Simulink was to be used for the in-vehicle testing, functions from the Matlab Optimization toolbox could not be used.

Instead, a global search algorithm was written in Embedded Matlab compatible code, based on [Georgieva and Jordanov, 2010]. A coarse global search is first made using a search grid based on the search area at hand, defined by the slope profile and the speed transition. More precisely the grid is based on $d_{DR_{max}}$, $d_{EB_{max}}$, $d_{tot_{min}}$ and $d_{DR_{min}}$ and designed to cover the interesting parts of the search area. Figure 3.8 displays an optimization sequence for the 120 km/h to 80 km/h deceleration over flat terrain discussed earlier. The purple circles constitute the global search pattern. When the search grid has been constructed, the cost function is evaluated for every point in the global search grid. The points with the lowest values of the cost function are then chosen for local searches. The algorithm used for local search is the classic Nelder-Mead Simplex Search Algorithm [Nelder and Mead, 1965]. The Simplex Algorithm uses simplexes, which are sets of points in the search space. In the case of two decision variables the simplexes consist of three points. The initial simplexes for the three local searches in Figure 3.8 are shown as red circles. Through a set of manipulations on the simplexes the Simplex Algorithm finds its way towards the local optimum. For further reference, see [Nelder and Mead, 1965]. The black circles shown in Figure 3.8 are drawn at the centers of the consecutive simplexes, showing the traces of the local optimizations, and the green circle shows the resulting optimum.

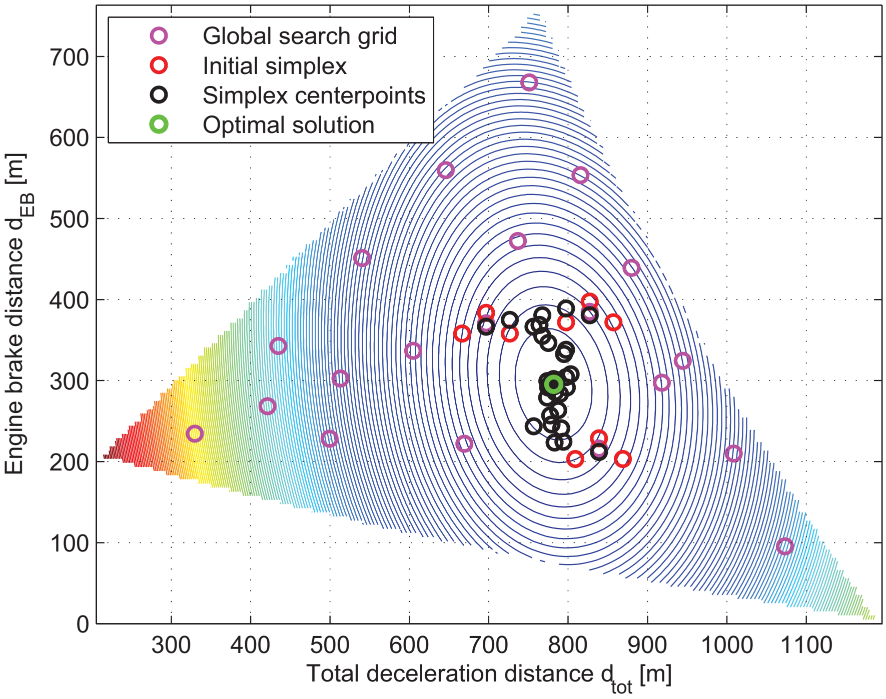


Figure 3.8 Optimization Search Sequence for deceleration from 120 km/h to 80 km/h

In the situation with flat terrain, all three local searches converge to the same point. In this situation the global search is of no big use. However, as already mentioned, the situation is another for rugged terrain. Figure 3.9 displays the optimization sequence for the deceleration from 120 km/h to 80 km/h that has already been considered if Figure 3.6. This time, a value of c_t which has its optimum in the vicinity of the large excluded area has been chosen for the contour plot. The color coding of the circles is the same as in Figure 3.8.

In this case, all three local searches find different local optimums, and the best one is taken as the global optimum. It is clear that without the global search, the result is heavily dependent on the starting point. Even with the global search, it is quite

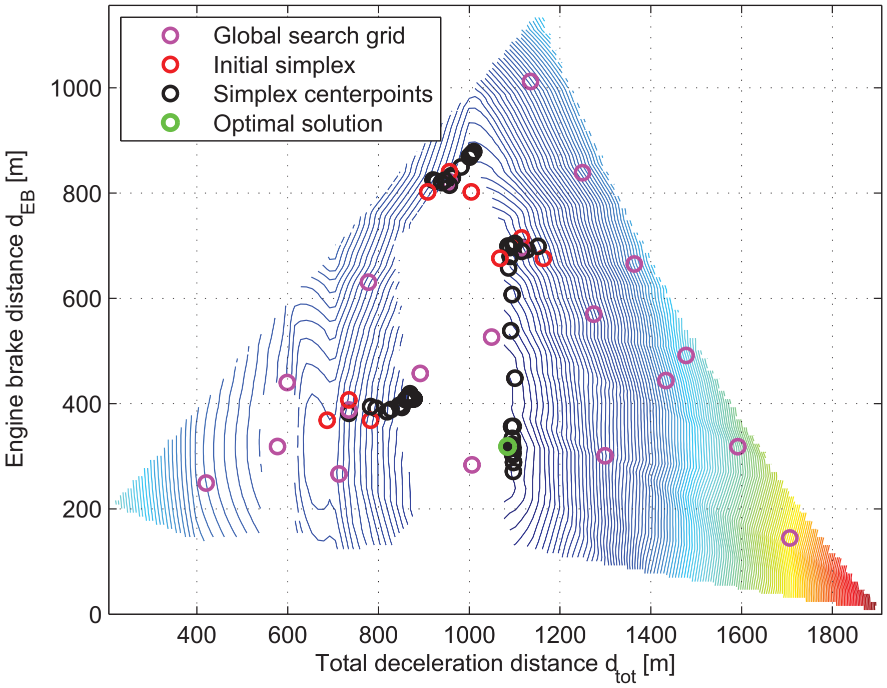


Figure 3.9 Optimization search sequence for deceleration from 120 km/h to 80 km/h in hilly terrain

likely that in situations with very steep 'up-and-down' terrain such as in Figure 3.9, the actual global optimum will not be found. However, accuracy comes at the cost of computation time, and with the course global search chances are good that a solution that is close to the global optimum in terms of cost function value can be found with a reasonable computational effort.

For the coarse global search strategy to work properly, it is important that no points are adjacent and that they are reasonably evenly spaced over the search area. Otherwise there is a risk that all local searches end up being made in about the same position. One remedy to this problem could be to choose points based not only on cost function, but to also consider distance from points already picked as an additional merit.

The optimal solution produced by the procedure above, for a given deceleration will be denoted $\bar{x}_{opt} = [d_{tot_{opt}}, d_{EB_{opt}}]$.

4. Deceleration Optimizer

This chapter contains a brief presentation of the Deceleration Optimizer implementation in Embedded Matlab. Throughout this chapter, variable names used in the code will be printed in `courier` font, and function names will be printed in *italic* font.

4.1 Navigation Information

At the core of the deceleration algorithm is the navigation information which is used to determine when a decrease in speed limit approaches and to calculate rolling trajectories over hilly terrain. The navigation information is presented in the form of three vectors, `vVect`, containing speed limit in m, `slopeVect` containing information about road grade in % and `sVect` containing distances in m to the events in `vVect` and `slopeVect`. The road grade in `slopeVect` is rounded to provide sections with approximate constant grade. In `vVect` speed limits can be both legal speed limits and speed limits imposed by sharp curves, narrow road sections et.c. An example of a possible setup of navigation information vectors can be seen in Table 4.1.

The first element in `sVect` is always 0, which represents the vehicle position. Every other element in `sVect` is positive and is the distance from the current vehicle position to the event in question. For a vector index `i`, `vVect(i)` and `slopeVect(i)` indicate speed and slope for the road section between `sVect(i)` and `sVect(i+1)`.

sVect	0	90	250	320	350	390	500	600	800
vVect	27.8	27.8	22.2	13.9	13.9	22.2	22.2	33.3	22.2
slopeVect	2	0	0	-1	0	1	0	0	-1

Table 4.1 Example of Navigation Information Vectors

In the example given in Table 4.1, the first elements in `vVect` and `slopeVect` indicate the conditions in the section between 0 m and 90 m. At 250 m, speed is reduced from 27.8 m/s to 22.2 m/s, and at 320 m speed is further reduced to 13.9 m.

For the Deceleration Optimizer there is no predefined length for the navigation information vectors. For correct triggering they obviously need to include distances up to $d_{DR_{max}}$ so that upcoming speed limit decreases can be seen early enough, and for accurate calculations distances up to d_{NL} must be covered.

In the implementation the length of the vectors has been fixed in the reconstructor which builds the navigation information vectors based on input from the navigation system. If there is a lot of variance in slope for a road section, this could mean that the prediction horizon is very short. In such a situation, the accuracy of the slope information should be decreased to increase the predictive distance to acceptable levels. This is, however, a vehicle implementation question that can be handled by the reconstructor independent of the Deceleration Optimizer itself.

4.2 Main function

```
function [dTotOpt, dEBOpt] = decelOpt(sigma_idle, cFF,
    drivingResData, downShiftMap, v_0, sVectIn, vVectIn,
    slopeVectIn, c_Time, tDRMin)
```

decelOpt.m is the main function of the optimization algorithm. It wraps the different parts of the optimization process together and calls upon a number of subfunctions during optimization. A flowchart describing the function is displayed in Figure 4.1. In *decelOpt.m*, one specific speed transfer at a time is being processed.

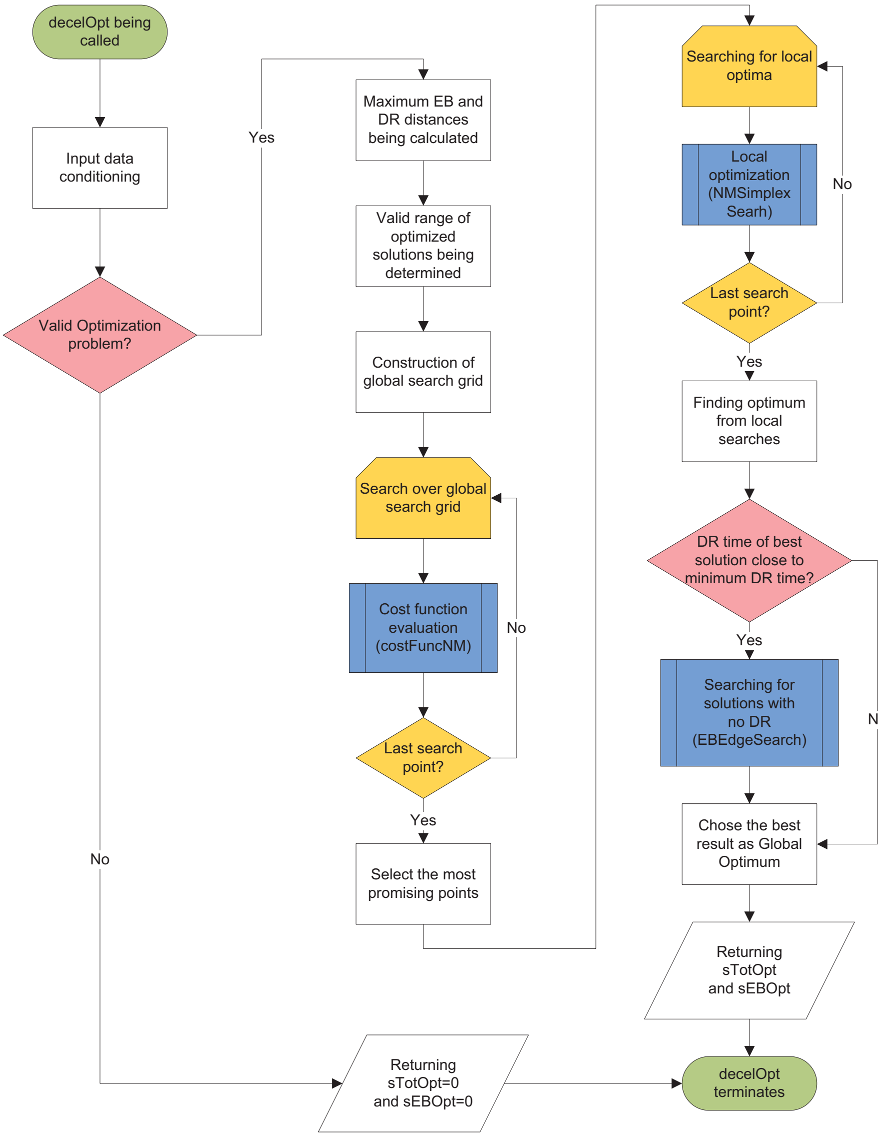

decelOpt being called
Input data conditioning
Valid Optimization problem?
Yes
No
Maximum EB and DR distances being calculated
Valid range of optimized solutions being determined
Construction of global search grid
Search over global search grid
Cost function evaluation (costFuncNM)
Last search point?
No
Yes
Select the most promising points
Searching for local optima
Local optimization (NMSimplex Searh)
Last search point?
No
Yes
Finding optimum from local searches
DR time of best solution close to minimum DR time?
Yes
No
Searching for solutions with no DR (EBEdgeSearch)
Chose the best result as Global Optimum
Returning sTotOpt and sEBOpt
Returning sTotOpt=0 and sEBOpt=0
decelOpt terminates

Figure 4.1 decelOpt.m flowchart

The first four arguments to *decelOpt.m* are vehicle dependent data, i.e. fuel consumption, driving resistance and gear switching parameters. The next four parameters, `v_0`, `sVectIn`, `vVectIn` and `slopeVectIn` contain information about the upcoming deceleration. The last two parameters, `c_Time` and `tDRMin` are tuning parameters for the deceleration strategy, with `c_Time` being the time cost coefficient c_t

and tDRMin the minimum allowed time in DR mode $t_{DR_{min}}$. The inputs to *decelOpt* as well as other function variables are listed and explained in Table 4.2

As can be deduced from the flowchart, *decelOpt.m* makes little more than setting up the optimization problem and handing tasks over to other functions. The most interesting part of *decelOpt.m* is the setting up of the global search grid over the optimization area and evaluation *costFuncNM.m* for the points in the grid. The most promising points are then selected for local optimization by *NMSimplexSearch.m*.

Parameter	Description
dOptTot	Optimal total deceleration distance (DR+EB+RB)
dOptEB	Optimal engine braking distance (EB+RB)
sigma_idle	Idle engine fuel consumption
sigma_v0	Constant speed fuel consumption at v_0 on level road
sigma_v0_slope	Constant speed fuel consumption at v_0 at a given slope
cFF	Fuel flow mapping for constant speed driving on level road
drivingResData	Driving resistance parameters (A_{DB}, B, $A_{EB_6}, \dots, A_{EB_1}$)
downShiftMap	Gear shifting map for downshifting in EB mode.
v_0	Initial vehicle speed v_0
sVectIn	Distances to changing points for speed limit or road grade
sVect	Derived from sVectIn, adjusted to fit algorithms
vVectIn	Current speed limits for every point in sVectIn
vVect	Derived from vVectIn, adjusted to fit algorithms
slopeVectIn	Road grade for every data point in sVectIn
slopeVect	Derived from slopeVectIn, adjusted to fit algorithms
c_Time	Time weight coefficient c_t
tDRMin	Minimum allowed time in DR mode
dDRMin	Minimum allowed DR distance, given by tDRMin and v_0
dDRMax	Maximum DR distance from current vehicle position
dEBMax	Maximum EB distance from current vehicle position
dTotMin	Minimum total dec. dist., given by the braking curve
dTotMax	Maximum total dec. dist., given by distance to next limit
errorTolBC	Error tolerance for brake curve intersection search
errorTolNM	Error tolerance for Nelder Mead Simplex search
errorTolES	Error tolerance for Golden Section search on the EB edge

Table 4.2 Function parameters in *decelOpt.m* and its subfunctions

4.3 Search and Cost Functions

```
function [xOpt xOptCost] = NMSimplexSearch(x0, errorTolNM,
    errorTolBC, maxCount, dDRMax, dDRMin, dEBMax, dTotMin, dTotMax,
    v_0, v_target, c_Time, sigma_v0, sigma_idle, drivingResData,
    downShiftMap, AVectSlope, sVect, objIndex)
```

NMSimplexSearch.m contains an implementation of the Nelder Mead Simplex Search algorithm [Nelder and Mead, 1965] and is being used for local optimization of the most promising samples from the global search. The function is being fed the $\bar{x}$ from the global search as starting point for the local search, as well as all the parameters required to call *costFuncNM.m*.

```
function [xOpt xOptCost] = EBEdgeSearch(errorTolES, errorTolBC,
    v_0, v_target, dTotMin, dTotMax, dEBMax, dDRMax, sigma_v0,
    sigma_idle, c_Time, drivingResData, downShiftMap, AVectSlope,
    sVect, objIndex)
```

EBEdgeSearch.m is used in the case when solutions returned from *NMSimplexSearch.m* have DR times that are close to the minimum allowed DR time. A lower bound for the DR time exists to prevent gear shifting into neutral for very short times before shifting back into drive mode, as explained in Section 3.4. When the optimum from *NMSimplexSearch.m* is found to be close to the minimum allowed DR time, it might be the case that a deceleration using purely EB would yield a lower value of the cost function. To determine if this is the case a search is being performed using *EBEdgeSearch.m* along the edge of the optimization area representing pure EB solutions. If this search yield a solution with a lower associated cost than *NMSimplexSearch.m*, then this solution is taken as the optimum.

The search algorithm being used for the one dimensional search along the edge is the method introduced in [Kiefer, 1953], usually referred to as Golden Section search.

```
function cost = costFuncNM(x, v_0, v_target, dDRMax, dDRMin,
    dEBMax, dTotMin, dTotMax, sigma_v0, sigma_idle, c_Time,
    drivingResData, downShiftMap, AVectSlope, sVect, objIndex,
    errorTolBC)
```

costFuncNM.m evaluates the fuel and cost time functions for a given decision variable $\bar{x}$, denoted `x` in the code, and checks for the validity of the current $\bar{x}$. If $\bar{x}$ represents a non-feasible solution then that solution is being punished by adding a high punishment term to the cost function. The punishment term is proportional to the distance from $\bar{x}$ to the feasible set. The main subfunctions of *costFuncNM.m* are *timeCost.m*, *constSpeedCons.m* and *ft_s.m*. The first two do what their names indicate and *ft_s.m* is used to compute the time spent in DR which in turn gives the idle engine fuel consumption for the deceleration profile defined by $\bar{x}$.

```
function [timeInt,x_Intersect] = timeCost(x, v_0, v_target,
    drivingResData, downShiftMap, AVectSlope, sVect, objIndex,
    dTotMin, errorTolBC)
```

timeCost.m plays a central role in the optimization since it computes the time cost associated with a deceleration profile given by $\bar{x}$. This means computing the integral depicted in Figure 3.3. Computing this integral means integrating the difference between the time-ideal driving curve $v_{t\_opt}(s)$ and the deceleration driving curve $v(s)$ from the DR starting point up to the intersection of the optimized deceleration with the braking curve. This integral has been displayed in Figure 3.4. However, this is most conveniently done by integrating each function separately. To do this, the intersection point $d_{intersect}$ between the deceleration profile and the braking curve must be found. The intersection point can be found as the solution to

$$v_{brake}(s, v_{target}) - v(s) = 0. \tag{4.1}$$

The root finding algorithm employed is the Modified Regula Falsi, decribed in

[Anderson and Björck, 1973]. The method is similar to the secant method, with the difference that the next point in an iteration is always chosen so that the last two points bound the root. This ensures stability but poses some issues with the rate of convergence, which is the main consideration for the modifications presented in [Anderson and Björck, 1973].

With $d_{intersect}$ for the current deceleration profile known, the integrals for time cost can be computed. This is done through numerical integration using Simpson's rule. This allows for good accuracy with a low number of evaluated points for $v(s)$ and $v_{brake}(s, v_{target})$.

```
function cC = constSpeedCons(dTot, dDRMax, AVectSlope, sVect,
    drivingResData, v_0, sigma_v0, objIndex)
```

constSpeedCons.m computes the constant speed driving consumption for a deceleration sequence with total deceleration distance `dTot`. The constant fuel consumption is computed as the amount of fuel being consumed from the point at distance `dDRMax` from the speed limit change up to the point at distance `dTot` from the speed limit change. See Figure 1.1 for a sample deceleration profile. In *constSpeedCons.m*, the different slope segments given by `sVect` and `AVectSlope` are being accounted for one by one, and an accumulated constant consumption is being computed as the sum of the consumption along each slope segment. To compute the fuel flow in a given segment, *sigmaSlope.m* is being called.

```
function sigma_v0_slope = sigmaSlope(A_slope, v_0, sigma_v0,
    drivingResData)
```

sigmaSlope.m computes the fuel flow for driving at constant speed `v_0` in a slope with additional deceleration `A_slope`. The computation is done according to the method explained in Section3.3.

4.4 Trajectory Calculation Functions

```
function v_s = fv_s(opt, s, v_0, drivingResData, downShiftMap,
    AVectSlope, sVect, posOffset)
```

fv_s.m computes the speed `v_s` for a vehicle that has travelled a distance `s` in DR or EB mode, beginning at speed `v_0`. The speed is computed over sections with constant slope, given by `AVectSlope` and `sVect`. In calculations for EB, gear switching is also taken into account. If `posOffset` is 0 then calculations are made assuming the deceleration sequence is started at position 0 in `sVect`, which represents the vehicle's current position. For positive values of `posOffset`, the computation is made from a point at distance `posOffset` from the current vehicle position, and this point is taken as the zero position for the calculation. This is important in the case where the slope is varying, where the starting position dictates the interaction between slope resistance and aerodynamic drag. On flat ground, the starting point has no influence on the deceleration trajectories.

During computation, the algorithm steps through the constant slope segments given by `sVect` and `AVectSlope`. For each segment, the end speed is computed using Equation (2.13). The end speed for the current segment is then used as the initial speed for the next segment. If the end speed for a segment drops below a gear shifting point, given in `downShiftMap`, the algorithm steps back to compute the distance at

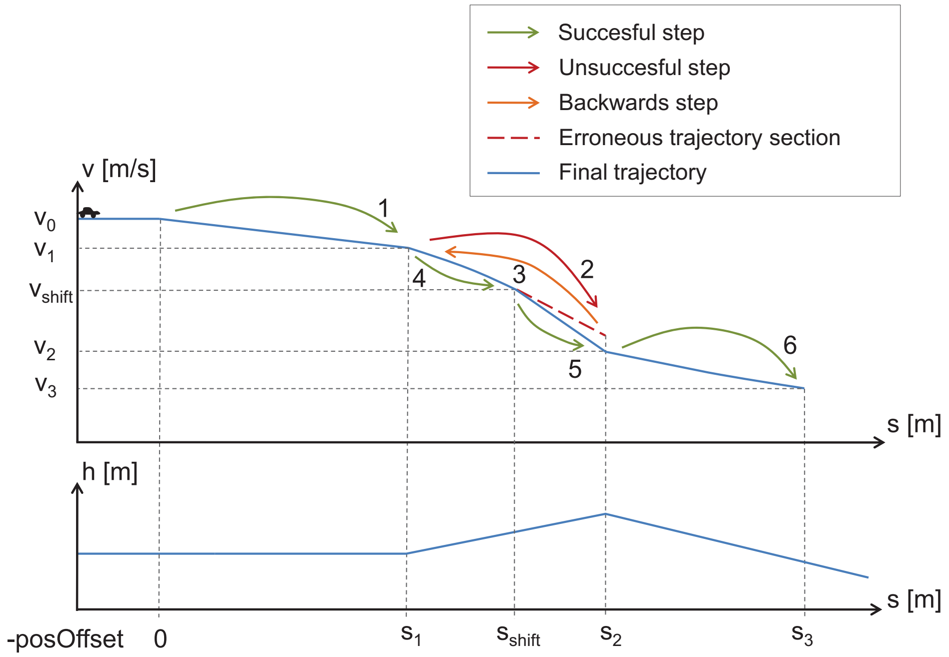


Figure 4.2 Illustration of a trajectory calculation in *fv_s.m*

which the gear switch would occur, using Equation (2.12). The current segment is then split into two segments where the end speed of the second part is computed with the gear switching speed as initial speed and using the *A* parameter associated with the new gear, chosen from `drivingResData`. An example of a speed trajectory calculation over a hill involving one gear shift is shown in Figure 4.2. The requested `s` is denoted s_3. In step 1 the end speed v_1 for the segment $0 \rightarrow s_1$ is successfully calculated. Thereafter, in step 2, the end speed of segment $s_1 \rightarrow s_2$ is computed, but since a gear shifting speed v_{shift} has been traversed the calculation needs to step back to (s_1, v_1). In step 4, the position s_{shift} at which the gear shift will occur is computed, after which the new end speed of segment $s_1 \rightarrow s_2$ is computed in step 5. Finally the requested speed v_3 is computed in step 6.

Whether DR or EB is being performed is decided by the `opt` argument which accepts either 'DR' or 'EB' as a string. If no valid string is given then computations are being performed for DR.

```
function s_v = fs_v(opt, v, v_0, drivingResData, downShiftMap,
    AVectSlope, sVect, posOffset)
```

fs_v.m is the inverse function of *fv_s.m* and computes the distance `s_v` for which the vehicle decelerates from `v_0` to `v`. As with *fv_s.m* the computation is made from a point at distance `posOffset` from the zero position in `sVect`. The computations are carried out by first building a combined vector of *A*-parameters for both slope and gear switches, along with a corresponding speed vector. This is done using *fv_s.m*. The speed vector indicates the speeds where the different *A*-parameters should be used and contains all switching speeds down to the requested speed `v`. This procedure is a way of getting around the problem that changes in slope are given at specified distances, while gear switches are defined at specified vehicle speeds. When the speed vector and *A*-vector have been built the distances travelled between each pair of elements in the speed vector are computed by Equation (2.12) and summed up.

```
function t_s = ft_s(opt, s, v_0, drivingResData, downShiftMap,
    AVectSlope, sVect, posOffset)
```

ft_s.m computes the time required to travel a distance `s` in EB or DR mode, beginning at speed `v_0` at position `posOffset` from the zero position in `sVect`. In Section 2.4, Equation (2.11) is given to compute speed as a function of time, but no inverse to (2.11) is given. This is because the inverse of (2.11)[1] is not unique. The two inverse functions both display discontinuities at varying places depending on `v_0`. It therefore seemed like a more trustworthy solution to create an inverse by numerical inversion of (2.11). This numerical inversion is performed by *ft_s_s_t.m*, which is described below.

The computations are being performed identically with those in *fv_s.m*, with the addition that *ft_s_s_t.m* is used to compute the travel time for each segment. The end speed of each segment still needs to be computed, in order to provide an initial speed for the next segment and to keep track of downward gear shifts.

```
function t_s = ft_s_s_t(s, v_0, A, B)
```

ft_s_s_t.m computes the travel time over a segment with constant A, length `s` and initial speed `v_0`. Since no inverse to (2.11) exists that is continuous over the desired range of values for `v_0` and A, (2.11) is being used and a numerical inverse is calculated. Calling the position for which the time should be calculated s_1, this corresponds to the root finding problem

$$s(t) - s_1 = 0 \tag{4.2}$$

where $s(t)$ is given by Equation (2.11). In order to find an initial bounding interval for the root, a course search along s(t) is first made. If s_1 is within the range of the deceleration in question a bounding interval is set up and t is thereafter found through Modified Regula Falsi as outlined in the description of *timeCost.m*.

```
function vEB = fv_sEB_dDR(dDR, s, v_0, drivingResData,
    downShiftMap, AVectSlope, sVect, posOffset)
```

fv_sEB_dDR.m is an auxiliary function used for computing v as a function of s for EB deceleration after a distance `dDR` of DR with initial speed `v_0`. `posOffset` is the offset from position 0 in `sVect` to the point where DR mode is initiated. *fv_sEB_dDR.m* simply calls *fv_s.m* two times in order to find the end speed for the DR section as initial speed for the EB section.

```
function s = fs_vtarget_v0(opt, v_target, v_0, drivingResData,
    downShiftMap, AVectSlope, sVect, dTotMax, objIndex)
```

fs_vtarget_v0.m computes the distance `s` required to achieve speed `v_target` at the target point starting out with speed `v_0` in deceleration mode selected by the `opt` argument. *fs_vtarget_v0.m* is used for computing the maximum DR and EB distances where no RB should be performed. Computations are performed backwards, starting at `v_target` at using Equation (2.13) with negative distances in order to find the speeds for every point in sVect up until `v_0`. If a gear shifting speed is passed, then calculations are made to find the corresponding position, just as for *fv_s.m*.

[1] Computed in Matlab's Symbolic Math Toolbox

```
function [vOut,xOut] = brakingCurve(xIn, v_target, vIn)
```

brakingCurve computes points on the braking curve. It has two mandatory and one optional arguments. For a given `xIn` and `v_target` the corresponding speed `vOut` is computed as `vOut` $= v_{brake}(-xIn, v_{target})$, that is `xIn` must be given as the absolute distance from the target point. If a third argument `vIn` is given, the corresponding distance `xOut` to the target point is computed, also as a positive number. As long as the function interface is correct then *brakingCurve* can be exchanged without modification to the remaining functions.

5. Simulink Implementation

The Deceleration Optimizer Matlab function performs optimization for one specific speed transfer at a time to produce optimal deceleration profiles for the given profile. However, in order to function in a vehicle some additional functionality is required to assist the Deceleration Optimizer. Therefore an auxiliary Simulink model has been built which is accounted for below. In the text, 'citation marks' have been used to mark out names of blocks and signals in the Simulink model. The exception is Deceleration Optimizer Shell, which will be taken as the name of the entire auxiliary function and written without citation marks.

The main tasks of the auxiliary model are:

- Monitoring of driving situation and upcoming speed transfers.
- Triggering of the Deceleration Optimizer when a speed transfer approaches.
- Feeding of current navigation data.
- Realization of DR and EB commands, based on optimal deceleration pattern and vehicle position.
- Issuing of RB command when vehicle trajectory intersects with the braking curve.
- Monitoring of the vehicle deceleration trajectory.

For simulation and testing purposes, a simulated vehicle environment has been implemented along with the Deceleration Optimizer Shell. It consists of a navigation system block which provides navigation information in the same format as the in-vehicle navigation system and a vehicle model implementing the longitudinal dynamics properties of an automobile with the ability to switch between DR, EB, RB and regular driving following a reference speed.

The top level of the simulation Simulink model is displayed in Figure 5.1. The Deceleration Optimizer Shell in the middle is connected to a Navigation system block and a vehicle model. In the vehicle implementation 'Deceleration Optimizer Shell' will be the same but connected to the vehicle navigation system and the speed regulator.

The signals from the navigation system are those described in Section 4.1. These signals feed into 'Deceleration Optimizer Shell'. Additionally, there is a separate slope vector for the vehicle model, allowing simulation of discrepancies between the navigation information given to the Deceleration Optimizer and the real driving situation. Additionally there is a selector for choosing c_t. This input allows to choose a value of c_t from a predefined set. The ambient air temperature, which affects air density, and the current vehicle speed are also provided. In the vehicle implementation these values will be provided from the CAN-bus.

The outputs from the Deceleration Optimizer Shell are the control signals that tell the vehicle whether to drive normally or to perform DR, EB or RB. These signals are denoted 'DRBool', 'EBBool' and 'RBBool' respectively, and are given as boolean signals that are mutually exclusive. Obviously, a car can only do one of the three tasks at once. A reference speed is provided through v_{ref},denoted 'vRef' in the Simulink model. The generation of v_{ref} will be discussed in the subsection on the 'Monitoring and Braking' subsystem. Additionally, 'info_Bus' provides information for use in a prototype display during in-vehicle function testing. 's_Next_Limit' and

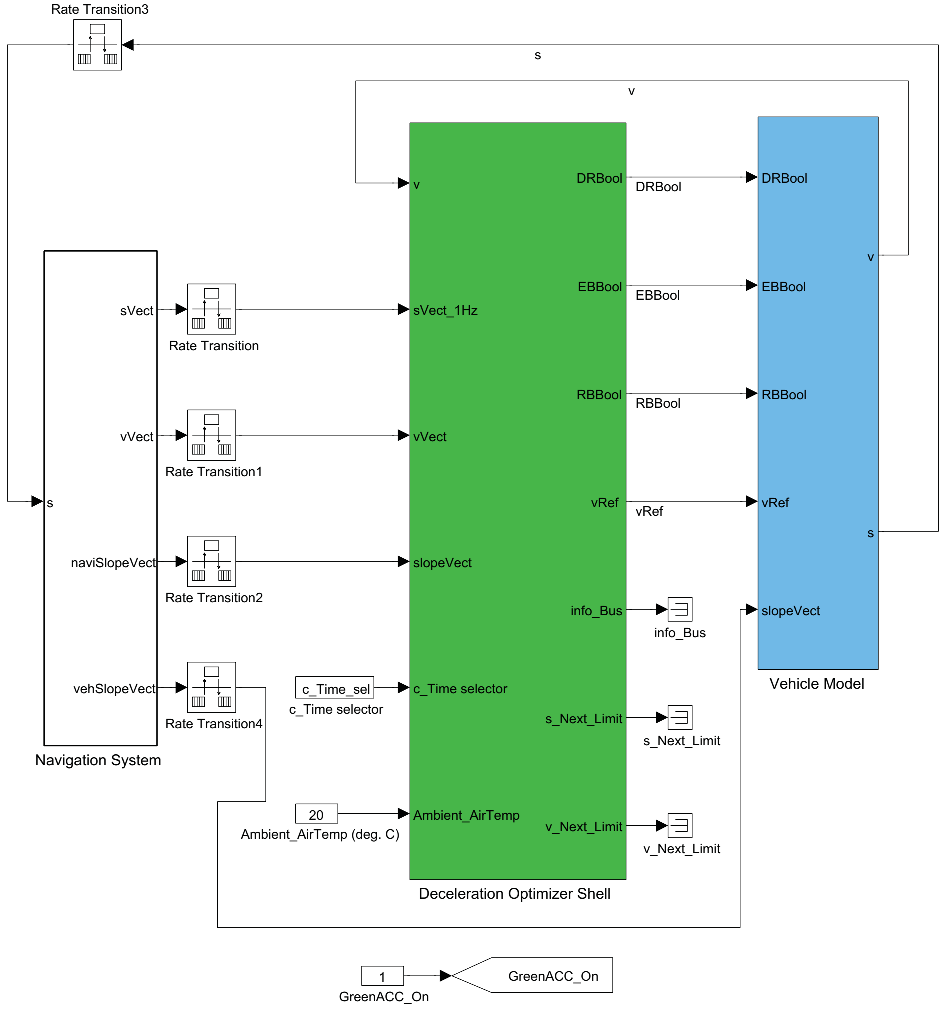


Figure 5.1 Top level of the simulation Simulink model

’v_Next_Limit’ are present because they are sent to the vehicle speed regulator. Finally there is a ’GreenACC_On’ signal which will be toggled by the speed regulator to allow triggering when the GreenACC is activated, in case there is an upcoming speed transfer close by.

All inputs variables to the simulation model have been defined in an external setup file, which is run before running a simulation.

5.1 Deceleration Optimizer Shell

The Deceleration Optimizer Shell contains several layers of subsystems. The first layer is displayed in Figure 5.2. It contains some auxiliary functions for treatment of the navigation information and selection of c_t, denoted ’c_Time’ in the Simulink model.

Frequency Transformer In the vehicle, navigation information arrives as packages from the navigation system to the reconstructor at an approximate frequency of 1 Hz. The arrival of packages is not precisely scheduled, instead the reconstructor is triggered by an interrupt each time a new package arrives. The Deceleration Optimizer Shell continuously monitors the upcoming driving situations in order to trigger

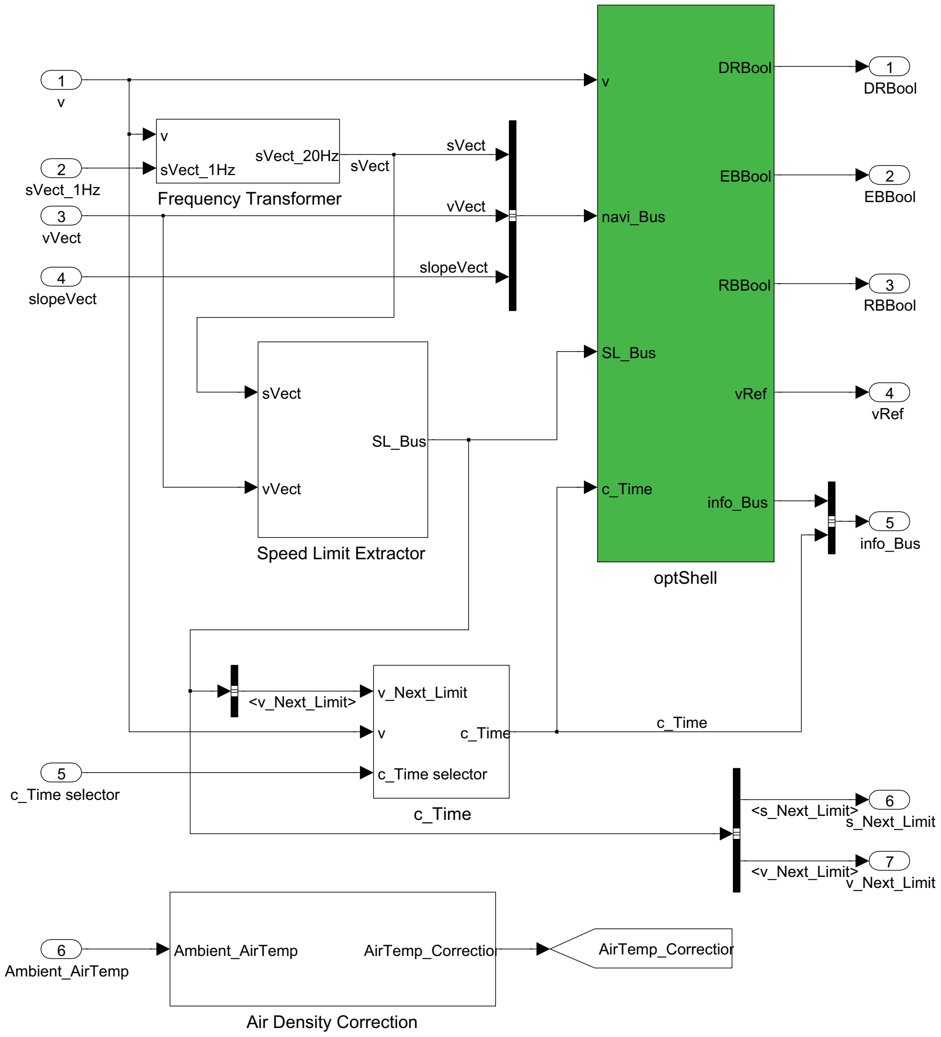


Figure 5.2 Deceleration Optimizer Shell Simulink model

the Deceleration Optimizer at appropriate times and to set the. An update frequency for the vehicle position of 1 Hz would introduce a lot of delay in the optimization triggering and in the subsequent execution of the optimal deceleration pattern.

The task of the ’Frequency Transformer’ is therefore to produce an signal ’sVect_20Hz’ with an update frequency corresponding to the Deceleration Optimizer Shell sample time of 50 ms. This is done by estimating the distance travelled between each update in ’sVect_1Hz’ by integrating the vehicle speed over time and subtracting the distance obtained from ’sVect_1Hz’. When ’sVect_1Hz’ is updated, the integrator is reset.

Speed Limit Extractor The task of this subsystem is to extract the distance to the next speed limit, its index in the navigation information vectors, the current speed limit and the speed of the next speed limit. The speed limit information is gathered in the ’SL_Bus’ signal.

c_Time The block chooses the value of c_t, denoted ’c_Time’ in the Simulink model, corresponding to the value of ’c_Time’ selector. It also performs the modifications to c_t described in Section 5.3.

Air Density Correction The air density, which affects aerodynamic drag as described by Equation (2.2), depends on air temperature and air pressure an in (2.3). The variation is usually slow and it can therefore be considered constant for the duration of each optimization. However, the difference in air density, which is proportional to drag, between a hot summer day (30 °C) and a cold winter night (−20 °C) is about 20 % at 1013 mbar air pressure. Since the ambient temperature is readily available in the vehicle a correction factor for the aerodynamic drag is calculated and applied to the B-parameter from Equation (2.7) where the car resistance data enters the Simulink model.

The optShell subsystem

decelOpt with Trigger The 'decelOpt with Trigger' subsystem contains the Deceleration Optimizer Embedded Matlab Function as well as the triggering system. The implementation of the Embedded Matlab function is discussed in Chapter 4. However, for the optimization to do what is intended it needs to be provided with current navigation information and it needs to be executed at suitable occasions. Since the optimization requires extensive computations it would not be appropriate to run it at the same sample rate as the rest of the model. To allow enough computing time it is run at a sample time of 500 ms. Optimizations should also only be performed at appropriate times. During a deceleration maneuver, the original $d_{tot_{opt}}$ and $d_{EB_{opt}}$ should be retained throughout the deceleration. If care is not taken, an optimization triggered in the middle of a deceleration at a lower speed than the original v_0 would overwrite the original deceleration profile and change the outcome. This has been solved through a triggered subsystem that contains the Deceleration Optimizer Embedded Matlab function, with five triggering conditions for different situations. During an ongoing deceleration, the triggers are disabled. The five triggering conditions in decelOpt with Trigger are:

- **Speed limit crossing** Each time the vehicle passes a change in speed limit, a trigger is issued. This is useful in the case where one deceleration phase is immediately followed by another one. An optimized deceleration profile for the second deceleration is then immediately obtained.
- **New speed limit approaches** The first trigger is made a short time before the distance to the next speed limit equals $d_{DR_{max}}$. This means that deceleration through maximum DR is still possible. At the same time the trigger should be performed as late as possible, to avoid decreases in vehicle speed between the optimization and the onset of the deceleration profile. The trigger condition is given by

  $$d_{NL} < d_{DR_{max}} + t_{trig} v$$

  where t_{trig} is the trigger preview time.
- **Speed increase from last trigger** When the vehicle accelerates after a trigger, the conditions of the optimization problem, and thereby also the optimal deceleration pattern, changes. Higher speeds generally generate longer deceleration profiles, and speed increases therefore has to be continuously monitored. The trigger condition is given by

  $$v > v_{trig} + v_{incTol}$$

  where v_{trig} is the speed for the last trigger and v_{incTol} is the tolerated speed increase before a new trigger is performed.

- **Speed decrease from last trigger** When speed decreases from the last trigger, the optimal deceleration profile for the lower speed will be shorter than the current one. In the case of a speed decrease it is therefore sufficient to check the speed right before the onset of the current deceleration pattern. The trigger condition is
$$(v < v_{trig} - v_{decTol}) \wedge (d_{NL} < d_{tot_{opt}} + t_{trig}v)$$
where v_{decTol} is the tolerance in speed decrease before a new optimization is performed.
- **GreenACC is turned on** When the GreenACC is being activated, an immediate optimization is required to make sure the Deceleration Optimizer produces an optimized deceleration also if a deceleration phase is nearby.

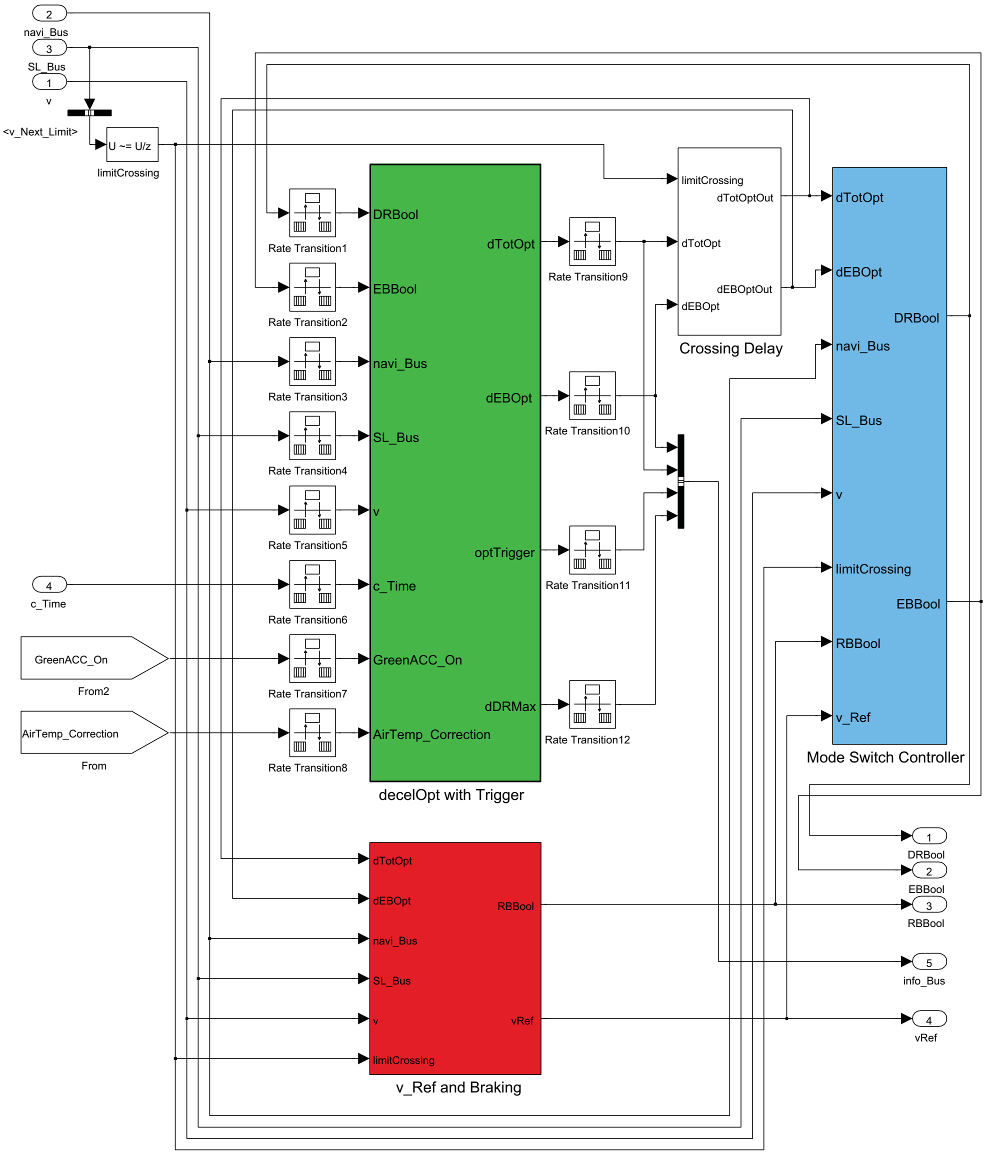


Figure 5.3 optShell subsystem

Mode Switch Controller The 'Mode Switch Controller' block basically performs the procedure described in the last part of Section 3.2. However, in the case where the vehicle's deceleration trajectory differs from the predicted, adjusting the DR → EB switching point offers the opportunity to compensate for speed deviations in the DR phase.

- **Vehicle travels faster than predicted in DR mode** When deceleration in DR mode is weaker than predicted, correction can be made through earlier shift into EB mode than originally intended. To this end, during deceleration in DR mode, the elongation of the intended EB deceleration path $v_{EB}(s)$ is continuously calculated for the current vehicle position. When the vehicle speed intersects this trajectory the switch into EB mode is made. The idea is that if the EB deceleration proceeds as predicted, the vehicle will be brought back to the optimal trajectory despite the deviance deviance in DR deceleration.
- **Vehicle travels slower than predicted in DR mode** When DR deceleration is stronger than predicted, correction can be made by delaying the switch into EB past the originally intended point. Since deceleration in DR mode in any case is weaker than deceleration in EB mode, this allows for the vehicle to catch up with the intended deceleration path. In this case, v_{ref} indicates the intended deceleration path, and switching into EB mode will be made when the vehicle speed intersects v_{ref}.

To summarize, switching from DR mode will be made when

$$\begin{cases} v(s) \geq v_{EB}(s) & \text{if} \quad d_{NL} > d_{EB_{opt}} \\ v(s) \geq v_{ref}(s) & \text{if} \quad d_{NL} \leq d_{EB_{opt}} \end{cases}$$

Crossing Delay The task of the 'Crossing Delay' block is to prevent malfunction in the case of a speed limit crossing when the next limit is close behind the first one. What might happen in this case, since the Deceleration Optimizer runs at a slower rate than the remaining model, is that the old values of $d_{tot_{opt}}$ and $d_{EB_{opt}}$ remain at the outputs of the 'decelOpt with Trigger' block. If $d_{tot_{opt}}$ is larger than the new d_{NL} the vehicle enter DR mode, and if the old $d_{EB_{opt}}$ is larger than d_{NL} then the vehicle enters EB mode, although this might not be the optimal deceleration for the new speed transfer. To avoid such situations, 'Crossing Delay' sets 'dTotOptOut' and 'dEBOptOut' to zero immediately after a change in speed limit is crossed, and enables feedthrough of 'dTotOpt' and 'dEBOpt' when it detects that a new optimization result is ready.

vRef and Braking subsystem

The 'vRef and Braking' subsystem is responsible for generating v_{ref} as predicted by the Deceleration Optimizer and for generation of 'RBBool'. The 'vRef and Braking' subsystem is displayed in Figure 5.4.

Brake Monitor This block monitors the vehicle trajectory to determine when it is time to start braking. Switching from EB to RB mode occurs when the vehicle trajectory intersects $v_{brake}(s, v_{target})$. From this point the vehicle speed regulator is supposed to track v_{ref} through RB. To keep 'RBBool' constantly high until the braking is finished, a state machine keeps track of the status of the braking. v_{brake} is also generated in 'Brake Monitor'.

Passive Deceleration For the sake of reference generation, this block contains a simple vehicle model which performs deceleration in DR and EB mode according to the current vehicle parameterization. That is, this model vehicle behaves exactly as predicted by the Deceleration Optimizer, when given the corresponding 'DRBool' and 'EBBool' control signals. The output speed $v_{rolling}$ is denoted 'v_Rolling' in the Simulink model.

Ref Gen Mode Switch Controller For the generation of a reference speed following the predicted deceleration profile, 'DRBool' and 'EBBool' are needed that follow the predicted pattern and that are not affected by deviations in vehicle deceleration. These are generated by 'Ref Gen Mode Switch Control'.

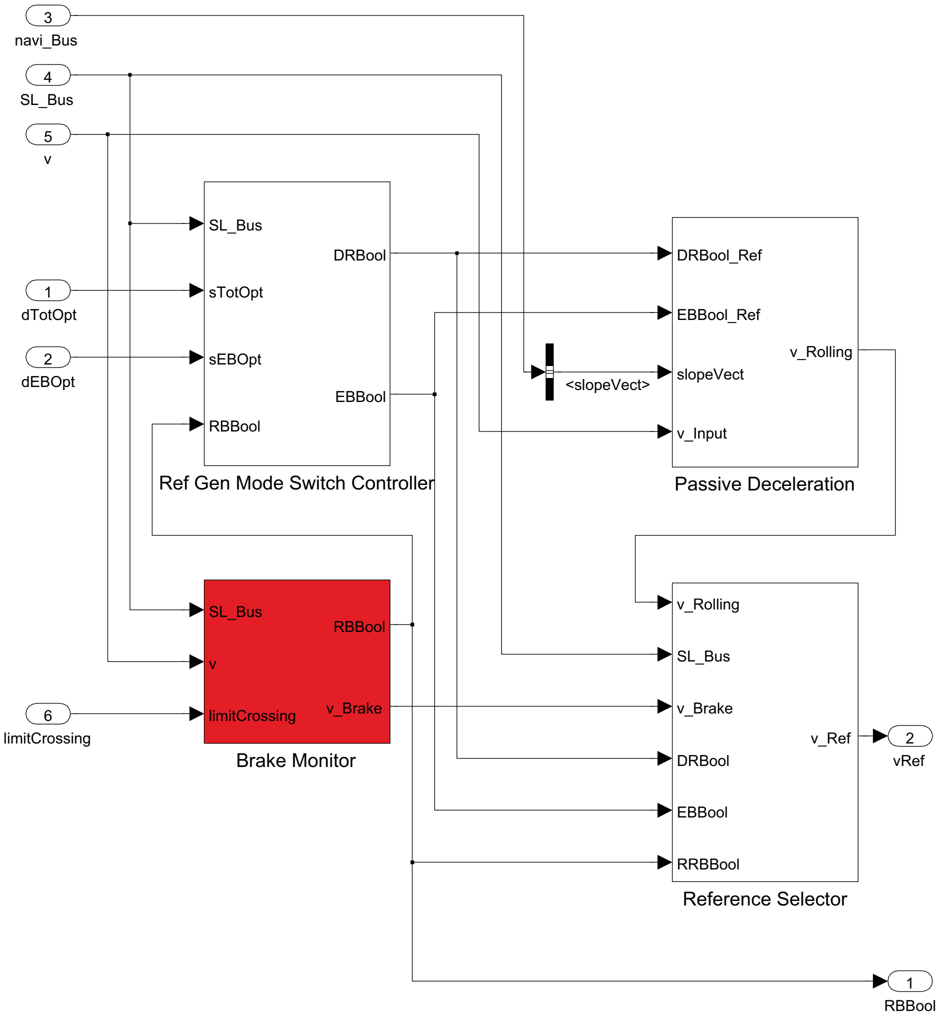


Figure 5.4 Monitoring and Braking subsystem

Reference Selector The purpose of the v_{ref} signal is threefold. It is also composed of three different signals, and which signal to use in which situation is dictated by the 'Reference Selector' block. During constant speed driving, v_{ref} supplies the speed regulator with the current speed limit, which is then the control objective. When a deceleration profile begins, the Deceleration Optimizer Shell issues 'RBBool' followed by 'EBBool'. The vehicle speed regulator then has no control objective other than keeping the vehicle in DR or EB mode. v_{ref} is then used to indicate how the vehicle would behave, according to the calculated vehicle trajectory, by switching to $v_{rolling}$. This makes it possible to detect abnormal deviances in the ongoing deceleration, that could be due to slopes not known by the navigation system, strong winds or other disturbances. When the deviance is outside of an acceptable interval, the speed regulator can then initiate emergency maneuvers such as resuming constant speed driving, to avoid excessively low speeds or a vehicle that completely comes to a halt.

The handling of such exceptions has not been further researched in this thesis project. Such considerations are indispensable for a final implementation, but in the case of a prototype function the main interest is a proof of principle and to demonstrate how a control strategy performs when everything does work. Therefore, the main effort has been put on implementation of the function itself, and work with the error handling has been limited to providing signals for detection and analysis of deviating behaviour.

When the switch from EB to RB is made, then v_{ref} assumes its third task, namely that of providing a reference speed for braking. v_{ref} then switches from $v_{rolling}$ to v_{brake}.

$$v_{ref} = \begin{cases} v_{limit} & \text{normal driving} \\ v_{rolling} & \text{DR or EB mode} \\ v_{brake} & \text{RB mode} \end{cases}$$

5.2 Vehicle Emulation

In order to test the Deceleration Optimizer in simulation, the behaviour of the systems connected to the Deceleration Optimizer Shell needs to be emulated. The word emulated is used because the objective is to produce correct external behaviour, without regard to how the systems really work. The object of interest is not the emulated vehicle components themselves, but how the Deceleration Optimizer reacts to their behaviour. What is needed is a navigation system model that mimics the output of the reconstructor and a vehicle model that reacts to the control signals from the Deceleration Optimizer and in turn provides feedback in the form of speed and position for the Deceleration Optimizer and navigation system.

Navigation System Emulation

The emulated navigation system provides sVect, vVect and slopeVect depending on the current position of the simulated vehicle, according to the description in Section 4.1. To simulate driving with the Deceleration Optimizer along a given section of road, the emulated navigation system takes distance, speed limit and slope vectors for the road section in question as input with zero position in the distance vector as the starting point of the road section. As the vehicle model traverses the road section, the 'Navigation System' outputs translated versions of the navigation information vectors according to the vehicle's position.

In order to investigate the influence of errors in the slope information in the navigation system, the possibility exists to provide differing slope vectors for the Deceleration Optimizer and for the vehicle model.

Vehicle Model

The vehicle model designed for the Deceleration Optimizer development is a minimalistic model featuring only what is needed for testing and debugging. The required capabilities are:

- Speed control to a constant v_{ref} during constant speed driving
- Passive deceleration in DR mode
- Passive deceleration in EB mode

- Active braking for tracking of v_{ref} in RB mode

Two subsystems, one for active driving and one for passive deceleration provides two different acceleration rates. Which one to use is determined by 'DRBool', 'EBBool' and 'RBBool'. The resulting acceleration is fed through two integrators in order to obtain speed and position of the emulated vehicle. The 'Vehicle Model' subsystem can be viewed in Figure 5.5.

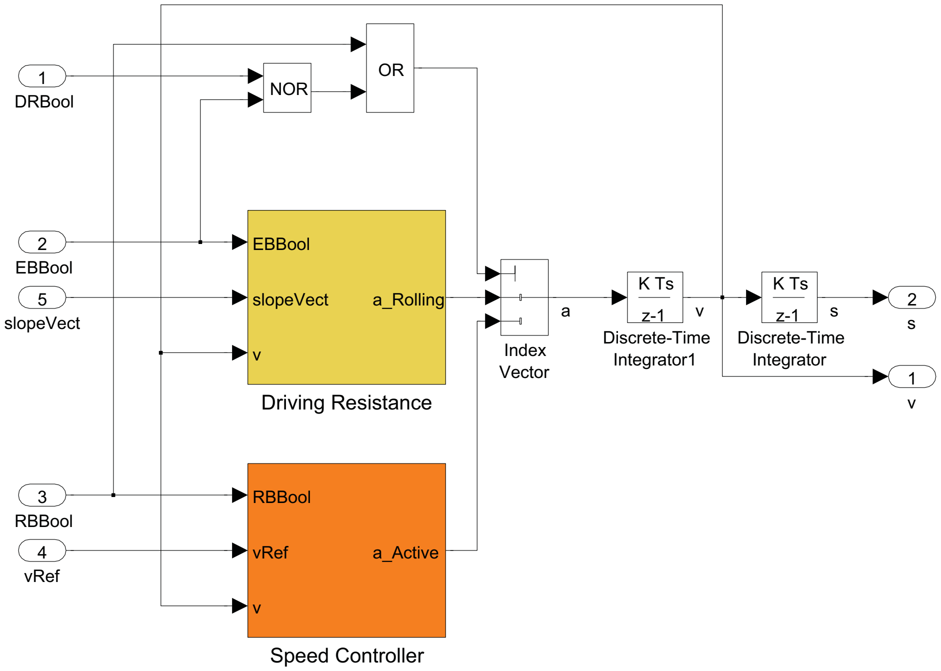


Figure 5.5 Simulink vehicle model

Speed Controller subsystem For testing of the Deceleration Optimizer it is not important how the vehicle accelerates. Speed control during constant speed driving was therefore implemented as a P-controller acting on the control error $e = v_{ref} - v$ with $a = \text{sat}_{a_{max}}(K_p e)$ where $\text{sat}_{a_{max}}$ is the saturation function with limit a_{max} and K_p is the controller gain. In the simulation environment this suffices to bring the vehicle speed to v_{ref} and achieve constant speed thereafter.

Braking according to v_{ref} was achieved using the same controller type with different saturation limit a_{max} and controller gain K_p. For the tracking problem the P-controller does not achieve perfect tracking, but with a high gain the result is good enough for the task at hand.

Passive Deceleration subsystem For passive deceleration in DR and EB mode, the different driving resistance components were calculated based on the current mode of operation and the vehicle speed. Slope resistance was calculated from the current slope provided from the navigation system. The resulting total acceleration was then given as output. The acceleration is usually negative, but in the case of downhill slopes positive acceleration in DR and EB mode occurs regularly.

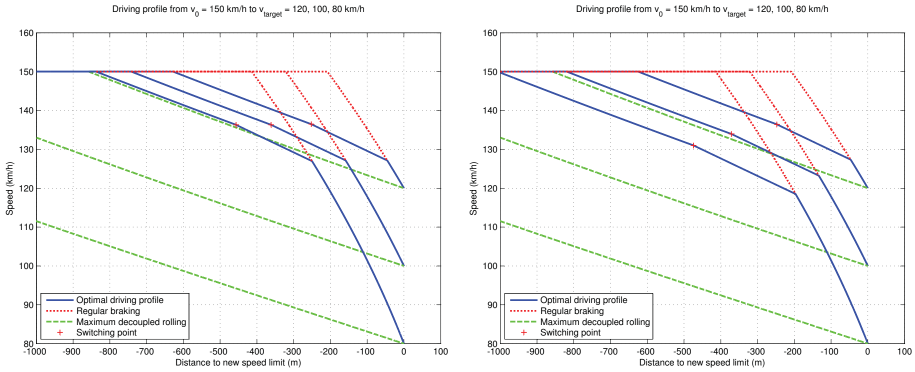
Additionally, for testing and debugging of the correction mechanism for the DR $\rightarrow$ EB switching point in the Deceleration Optimizer Shell, deviance in the A and B parameters compared to those given to the Deceleration Optimizer Shell can be induced.

5.3 Choosing Time Cost Weight Coefficient

Apart from the choice of an appropriate cost function for the time cost, the time cost coefficient c_t is the most important design parameter for the tuning of the Deceleration Optimizer. The process of adjusting c_t is fundamentally a matter of selecting which solution from the set of Pareto optimal solutions to use. Choosing a high value for c_t corresponds to a preference for speedy driving in pace with the surrounding traffic. A low value for c_t means that the time cost is depreciated in favor for fuel cost, so that longer deceleration phases are accepted to obtain greater fuel savings.

Another aspect to the choice of c_t is that of consistent behaviour of the Deceleration Optimizer. It is necessary that the driver understands what it does and subjectively experiences that the generated deceleration profiles are consistent over the entire range of possible speed transfers.

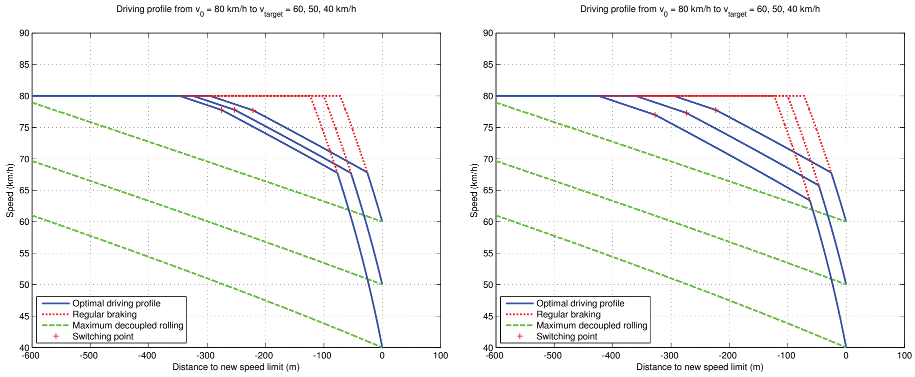
At first a constant time cost seems like an appropriate choice. However, as can be seen from figures 5.6(a) and 5.7(a), this has as consequence a behavior of the chosen Pareto optimal solution to depend heavily on v_0 but very weakly on v_{target}. It seems more intuitive to allow longer deceleration phases for big changes in driving speeds.



(a) Constant c_t

(b) Varying c_t

Figure 5.6 Deceleration profiles from 150 km/h



(a) Constant c_t

(b) Varying c_t

Figure 5.7 Deceleration profiles from 80 km/h

A speed transfer is characterized by an initial speed v_0 and a target speed v_{target}. To get a better representation of what we try to achieve, a useful change of coordinates will be to parametrize a given speed transfer as v_0 and relative speed decrease

$$v_{diff} = (v_0 - v_{target})/v_0.$$

By choosing c_t based on these parameters, the relative length of deceleration profiles for different v_0 and v_{diff} can be manipulated.

The generating function will be constructed as

$$c_t = c_{t_{base}} \cdot f(v_0) \cdot f(v_{diff})$$

where $f(v_0)$ and $f(v_{diff})$ will be used to make deceleration profiles subjectively consistent over different initial speeds and relative speed decreases, and c will be used to adjust the overall length of the deceleration profiles.

As a starting point for constructing c_t, a constant value was chosen. Thereafter, $f(v_0)$ and $f(v_{diff})$ were tuned by using second degree polynomials in v_0 and v_{diff} respectively and by thereafter examining the resulting deceleration patterns. Since there is no objective answer to the question what a good set of deceleration patterns really look like, the tuning process is by necessity subjective. To tell if c_t is well tuned, the end result needs be examined and verified through experimentation in a test vehicle. What is important is to find out what could work for a majority of the BMW customers, or at least those who want to use the GreenACC, something that requires thorough testing with a lot of drivers. The aim of this project is, however, mainly to find ways to handle these consideration and to get a rough feeling for what settings that are reasonable. The final deceleration profiles for can be compared with the ones for constant c_t in figures 5.6(a)-5.6(b) and figures 5.7(a)-5.7(b).

6. Simulation and Efficiency Improvements

6.1 Simulation Environment

At EG-61, there is an extensive set of Dymola libraries for simulation of conventional, hybrid and electric vehicles. Dymola is a graphical implementation and compiler for the Modelica language. Modelica is an object oriented modeling language that supports simulation of systems made up of many different types of interacting components. Its multi-domain capabilities as well as its focus on reuse and extensive standard model libraries makes it well suited for, among others, applications in the Automotive Industry [Otter and Elmqvist, 2001]. It differs from block oriented modeling tools such as Simulink, in its non-casual modeling capabilities, which means that signal flow is bidirectional and interactions are sorted out during the actual solving of the systems of equations. This allows for more accurate and involved modeling of physical systems, since the causal direction of an interaction must not be predefined in beforehand [Modelica Association, 2000]. Since the work at EG-61 is focused on fuel efficiency and consumption reduction, the libraries used by the department are mainly focused on fuel consumption. This does not mean that simulation result will be correct in terms of absolute fuel consumption. They do, however, give a good representation of relative fuel consumption, so that different control strategies and driving patterns can be compared with good accuracy. A printout of the top level of the EG-61 full vehicle model is shown in Figure 6.1

The topology of the model is such that the different subsystems of the vehicle have been implemented as separate model blocks with connections representing physical, electrical or digital interconnections between the components. In the model there is also a driver block and a track block, which allows for simulation of different driving conditions with various driver behaviours. Particularly important are the different official driving cycles used for consumption testing, the European NEDC[1], the American FTP-75[2] and the Japanese driving cycle *10-15 Mode*. The benefits of the Deceleration Optimizer cannot, however, be assessed by simulation of any of the official driving cycles. This is because these driving cycles are defined through speed profiles over time that need to be very closely followed during testing. The fuel savings of the Deceleration Optimizer are due to optimization of the speed profile over distance and must thus be simulated under more flexible driving conditions.

6.2 Dymola Simulation

One option for assessing the consumption benefits of the Deceleration Optimizer would be to incorporate it in a Dymola driver or track model and have the function run along with the Dymola simulation. Since the aim of the project is to get a prototype function running in an experiment vehicle, merging the function with the Dymola

[1] New European Driving Cycle

[2] Federal Test Procedure 1975

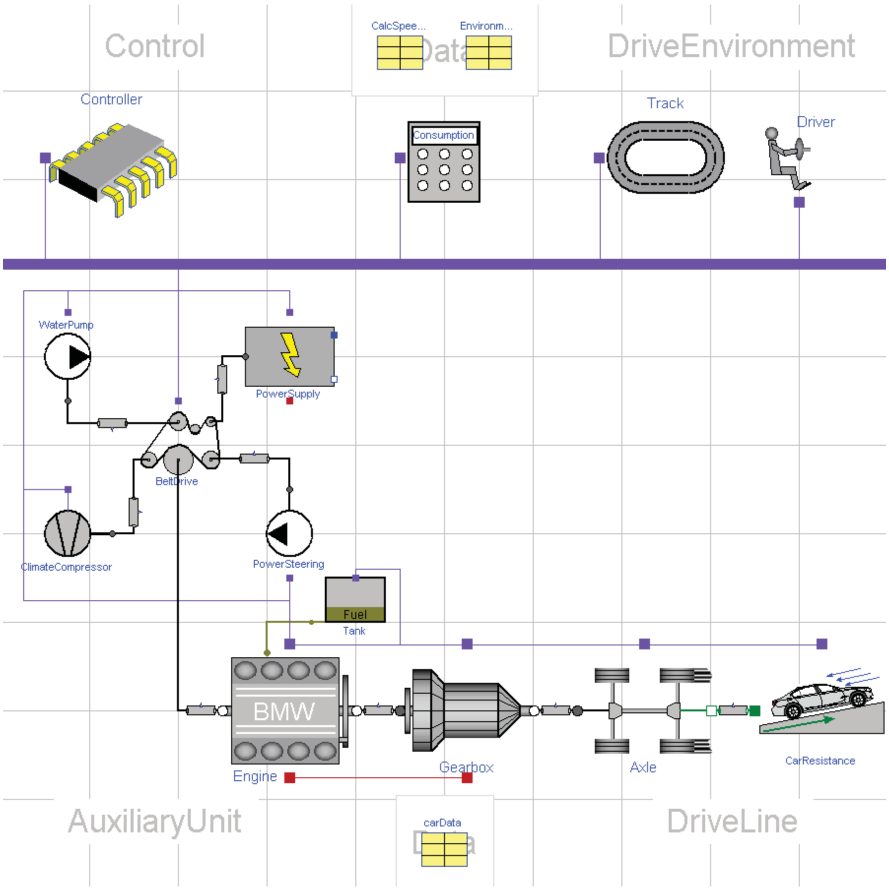


Figure 6.1 BMW Conventional Vehicle Dymola Model

model did not seem like time well spent. Instead, current functionality in the EG-61 Dymola vehicle model was used to run simulations for consumption assessment.

Among the various options for the Drive Environment of a vehicle model is included the possibility of simulating a custom speed profile, defined over either time or distance, referred to as a *custom track*. This allows for a two-step procedure for simulating driving with the Deceleration Optimizer over a given stretch of distance. First, speed limit, slope and distance vectors must be obtained for the trip in question. Thereafter, driving can be simulated in the Deceleration Optimizer Simulink model described in Chapter 5. A speed profile over distance, as well as DR and EB signals is then obtained. Using this output custom tracks can be generated for use in the Dymola vehicle simulation model, with DR and EB sequences being performed by the virtual driver.

6.3 Driving Environment

On important consideration for simulating fuel consumption is what driving environment to chose. Obviously, what driving environment to choose depends on the function to be tested. At the same time, when looking at potential customer fuel savings it is important to try not to idealize the driving conditions, since this will lead to misleading results that will compare poorly to everyday driving. For the Deceleration Optimizer, highway driving with varying speed limits and low traffic density is the optimal driving situation. In city traffic with crossings and traffic lights, it will in its present form be of very little use since the driver will have to intervene in a lot of situations. To get results that would represent a commuter whose daily driving is composed of both highway and city driving, it is thus necessary to simulate a mix of the two and avoid intervention from the Deceleration Optimizer in situations where it is likely not usable in reality, e.g. city driving. Starting at the BMW FIZ[3], there is a circuit of 43 km running through the outskirts of Munich in a mixture of highway, freeway and city driving. This circuit is internally referred to as the *Münchener Nordrunde*[4]. It is commonly used by BMW for evaluation of prototype functions and driving experiments, both because of its balanced mixture of driving conditions and its location close to FIZ. A printout from Google Maps displaying the Münchener Nordrunde is shown in Figure 6.2.

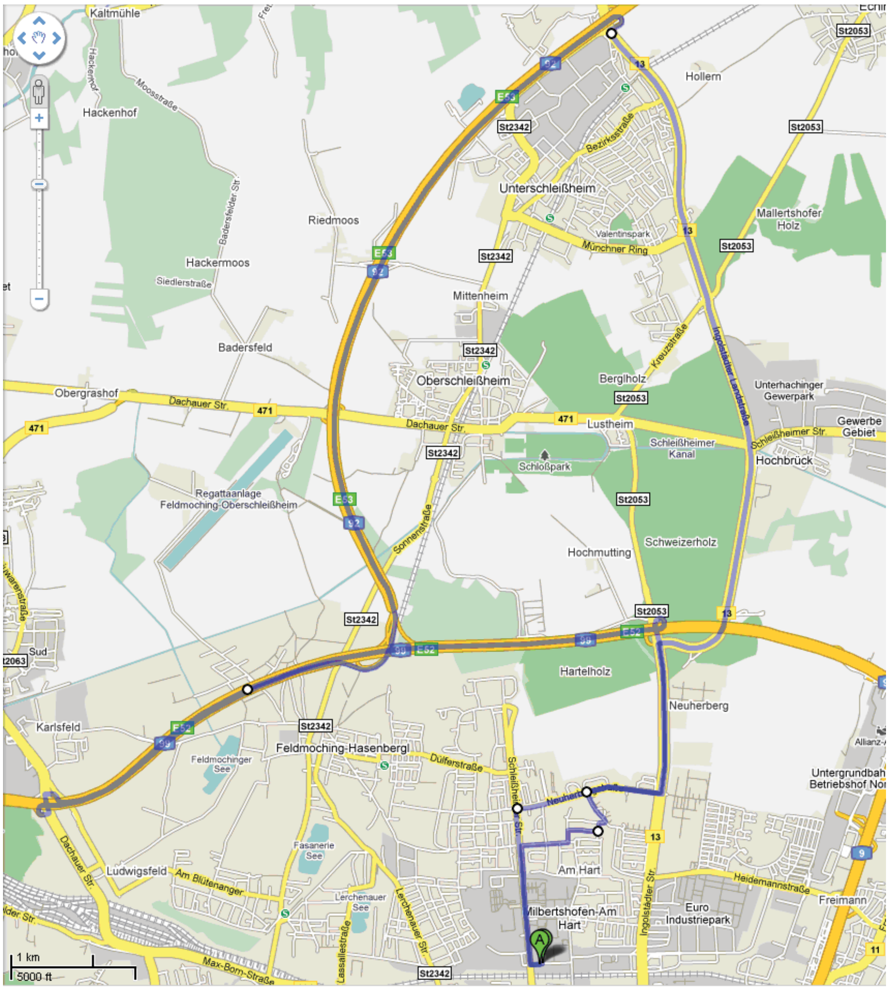


Figure 6.2 Münchener Nordrunde

City Driving and Traffic Lights

To make simulation results represent what one could expect from using the Deceleration Optimizer in real driving, care needs to be taken to make the simulated driving patterns similar to real ones. As already stated, the Deceleration Optimizer cannot be used in cities to any meaningful extent. In the simulations, it was therefore disabled

[3] BMW Forschungs und Innovationszentrum

[4] Munich's North Circuit

for any initial speed under 49 km/h and any target speed 31 km/h. Another question is how to handle traffic lights and city traffic turns. In real driving, the driver will sometimes have to stop in these situations, and sometimes it will be possible to keep on going without slowing down. Stopping increases fuel consumption and traveling time quite dramatically. To obtain a similar fuel and time consumption for the city driving part of the simulation, the simulation speed limit was set to 10 km/h at traffic lights and turns, forcing the simulated vehicle to slow down through braking, thus increasing fuel consumption and travel time.

6.4 Simulation Results

Using a speed limit profile over distance as input for the navigation system emulation described in 5.2, this circuit was simulated in the Deceleration Optimizer Simulink model. Using the output from the vehicle model and the Deceleration Optimizer custom tracks were generated. Automating this process through a Matlab script allowed for production of several custom tracks with varying values of $c_{t_{base}}$, yielding proportionally the same variation in the final c_t values for each speed transfer. The generated custom tracks where subsequently simulated through the Dymola vehicle model. The different values for c_t resulted in different lengths of the resulting deceleration patterns for each deceleration. A small c_t means time is depreciated and will lead to long and fuel conserving deceleration profiles, while a big c_t means short decelerations and low fuel savings. For a complete evaluation of the performance of a certain c_t setting, the fuel consumption simulations must be coupled with driving experiments that provide the input of what it actually feels like to drive the car.

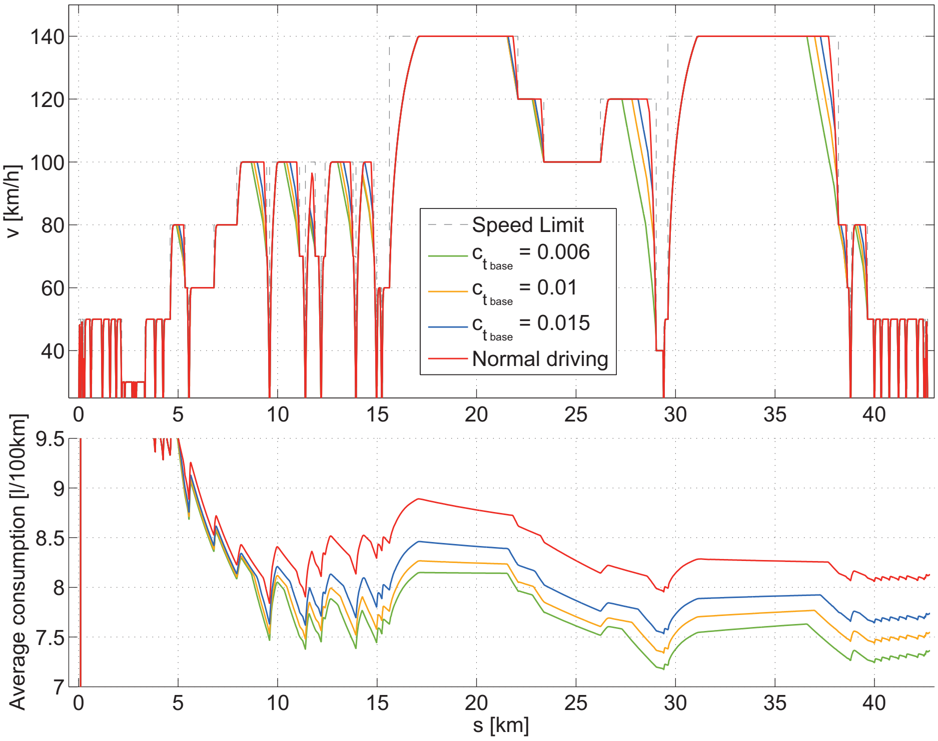


Figure 6.3 Simulated driving profiles on the Münchener Nordrunde

Figure 6.3 shows the simulated speed and the average consumption over the entire Münchener Nordrunde. Readily identifiable are the city, highway and freeway sections of the circuit. Noteworthy is also that for each decrease in c_t, a substantial increase in fuel savings can be obtained. The results concerning fuel consumption and time loss are displayed in Table 6.1. Interesting is the relation between fuel savings and time loss for the three driving profiles with predictive deceleration. Accepting marginal time losses enables quite large fuel savings.

Base time weight $c_{t_{base}}$	Consumption	Savings	Trip time	Time loss
Normal driving	3.48 l	0.0 %	35.3 min	0.0 %
15×10^{-3}	3.31 l	4.80 %	35.5 min	0.6 %
10×10^{-3}	3.23 l	7.15 %	35.8 min	1.4 %
6×10^{-3}	3.15 l	9.40 %	36.1 min	2.5 %

Table 6.1 Simulation results

Base Case Braking Curve

The base case is a 'Normal Driving' driving pattern where deceleration is made according to the braking curve with no predictive deceleration. This driving pattern was produced using the Deceleration Optimizer with $c_{t_{base}} = 1$. This is sufficiently high to yield deceleration profiles with no predictive deceleration.

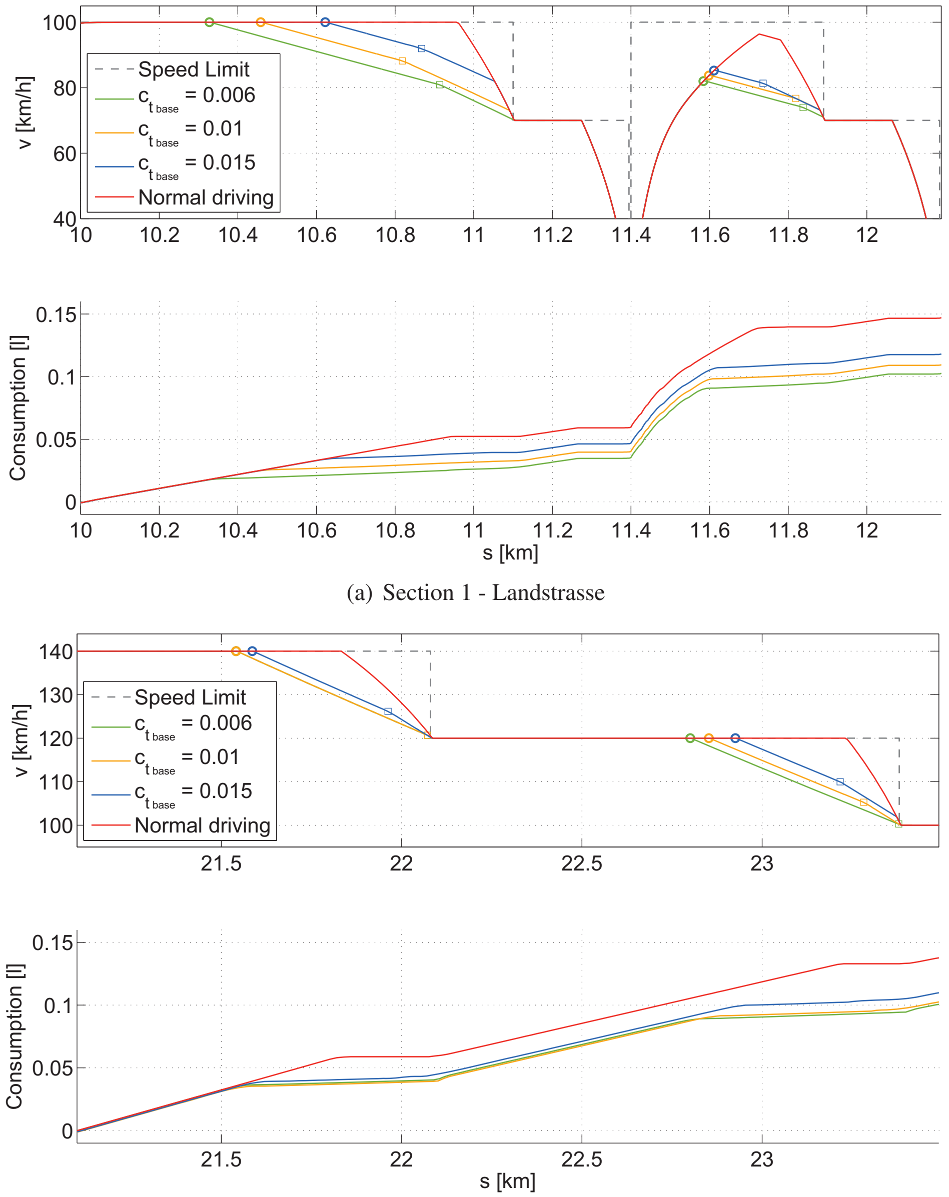
In order for the base case to represent normal driving, it is crucial that the braking curve corresponds to a regular driving behaviour. If a braking curve featuring very hard braking would be used, then fuel consumption for the base case would be higher than reasonable, and in comparison the predictive deceleration driving patterns would look better than they really are in terms of fuel consumption, and worse than they really are in terms of travel time. Therefore, a braking curve obtained from recorded driving experiments on the Münchener Nordrunde has been used. The braking curve was produced as an average of the deceleration rates at different speeds from multiple recorded drives. The same braking curve was used as the time optimal deceleration pattern for the Deceleration Optimizer.

Selected Road Sections

Figure 6.4 displays two selected parts of the circuit, one Landstrasse section and one Autobahn[5] section. The speed over distance is displayed as well as the consumption over the selected section. The switching points from constant speed driving to DR mode have been marked with circles and the switching points between DR and EB mode have been marked out with squares.

Figure 6.4(a) displays a Landstrasse section where the constant speed driving at 100 km/h is interrupted by two traffic lights with foregoing 70 km/h speed limits. In Figure 6.4(a) it is apparent how the three settings of c_t yield different lengths of the deceleration pattern. For $c_{t_{base}} = 6 \times 10^{-3}$, braking is almost entirely avoided through DR and EB. The DR distance is quite important, as opposed to the driving pattern for $c_{t_{base}} = 10 \times 10^{-3}$, where almost the same terminal velocity is obtained with substantially more EB. For $c_{t_{base}} = 15 \times 10^{-3}$, the deceleration is quite constrained and a lot of braking is being performed. For the second part with 100 km/h there is not enough time to accelerate to the speed limit before deceleration must be initiated. The long and the medium deceleration profiles perform mainly DR, while the short profile used more EB and commences deceleration later. In the fuel consumption plot, the difference in fuel consumption can be clearly seen in the two deceleration phases.

[5] Highway and Freeway



(a) Section 1 - Landstrasse

(b) Section 2 - Autobahn

Figure 6.4 Closeups on two sections from the Münchener Nordrunde

In Figure 6.4(b), a driving sequence from a Freeway section is displayed. For the first deceleration, all the driving profiles transfers to the correct speed with no braking. What differs is the relation between DR and EB. Both the long and the medium deceleration profile is made by pure DR. Since the profit of using DR instead of EB increases with high speeds[6], this makes sense. The second speed transfer involves a little braking for the short deceleration profile but otherwise displays the same pattern.

This similarity in behaviour for the three different values of c_t are due to the high speeds. For speeds as in Figure 6.4(b), the fuel savings grow relatively large compared to the deviance from the normal driving speed and early deceleration is therefore always profitable. When comparing figures 6.4(a) and 6.4(b), the difference

[6] See Section 3.3

in fuel consumption between the three different driving patterns is larger for Figure 6.4(a). This shows that for the short deceleration driving pattern, when c_t is made quite large, fuel is mainly saved for higher speeds, where the potential for fuel savings is larger.

Comparison of Different Deceleration Strategies

Another perspective of the different Deceleration Optimizer driving profiles is given in Figure 6.5. Here, the consumption results in terms of percental improvement over the normal driving pattern as a function of driving time are displayed for the three different deceleration strategies discussed above. Also displayed is the normal driving profile and results for maximum DR and maximum EB strategies. Maximum DR and maximum EB driving have been simulated under the same constraints as the Deceleration Optimizer strategies, that is with no predictive deceleration for city driving sections. In coherence with the assumptions of this thesis project, maximum DR strategy seems to be the fuel optimal end point of the Deceleration Optimizer, at least for decelerations down to speeds around 50 km/h which has been simulated for the results above. Maximum EB strategy, which is sometimes proposed as the most efficient mean of deceleration, is clearly inferior to the optimized deceleration strategies.

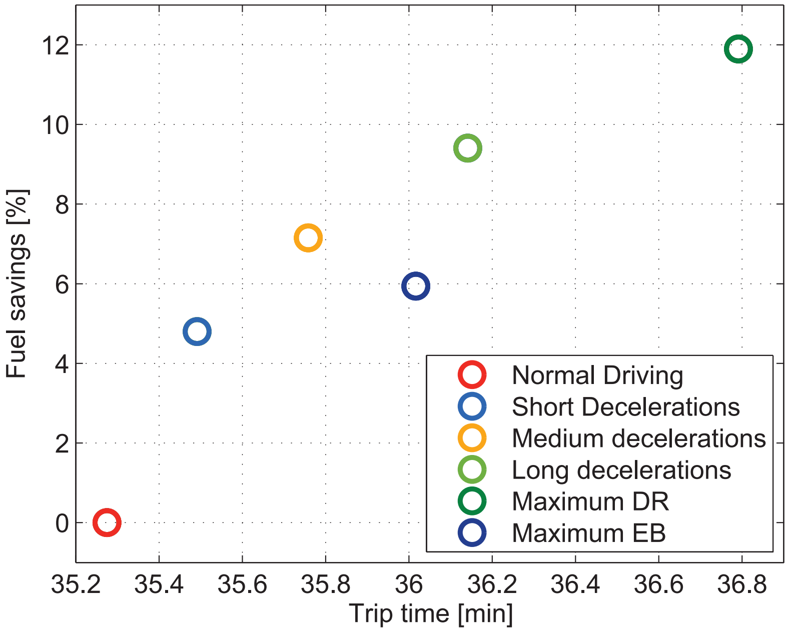


Figure 6.5 Different driving strategies for the Deceleration Optimizer compared with maximum DR and maximum EB strategies

7. Vehicle Implementation and Experimental Results

The prototype function was developed using Embedded Matlab and Simulink. The implementation of the Deceleration Optimizer and the auxiliary Simulink model that was used for the vehicle testing has been covered in Chapter 4 and Section 5.1. To make the optimizer work in a test vehicle, some additional systems are required. Most notably an interface to the navigation system and a speed regulator capable of performing the DR, EB and RB commands. First a short introduction of the hardware system used in vehicles for prototype function testing will be given.

7.1 Hardware

Autobox

After merging the Reconstructor, the Deceleration Optimizer and the Speed Regulator into one Simulink model the entire prototype GreenACC model was compiled using Real-Time Workshop into code runnable in the BMW experimental vehicles. The target system where the prototype GreenACC should run is a real-time controller board mounted in an Autobox, made by the firm dSpace[1]. The Autobox provides additional real-time computing power which can be used for rapid prototyping and for running prototype functions without compromising the vehicles regular control system. The Autobox has access to the CAN-bus and can override the standard control units at various levels in the control structure.

CAN-bus

The CAN-bus is a serial bus that is widely used for communication between the various control units in a modern vehicle. CAN is a standardized protocol originally developed by Bosch and was designed specifically for use in automotive applications. In the BMW 5-series, several CAN-buses with different communication speeds are used, depending on the real-time demands of the connected control units. Obviously the injection control unit has higher demands on speed than the air conditioning. In this project the hardware implementation issues have not been in focus and therefore reference has been and will be made to the CAN-bus as a unit. This should be considered as a logical construction, representing the entire communication system of the vehicle without taking care of the actual layout of the hardware.

7.2 Navigation System

For the GreenACC a navigational system called ADAS-RP[2] is being used. ADAS-RP is being developed by a maps and navigation company called Navteq[3]. It estimates the current vehicle position, predicts future driving paths and proposes suitable speed trajectories for these paths. The maps used in ADAS-RP contain information about,

[1] http://www.dspaceinc.com/ww/en/inc/home/products/hw/accessories/autobox.cfm

[2] Advanced Driver Assistance Systems Research Platform

[3] www.navteq.com

among others, street type, legal speed limits, curves, crossings and slopes. For the Deceleration Optimizer, speed limits, legal as well as those imposed by curves and other events, and slope information are the most important inputs.

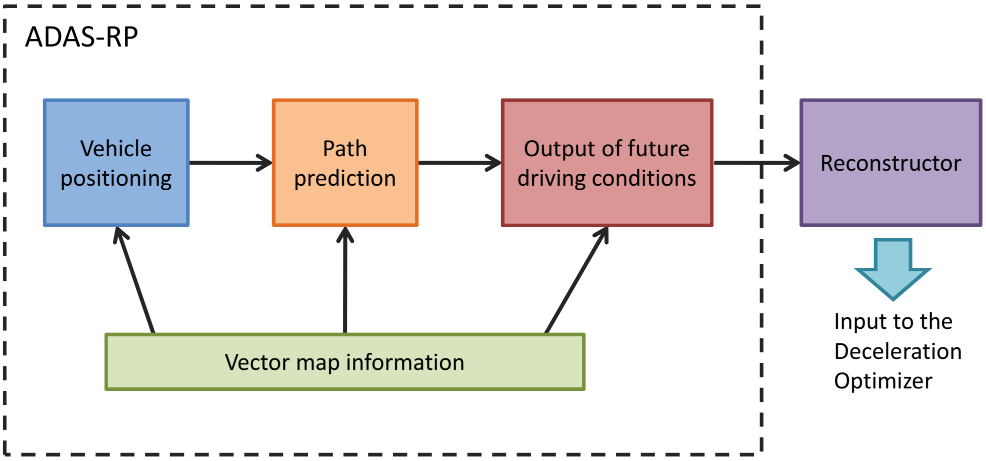


Figure 7.1 Conceptual layout of the Navigation System

Based on information from a GPS receiver and optionally other positioning systems such as inertial positioning devices, the vehicle's position is being calculated. In the present case the need for accuracy is not extraordinary high, and a regular GPS receiver has been used as positioning source. Thereafter a predicted future driving path is calculated. In the most straightforward case, this path is indirectly provided by the driver by entering the destination in the navigation system before starting the trip. ADAS-RP then calculates a driving path to the target and assumes the driver will follow this path. If no destination has been entered by the driver, the path is estimated based on earlier driving sequences in the same area or if no driving history is available based on the layout and importance of the roads ahead. Vectors containing information about distance, advised speed and event type for the different events along the predicted path are then being assembled.

Before being propagated to the Deceleration Optimizer, this information is being processed in a reconstructor that transforms it into a suitable format for the optimization calculations, as described in 4.1. The reconstructor also reads the current vehicle speed and other required inputs from the CAN-bus and transmits them to the Deceleration Optimizer Shell.

7.3 Speed Regulator

To effectuate a deceleration profile dictated by the Deceleration Optimizer a slightly more competent speed controller than the conventional one is required. Apart from following a set reference speed, it must be able to switch to and from neutral gear and to brake when need be. The ACC available in series production of the BMW 5-series already possesses the ability of braking to maintain a safe distance to a vehicle in front. This braking path can be manipulated to be used for the Deceleration Optimizer.

Automatically switching from drive to neutral mode on an automatic gearbox requires either a way of mechanically manipulate the mode selection lever or a mechanically decoupled signal path from lever to gearbox. As discussed in Section 2.1, the mode switching lever on the automatic gearbox version of the 5-series is mechanically decoupled from the gearbox and communicates via the CAN-bus. In the rapid-prototyping vehicles such as the BMW 535i used in the present study, a gateway has been placed between the mode selection lever and the rest of the CAN-bus. The gateway is a processor unit that enables rerouting and emulation of CAN-messages from and to the lever, and so allows for a prototype speed regulator to intercept this communication path and send commands to the gearbox independent of the mode selection lever.

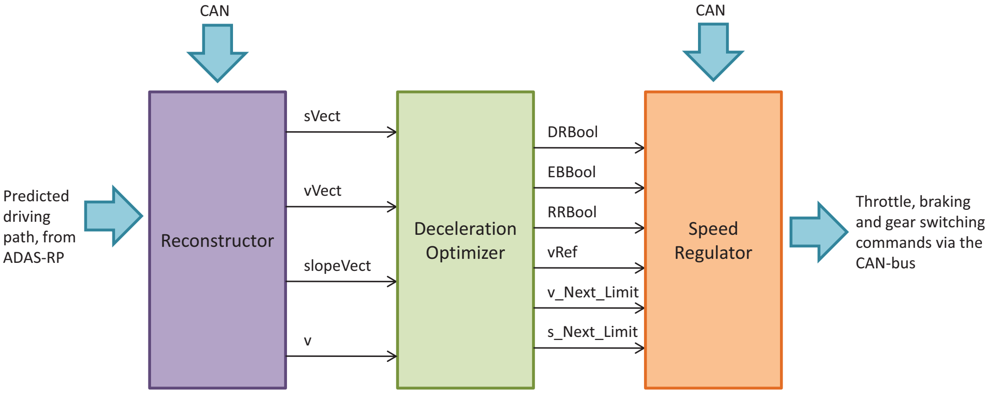


Figure 7.2 Interfaces to the Deceleration Optimizer

With the purpose of enabling experimentation with the GreenACC a speed regulator that incorporates the required functions described above was developed concurrently with the Deceleration Optimizer by BMW Group Research and Technology. This speed regulator builds upon the series production ACC but additionally accepts signals for DR, EB and RB modes. While in DR and EB mode the speed regulator is passive. In RB mode braking is performed but due to limitations in the regular ACC only at a low level. Full tracking of v_{ref} is therefore currently not achieved. During acceleration acceleration is performed according to the parametrization of the series ACC. Naturally, for the GreenACC as a whole it is important to carefully choose efficient rates of acceleration. For testing of the Deceleration Optimizer it is, however, no important matter.

The layout and interfaces between the Reconstructor, Speed Regulator and the Deceleration Optimizer is shown in Figure 7.2

7.4 Experimental Results

Testing of Non-Automated Deceleration Optimizer

The first in-vehicle testing of the Deceleration Optimizer was carried out while driving manually, but with the Deceleration Optimizer coupled with the reconstructor running in the Autobox. The outputs of the Deceleration Optimizer were observed and DRBool and EBBool were adhered to as well as possible by manual mode switching using the control lever of the automatic gearbox. Although manual switching was not very accurate, the experiment showed that the Deceleration Optimizer was running as expected, triggered at upcoming decreases in speed limit, calculated deceleration patterns and reacted reasonably to the errors induced by the delays in the manual switching. Figures 7.3, 7.4 and 7.5 show three recordings of deceleration sequences.

Since manually braking according to a braking curve was judged as very hard to accomplish, the Deceleration Optimizer was tested using a low value for c_t, yielding long decelerations with almost no RB.

Figure 7.3 displays a deceleration sequence where the vehicle decelerates more than the predicted trajectory shown in v_{ref}. The vehicle speed is shown as the solid line in the first subplot, while v_{ref} is dashed. In the second subplot, DRBool is shown as the dash-dotted line and EBBool as the dotted. In the three figures, it can be clearly seen how v_{ref} adjusts itself to the current vehicle speed when DRBool goes high. This is the intended behaviour. During constant speed driving v_{ref} indicates the current speed limit, and during deceleration the predicted vehicle speed. However, when the optimization is triggered, the vehicle speed at the time of the trigger is used for v_0, not v_{ref}. In Figure 7.3, the lower than predicted vehicle speed causes DRBool to stay high, to eliminate the error between the vehicle speed and v_{ref}, as discussed in Chapter 5. At the end of the deceleration, when v_{ref} has switched to EB mode, the vehicle catches up with v_{ref}.

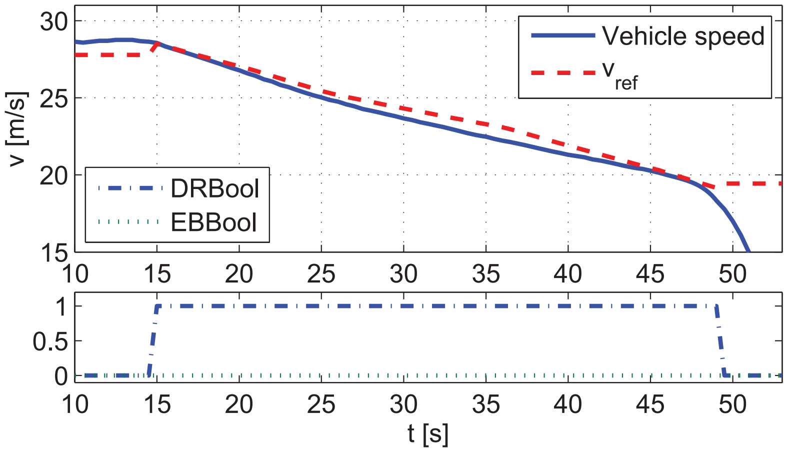


Figure 7.3 DR dominated deceleration

The next deceleration sequence, displayed in Figure 7.4, shows a deceleration where the driver accidently keeps accelerating after DRBool has gone high. This makes the vehicle speed quite a bit higher than v_{ref}, and therefore the Deceleration Optimizer switches into EB mode after only a short while in DR mode.

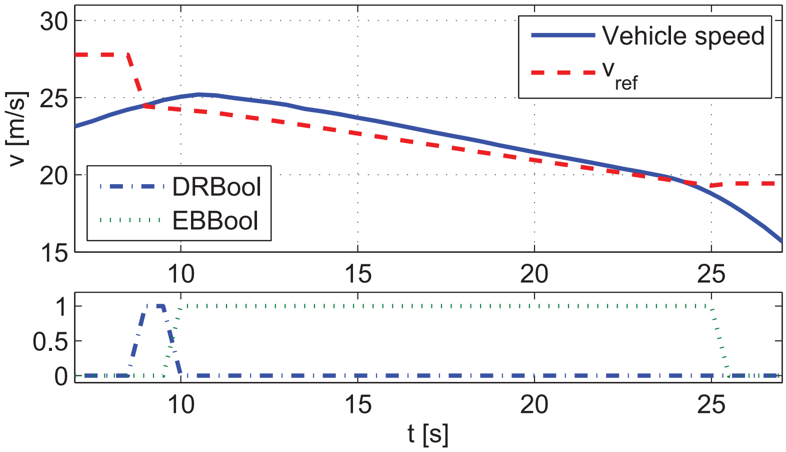


Figure 7.4 EB dominated deceleration

Finally, Figure 7.5 displays a deceleration where the manual switching was quite successful in following the predicted deceleration pattern. A combination of DR and EB is being performed.

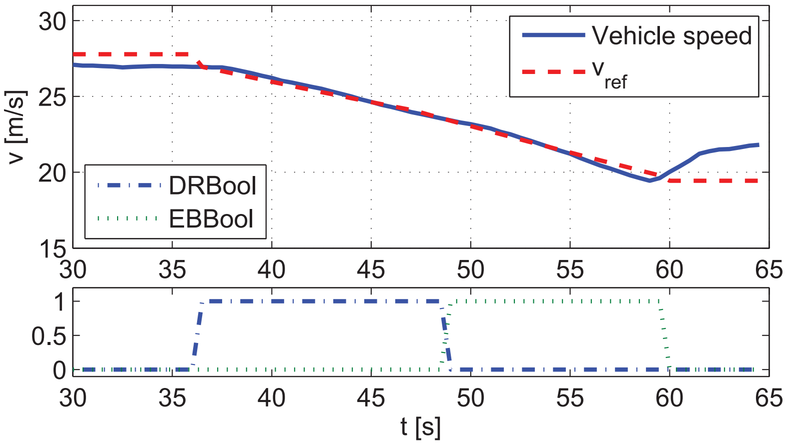


Figure 7.5 Mixed deceleration

Prototype GreenACC

At the end of this thesis project, time was running out faster than expected and the merging of the Deceleration Optimizer and reconstructor with the speed regulator was delayed until the last week. Unfortunately, the combined function was initially not prepared for parameter recording, and therefore no measurements of the driving tests with the automated prototype GreenACC could be made.

However, what could be made were subjective driving experiments to get a feeling for what driving with the GreenACC feels like. The three c_t settings from Chapter 6 were tested, and all of them were found drivable. Since this initial testing was restricted to a small group of people, one should be careful with definite conclusions drawn from this test, but it was in any case clear that the c_t setting with short decelerations caused very little discernible deviation from desired driving speed and should be acceptable for most people.

8. Fuel Efficient Driving on the Nürburgring

Along with the increased interest for fuel efficient driving, racing competitions aiming at efficient driving have appeared alongside with regular competitions. Such an event is the RCN Green Challenge[1] that is organized on the Nordschleife[2] of the Nürburgring racing track in south western Germany. During 2009 three races were held and another three races are planned for 2010. Within the departments at BMW occupied with fuel efficiency and predictive energy management there is interest in participating in the RCN Green Challenge, in order to promote the work being done in the field. This provides an interesting opportunity to try the Deceleration Optimizer as a tool to find fuel efficient driving patterns for the Nürburgring.

8.1 The Challenge

Competition Rules

The RCN Green Challenge has a rule set that is made up to allow different cars to compete under equal conditions. Based on its weight-to-power[3] ratio, each car is assigned a base lap time t_{base}. The power in question is the nominal engine power specified by the manufacturer. The Nürburgring should then be traversed five times with decreasing lap times t_{lap} based on t_{base}. Each lap has to be completed with a difference of no more than $\pm 5\,\mathrm{s}$ from t_{lap}. Finally the track is traversed one sixth time with only an upper bound on the lap time, well above the five fixed lap times. t_{base} and weight-to-power ratios for some selected cars of the BMW Group can be viewed in table 8.1.

When the six laps have been completed, the accumulated consumption is measured and points are distributed based on the vehicle's performance compared with its NEDC[4] consumption. This means that in terms of point distribution it is not necessarily advantageous to choose a fuel efficient vehicle. Rather, one that is poorly optimized for the NEDC would be a good option. For the sake of good publicity it might, however, be better to use a car that has an environmentally appealing profile. For a relatively powerful car like the BMW 123d, t_{base} corresponds to an average speed over the track of about 92 km/h. Considering the quantity of sharp curves, of which a number requires speeds under 60 km/h, achieving an average fuel consumption at or under the nominal NEDC consumption constitutes a considerable challenge.

The Nürburgring

The Nordschleife of the Nürburgring is 20.8 km long and features numerous curves and hills. The maximum grades are 17 % uphill and 11 % downhill[5]. The height difference over the track is about 300 m. To use the Deceleration Optimizer to compute optimal driving profiles for the Nürburgring, the required information is the speed

[1] http://www.greenchallenge.de/

[2] Northern loop

[3] (kg/kW)

[4] New European Driving Cycle

[5] http://www.nuerburgring.de/ft/nordschleife/streckendaten-nordschleife.html

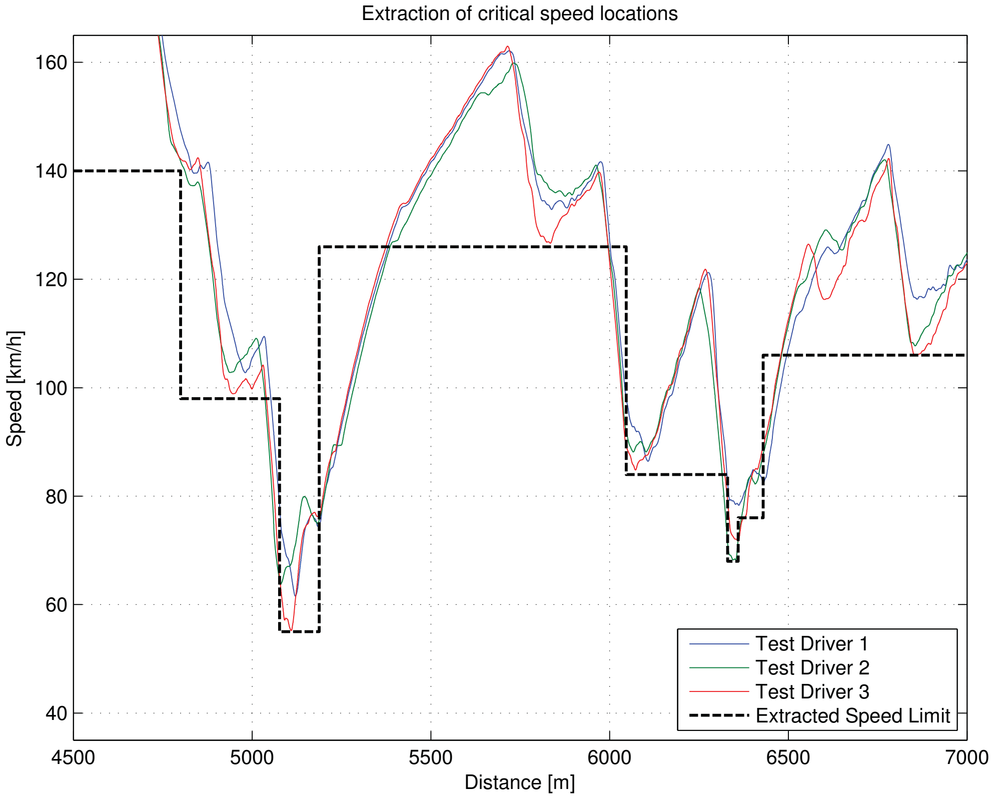


Figure 8.1 Competitive driving on the Nordschleife

constraints along the track. These constraints are due to sharp curves which must be traversed at relatively low speeds. To obtain a speed constraints profile for the Nordschleife, recorded driving profiles from drivers racing for time on the Nürburgring were studied.

The speed over distance over a selected stretch of the Nordschleife for three different runs can be viewed in Figure 8.1. Studying these measurements the points requiring limited speeds could be obtained and a speed limit profile was constructed. The speed limit profile for the selected stretch of the Nordschleife is also displayed in Figure 8.1.

Vehicle Selection

Choosing an appropriate vehicle for the competition is apparently an important part of competitive participation. The left part of Table 8.1 indicates that it is advantageous to choose a vehicle which has a weight-to-power ratio at the lower end of an interval. The right part of Table 8.1 contains weight-to-power ratios and t_{base} for seven BMW diesel models. The weight used for calculation was the weight of the empty vehicle with 150 kg added for fuel and driver. It was decided that the ambition should be to compete with a vehicle with low total consumption. Therefore only diesel models, which generally have lower consumption, were studied. Among the vehicles listed in 8.1, the 123d Coupé stands out as a viable option. It is right at the lower end of a weight-to-power interval and is a small car with relatively low consumption. The following optimization has therefore been carried out for the BMW 123d. It was decided that the 123dA, which is the model with automatic gearbox, should be initially studied. This is because the 123d with manual gearbox features a function for automatically shutting off the engine when the vehicle stands still and for automatic starting once the driver pushes the gas pedal. This generates significantly lower NEDC consumption, but gives no advantage during driving. Since the Green Challenge will require continuous driving, the automatic gearbox model should be used, which does not have automatic engine start/stop functionality, and therefore a higher NEDC consumption which can be more easily achieved at the Nürburgring.

(kg/kW)	t_{base} (s)
<9	760
>9<11	795
>11<13	810
>13<15	825
>15<18	840
>18<21	860
$\geq$21	885

Car	(kg/kW)	t_{base} (s)
120d Coupé	12.3	810
123d Coupé	11.0	810
325d Coupé	12.0	810
330d Coupé	9.7	795
335d Coupé	8.6	760
520d Touring	14.0	825
525d Touring	13.0	825

Table 8.1 Lap times for different weight to power categories and a listing of base lap times and weight-to-power ratios for some BMW models.

8.2 Modifications to the Deceleration Optimizer

To find a fuel efficient driving pattern for the Nürburgring, optimizing the deceleration phases for each speed transfer is clearly not enough. Since each lap needs to be completed in a predefined time, one first question is what time weight coefficient c_t that should be used. Then, this parameter needs to be matched with an appropriate acceleration strategy, also adopted to minimizing fuel consumption over time. Finding a good combination of reasonable acceleration and fuel efficient deceleration, while accomplishing the correct t_{lap} for each lap, is the key to finding a competitive driving pattern.

Modified Time Cost Function

When running the Nürburgring on a given time with the aim of lowest possible fuel consumption, a time cost function reflecting deviance from surrounding traffic is ill-suited for the optimization task. What is interesting in this case is the actual time loss compared to a driving profile with constant speed followed by hard braking. Therefore a time cost function expressing the pure time loss of a deceleration profile has been constructed. Rephrasing Equation (3.9) the following expression for Φ_t is obtained.

$$\Phi_t = \int_0^{d_{max_{DR}}} \left(\frac{1}{v(d)} - \frac{1}{v_{t\_opt}(d)} \right) \mathrm{d}d \qquad (8.1)$$

Along with Φ_t, a strategy for choosing c_t is needed. In this case it is, however, desired to compare fuel consumption directly to travel time. Therefore constant values of c_t will be appropriate, so that fuel savings will be weighted against time losses equally for all speed transitions. Also notice that due to the difference in time cost function, the appropriate range of values for c_t will be different than for the time cost defined in Section 3.3.

Acceleration Strategy

The question of optimal acceleration given a time constraint is a very involved question. For the generation of driving patterns for the Nürburgring, a simple yet versatile acceleration model was implemented. An acceleration curve for flat ground was defined as a function of vehicle speed. It was composed of three part:

- Constant acceleration a_1 between $v = 0$ and $v = v_1$
- Constant power P_{const} for speeds higher than v_2
- Transition phase between v_1 and v_2

Constant Acceleration The constant acceleration phase was taken as a result from an earlier investigation at EG-61 concerning optimal acceleration. In this investigation, vehicles were simulated running a stretch of 1 km with varying rates of acceleration. This study showed that up to about 50 km/h, a constant acceleration between 0.5 m/s^2 and 1.5 m/s^2 would yield fuel consumptions within 0.5 % of the minimal consumption over the distance. The optimal acceleration then decreased down to zero at about 75 km/h, from where further acceleration would increase fuel consumption over the distance traveled. However, because of the time constraint posed by t_{lap} the acceleration needs to be higher than the optimum one. Since vehicle speed will be smaller than 75 km/h only for a few times during the race, no further effort was put into optimizing the acceleration mapping for low speeds, and loosely based on the investigation mentioned above a_1 was set to 2 m/s^2 and the end of the constant acceleration interval v_1 was put to 50 km/h

Constant Power Since the Nürburgring has to be traversed in a given time, with an average speed of more than 90 km/h, acceleration above the fuel optimal point of operation is required. To achieve an element of optimization in determining this acceleration, an acceleration rate based on constant engine output power P for speeds over $v_2 = 75$ km/h was constructed.

The acceleration of a vehicle is given in Section 2.4 and will here be written as

$$a = \frac{F_{motor}}{m} + A_{DR} + Bv_2^2. \tag{8.2}$$

The difference here from the presentation in Section 2.4 is that F_{motor} is usually positive and that for positive accelerations the sum of F_{motor} and F_{slope} is bigger than the sum of resistive components F_{air} and F_{drag}. Since $P = Fv$ we have that $F_{motor} = P/v$ while pushing the throttle. Inserted in Equation (8.2) this yields

$$a = \frac{P}{mv} + A_{DR} + Bv_2^2. \tag{8.3}$$

Transition interval To connect constant acceleration part with the constant power part a linear transition was made connecting (v_1, a_1) and (a_2, v_2) where a_2 is given by Equation (8.3), yielding $a_2 = \frac{P}{mv_2} + A_{DR} + Bv_2^2$.

The resulting acceleration as a function of speed is the given by Equation (8.4)

$$\begin{cases} a = 2 & v \le v_1 \\ a = k_a v + m_a & v_1 < v \le v_2 \\ a = \dfrac{P}{mv} + A_{DR} + Bv^2 & v > v_2 \end{cases} \tag{8.4}$$

where

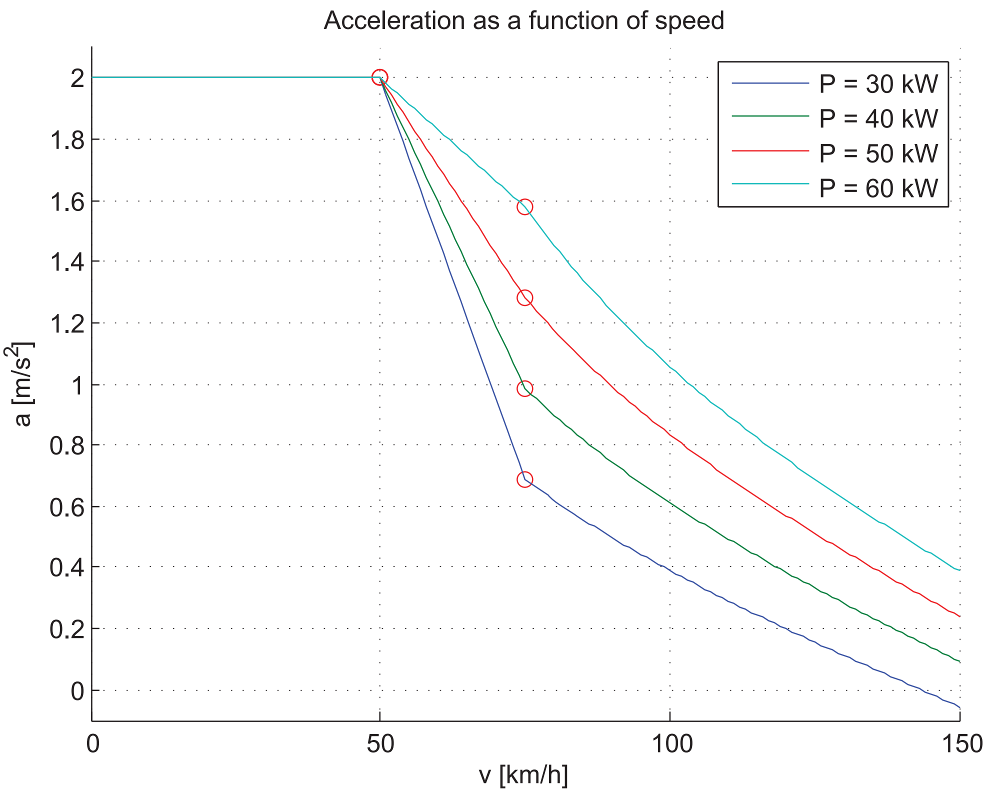


Figure 8.2 Acceleration as a function of speed

$$k_a = (a_2 - a_1)/(v_2 - v_1)$$

$$m_a = (a_1 - k_a v_1)$$

Acceleration as a function of speed according to (8.4) for various constant values of P is displayed in Figure 8.2, with $v_1 = 50\,\mathrm{km/h}$, $a_1 = 2\,\mathrm{m/s^2}$ and $v_2 = 75\,\mathrm{km/h}$.

8.3 Finding Fuel Efficient Driving Patterns

Finding an efficient driving pattern is a complex optimization problem on a larger scale than that of the Deceleration Optimizer, with the fuel consumption as the objective function and the trip time as a constraint on the solutions. The feasible set is composed of every possible driving pattern over the Nürburgring that meets the time constraint. The problem is that the number of decision variables required to describe every possible solution is virtually infinite. The discussion in Section 8.2 aims at finding decision variables that describe a subset of the feasible set, that still contain solutions that are close to the optimal solution of the original problem. The optimization procedure below was carried out for a BMW 123dA, using a $t_{lap} = 795\,\mathrm{s}$ which is the time for the third out of the five laps with progressively decreasing lap times.

Maximum speed

One additional parameter that will be used in the search for a fuel optimal driving pattern is a maximum driving speed v_{max}. The idea is that to minimize fuel consumption, the speed should be kept close to constant [Chang and Morlok, 2005]. Because of sharp curves along the track, the time constraints cannot be met by maintaining a constant speed. However, introducing a maximum speed v_{max} means that parts of the track where high speeds are possible will be traversed at constant speed. v_{max} will be used as an upper bound for the speed limit profile for the Nürburgring.

Modifications to Acceleration

The acceleration described as in Equation (8.4) does not take into account the slope of the road. In a steep grade an acceleration of $2\,\mathrm{m/s^2}$ might correspond to pushing the engine to its limit, and if the road is steep enough the prescribed acceleration will not be possible to obtain. Also, if the vehicle is traveling at v_{max} in a downhill section that is steep enough that braking is required to keep the speed at v_{max}, then it makes more sense to let the vehicle passively accelerate down the slope. The acceleration obtained from Equation (8.4) has therefore been adjusted with the acceleration due to gravity $a_{slope} = -g\sin\alpha$. For uphill sections the minimum acceleration has finally been set to 0. This means that in the case of steep uphill sections where the current P is not high enough to maintain constant speed, an increase in engine power will be allowed to avoid deceleration.

Search Method

Three decision variables, c_t, P and v_{max} have now been identified. Let $\bar{x} = [c_t, P, v_{max}]$. Feeding $\bar{x}$ to the Deceleration Optimizer Simulink model, described in Chapter 5, with the modified acceleration pattern mentioned above, a speed trajectory for the entire Nürburgring can be obtained. However, this trajectory is determined completely by $\bar{x}$ and the travel time is a function of $\bar{x}$ with no consideration to t_{lap}. To achieve the required t_{lap} it is thus necessary to perform some kind of search for vectors $\bar{x}$ that fulfills this time constraint. This can be done by fixing two of the components in $\bar{x}$ and using the third parameter as search parameter. In the following, c_t and P were fixed while v_{max} was used as search parameter for finding the correct t_{lap}. The search method employed was the bisection method. The maximum v_{max} allowed was $140\,\mathrm{km/h}$ and minimum v_{max} was taken as $95\,\mathrm{km/h}$. For some combinations of P and c_t, representing slow accelerations and long deceleration sequences, t_{lap} could not be accomplished at all for $v_{max} < 140\,\mathrm{km/h}$. The search was then aborted and the inappropriate combination of P and c_t was discarded. Solutions with v_{max} close to $140\,\mathrm{km/h}$ displays poor performance, so looking for solutions with even higher v_{max} is not interesting in the quest for a fuel efficient driving pattern. For the lower bound at $95\,\mathrm{km/h}$, the maximum speed is getting close to the required average speed to complete the track on time. Within the range of c_t and P tested, no solution obtained had a v_{max} under $100\,\mathrm{km/h}$. Therefore $95\,\mathrm{km/h}$ is a lower bound that poses no serious restrictions on the solutions. A flow diagram illustrating the steps in the search process in shown in Figure 8.3.

From the simulation results, custom tracks were generated for simulation in Dymola, from where the accumulated fuel consumption for each driving pattern was finally extracted. This brute force method could have been avoided for some more efficient method if Dymola simulations could be controlled from the Matlab environment. Unfortunately such an interface was not available and the initialization of the Dymola simulations had to be done manually. More information on the Dymola simulation environment can be found in Chapter 6.

Resulting Driving Patterns

In figures 8.4 and 8.5, three driving patterns over two different sections of the Nürburgring are displayed. The figures allow for observation of the influence of the decision variables P, c_t and v_{max} on the driving profiles. The first subplot of each figure contains plots of vehicle speed over distance for the three driving patterns. In the second

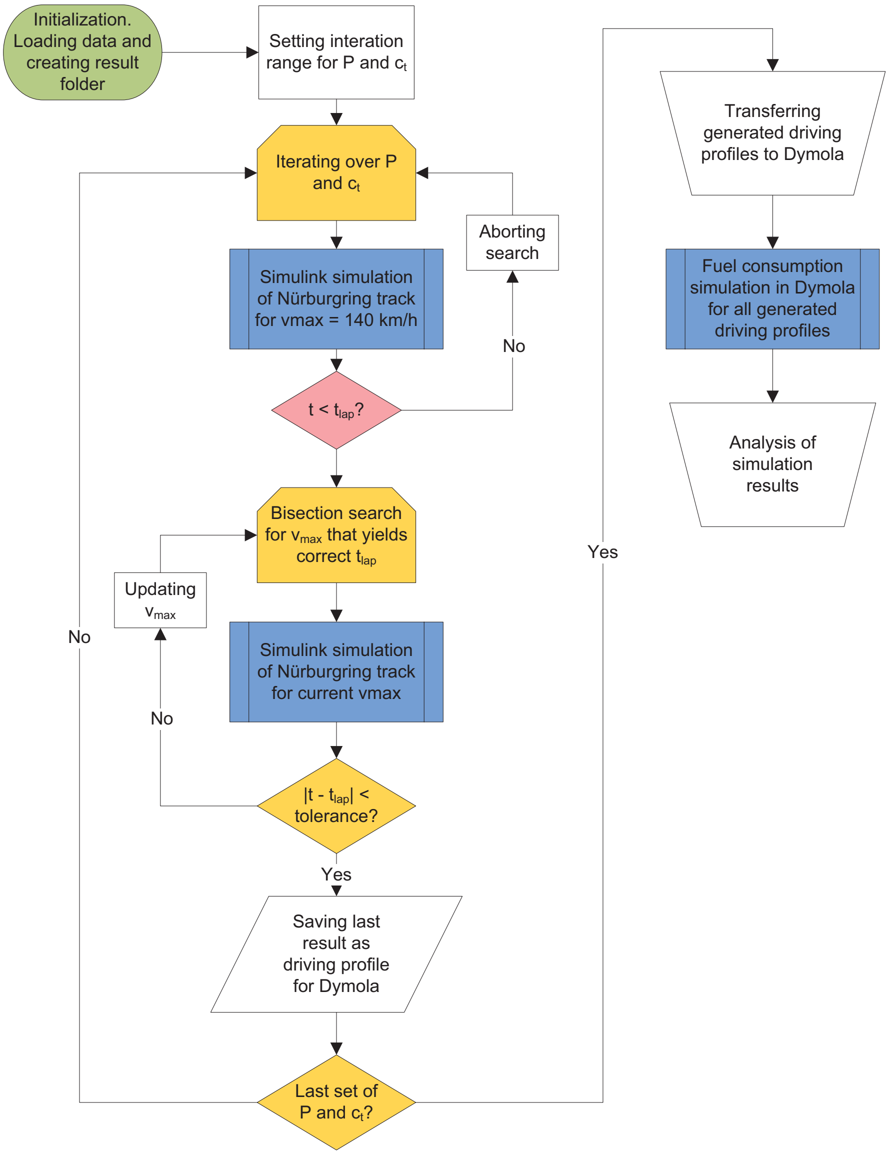

Initialization. Loading data and creating result folder
Setting interation range for P and c_t
Iterating over P and c_t
Simulink simulation of Nürburgring track for vmax = 140 km/h
t < t_{lap}?
No
Aborting search
Bisection search for v_{max} that yields correct t_{lap}
Updating v_{max}
Simulink simulation of Nürburgring track for current vmax
|t - t_{lap}| < tolerance?
No
Yes
Saving last result as driving profile for Dymola
Last set of P and c_t?
No
Yes
Transferring generated driving profiles to Dymola
Fuel consumption simulation in Dymola for all generated driving profiles
Analysis of simulation results

Figure 8.3 The optimization process for finding fuel optimal driving profiles for traversing the Nürburgring at a given time

subplot, the accumulated fuel consumption for the different driving patterns during this section are displayed. The third subplot displays the road height profile of the selected section with the height in meters relative to the starting point of the track. The performance in time between the three profiles can also be compared by looking at their relative speeds over the section. The switching points between different modes of operation are also marked in the first subplot of each figure. All three driving patterns complete the track in the same prescribed t_{lap}.

Example 1 All three driving patterns complete the track in $t_{lap} \pm 1\,\mathrm{s}$. Pattern number 2, which is drawn as a solid line, is the optimal driving pattern. The two others are extremes picked to clarify the influence of the different decision variables. During the acceleration phase at $s = 1.5\,\mathrm{km}$ the difference in P can be clearly seen. Pattern number 3 performs a hard acceleration but only up to about $100\,\mathrm{km/h}$ where it keeps accelerating only due to the steep downhill slope. Number 2 accelerates more calmly, then rolls down the slope. Number 1 accelerates even less but keeps accelerating up

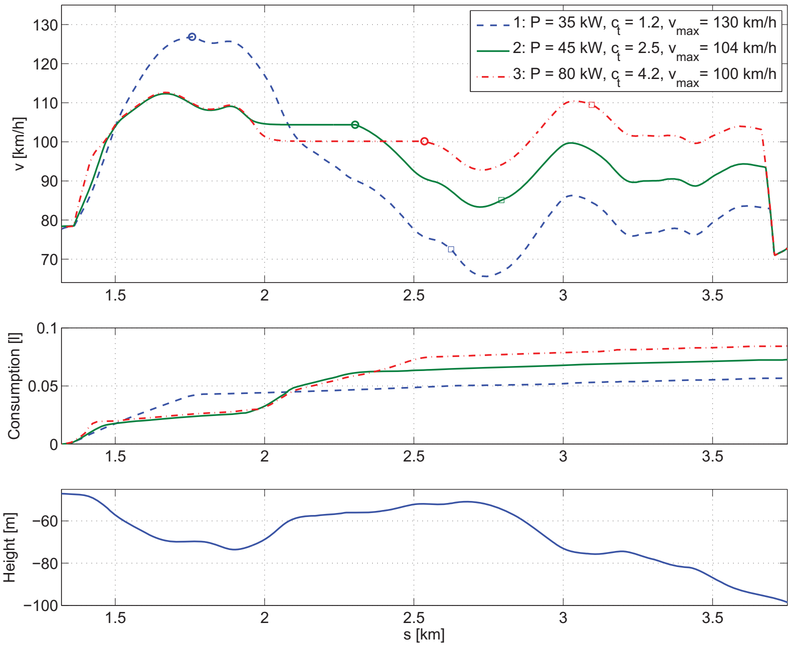


Figure 8.4 Example 1 - Influence of the decision variables on the resulting driving patterns. Switching from normal driving to DR is marked with circles and switching from DR to EB is marked by squares.

to the point where it switches into DR. The switch occurs early compared to 2 and 3, because of the low value of c_t for pattern number 1. In the uphill section at 2 km, number 2 and 3 decelerate to their v_{max} and runs at constant speed up the hill. Number 1, which is in DR mode, decelerates heavily. Between 2.5 km and 3 km, number 2 and 3 also switch into DR. In the second half of the section, all three driving patterns perform EB or DR mode, at different speeds. It should be noted that during this specific section, number 1 and number 2 both loose time compared to number 3. Since the total time is the same, this means there are other sections where number 3 travel slowlier than the other two. On the other hand, number 1 consumes less fuel than number 2 and 3.

Example 2 Figure 8.5 displays the same three driving pattern for a section near the end of the Nürburgring track. The section is a long stretch of relatively straight road. Here, the relation between the performance of the driving profiles is different from Example 1. The high maximum speed of pattern number 1 allows for compensation for the relative time loss earlier in the track. This leads to a higher fuel consumption that for number 2 and 3, although number 1 saves a lot of fuel towards the end of the section thanks to DR over a long distance. Number 2 also gains time relative to pattern number 3, thanks to it's higher maximum speed, but still consumes a little less fuel than number 3 by DR towards the end of the section. Number 3 saves a little fuel up until 19.5 km, compared to number 2, but keeps on driving at constant speed for a while longer and therefore ends up consuming more fuel than number 2 over the entire section.

For both sections, but most clearly in Figure 8.4, it can be seen that during the EB phases, although consumption during EB should be 0, the consumption increases slightly. That is because in the present state of the Dymola conventional vehicle models it is not possible to impose pure EB. That is, the simulated driver can not be made to completely refrain from braking or pushing the throttle during EB. Even for an

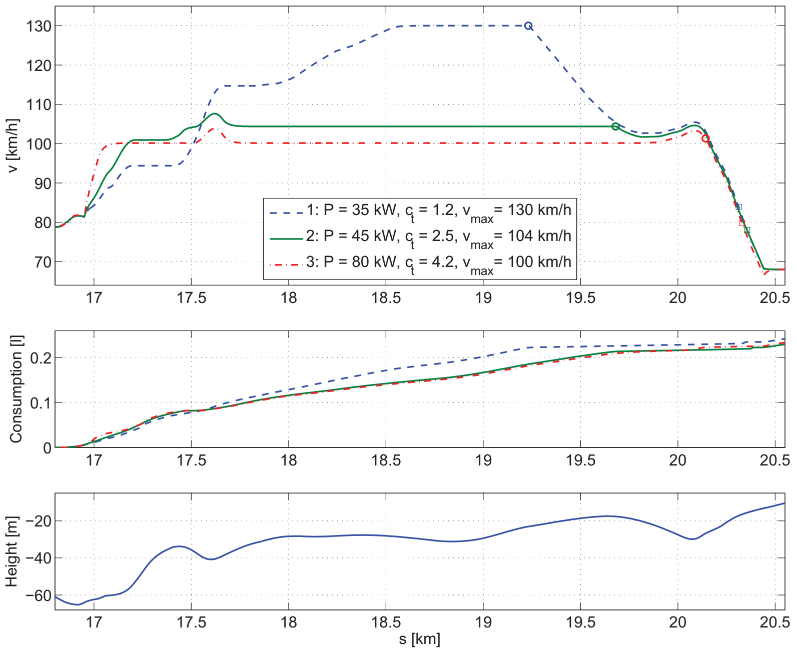


Figure 8.5 **Example 2** - Influence of the decision variables on the resulting driving patterns. Switching from normal driving to DR is marked with circles and switching from DR to EB is marked by squares.

ideal parametrization of the EB phase, the consumption is about the same as the idle consumption during DR. This is obviously a weakness in the simulation environment when studying the benefits of the GreenACC, and the ability to simulate pure EB deceleration will hopefully be implemented in the future.

Optimization results and Fuel Consumption

In figure 8.6, a contour plot of the average fuel consumption for the simulated Nürburgring driving profiles is shown. The contour plot has been drawn with c_t on the x-axis and P on the y-axis. The color bar displays the average consumption for the different contours and the optimal deceleration among those simulated is marked out with a red circle. For each pair (c_t, P) there is an associated value of v_{max} which yields the correct t_{lap}. In the lower left corner, with weak accelerations and long decelerations, are driving patterns with high values of v_{max}. The blank space represents combinations (c_t, P) for which driving times are still to long for $v_{max} = 140\,\text{km/h}$. For higher values of P and c_t the v_{max} required to finish the track in correct time is lower, with the lowest v_{max} associated with the solutions in the upper right corner.

The contour plot is quite rugged and irregular. The reason is that there is a lot of interaction between the slope and the curves of the track and the driving profile. To a certain extent these interactions are random, but when regarding the entire optimization area there is still a distinct basin featuring the most efficient driving patterns.

The consumption values in Figure 8.6 were obtained from simulation with the climate compressor on. During the Green Challenge the climate compressor will be turned off, and when simulating the best driving pattern without climate compressor the average fuel consumption is 5.55l/100km. The simulated NEDC consumption for the 123dA is 6.05l/100km, which means the driving profile obtained offers an improvement over the NEDC consumption of about 8 %. Considering that last year's

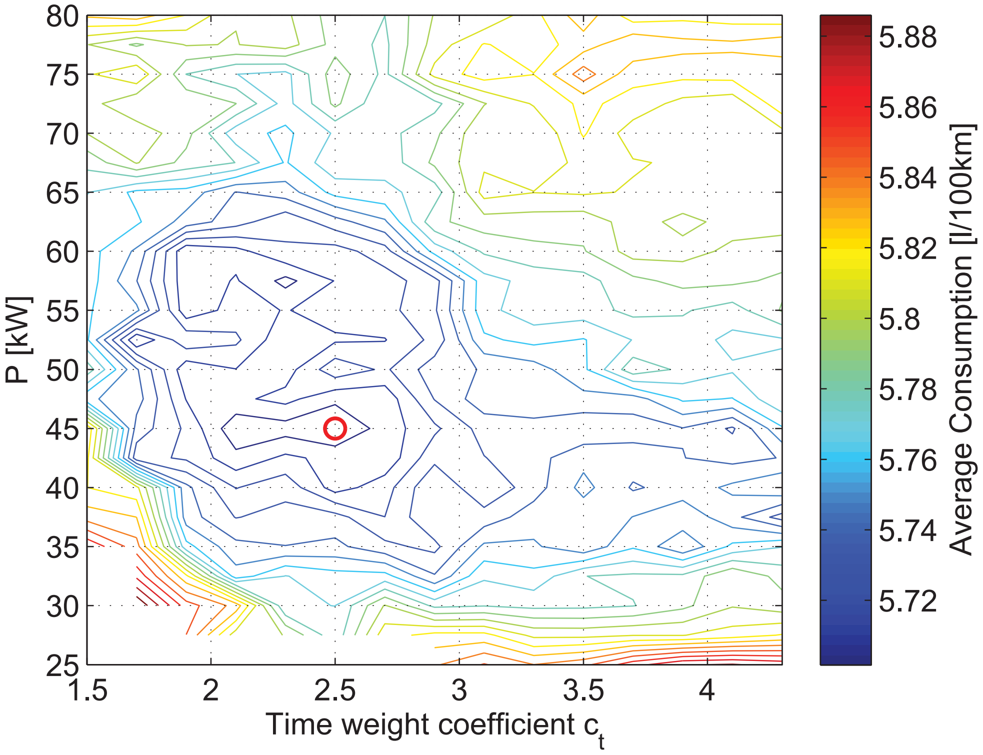


Figure 8.6 Contour plot of Nürburgring simulation results

winner[6], an Audi TT, completed the competition with an average consumption right at it's NEDC consumption[7] of 7.7 L/100km, this constitutes a competetive figure.

[6] http://www.oneighturbo.com/audi/audi-tt/audi-wins-2nd-heat-of-%E2%80%98rcn-green-challenge%E2%80%99-held-at-the-nurburgring/

[7] The improvement of 28 % mentioned in the article referes to the city part of the NEDC cycle

9. Conclusions and Future Work

In this master thesis project, a control strategy for automatic preemptive deceleration using navigation information has been elaborated, for use in an energy efficient cruise controller titled GreenACC. The control strategy involves optimization over the set of possible decelerations in order to find Pareto optimal solutions in terms of fuel consumption and time loss. The control strategy has thereafter been implemented in Embedded Matlab and Simulink. The fuel saving capacity of the resulting Deceleration Optimizer has been evaluated in simulation and preliminary driving experiments with the prototype function running in a test vehicle has been carried out. The initial driving tests look promising and the simulated fuel efficiency improvements indicate potential for substantial fuel savings with minor time losses. The main advantage of the Deceleration Optimizer is the ability to compute reasonable deceleration profiles for any speed transfer and slope configuration, and that the length of the decelerations can be adjusted online through varying the time weight coefficient c_t.

A lot of improvements and further development is possible. Of course, more thorough vehicle testing is necessary to ensure stable behaviour in the vehicle. To capture the subjective feeling of driving with the GreenACC, driving tests need to be carried through with drivers trying out different parameters settings.

On the technical side, one weakness of the optimization in its current shape is that only one speed transfer at a time is considered. When several decreases in speed limit follows close to each other, the second speed limit decrease will not be considered until the vehicle has passed by the first one. In this situation, it would be beneficial to consider the entire deceleration scenario and look for an optimal solution involving all of the adjacent speed decreases.

Another interesting question is how an implementation in series production should be made. The option of running the optimization algorithm in the series GreenACC would provide interesting possibilities of customization through adjustment of c_t. The driver could then choose a desired level of fuel consumption versus time loss. Automatic adjustment to c_t could also be made based on additional parameters such as road type and traffic density.

If the optimization algorithm is too computationally expensive to run online in the series cruise controller, then the Deceleration Optimizer could prove useful as a template for simplified control strategies aiming at the same objective. In short, for flat roads the optimized decelerations could easily be translated into a mapping of deceleration distances for different speed transfers. For hilly sections the interaction between aerodynamic drag and slope resistance makes things more complicated, making the Deceleration Optimizer a valuable tool for evaluation of simplified control strategies.

10. Bibliography

Anderson, N. and r. Björck (1973): “A new high order method of regula falsi type for computing a root of an equation.” *BIT Numerical Mathematics*, **13:3**, pp. 253–264.

Braess, H.-H. and U. Seiffert (2005): *Vieweg Handbuch Kraftfahrzeugtechnik, 4. Auflage*. Vieweg & Sohn Verlag/GWV Fachverlage GmbH, Wiesbaden.

Chang, D. J. and E. K. Morlok (2005): “Vehicle speed profiles to minimize work and fuel consumption.” *Journal of Transportation Engineering*, **131:3**, pp. 173 – 182.

Collete, Y. and P. Siarry (2003): *Multiobjective Optimization*. Springer-Verlag, Berlin, Heidelberg.

Ehrgott, M. (2005): *Multicriteria Optimization, Second Edition*. Springer-Verlag, Berlin,Heidelberg.

Georgieva, A. and I. Jordanov (2010): “A hybrid meta-heuristic for global optimisation using low-discrepancy sequences of points.” *Computers & Operations Research*, **37:3**, pp. 456–469.

Hellström, E., M. Ivarsson, and J. Åslund (2009): “Look-ahead control for heavy trucks to minimize trip time and fuel consumption.” *Control Engineering Practice*, **17:2**, pp. 245–255.

Kiefer, J. (1953): “Sequential minimax search for a maximum.” *Proceedings of the American Mathematical Society*, **4:3**, pp. 502–506.

Lechner, G. and H. Naunheimer (1994): *Fahrzeuggetriebe*. Springer-Verlag, Berlin, Heidelberg, New York.

Modelica Association (2000): “Modelica - A Unified Object-Oriented Language for Physical Systems Modeling, Tutorial.” `http://www.modelica.org/documents/ModelicaTutorial14.pdf`. *Accessed 2010.03.30*.

Nelder, J. A. and R. Mead (1965): “A Simplex Method for Function Minimization.” *The Computer Journal*, **7:4**, pp. 308–313.

Otter, M. and H. Elmqvist (2001): “Modelica, Language, Libraries, Tools, Workshop and EU-Project RealSim.” `http://www.modelica.org/documents/ModelicaOverview14.pdf`. *Accessed 2010.03.30*.

Schwarzkopf, A. B. and R. B. Leipnik (1977): “Control of highway vehicles for minimum fuel consumption over varying terrain.” *Transportation Research*, **11:4**, pp. 279 – 286.

Sli, E. and D. F. Mayers (2003): *An Introduction to Numerical Analysis*. Cambridge University Press, Cambridge.

Wallentowitz, H. (2001): *Längsdynamik von Kraftfahrzeugen, 5. Auflage*. Forschungsgesellschaft Kraftfahrwesen Aachen mbH, Aachen.

Wolf-Heinrich, H., Ed. (2005): *Aerodynamik des Automobils*. Vieweg & Sohn Verlag/GWV Fachverlage GmbH, Wiesbaden.

Woll, T. (2005): “Theorie der Fahrwiederstände.” In Wolf-Heinrich, Ed., *Aerodynamik des Automobils*, pp. 125–130. Vieweg & Sohn Verlag/GWV Fachverlage GmbH, Wiesbaden.

A. Nomenclature

A.1 Abbreviations and Expressions

Deceleration profile	A pair of decision variables $\overline{x} = [d_{tot}, d_{EB}]$ and the associated function of speed over distance from $-d_{DR_{max}}$ to 0.
Driving pattern	The speed trajectory over distance and the corresponding modes of operation for a vehicle driving along some section of road.
Average Consumption	Average fuel consumption for some driving sequence, measured in $\mathrm{l/km}$.
Fuel Consumption	Accumulated consumption over some time or distance, measured in volume.
Fuel Flow	Instantaneous fuel consumption, measured in $\mathrm{g/s}$
Target point	A target position and speed for a deceleration profile
ADAS-RP	Advanced Driving Assistance System Reasearch Platform
CAN	Controller Area Network, vehicle communications bus
GreenACC	Green Adaptive Cruise Controller
DR	Decoupled Rolling
EB	Engine Braking
RB	Regular Braking

A.2 Notations

Notation	Description
A	Parameter for the speed independent term in acceleration calculations
A_α	Added acceleration term for slope of angle α
$A_{EB_{flat}}$	A-parameter for EB with no slope
A_{EB}	A-parameter for EB, including slope
A_{DR}	A-parameter for DR, including slope
A_f	Frontal area
a	Acceleration
α	Slope angle
B	Parameter for v^2 dependent term in acceleration calculations
C_w	Drag coefficient
C_{punish}	Constant multiplying the factor proportional to the error in Φ_{punish}
c_t	Time cost weight coefficient
$c_{t_{base}}$	Constant factor for calculation of c_t
d_{EB}	Engine braking distance
$d_{EB_{max}}$	Maximum EB deceleration distance possible
$d_{EB_{opt}}$	Optimal engine braking distance
d_{DR}	Distance in decoupled rolling mode, given as $d_{tot} - d_{EB}$
$d_{DR_{max}}$	Maximum DR deceleration distance possible
$d_{DR_{min}}$	Minimum distance in DR mode, given by v_0 and $t_{DR_{min}}$
$d_{intersect}$	Distance from target point to intersection $v(s)$ with $v_{brake}(s)$
d_{NL}	Distance to next speed limit
d_{tot}	Total deceleration distance
$d_{tot_{max}}$	Maximum available distance for deceleration
$d_{tot_{min}}$	Minimum deceleration distance, imposed by $v_{brake}(s, v_{target})$
$d_{tot_{opt}}$	Optimal total deceleration distance
F_{drag}	Aerodynamic resistance force
F_{motor}	Friction torque resistance force
F_N	Normal force
F_{roll}	Rolling resistance force
F_{slope}	Slope resistance force
$\bar{f}$	Objective functions
g	Gravitational constant
$\bar{g}$	Equality constraints
$\bar{h}$	Inequality constraints

m	Total vehicle mass
m_{unl}	Unloaded vehicle mass
μ_r	Rolling resistance coefficient
N_i	Gear ratio for i:'th gear
N_{rag}	Rear axle gear ration
ω	Engine speed
ω_{idle}	Idle engine speed
P	Engine power
$P_{(v_0,\alpha)}$	Required engine power for driving at v_0 at angle α
p	Ambient air pressure
Φ_{DR}	DR fuel consumption for a given deceleration
Φ_f	Fuel consumption cost function
$\Phi_{f_{const}}$	Fuel consumption for constant driving segment of a deceleration
Φ_{punish}	Punishment term to cost function to exclude undesired solutions
Φ_t	Time cost function
Φ_{tot}	Total cost function
r_w	Wheel radius
ρ	Air density
ρ_p	Slope fuel consumption ratio
s	Distance
σ_{idle}	Engine idle fuel flow
σ_{v_0}	Constant speed driving fuel flow for flat terrain
$\sigma_{(v_0,\alpha)}$	Constant speed driving fuel flow for angle α
T	Air temperature
t	Time
t_{DR}	Time in decoupled rolling mode for a given deceleration profile
$t_{DR_{min}}$	Minimum time allowed in DR mode
τ_f	Friction torque
v	Speed
v_0	Initial speed for a deceleration
v_{brake}	Braking curve
v_{target}	Deceleration target speed
v_{t_opt}	Time optimal deceleration profile
$\hat{v}$	Unit vector
$\bar{x}$	Decision variables
ζ	Slope fuel consumption correction factor