

Explicit use of road topography for model predictive cruise control in heavy trucks

Master's thesis
performed in **Vehicular Systems**

by
Erik Hellström

Reg nr: LiTH-ISY-EX--05/3660--SE

21st February 2005

Explicit use of road topography for model predictive cruise control in heavy trucks

Master's thesis

performed in **Vehicular Systems,**
Dept. of Electrical Engineering
at **Linköpings universitet**

by **Erik Hellström**

Reg nr: LiTH-ISY-EX--05/3660--SE

Supervisor: **PhD Student Anders Fröberg**
Linköpings universitet

Examiner: **Professor Lars Nielsen**
Linköpings universitet

Linköping, 21st February 2005



Avdelning, Institution
Division, Department

Vehicular Systems,
Dept. of Electrical Engineering
581 83 Linköping

Datum
Date

21st February 2005

Språk

Language

Svenska/Swedish

Engelska/English

Rapporttyp

Report category

Licentiatavhandling

Examensarbete

C-uppsats

D-uppsats

Övrig rapport

ISBN

—

ISRN

LiTH-ISY-EX--05/3660--SE

Serietitel och serienummer

Title of series, numbering

ISSN

—

URL för elektronisk version

<http://www.vehicular.isy.liu.se>

<http://www.ep.liu.se/exjobb/isy/05/3660/>

Titel Explicit användning av vägtopografi för modellprediktiv farthållningsfunktion i tunga fordon

Title Explicit use of road topography for model predictive cruise control in heavy trucks

Författare Erik Hellström
Author

Sammanfattning

Abstract

New and exciting possibilities in vehicle control are revealed by the consideration of topography through the combination GPS and three dimensional road maps. This thesis explores how information about future road slopes can be utilized in a heavy truck with the aim at reducing the fuel consumption over a route without increasing the total travel time.

A model predictive control (MPC) scheme is used to control the longitudinal behavior of the vehicle, which entails determining accelerator and brake levels and also which gear to engage. The optimization is accomplished through discrete dynamic programming. A cost function is used to define the optimization criterion. Through the function parameters the user is enabled to decide how fuel use, negative deviations from the reference velocity, velocity changes, gear shifts and brake use are weighed.

Computer simulations with a load of 40 metric tons shows that the fuel consumption can be reduced with 2.5% with a negligible change in travel time, going from Linköping to Jönköping and back. The road slopes are calculated by differentiation of authentic altitude measurements along this route. The complexity of the algorithm when achieving these results allows the simulations to run two to four times faster than real time on a standard PC, depending on the desired update frequency of the control signals.

Nyckelord

Keywords

topography,MPC,dynamic programming

Abstract

New and exciting possibilities in vehicle control are revealed by the consideration of topography through the combination GPS and three dimensional road maps. This thesis explores how information about future road slopes can be utilized in a heavy truck with the aim at reducing the fuel consumption over a route without increasing the total travel time.

A model predictive control (MPC) scheme is used to control the longitudinal behavior of the vehicle, which entails determining accelerator and brake levels and also which gear to engage. The optimization is accomplished through discrete dynamic programming. A cost function is used to define the optimization criterion. Through the function parameters the user is enabled to decide how fuel use, negative deviations from the reference velocity, velocity changes, gear shifts and brake use are weighed.

Computer simulations with a load of 40 metric tons shows that the fuel consumption can be reduced with 2.5% with a negligible change in travel time, going from Linköping to Jönköping and back. The road slopes are calculated by differentiation of authentic altitude measurements along this route. The complexity of the algorithm when achieving these results allows the simulations to run two to four times faster than real time on a standard PC, depending on the desired update frequency of the control signals.

Keywords: topography,MPC,dynamic programming

Preface

This thesis completes my studies for a Master of Science degree in Applied Physics and Electrical Engineering. The thesis work has been an interesting and rewarding effort that I have enjoyed much.

The use of modern technology, control theory and mathematics to enable more efficient utilization of energy resources is a worthy and thrilling motive. My thesis explores a possibility of using GPS, three dimensional road maps and on line computers with the aim at reducing the fuel consumption in a heavy truck without increasing the total travel time.

Outline

The purpose and method of the thesis are presented in the introductory chapter. In the following chapter a vehicle model is derived that is used for evaluating control strategies but is also the basis for the model used in the control algorithm. The obtained model can be characterized as a hybrid system.

The third chapter contains a review of optimal control methods for hybrid systems. The review leads to conclusions about how to approach the problem at hand. Chapter four shortly presents the mathematical foundation of dynamic programming.

The control algorithm is introduced in chapter five. A model predictive control (MPC) scheme is used where the optimization is carried out through discrete dynamic programming. In chapter six simulation results are reported. Results from both artificial and authentic road maps are discussed and illustrated. Artificial road sections are used to illustrate specific behavior of the controller whereas authentic sections primarily are used to show the magnitude of effects on fuel consumption and travel time.

In the seventh chapter conclusions are drawn from the achieved results. The final chapter indicates future work and extensions to the thesis.

Acknowledgment

I express my gratitude to my dedicated supervisor Anders Fröberg. My examiner Professor Lars Nielsen and the staff at Vehicular Systems are acknowledged for allowing me to become a part of the division. I am grateful for the fruitful and inspiring discussions we have had. Magnus Pettersson at Scania is recognized for sketching the problem which is the basis for this thesis.

A further acknowledgment is due to my grandfather John who passed away during my thesis work. Your humbleness and indomitable courage will continue to inspire me and you will always be a part of me. I would finally like to express my love and appreciation for my family, friends and for Gabriella.

Erik Hellström
Linköping, February 2005

Contents

| | |
|--|-----------|
| Abstract | v |
| Preface and Acknowledgment | vi |
| 1 Introduction | 1 |
| 1.1 Purpose | 1 |
| 1.2 Method | 2 |
| 2 Modeling | 4 |
| 2.1 Engine | 4 |
| 2.1.1 Approximations | 6 |
| 2.2 Driveline | 7 |
| 2.3 Longitudinal forces | 9 |
| 2.4 Complete driveline model | 9 |
| 2.5 Cruise control | 10 |
| 2.6 Fuel consumption | 12 |
| 2.7 Model implementation | 12 |
| 3 Control of a hybrid system | 13 |
| 3.1 Modeling | 13 |
| 3.2 Optimal control | 14 |
| 3.2.1 Variational methods | 14 |
| 3.2.2 Dynamic programming | 14 |
| 3.2.3 Model Predictive Control | 15 |
| 3.3 Conclusions | 16 |
| 4 Dynamic Programming | 18 |
| 4.1 System | 18 |
| 4.2 Algorithm | 19 |
| 5 Control algorithm | 22 |
| 5.1 Objective and constraints | 23 |
| 5.2 Problem representation | 23 |
| 5.3 State augmentation | 24 |

| | | |
|----------|---|-----------|
| 5.4 | Determining the search space | 25 |
| 5.4.1 | Gear selection | 25 |
| 5.4.2 | Considered velocities | 26 |
| 5.4.3 | Pedal and brake selection | 27 |
| 5.5 | Cost function | 28 |
| 5.6 | Algorithm | 29 |
| 5.7 | Complexity | 31 |
| 6 | Simulations | 33 |
| 6.1 | Parameters | 34 |
| 6.2 | Constant slope | 35 |
| 6.3 | Crest and depression | 39 |
| 6.4 | Authentic road sections in detail | 45 |
| 6.5 | From Linköping to Jönköping and back | 48 |
| 6.6 | Neutral gear | 48 |
| 6.7 | Complexity | 49 |
| 7 | Conclusions | 51 |
| 8 | Future work and extensions | 53 |
| | References | 54 |
| A | Simulink model | 56 |
| B | Calculations | 59 |
| B.1 | Vehicle model | 59 |
| B.2 | Pedal and brake level | 60 |
| B.3 | Numerical stability of Euler's method | 61 |
| B.3.1 | Linearization | 62 |
| B.3.2 | Test problem | 62 |
| B.3.3 | Conclusions | 63 |
| B.4 | Accuracy in Euler's method | 63 |
| B.4.1 | Conclusions | 64 |

Chapter 1

Introduction

Use of GPS in a vehicle combined with three dimensional road maps can give information about the road ahead. These data present new possibilities in vehicle control. For example, with information on future road slopes, the future longitudinal load can be predicted. The predictions can then be used in a control system that optimizes the control signals with the aim at reducing the fuel consumption. This thesis explores this concept.

A typical scenario is where the vehicle is kept at a constant reference velocity by a cruise controller. If the velocity is allowed to vary along the road, potential for saving fuel is revealed. The velocity could for example be lowered when there is a downhill slope ahead in which the vehicle will accelerate to a velocity above the set point. This would save fuel and possibly also brake use. If the road slope is sufficient, using the neutral gear will allow the vehicle to accelerate faster and by that increase the mean velocity over a route. If a considerable uphill is ahead, it may be advantageous to accelerate before the hill is reached. This increases the mean velocity and may also reduce the time needed running on a lower gear. All in all, there is a potential of saving fuel and at the same time not affect the travel time of the same amount.

1.1 Purpose

The goal of this thesis is to assess possible fuel savings with strategies for adaptive cruise control that make use of topographic information. Vehicle models are obtained and used in deriving control strategies and to evaluate their impact on vehicle dynamics and fuel consumption. A simulation environment is built to facilitate the assessment of control strategies.

1.2 Method

A longitudinal vehicle model is first derived. This has two fields of application. First, to be able to evaluate vehicle dynamics and fuel consumption, the model is implemented in MatLab/Simulink. Control algorithms are implemented in C++ or MatLab m-scripts and run as so called s-functions in Simulink. Second, the model with simplifications is used in algorithms when predicting the vehicle dynamics and the fuel consumption that would be the effect of a specific set of control signals.

A survey of relevant literature is made. Optimal control of the class of systems that the vehicle model belongs to are reviewed in order to decide how to approach the problem. A control algorithm is then developed and evaluated.

Evaluation of control algorithms is made with both artificial and authentic road maps. Short artificial and selected parts of authentic roads are used to study the controller behavior in detail. Longer authentic roads are used to assess the magnitude of effects on the fuel consumption.

The road slope α is defined as

$$\alpha = \frac{\Delta h}{\Delta L} = \frac{h_2 - h_1}{p_2 - p_1},$$

see figure 1.1. Note that positive road slope, $\alpha > 0$, means an uphill, $h_2 > h_1$, and conversely.

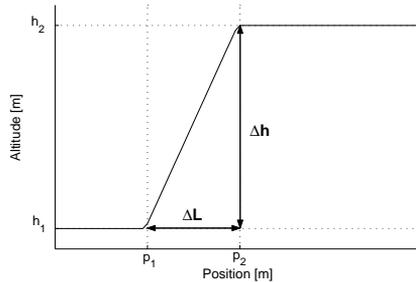


Figure 1.1: Artificial road section with one hill.

By varying the slope and length of the hill in figure 1.1, a set of simple sections is created.

In order to receive a closer to reality road section than the hill with constant slope, the slope is let to vary linear with position. This creates a depression if the slope increases from a negative value. A crest is received if the slope decreases from a positive value. An example of a depression is shown in figure 1.2. The length L is 500m and the slope increases from -2% to 2%. The length L and the start and end points of the slope are varied to create different sections.

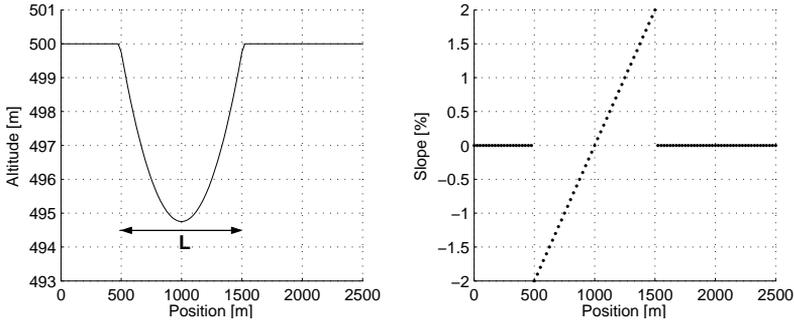


Figure 1.2: Artificial road section with one depression.

In order to receive a good assessment of the fuel consumption with a control strategy, a longer road section than the ones mentioned above is needed. This section should also in its configuration be close to a road where the vehicle typically travels. In this thesis, measurements on the road between the Swedish cities of Linköping and Jönköping have been used for this purpose. In figure 1.3 the altitudes above sea level along this road are shown.

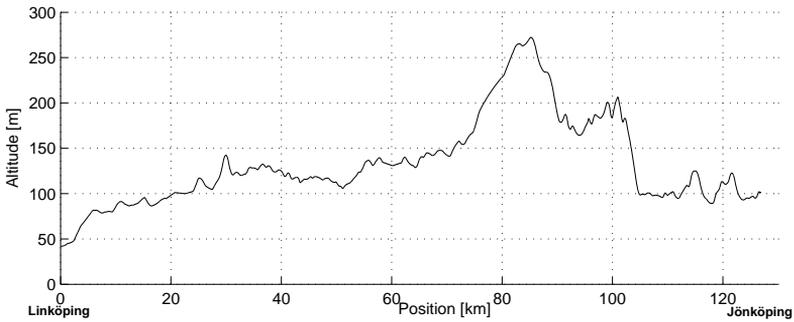


Figure 1.3: The road between Linköping and Jönköping

Chapter 2

Modeling

A model for the longitudinal dynamics of the vehicle is obtained. The torque generation in the engine is modeled with a set of simple functions. Driveline models are derived with elementary mechanics. Further, the longitudinal forces acting on the vehicle are modeled. These steps result in a complete basic driveline model.

Driveline modeling is well covered in Nielsen and Kiencke (2000) and Nielsen and Eriksson (2003) and are the foundation of this chapter.

2.1 Engine

The process in an internal combustion engine produces power and emissions from fuel and air. In the application at hand, only the generated power and consumed fuel are of interest. With this aim, a rectangular engine map with torque and fueling bounds is used to model the process. This map is made up of steady state measurements. This means among other things that the internal friction from the engine is included in the map. In figure 2.1 the

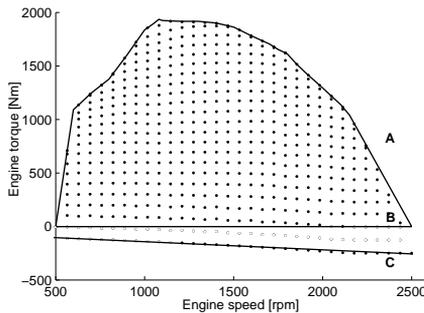


Figure 2.1: An engine map

used engine map and bounds are shown. Each dot represents one steady state measurement with constant engine speed N and constant engine fueling δ . The map can be expressed as $T_{map}(N, \delta)$. Curve A is the upper torque bound $T_{max}(N)$. Curve B shows idle operation. Finally, curve C is valid during engine overrun, which is when the engine is pulled by the vehicle. This torque is denoted as $T_{drag}(N)$, the drag torque. An upper limit that restricts the possible fueling is finally introduced and denoted $\delta_{max}(N)$.

For control purposes a pedal P and a gear signal G are introduced. The pedal signal is seen as normalized fueling and is therefore restricted to values between zero and one. The gear signal is a number in the set of available gears H that maps to gear parameters which are introduced later in section 2.2. For now it is sufficient to notice that neutral gear is gear number zero.

With use of the presented control signals the fueling function δ is constructed as

$$\delta(N, P, G) = \begin{cases} P\delta_{max}(N) & G \neq 0 \\ \delta_{idle} & G = 0 \end{cases} \quad (2.1)$$

where δ_{idle} is the idle fueling. It is assumed that the fueling becomes the idle fueling immediately when the gear signal becomes zero. The engine torque T_e can now be expressed as

$$T_e(N, P, G) = \begin{cases} T_m(N, P) & P > 0, G \neq 0 \\ T_{drag}(N) & P \leq 0, G \neq 0 \\ 0 & G = 0 \end{cases} \quad (2.2)$$

$$\text{where } T_m(N, P) = \min \{T_{map}(N, P\delta_{max}(N)), T_{max}(N)\}$$

The process is now described by a set of functions. The five basic functions are

| | |
|----------------------|------------------------|
| $T_{map}(N, \delta)$ | Rectangular engine map |
| $T_{max}(N)$ | Upper torque bound |
| $T_{drag}(N)$ | Drag torque |
| $\delta_{max}(N)$ | Upper fueling bound |

and by introducing the control signals

| | |
|----------------|----------------------------------|
| $P \in [0, 1]$ | Pedal signal, normalized fueling |
| $G \in H$ | Gear number |

the following functions are defined

| | |
|-------------------|------------------|
| $\delta(N, P, G)$ | Fueling function |
| $T_e(N, P, G)$ | Engine torque |

The operating range of the engine in terms of the speed N is set to $N \in [600, 2500]$ rpm. These functions are defined on this range.

2.1.1 Approximations

The set of functions above originates from measurements and are therefore sets of discrete values. To deal with the system analytically it is useful to describe the functions as continuous rather than discrete. This is achieved by fitting the data to polynomial functions of varying degrees. The coefficients are calculated with the least square method. The calculated values of the coefficients are to be found in appendix A.

The engine map is fairly linear in N and δ . Therefore, the following approximation is chosen

$$\hat{T}_{map}(N, \delta) = a_e N + b_e \delta + c_e \quad (2.3)$$

This approximation provides a good agreement with the measurement data. A model for the drag torque $T_{drag}(N) = T_{map}(N, 0)$ is obtained using only measurements when the fueling δ is zero. As seen in figure 2.1 (curve C), a linear model seems sufficient. Accordingly, set

$$\hat{T}_{drag} = a_d N + b_d \quad (2.4)$$

The upper torque and fueling bounds appear to be well approximated by

$$\hat{T}_{max}(N) = a_T N^2 + b_T N + c_T \quad (2.5)$$

$$\hat{\delta}_{max}(N) = a_\delta N^2 + b_\delta N + c_\delta \quad (2.6)$$

as can be seen in figure 2.2. To further simplify the expression for engine

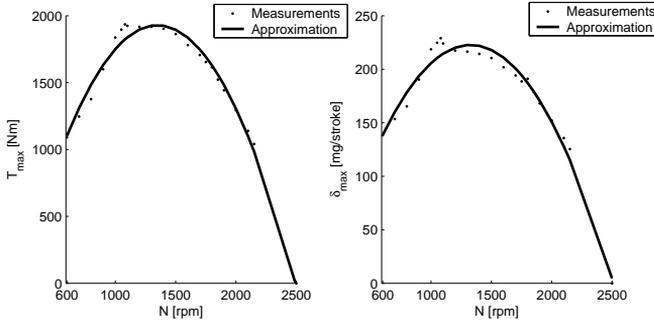


Figure 2.2: Upper torque and fueling bounds.

torque in equation (2.2), the upper torque bound can be neglected. It is thus assumed that

$$\hat{T}_{map}(N, \hat{\delta}_{max}(N)) = \hat{T}_{max}(N) \quad (2.7)$$

holds for all N . The agreement of this equality is shown in figure 2.3.

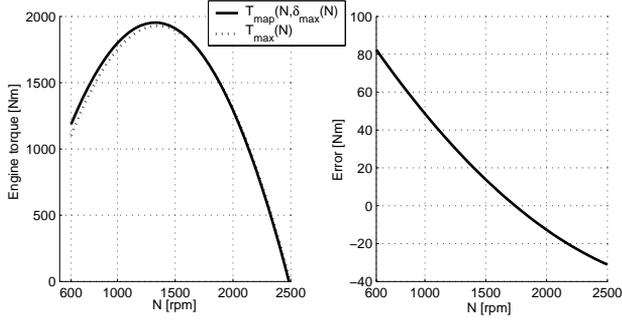


Figure 2.3: Left: The left and right hand side of (2.7). Right: The error of this assumption.

By using equations (2.1) and (2.3) to (2.7), (2.2) can now be approximated as

$$\hat{T}_e(N, P, G) = \begin{cases} a_e N + b_e P \hat{\delta}_{max}(N) + c_e & P > 0, G \neq 0 \\ a_d N + b_d & P \leq 0, G \neq 0 \\ 0 & G = 0 \end{cases} \quad (2.8)$$

2.2 Driveline

A vehicular driveline with torque and angle labels is shown in figure 2.4. The driveline is assumed stiff. The transmission is modeled in a very simple way with a gear ratio and efficiency.

Engine The set of functions in the previous section models T_e , the produced torque in the combustion and the internal friction from the engine. The external load arise from the clutch, T_c . Newton's second law of motion gives

$$J_e \ddot{\vartheta}_e = T_e - T_c$$

The mass moment of inertia of the engine is J_e and the angle of the flywheel is ϑ_e .

Clutch The clutch is assumed stiff, which yields

$$\begin{aligned} T_c &= T_t \\ \vartheta_e &= \vartheta_c \end{aligned}$$

Transmission Transmission inertia is disregarded. A gear number G maps to two parameters, the conversion ratio i_t and an efficiency η_t that models energy losses. This gives

$$\begin{aligned} T_t i_t \eta_t &= T_p \\ \vartheta_c &= i_t \vartheta_t \end{aligned}$$

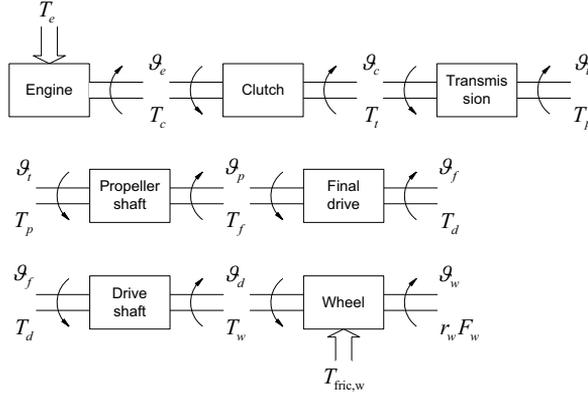


Figure 2.4: A vehicular driveline. Figure adapted from Nielsen and Eriksson (2003, p. 110)

where i_t is the current gear's conversion ratio.

Propeller shaft The propeller shaft is presumed stiff, which gives

$$\begin{aligned} T_p &= T_f \\ \vartheta_t &= \vartheta_p \end{aligned}$$

Final drive In like manner as for the transmission, inertia is neglected and losses are modeled by an efficiency η_f . This brings forth

$$\begin{aligned} T_f i_f \eta_f &= T_d \\ \vartheta_p &= i_f \vartheta_f \end{aligned}$$

where i_f is the conversion ratio in the final drive.

Drive shafts The drive shafts is assumed stiff and therefore

$$\begin{aligned} T_w &= T_d \\ \vartheta_f &= \vartheta_d \end{aligned}$$

Wheel Neglecting the wheel friction $T_{fric,w}$, then

$$\begin{aligned} J_w \ddot{\vartheta}_w &= T_w - k_b B - r_w F_w \\ \vartheta_d &= \vartheta_w \end{aligned}$$

where J_w is the wheel inertia and r_w is the wheel radius. F_w is the resulting friction force at the wheel. The brake torque is simply modeled as $k_b B$ where B , $B \in [0, 1]$ is the brake control signal and k_b is a constant parameter.

2.3 Longitudinal forces

The vehicle is affected by the following longitudinal forces.

Aerodynamic drag F_a is estimated by

$$F_a = \frac{1}{2} c_w A_a \rho_a v^2$$

where c_w is the air drag coefficient, A_a is the maximum cross section area of the vehicle, ρ_a is the air density and v is the velocity of the truck.

Rolling resistance F_r is modeled as being proportional, with the rolling resistant coefficient c_r , to the normal force of the vehicle on the tires F_N . This gives

$$F_r = c_r F_N$$

$$\text{where } F_N = mg \cos \alpha$$

The road slope is α and m is the mass of the truck.

Gravitational force F_g is simply

$$F_g = mg \sin \alpha$$

where α is the road slope and m is the mass of the truck.

Newton's second law now yields

$$m\dot{v} = F_w - F_a - F_r - F_g$$

where F_w is the resulting friction force at the wheel.

2.4 Complete driveline model

Assume that the current gear G is another than neutral gear. The vehicle velocity v is then

$$v = \dot{\vartheta}_w r_w = \frac{r_w}{i_t i_f} \dot{\vartheta}_e$$

where r_w is the wheel radius. With this identity, the equations in this chapter can be combined to yield

$$\dot{v} = \frac{r_w}{J_w + mr_w^2 + \eta_f i_f^2 \eta_t i_t^2 J_e} \left(\eta_t i_t \eta_f i_f T(v, P, G) - k_b B - \frac{1}{2} c_w A_a \rho_a r_w v^2 - mgr_w (c_r \cos \alpha + \sin \alpha) \right)$$

where T is the engine torque.

When neutral gear is used, no torque is fed to the transmission and combining the equations yields

$$\dot{v} = \frac{r_w}{J_w + mr_w^2} \left(-k_b B - \frac{1}{2} c_w A_a \rho_a r_w v^2 - mgr_w (c_r \cos \alpha + \sin \alpha) \right)$$

A comparison between the two equations above reveals that neutral gear is equivalent to a conversion ratio i_t which is zero. Neutral gear will therefore be assigned a ratio of zero.

The relationship between the engine speed and vehicle velocity is defined as

$$N = \frac{30}{\pi} \dot{v}_e = \begin{cases} \frac{30}{\pi} \frac{i_t i_f}{r_w} v & G \neq 0 \\ N_{idle} & G = 0 \end{cases} \quad (2.9)$$

where N_{idle} is the idle engine speed. It is assumed that the engine speed becomes the idle speed immediately when the gear signal becomes zero.

A complete driveline model can now be stated as

$$\dot{v} = \frac{r_w}{J_w + mr_w^2 + \eta_f i_f^2 \eta_t i_t^2 J_e} \left(\eta_t i_t \eta_f i_f T(v, P, G) - k_b B - \frac{1}{2} c_w A_a \rho_a r_w v^2 - mgr_w (c_r \cos \alpha + \sin \alpha) \right)$$

where $T(v, P, G) = T_e(N, P, G)$ with the engine speed N according to (2.9). The engine torque T_e is described by equation (2.2) or the approximation in (2.8). The control signals are shown in table 2.1. A gear number G maps to a

| Variable | Description |
|----------------|---|
| $P \in [0, 1]$ | Pedal signal, normalized fueling. |
| $B \in [0, 1]$ | Brake signal. |
| $G \in H$ | Gear signal. H is the set of available gears. |

Table 2.1: Control signals in the driveline model.

conversion ratio i_t and an efficiency η_t . Gear number zero represents neutral gear and has a ratio of zero.

2.5 Cruise control

The cruise controller is made up of two PI controllers. One for the pedal signal P and one for the brake signal B . The benefit of this is that it's possible to have different set points. Since the speed is allowed to increase above the reference v_{ref} , it's not desirable to brake until the maximum allowed velocity v_{max} is reached. With the actual speed labeled v_{act} the pedal signal P can be

stated as

$$P(t) = \begin{cases} 0 & f_P(t) \leq 0 \\ f_P(t) & 0 < f_P(t) < 1 \\ 1 & f_P(t) \geq 1 \end{cases} \quad (2.10)$$

$$\text{where } f_P(t) = K_P (v_{ref} - v_{act}) + I_P \int_0^t (v_{ref} - v_{act}) dt$$

and the brake B signal as

$$B(t) = \begin{cases} 0 & f_B(t) \leq 0 \\ f_B(t) & 0 < f_B(t) < 1 \\ 1 & f_B(t) \geq 1 \end{cases} \quad (2.11)$$

$$\text{where } f_B(t) = K_B (v_{max} - v_{act}) + I_B \int_0^t (v_{max} - v_{act}) dt$$

When implementing the cruise controller, care should be taken to prevent integrator wind-up. Wind-up is always an issue when dealing with controllers including integrator states, but with the configuration of two PI controllers it is easy to realize that it becomes even more important. As long as the accelerator controller is active, the integrator state of the brake controller will decrease and vice versa. The integrator states are therefore saturated.

Gear selection is simply made based upon the current engine speed. For each gear, two engine speed values are stored. The lower value is the threshold where the next lower gear is chosen. When the upper engine speed value is reached, the next higher gear will be used. In figure 2.5 is the engine speed plotted against the vehicle velocity for the six highest gears. The engine's operating range is between 600 and 2500 rpm. The first gear shifting points are marked in the figure by arrows.

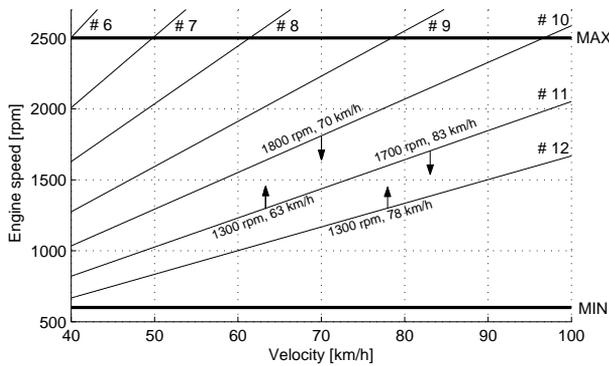


Figure 2.5: Gear selection

2.6 Fuel consumption

The mass flow of fuel \dot{m}_f [g/s] is determined by the fueling δ [mg/stroke] in equation (2.1) and the engine speed N [rpm]. The mass flow in [g/s] is then

$$\begin{aligned}\dot{m}_f(N, \delta) &= c_f N \delta \\ c_f &= \frac{1}{6 \cdot 10^4} \frac{n_{cyl}}{n_r}\end{aligned}\quad (2.12)$$

where n_{cyl} is the number of cylinders and n_r is the number of crankshaft revolutions per stroke. With (2.13) and the fueling according to equation (2.1) the fuel consumption model is

$$\begin{aligned}m_f(N, P, G) &= \int \dot{m}_f(N, P, G) dt \\ \dot{m}_f(N, P, G) &= \begin{cases} c_f N P \delta_{max}(N) & G \neq 0 \\ c_f N_{idle} \delta_{idle} & G = 0 \end{cases}\end{aligned}\quad (2.13)$$

where N_{idle} is the idle engine speed and δ_{idle} is the idle fueling. With (2.9) the consumption model is made dependent on the velocity v instead of the engine speed N .

2.7 Model implementation

The model equations are implemented in MatLab/Simulink. The complete driveline model in (2.10) is used together with the engine map (2.2). Further, the approximations of the upper torque bound (2.5), fueling bound (2.6) and the drag torque (2.4) are used. The vehicle is controlled by the cruise controller in (2.10) and (2.11). Gear selection is made as described above in section 2.5. The fuel consumption is modeled according to (2.13). See appendix A for details.

Chapter 3

Control of a hybrid system

This thesis explores the explicit use of topographic information in a controller. A prerequisite is that the choice of route is known. This information must be given by the driver or be guessed by the system. A route is represented by a finite set of information. The problem at hand is thus a control problem over a finite horizon.

It is assumed that the vehicle model is deterministic and the information about the route is exactly known. With sensor errors, imperfect topographic information, model errors and other disturbances uncertainty is however introduced. In this thesis the disturbances is assumed to be approximately zero in order to simplify matters. Robustness is nevertheless of importance and should be investigated when a controller is designed. One way to handle a situation where the calculated control signals not are suitable for direct use, is to let the optimal controller only generate reference trajectories. An external controller that are more robust against disturbances is then used to carry out tracking of these trajectories.

In this chapter, methods for optimal control of hybrid systems are reviewed. Modeling of hybrid system is first shortly discussed. The core is a survey of different control approaches. Finally, conclusions are made that lead to the choice of course taken in this thesis.

3.1 Modeling

The vehicle model in chapter 2 is a hybrid system, which means that it is a dynamic system with both continuous and discrete parts. The position and velocity make up two continuous states. Accelerator and brake levels are continuous input signals. The gear signal is discrete. Further is the evolution of the velocity state non-linear.

The approach taken when formulating a mathematical model of a hybrid system should, as in all modeling, depend strongly on the purpose of the

model. Hybrid automata have been the common approach in the computer science community to model a hybrid system. By letting continuous dynamics evolve within each of the states, the automata model is extended to deal with hybrid systems. In automatic control, a more algebraic approach has commonly been used. The discrete parts are described by algebraic equations and constraints. (Camacho and Bordons, 2004; Hedlund, 2003)

3.2 Optimal control

A general way of tackling a control problem is to seek the control law that is optimal according to some stated criteria. It is however often difficult to quantify the value of controller performance. Optimal control is a method to achieve an optimal controller in a systematic way. The desired properties are mathematically stated in a cost function. It can depend on the state variables and the control signals in the system. The control law that minimizes the cost function subject to system constraints is the optimal controller. The resulting mathematical problem can however be hard to solve. It is therefore common with various approximations, for example simplification of the original problem formulation.

3.2.1 Variational methods

For continuous system the maximum principle (Ljung and Glad, 2003) state necessary optimality conditions. In Piccoli (1999); Sussmann (1999) the maximum principle is extended to hybrid systems where the discrete switching is autonomous. The maximum principle is however limited to be used on the continuous evolution of the system during time intervals when no discrete switching occur.

3.2.2 Dynamic programming

The basis for dynamic programming (DP) is the principle of optimality. For a discrete dynamic system, the DP algorithm, stated in section 4.2, finds the optimal control policy. For a deterministic finite-state problem, the DP problem is equivalent with a shortest path problem in a directed graph. (Bellman, 1957; Bertsekas, 1995)

The continuous counterpart of the DP algorithm is the Hamilton-Jacobi-Bellman (HJB) equation. This is a partial differential equation that state constraints on the optimal value function and the corresponding control signal. The equation can be used to give sufficient conditions for optimality (Bertsekas, 1995; Ljung and Glad, 2003). In Bensoussan and Menaldi (1997); Branicky and Mitter (1995) the HJB equation for hybrid systems is formulated.

With DP, analytical solutions can be obtained but it often requires rather simple models. In many realistic problems the viable approach is a numeric solution. Unfortunately, the number of calculations needed for a complete solution is often far too great for a usable algorithm. Bellman called this *the curse of dimensionality*. The curse lies in that, with a forthright approach, the required amount of computations grows exponentially with the dimensions of the state, control and disturbance spaces. In order to end up with a tractable algorithm, a suboptimal method is thus generally necessary. Care should then be taken to find an acceptable balance between algorithm complexity and performance. In contrast, the power of DP is its wide scope of applicability in that it manages complex constraint sets such as integer or discrete sets. (Bertsekas, 1995)

3.2.3 Model Predictive Control

Model predictive control (MPC) make use of a model of the process to predict future outputs as a function of possible control signals. The control signal that is optimal according to a criterion is then chosen. Formally, the methodology used by a MPC controller is as follows:

1. The process model is used at each instant t to predict future outputs y for a determined prediction horizon M . The outputs $y(t+k|t)$, $k = 1 \dots M$ depends on future control signals $u(t+k|t)$, $k = 1 \dots N$ for a determined control horizon N and the measurable disturbances that are know at time t .
2. Formulate a criterion based on predictions, control signals and measurements. Optimize with respect to the control signals.
3. Send the optimal control signal $u(t|t)$ to the process and disregard from the rest of the calculated control signals.
4. At the next sample hit, repeat from step 1.

All but the first calculated control signal is rejected because at the next sample hit, step one is repeated and $u(t+1|t+1)$ will then in general be different from $u(t+1|t)$ owing to the fact that new information will be available. (Camacho and Bordons, 2004; Ljung and Glad, 2003)

The fundamental parts of MPC are the process model and the optimizing algorithm. The model must be accurate enough for the predictions and at the same time simple enough for a simple and fast implementation. The criterion, or cost function, together with the chosen model is decisive for which algorithm to use.

In Bemporad and Morari (1999) the algebraic modeling approach mentioned above in section 3.1 is used when proposing a MPC scheme for mixed logical dynamical (MLD) systems. A MLD system is described by linear dynamic equations and linear inequalities containing real and integer variables.

A linear hybrid system is one example of a MLD system. With a quadratic cost function the optimization is carried out through mixed integer quadratic programming (MIQP).

Another method of modeling a hybrid system is by piecewise affine systems (PWA) (Camacho and Bordons, 2004). The state and control space is partitioned and in each subset the system is described by affine functions, that is the sum of a linear function and a constant. Integer variables are then used to switch among the models. In Heemels et al. (2001) the equivalence of different hybrid dynamical models is shown. Among the models are MLD and PWA systems. It is thus possible to transfer tools for one class of systems to another.

The system at hand is a non-linear hybrid system. It is possible to model the hybrid nature and to approximate the non-linearities arbitrary well with a PWA description. As mentioned above a PWA system is equivalent to a MLD, and MIQP can be used for optimization. A PWA description of the vehicle system should contain a number of regions corresponding to velocity intervals for each one of the gears. With 3 speed intervals for 6 gears, there are 18 regions. A sampling interval in the region of seconds and a prediction horizon of probably more than ten steps are needed. Solving a MPC problem with horizon N with a PWA model with S regions using MIQP renders a maximum number of QP problems of S^N , which is the same as the number of combinations for the integer variables used to switch models in the PWA description (Camacho and Bordons, 2004). In this example, that would give $18^{10} \approx 3.6 \cdot 10^{12}$ QP problems which is too many for a real time controller. It is thus necessary to reduce the number of combinations of the integer variables. In Peña et al. (2003) the reachable regions in the next few sample times from the current regions are determined. This makes a reduction in the number of QP problems to solve.

The MPC scheme does not specify the optimization algorithm to use. For a non-linear hybrid system, the optimization can be characterized as dynamic mixed-integer optimization (MIDO). The most frequent appeared approach to solutions of MIDO problems are based on decomposition principles. One method is to convert the MIDO problem into a mixed-integer non-linear program (MINLP). The MINLP can then be solved by means of a standard algorithm, such as branch and bound, outer-approximation or cutting plane methods. (Bansal et al., 2003; Biegler and Grossmann, 2004a,b)

3.3 Conclusions

When formulating the vehicle model in chapter 2 the algebraic modeling approach is chosen.

Solving an optimal control problem for a hybrid system is a formidable task. All DP algorithms suffer from the curse of dimensionality, which means that the complexity grows exponentially with the problem dimensions. The

approach with a PWA description and a MIQP solver is interesting but it is feared that the complexity, which is roughly estimated above, would be too great for the current application. MIQP is an extension of MINLP methods (Biegler and Grossmann, 2004b) and therefore general MINLP methods are also suspected to lead to algorithms that are too complex.

It is not anticipated that a control law can be computed at each instant, for the entire information horizon that is present. This leads to solving smaller problems repeatedly and the MPC scheme is thus chosen.

All mentioned algorithms are suspected to render into great complexity. Therefore it is expected that whatever the method, the space that is searched for an optimal solution needs to be reduced. DP handles constraints on the state and control spaces easily. In the beginning of this chapter, it was stated that all but the disturbances that are measurable are neglected. If the system is discretized, the assumption about the disturbances makes it possible to formulate the DP problem as a shortest path problem. Restrictions on the search space are then imposed with easiness.

The approach taken in this thesis is a MPC scheme with discrete DP as optimizer.

Chapter 4

Dynamic Programming

The mathematical foundation of dynamic programming (DP) is shortly presented in this chapter. The ideas that lie behind DP are old. However, it was Bellman with his works, published in 1957 and 1962, that started to unify the theory and showed the wide scope of applicability of DP. Besides the books by Bellman, more recent publications (Bertsekas, 1995; Denardo, 1982) have formed the basis for this chapter.

4.1 System

A discrete dynamic system is described as

$$x_{k+1} = f_k(x_k, u_k, w_k), \quad k = 0, 1, \dots, N-1 \quad (4.1)$$

$$\begin{aligned} x_k \in S_k & \quad \text{is the state} \\ u_k \in U_k & \quad \text{is the control} \\ w_k \in D_k & \quad \text{is the random disturbance} \end{aligned}$$

The state x_k is said to belong to stage k .

A policy

$$\pi = \{\mu_0, \mu_1, \dots, \mu_{N-1}\}$$

is considered to be a sequence of functions that transform states x_k into controls u_k , that is

$$u_k = \mu_k(x_k)$$

The set Π contains policies for which

$$\mu_k(x_k) \in U_k \quad \forall x_k, k$$

holds. A policy $\pi \in \Pi$ is called an admissible policy.

Weighting functions, ζ_k, ζ_N where

$$\zeta_k(x_k, u_k, w_k), \quad k = 0, 1, \dots, N-1 \quad \text{and} \quad \zeta_N(x_N)$$

are used to define the cost of a policy.

With an initial state x_0 together with an admissible policy $\pi \in \Pi$, the cost is

$$J_\pi(x_0) = \zeta_N(x_N) + \sum_{k=0}^{N-1} \zeta_k(x_k, \mu_k(x_k), w_k) \quad (4.2)$$

This quantity is to be minimized. The policy that achieves this is called an optimal policy π^* . With an initial state x_0 this means that

$$J_{\pi^*}(x_0) = \min_{\pi \in \Pi} J_\pi(x_0)$$

and the optimal cost is

$$J^*(x_0) = \min_{\pi \in \Pi} J_\pi(x_0)$$

4.2 Algorithm

The foundation for dynamic programming is a simple principle and it is stated below.

Principle of Optimality

Let $\pi^* = \{\mu_0^*, \mu_1^*, \dots, \mu_{N-1}^*\}$ be an optimal policy for the system in (4.1) and assume that π^* has been used up to stage i . Thus, the state is x_i and the subproblem to minimize the *cost-to-go* from stage i to N is faced.

$$J_{x_i}^* = \zeta_N(x_N) + \sum_{k=i}^{N-1} \zeta_k(x_k, \mu_k(x_k), w_k).$$

Then the truncated policy $\{\mu_i^*, \mu_{i+1}^*, \dots, \mu_{N-1}^*\}$ is optimal for this subproblem.

To draw a parallel with the application in this thesis, assume that the optimal policy for controlling a vehicle on the track from Linköping to Uppsala is sought. If the optimal policy going from Stockholm to Uppsala is known, that policy is also optimal on this part of the route from Linköping to Uppsala. The problem faced is then to find an optimal policy between Linköping and Stockholm to receive a policy for the entire route.

The quantity $J_k(x_k)$ is interpreted as the optimal cost for the subproblem having $(N - k)$ number of stages, starting at x_k and ending in x_N . The entity $J_k(x_k)$ will be called the *cost-to-go* at stage x_k , step k . Following the principle above, a simple and intuitive algorithm can be formulated. It starts with the subproblem containing the two last stages, $N - 1$ and N . Solving this subproblem gives μ_{N-1}^* and the *cost-to-go* from stage $N - 1$. Henceforth the

remaining stages are included one by one in the subproblem. This algorithm is stated below.

DP Algorithm

1. Let $J_N(x_N) = \zeta_N(x_N)$.
2. Let $k = N - 1$.
3. Find $u_k^* = \mu_k^*(x_k)$ that satisfies

$$J_k(x_k) = \min_{u_k \in U} \{ \zeta_k(x_k, u_k, w_k) + J_{k+1}(f_k(x_k, u_k, w_k)) \}.$$

4. Let $k = k - 1$.
 5. If $k \geq 0$ goto 3.
 6. For every initial state x_0 , the optimal cost is $J^*(x_0) = J_0(x_0)$.
 7. The policy $\pi^* = \{ \mu_0^*, \mu_1^*, \dots, \mu_{N-1}^* \}$ is optimal.
-

The optimality of the algorithm can be proved by induction (see Bertsekas, 1995).

The system (4.1) is deterministic if the random disturbance w_k can take only one value. The evolution of the system is then exactly predictable and there is no gain in using feedback. The minimization could thus be done over sequences of control vectors rather than over admissible policies. (Bertsekas, 1995)

The evolution of a deterministic problem with a finite state space, under the influence of different control signals, can be represented in a directed graph. An arc represents a transition between states in successive stages and is associated with a cost for this transition. The cost of an arc can be viewed as the length of that arc. Through this, the deterministic problem is made equivalent with a shortest path problem in a graph. The final stage is handled by a virtual terminal node t and connecting each state x_N in stage N with arcs having a cost of $g_N(x_N)$.

The DP algorithm for this formulation is stated below (Bertsekas, 1995). The new notation introduced is

$$\begin{aligned} a_k^{i,j} &= \text{transition cost at step } k \text{ from state } i \in S_k \text{ to state } j \in S_{k+1} \\ a_N^{i,t} &= \text{terminal cost of state } i \in S_N. \end{aligned}$$

The cost $a_k^{i,j}$ is equal to $\zeta_k(i, u_k^{i,j}, w_k)$, where $u_k^{i,j}$ is the control that cause the transition from state i to j . The terminal cost of state i is equal to $g_N(i)$.

An example network is depicted in figure 4.1.

Shortest path DP Algorithm

1. Let $J_N(i) = a_N^{i,t}$, $i \in S_N$.

2. Let $k = N - 1$.

3. Let

$$J_k(i) = \min_{j \in S_{k+1}} \{a_k^{i,j} + J_{k+1}(j)\}, \quad i \in S_k.$$

4. Repeat step 3 for $k = N - 2, N - 3, \dots, 1$.

5. The optimal cost is $J_0(s)$ which is equivalent to the length of the shortest path from s to t .

6. The control sequence along the shortest path is optimal.

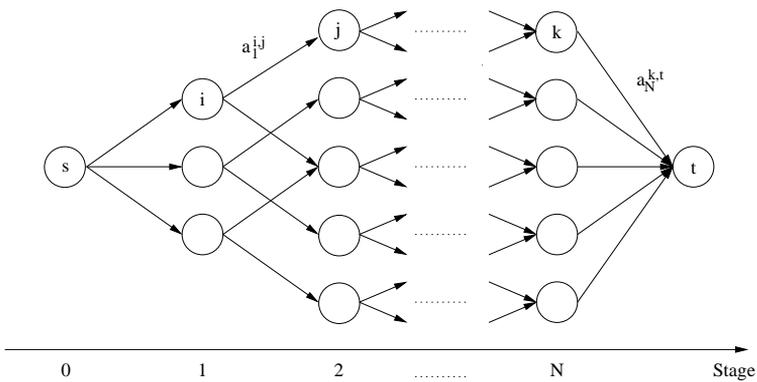


Figure 4.1: Evolution of a deterministic system depicted as a transition graph.

Chapter 5

Control algorithm

This chapter explains the design of the algorithm to be used as the optimization component in a MPC-scheme. The objectives of the controller and the constraints on the solution are first identified. Further is an effort made to reduce the search space for dynamic programming. A function is defined that weigh control actions versus their consequences. This cost function is decisive for which state trajectory that is rated as optimal. The algorithm is finally summarized.

An overview of the controller structure is seen in figure 5.1. Input to the controller is feedback from the vehicle and information from a database. The most important information the database contains is road maps including calculated road slopes along the way. Besides this knowledge, it is possible to

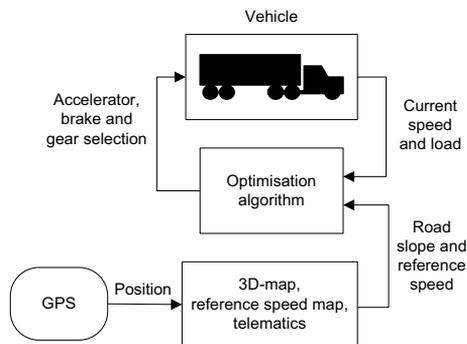


Figure 5.1: The controller structure.

get more information that is useful as well, with aid of the position feedback from the GPS. With a reference speed map, the velocity set point can be adjusted automatically. If the curvature of the road is known and stored, it can be used to lower the reference speed in road segments with a high curvature.

Information about the current traffic situation could for example be retrieved and also be used in determining the set point. In this thesis the focus is however on utilizing the road slope information.

5.1 Objective and constraints

The main objective of the system is to keep the vehicle in an allowed range of velocities with a minimum use of fuel. Denote the reference velocity v_{ref} [km/h]. The vehicle velocity is allowed to decrease with v_{dec} and increase with v_{inc} from v_{ref} . The constraint on the vehicle speed v can then be expressed as

$$v_{ref} - v_{dec} \leq v \leq v_{ref} + v_{inc} \quad (5.1)$$

The lower limit will be denoted $v_{min} = v_{ref} - v_{dec}$ and the upper limit $v_{max} = v_{ref} + v_{inc}$.

It can not be expected that a heavy truck can be kept above the minimum allowed speed v_{min} , on all possible road configurations. The lower bound is therefore treated as a soft constraint. This means that the solution will not be infeasible if the constraint does not hold but violation will be penalized.

The brake system is assumed to be effective enough for the road ahead and the upper bound is thus treated as a hard constraint that is not allowed to be broken. Brake use should further at least not increase compared to traditional cruise control.

Gear switching is a process that takes an amount of time that is not negligible. During the shifting process the engine can not propel the vehicle and kinetic energy is then lost. Because of the required time, frequent gear changing is neither possible nor desirable. The model at hand does not, for simplicity reasons, contain these dynamics. A limit to the shifting frequency is therefore needed.

5.2 Problem representation

To model the vehicle, the complete driveline model in (2.10) with the approximation (2.8) of the engine torque is used. The fuel consumption is modeled according to (2.13) together with the approximation of the upper fueling bound in (2.6). This gives a state description with velocity as the only state, three control signals and the slope as one measurable disturbance. The output signals are the velocity and the fuel flow.

$$\begin{aligned} \dot{v} &= f_1(v, \underline{u}, \alpha) \\ y_1 &= v \\ y_2 &= f_2(v, \underline{u}, \alpha) \end{aligned} \quad \underline{u} = [P \ B \ G]^T \quad (5.2)$$

The road profile map is position dependent rather than time dependent, as is the vehicle model. This is handled by transforming the latter to a position

dependent model by the following simple rewrite.

$$\frac{dv}{dt} = \frac{dv}{ds} \frac{ds}{dt} = \frac{dv}{ds} v \Rightarrow \frac{dv}{ds} = \frac{1}{v} \frac{dv}{dt}, \quad v \neq 0 \quad (5.3)$$

The event of a vehicle stop, $v = 0$, is not included in the problem formulation. It is thus presumed that it is possible to keep a velocity greater than zero at all time.

The optimization problem at hand is to be solved numerically by means of dynamic programming (DP). A discrete model is therefore needed.

The stage grid in DP is set to $S [m]$. Divide S into M parts. Let $h = \frac{S}{M}$ and denote $v_k = v(kh)$. It is further assumed that the inputs and the disturbance is constant during S , that is

$$\begin{aligned} \underline{u}(s) &\equiv \underline{u}_k \quad \forall s \in [kS, (k+M)S[\\ \alpha(s) &\equiv \alpha_k \end{aligned}$$

Euler's numerical integration method with step length h and the velocity assumption then gives

$$v_{k+1} = v_k + \frac{h}{v_k} f_1(v_k, \underline{u}_k, \alpha_k) \quad k = 0, 1, \dots, M \quad v_k > 0 \quad \forall k \quad (5.4)$$

To determine the fuel mass consumed the output signal y_2 is integrated. Applying Euler's method again with the step length h yields

$$m_{f,k+1} = m_{f,k} + \frac{h}{v_k} f_2(v_k, \underline{u}_k, \alpha_k) \quad k = 0, 1, \dots, M \quad v_k > 0 \quad \forall k \quad (5.5)$$

See appendix B concerning accuracy and stability issues for the use of Euler's method on the vehicle model.

A stiff driveline and a transmission modeled with a ratio and an efficiency are assumed. The dynamics in a gear change is neglected. A gear shift is then only an instantaneous shift of the ratio and efficiency parameters.

5.3 State augmentation

A limit on the maximum possible gear shifting frequency is wanted. The bound is denoted k_{lim} and defined in terms of stages, of length S , that must pass before a change of gear is allowed. This restriction reduces the available control actions, but raises the need for information in a state. This is achieved by expanding the state vector. The gear that has brought the system to the current state is included. A counter that keeps track of the number of stages passed since the last gear change is stored in a state together with the state vector. The constraint is then easily imposed by not allowing a gear shift if the counter is less than the limit k_{lim} .

The enlarged discrete system description is then

$$\begin{aligned}
 v_{k+1} &= v_k + \frac{h}{v_k} f_1(v_k, \underline{u}_k, \alpha_k) \\
 g_{k+1} &= g_k \\
 c_{M+1} &= \begin{cases} 1 + c_0 & , \quad g_{M+1} = g_0 \\ 1 & , \quad g_{M+1} \neq g_0 \end{cases} \\
 m_{f,k+1} &= m_{f,k} + \frac{h}{v_k} f_2(v_k, \underline{u}_k, \alpha_k) \\
 k &= 0, 1, \dots, M \quad v_k > 0 \quad \forall k
 \end{aligned} \tag{5.6}$$

where g denote the gear number and c the counter.

5.4 Determining the search space

Due to the curse of dimensionality, it is important to reduce the search space. The search space consists of the values of velocities and control signals which are considered by the DP algorithm.

The allowed velocities, specified in (5.1), are a first limitation. The lower bound is however treated as soft, and therefore it may be necessary to take additional velocities into account. The initial state is known and with this information the first part of the search space can be reduced, because it is not certain that all of the allowed velocities are reachable in one step. If constraints are set on possible final states the possible velocities can be restricted further.

The set of gears can be limited by introducing an allowed range for the engine speed. The brakes will only be applied if the upper bound v_{max} otherwise would be reached. The pedal signal will be calculated rather than discretized.

The rest of this section explains in detail how the search space is determined.

5.4.1 Gear selection

A gearbox in a heavy truck can have more than ten gears. But with a given velocity only a subset of these are applicable. With a constraint on the engine speed N it is possible to select a set of usable gears in a state. A gear G is a number that maps to a conversion ratio i_t and an efficiency η_t . In a state with the velocity v_k , the set of usable gears G_{v_k} is then defined as

$$G_{v_k} = \{G \mid N_{min} \leq N(v_k, G) \leq N_{max}\} \cup \{0\} \tag{5.7}$$

where $N(v_k, G)$ is the engine speed at vehicle velocity v_k and gear number G with parameters $\{i_t, \eta_t\}$,

$$N(v_k, G) = \frac{30}{\pi} \frac{i_t i_f}{r_w} v_k, \quad G \neq 0$$

where i_f is the final drive ratio and r_w the wheel radius. Note that neutral gear is gear number zero and modeled by a ratio i_t which equals zero.

5.4.2 Considered velocities

In order to determine the velocities to consider the reachable velocities, with consideration of the allowed range, from the initial state along the horizon is first calculated. This gives an interval of velocities for each stage. The lower bound in the last stage is then increased to the reference velocity v_{ref} , or set equal to the higher bound if that goes below v_{ref} . This is done in order to prevent the solutions of always ending in a state with a velocity lower than the set point. In general, that would save fuel but it is not a desirable behavior. With this restriction it is possible to go through the interval backwards from the last stage and remove states from where it is not possible to reach one of the allowed velocities in the last stage. An example is shown in figure 5.2. The gray area is the part of the state space that will be considered. The darker area is the velocities that are removed when going backwards from the last stage.

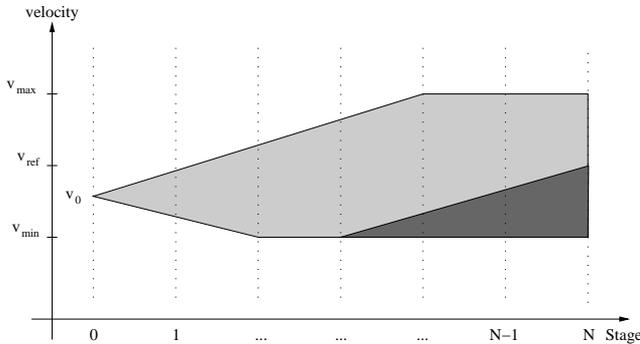


Figure 5.2: The velocity state space

For a fix initial velocity and gear it is clear that maximum fueling, $P = 1$ gives the highest end velocity. Which gear that lead to the highest speed is not as evident. It can however easily be determined by simulating. Let a vector $\underline{v}_{P=1}$ contain these simulated velocities. The highest reachable speed v^+ is then the maximum of the speeds attained with each of the gears and $P = 1$.

$$v^+ = \max \{ \underline{v}_{P=1} \}$$

If brake use is disregarded, the lowest speed v^- that can be attained is when the pedal P is zero. The drag torque is modeled as linear in engine speed. The gear with the highest ratio gives the highest engine speed and hence the greatest drag torque and the lowest vehicle speed.

The set of reachable velocities is accordingly $[v^-, v^+]$. The constraints on the velocity, specified in (5.1), must then be imposed. Denote the set of velocities that are to be considered $[v_{low}, v_{high}]$. The higher limit v_{high} is

$$v_{high} = \min \{v^+, v_{max}\}$$

If the high limit exceeds the minimum allowed speed, then the lower bound is set to

$$v_{low} = \min \{ \max \{v^-, v_{min}\}, v_{max} \}, v^+ \geq v_{min}$$

That is, the limit is set to v^- as long as it not violates the bounds v_{min} and v_{max} . It is however a possibility that $v^+ < v_{min}$, and then the lower limit is set to the minimum of the speeds $\underline{v}_{P=1}$ that was attained with maximum fueling.

$$v_{low} = \min \{ \underline{v}_{P=1} \}, v^+ < v_{min}$$

The interval $[v_{low}, v_{high}]$ is first calculated for each stage, starting at the initial state in the first stage. The lower bound in the last stage is increased to the reference velocity v_{ref} , or set equal to the higher bound if that goes below v_{ref} . The intervals are then processed once again, starting out at the new lower bound in the last stage. The lower bound in the previous stage is increased to the minimum velocity required to reach the new bound in the last stage. This calculation is then repeated backwards through all stages.

There is now, for each stage, a set of velocities which are to be considered, $[v_{low}, v_{high}]$. This is a subset of the reachable velocities. The set in stage k is discretized in constant steps of δ . This make up a set V_k

$$V_k = \{v_{low}, v_{low} + \delta, v_{low} + 2\delta, \dots, v_{high}\} \quad (5.8)$$

5.4.3 Pedal and brake selection

The algorithm will generate grid points in neighboring stages and try to find feasible control actions between all of these, see section 5.6. Grid points consist of velocity and gear values. The subproblem is therefore to find a feasible control action, if possible, between two specified states.

Assume that the grid point in the current stage is $\{v_0, g_0\}$ and $\{v_1, g_1\}$ in the next stage. The required pedal signal P can then be computed from the system equation (5.4). If

$$-\epsilon \leq P \leq 1 + \epsilon, \quad \epsilon > 0$$

holds, a feasible control action is found. P is then limited to the interval $0 \leq P \leq 1$. With the calculated value of P , the system (5.6) is simulated with initial velocity v_0 and gear g_1 . This gives the fuel mass required for this choice of pedal level which is used to calculate the cost.

If no feasible pedal level is found and $|v_1 - v_{max}| < \delta$, a brake level is computed. The brake level that transforms the system from v_0 to v_1 is calculated from the system equation (5.4).

The parameter ϵ determines the values of the calculated pedal level that are treated as minimum ($P = 0$) and maximum ($P = 1$) fueling respectively. It is set by the user. Details about the preceding calculations are found in appendix B.

5.5 Cost function

With basis of the objectives, the weighting function ζ_k is set to

$$\zeta_k(v_k, v_{k+1}, \underline{u}_k, \underline{u}_{k+1}, \alpha_k) = [Q_1, Q_2, Q_3, Q_4, Q_5] \begin{bmatrix} m_{f,k} \\ \kappa(e_k)e_k^2 \\ |v_k - v_{k+1}| \\ \kappa(|G_k - G_{k+1}|) \\ B_k \end{bmatrix} \quad (5.9)$$

where $e_k = v_k - v_{ref,k}$ and κ is a step function

$$\kappa(t) = \begin{cases} 1 & , \quad t > 0 \\ 0 & , \quad t \leq 0 \end{cases} \quad (5.10)$$

First, the required fuel mass $m_{f,k}$ is included (Q_1). Only velocities above the reference speed is penalized (Q_2). It is desirable that the speed increases above the reference in order to gain kinetic energy, if that is advantageous in a long view. Velocity changes are penalized (Q_3) in order to receive a smooth control. Besides the shifting limit k_{lim} , gear changes are penalized in the cost function as well (Q_4). A big k_{lim} increases the risk of not finding a feasible solution, see section 5.6 below. With an extra cost on gear shifts, the need of a large k_{lim} is reduced. Finally is the brake level penalized (Q_5). The effect of the penalizing factors is stated in table 5.1 for clarity.

| Factor | Penalizes |
|--------|--|
| Q_1 | Fuel use |
| Q_2 | Negative deviation from the reference velocity |
| Q_3 | Velocity changes |
| Q_4 | Gear shifts |
| Q_5 | The use of brakes |

Table 5.1: Penalizing factors

The terminal cost is set to zero

$$\zeta_N = 0$$

The set of velocities that are considered in the last stage is limited, as described in section 5.4. It is not necessary nor desirable to penalize these allowed states. It is preferred to end up in a velocity greater than the lowest allowed, if that is advantageous considering the entire horizon.

5.6 Algorithm

The algorithm is started once every S meter. When the vehicle has traveled S meter, the calculated optimal control signal is applied. If the vehicle position is s , the algorithm calculates the control signal to apply when $s \in [s + S, s + 2S[$. The time available for the computation is the transport time when $s \in [s, s + S[$.

The first step is to predict the vehicle velocity after S meter with the current control signals. This predicted initial state is used as the starting point when determining the velocity range V_k which is to be considered in each stage k , as it is described in section 5.4 above.

Because of the principle of optimality, see chapter 4, and the fact that the MPC scheme only uses the first control of the calculated ones, it is not necessary to store any information about subsequent stages when the optimal costs has been computed for the states in a stage. This saves a lot of memory, especially if the number of steps is great.

A state i is made up of a velocity v and a gear number g , $i = \{v, g\}$. The counter c is merely used to make sure that the limit k_{lim} is not violated. The possible states $i \in S_k$ in stage k will be generated from the velocity range V_k and the set of gears G ,

$$S_k = \{\{v, g\} \mid v \in V_k, g \in G_v\}$$

The counters in the states of the last stage, c_N^j , $j \in S_N$, are set to k_{lim} . For each state in stage k , feasible control actions are sought which transforms the system into the states in stage $k + 1$. The feasible control action with the lowest cost is the optimal control from the current state.

The initial node is denoted s and the terminal node t . The notation for the costs is

$$\begin{aligned} a_k^{i,j} &= \text{transition cost at step } k \text{ from state } i \in S_k \text{ to state } j \in S_{k+1} \\ a_N^{i,t} &= \text{terminal cost of state } i \in S_N. \end{aligned}$$

The cost $a_k^{i,j}$ is equal to $\zeta_k(i, j, \underline{u}_k^{i,j}, \alpha_k)$, defined in (5.9). The control $\underline{u}_k^{i,j}$ causes the transition from state i to j with a road slope of α_k . The terminal cost of state i is equal to $g_N(i)$ which is set to zero according to (5.10). If there is no control that transforms the system from state i to j at stage k , $a_k^{i,j}$ is set to infinity. With a numerical approach, an infinite cost means a very large number in comparison with other transition costs.

The algorithm is summarily stated below.

DP algorithm

1. Let $J_N(i) = a_N^{i,t} = 0$.

2. Let $k = N - 1$.

3. Let

$$J_k(i) = \min_{j \in S_{k+1}} \left\{ a_k^{i,j} + J_{k+1}(j) \right\}, \quad i \in S_k.$$

A control action $u_k^{i,j}$ that transforms the system from state $\{v^i, g^i\}$, $i \in S_k$ to a state $\{v^j, g^j\}$, $j \in S_{k+1}$ is only allowed if $g^i = g^j$ or $(1 + c_{k+1}^j) \geq k_{lim}$. The counter of state i , c_k^i is set to $(1 + c_{k+1}^j)$ if $g^i = g^j$ and 1 otherwise.

4. Repeat step 3 for $k = N - 2, N - 3, \dots, 0$

5. The optimal cost is $J_0(s)$ and the sought control is the optimal control set from s .

When the sets of considered velocities V_k , $k = 1, \dots, N - 1$ are determined, the counter of the initial state, c_0^s , is taken into account. It not possible to regard the shifting limit in stages k , where $k \geq (k_{lim} + c_0^s)$, because it is not known beforehand if and where the optimal path contains a gear shift. This introduces a risk of not finding a feasible solution. The possibility is of course increased with a shifting limit of many stages. If no feasible solution is found, the controller outputs the last feasible control.

The behavior of the algorithm is controlled by a number of parameters. These are summarized in table 5.2.

| Parameter | Function |
|------------|--|
| S | The stage grid [m] |
| N | The number of steps of length S taken in the algorithm |
| h | The step length used to integrate the system equations [m] |
| δ | Velocity discretization [km/h] |
| ϵ | Accuracy used when calculating a pedal level |
| v_{inc} | Maximum increase above the reference speed [km/h] |
| v_{dec} | Minimum decrease below the reference speed [km/h] |
| k_{lim} | Minimum number of steps of length S before a gear shift |
| N_{min} | Lower bound on the operating range of the engine [rpm] |
| N_{max} | Upper bound on the operating range of the engine [rpm] |

Table 5.2: User parameters

5.7 Complexity

During the first phase of the algorithm, the set of velocities to be considered is determined. This requires the horizon to be searched twice. Two velocities are calculated at every step. This action is thus linear with the number of steps.

A state has a unique combination of a velocity and gear number. The possible number of states is thus dependent on the range of allowed velocities, the discretization δ and the number of gears. The range is

$$v_{max} - v_{min} = v_{inc} + v_{dec}$$

at the most according to section 5.4. The maximum number of states n is then

$$n = k_g \left(\left\lfloor \frac{v_{inc} + v_{dec}}{\delta} \right\rfloor + 1 \right)$$

where k_g is the maximum number of gears that is applicable at one stage. The limit on gear shifting frequency k_{lim} reduces the possible number of states but this is not taken into account.

The minimization of the cost function in the DP algorithm will at every step i process every combination of states in stage i and $i + 1$. The number of possible combinations is the product of the numbers of states in the two stages. At every step, the number of operations will therefore be proportional to the square of the number of states n^2 . The total number of operations for the minimization over a horizon of N steps is then proportional to Nn^2 .

The preceding reasoning leads to that the complexity O as function of the number of steps N can be approximated by

$$O(N) = kNn^2 = kNk_g^2 \left(\left\lfloor \frac{v_{inc} + v_{dec}}{\delta} \right\rfloor + 1 \right)^2 \quad (5.11)$$

where k is a constant. The complexity is linear with the horizon length. This is a direct consequence of the fact that the allowed velocities and thereby the number of states are limited.

Assume that the parameters v_{inc} and v_{dec} are determined but N and δ are not yet chosen. Let there be two different set of parameters, $N_1 \delta_1$ and $N_2 \delta_2$. The complexity ratio of $O(N_1)$ and $O(N_2)$ is

$$\frac{O(N_1)}{O(N_2)} = \frac{N_1 k k_g^2 \left(\left\lfloor \frac{v_{inc} + v_{dec}}{\delta_1} \right\rfloor + 1 \right)^2}{N_2 k k_g^2 \left(\left\lfloor \frac{v_{inc} + v_{dec}}{\delta_2} \right\rfloor + 1 \right)^2} \approx \frac{N_1}{N_2} \left(\frac{\delta_2}{\delta_1} \right)^2 \quad (5.12)$$

If the velocity discretization is made twice as accurate the horizon must then be made four times as short in order to achieve the same computational complexity.

Including the neutral gear in the set of gears that can be chosen obviously increases the complexity due to the fact that the possible number of states increases. To estimate the magnitude of the raise, let the complexity function depend on the parameter k_g . If k_g is increased by m , the complexity ratio is

$$\frac{O(k_g + m)}{O(k_g)} = \frac{Nk(k_g + m)^2 (\lfloor \frac{v_{inc} + v_{dec}}{\delta} \rfloor + 1)^2}{Nk k_g^2 (\lfloor \frac{v_{inc} + v_{dec}}{\delta} \rfloor + 1)^2} = \left(1 + \frac{m}{k_g}\right)^2 \quad (5.13)$$

With aid of figure 2.5 the number of gears that is applicable in neighborhood of one velocity is estimated to three. The complexity increase according to (5.13) with $m = 1$ then becomes

$$\frac{O(k_g + 1)}{O(k_g)} = \left(1 + \frac{1}{3}\right)^2 = \frac{16}{9} \approx 1.8$$

A rather large increase is the result. The gear shifting limit k_{lim} has been neglected and therefore the value can be seen as an upper bound to the complexity increase. If N is much larger than k_{lim} the ratio will be closer to the approximated value than if the values of N and k_{lim} are more alike.

The stage grid S do not directly affect the complexity of the algorithm but will of course increase the computational effort needed for the MPC controller by determining how often the algorithm is started.

Chapter 6

Simulations

In this chapter the results of a number of simulations on different road sections are reported. First the used parameters values are described and then results from both artificial and authentic road maps are discussed and illustrated. The magnitude of the gain in the possibility of using the neutral gear is investigated. Finally, the the computational complexity of the algorithm is examined.

To assess the performance on different road sections, the relative difference in the fuel consumption $\Delta fuel$ and the travel time $\Delta time$ are calculated as follows

$$\begin{aligned}\Delta fuel &= \frac{fuel_{MPC} - fuel_{PI}}{fuel_{PI}} \\ \Delta time &= \frac{time_{MPC} - time_{PI}}{time_{PI}}\end{aligned}$$

where the subscript refers to the respective controller.

When studying the computational complexity a ratio q is determined between the required computer time and the time simulated during that time.

$$q = \frac{\text{computer time}}{\text{simulated time}}$$

If it takes the computer t_1 s to simulate t_2 s, the ratio is $q = \frac{t_1}{t_2}$. A ratio of one thus means that the simulation precisely runs in real time. A ratio greater than one means that the simulation demands more time than the actual time simulated and conversely.

Simulation times reported are relative to a PC with a Intel Celeron 2.6GHz processor and 480Mb of RAM running Windows XP SP2. The MatLab version used is 6.5.1, Release 13.

6.1 Parameters

The resolution in position of the road measurements is 25m which is about the length of a heavy truck. This length is used as basis when choosing the stage grid S . Simulations with a grid of 10, 25 and 50m are reported. The step length used to integrate the system equations h is set to S and in appendix B the accuracy in predicted velocities with these step lengths are discussed. This discussion is the basis for selecting the velocity discretization. When studying controller behavior in detail, a prediction horizon of 1000m is used. When varying the prediction horizon it appears that a length of 1000m is necessary to achieve satisfactory results, see section 6.5.

The algorithm parameters that are used if nothing else is specified, are found in table 6.1. The minimum number of steps before a gear shift is allowed, k_{lim} is adjusted with the stage grid S so that $S \cdot k_{lim}$ equals 200m. Only one step is taken when using the Euler method to integrate the system equations which means that the step length h is set equal to the stage grid S .

| Parameter | Function | Value |
|-------------|--|-----------------|
| $S \cdot N$ | Horizon = Grid x Steps | 1000m |
| h | Step length when integrating | S |
| δ | Velocity discretization | 0.1 or 0.2 |
| ϵ | Accuracy when calculating a pedal level | 0.1 |
| v_{inc} | Max. increase above the reference | 5km/h |
| v_{dec} | Min. decrease below the reference | 5km/h |
| k_{lim} | Min. number of steps before a gear shift | $\frac{200}{S}$ |
| N_{min} | Lower bound on engine operating range | 1000rpm |
| N_{max} | Upper bound on engine operating range | 2000rpm |

Table 6.1: User parameters

The values of the penalization factors are not easily determined. The used values have been chosen by simulating simple road sections as a straight road and the artificially created road maps described in section 1.2 and tuning the parameters to get satisfactory behavior of the controller. When a set of values have been determined through these simple sections, the controller performance is checked on pieces of the authentic road maps available. The process is repeated until the performance is good enough. All penalization factors and their values used in simulations are stated in table 6.2.

The results and the values of the penalization factors naturally depends on the vehicle parameters used. The most important parameter is the mass of the vehicle. The vehicle mass has been chosen to 40 metric tons and the reference velocity is set to 85km/h in all simulations. The complete set of vehicle parameters and their values are to be found in appendix A.

| Factor | Penalizes | Value |
|--------|---|--|
| Q_1 | Fuel use | 2 |
| Q_2 | Negative deviation from the reference speed | 5 (if $\delta=0.1$) or 7 (if $\delta=0.2$) |
| Q_3 | Velocity changes | 15 |
| Q_4 | Gear shifts | 15 |
| Q_5 | The use of brakes | 100 |

Table 6.2: Penalization factors

6.2 Constant slope

Artificial road sections with one ascent or descent are created. The section begins with 500m straight road followed by L m with constant slope α and is finally ended with another 1500m straight road, see figure 6.1. Simulations are run on sections where the length L and slope α are varied. The results are shown in figure 6.2. In ascents ($\alpha > 0$), the effect on fuel consumption

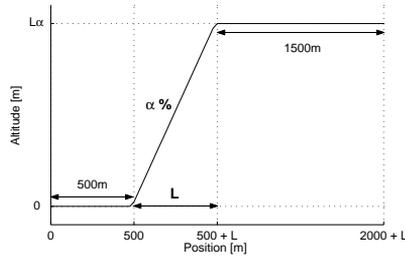


Figure 6.1: Artificial road section with one hill.

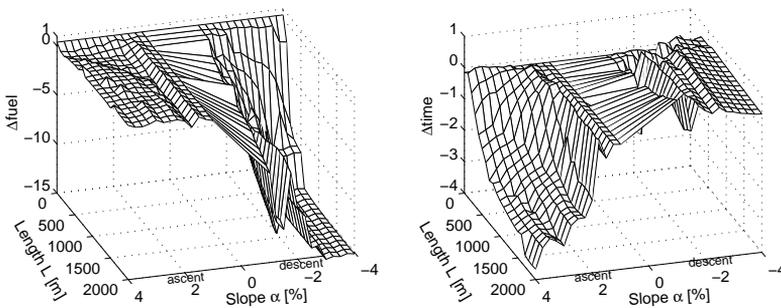


Figure 6.2: The effect on fuel consumption and travel time in ascents and descents with varying slope and length.

is small but the travel time is reduced. The mean velocity is thus increased.

The increase grows with the length of the hill L and the main cause of this in the longer hills is that the controller accelerates the vehicle before the uphill. An example of this effect in the section with $L = 500m$ and $\alpha = 3.5\%$ is shown in figure 6.3. With the extra fuel used for the acceleration before the hill it is possible to keep a higher velocity throughout the ascent and the time required on a lower gear is reduced. This behavior is primarily determined by the ratio between the penalization of fuel use (Q_1) and negative deviation from the reference velocity (Q_2) in the cost function, see (5.9) in section 5.5.

The controller only accelerate the truck before the the hill if it is steep enough. In figure 6.4 the length is still 500m as in the former example, but the slope is only 2%. The sole difference between the MPC and PI controller appear after the hill and is due to the fact that the integrator part of the PI controller is saturated (with its maximum value). This effect is also seen in the previous example in figure 6.3.

The use of fuel is noticeably reduced in descents ($\alpha < 0$) according to figure 6.2. The reduction increases with steeper and longer descents and the main cause of this is that the controller lets the velocity sink below the reference before the descent. The travel time is generally slightly increased because of the lower velocity before the downhill. If the maximum velocity is reached going downhills, the need of braking is lowered owing to the lower velocity. The relationship in the cost function, see (5.9) in section 5.5, between the penalization of fuel use (Q_1), negative deviation from the reference velocity (Q_2) and brake use (Q_5) is primarily decisive of when to lower the velocity.

Examples of descents with slope $\alpha = -3\%$ and lengths L of 300 and 500m are seen in figures 6.5 and 6.6. In the short downhill, see figure 6.5, neutral gear is used from about 200m before to 200m after the downhill slope. The gain in form of kinetic energy is evidently greater than the fuel used to run the engine on idle. If the neutral gear was disengaged in the 500m downhill, see figure 6.6, the maximum velocity would be reached earlier and increase the need for braking. In both examples the brake use is lowered compared to the PI-controller. The difference between the MPC and PI controller after the descent is owing to the fact that the integrator part of the PI controller is saturated (with its minimum value).

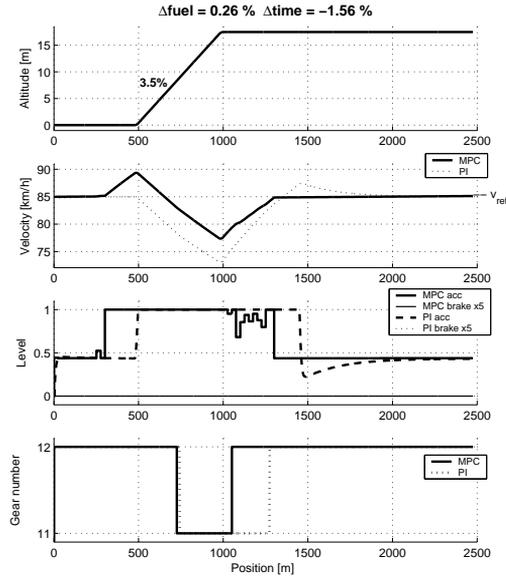


Figure 6.3: A 500m ascent with slope of 3.5%. The MPC controller accelerates the vehicle before the ascent.

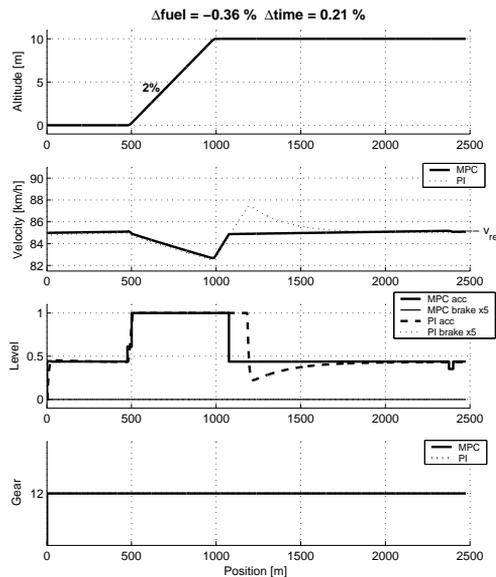


Figure 6.4: A 500m ascent with slope of 2%. The difference is due to a saturation effect in the PI controller.

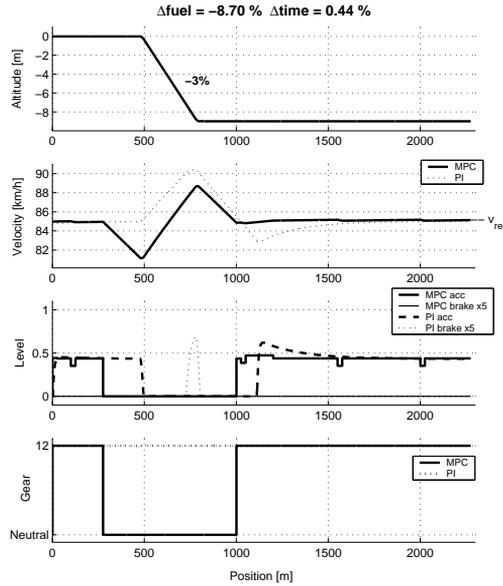


Figure 6.5: A 300m descent with slope of -3%. Fuel and brake use is lowered by the MPC controller by letting the vehicle slow down before the descent.

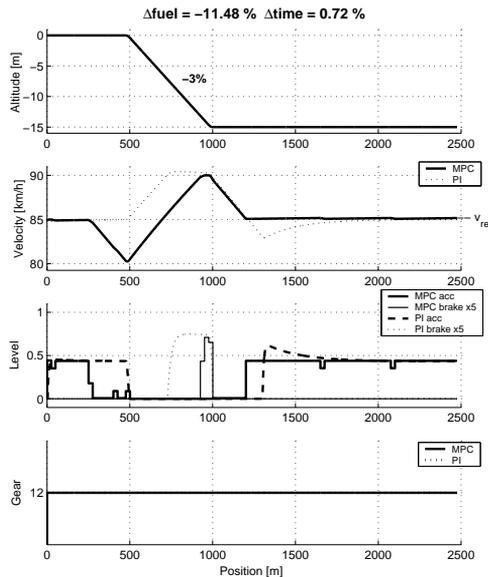


Figure 6.6: A 500m descent with slope of -3%. Fuel and brake use is lowered by the MPC controller by letting the vehicle slow down before the descent.

6.3 Crest and depression

As described in section 1.2, artificial crests and depressions are created by letting the slope vary linear with position. The crest or depression is L m long and starts with 500m and ends with 1000m straight road. If the slope increases linear with position from α to β where $\alpha < 0$ and $\beta > 0$, a depression is received. If the slope decreases linear with position from α to β where $\alpha > 0$ and $\beta < 0$, a crest is created. The absolute values of α and β are set equal and $\beta = |\alpha|$ is then varied together with the length L in order to create a set of different road sections. A positive value of β therefore means a depression and a negative value means a crest, see figure 6.7. The algorithm is applied to a set of these sections and the results are shown in figure 6.8. The fuel

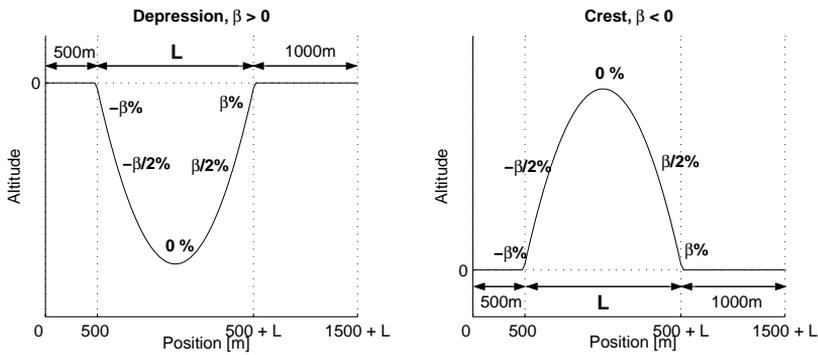


Figure 6.7: Artificial road section with a depression or a crest.

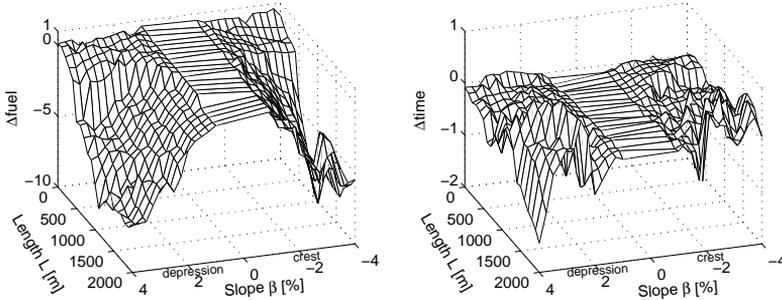


Figure 6.8: The effect on fuel consumption and travel time in crests and depressions with varying slopes and lengths.

consumption is reduced in both crests and depressions if the slope is greater than 1% or less than -1% and the length L is longer than about 1000m. The

travel time is rather moderately affected. The greatest difference occurs in long and steep depressions according to figure 6.8.

Figure 6.9 and 6.10 shows a crest and a depression, each of 1500m length. The neutral gear is used in both cases. On the top of the crest in figure 6.9, the vehicle is run on neutral gear until the road is straight. At the top the velocity sinks below the reference but as the downhill slope gets steeper the vehicle accelerates back to the reference velocity on neutral gear. In the depression, neutral gear is used from the beginning to the bottom of the depression. This allows the vehicle to accelerate faster than running on fuelcut which can be seen in figure 6.10.

The use of neutral gear is mainly governed by the relationship between the penalizing factors for fuel use (Q_1), negative deviation from the reference velocity (Q_2) and gear shifts (Q_4) in the cost function, see (5.9) in section 5.5.

In figures 6.11 and 6.12, the previous crest and depression are made twice as steep. This evidently reveals a greater potential for saving fuel.

In the the crest, figure 6.11, the controller accelerates the vehicle before the uphill begins. This leads to a higher velocity from the foot to the top of the crest. Before the top is reached, the accelerator level is decreased despite that the velocity is much below the reference. The coming downhill will accelerate the vehicle above the reference anyway and the lower velocity on the top of the crest will reduce the need for braking. The lower gear is kept throughout the crest to further reduce the need of braking. The vehicle is finally let to slow down to the reference velocity on the neutral gear. Neutral gear makes the retardation slightly smaller.

In the depression in figure 6.12, the vehicle is let to slow down before the downhill begins. This saves fuel and the need for braking is lowered later in the downhill . The controller increases the pedal level at the bottom of the depression, despite that the velocity is above the reference. This acceleration, in the MPC case, allows the truck to keep a higher velocity throughout the downhill on the same gear whereas the PI controller shifts to a lower gear when the uphill gets steep.

The acceleration before an ascent and retardation before a descent as it appear in the crest and depression in figures 6.11 and 6.12 are the same effects as described above in the case of a constant slope, see figures 6.3, 6.5 and 6.6.

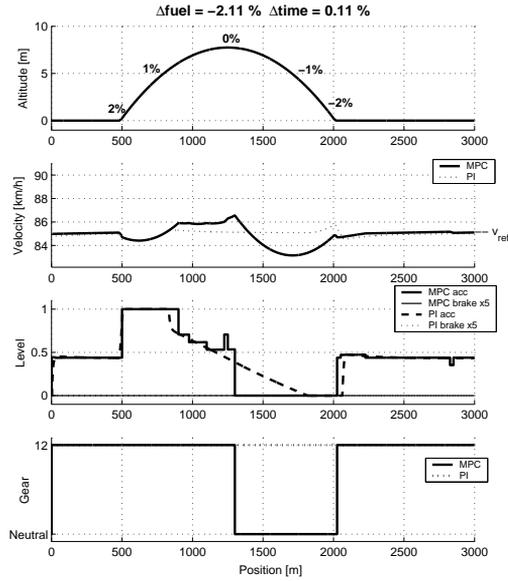


Figure 6.9: A 1500m crest. The MPC controller lets the vehicle run on neutral gear as the downhill slope gets steeper.

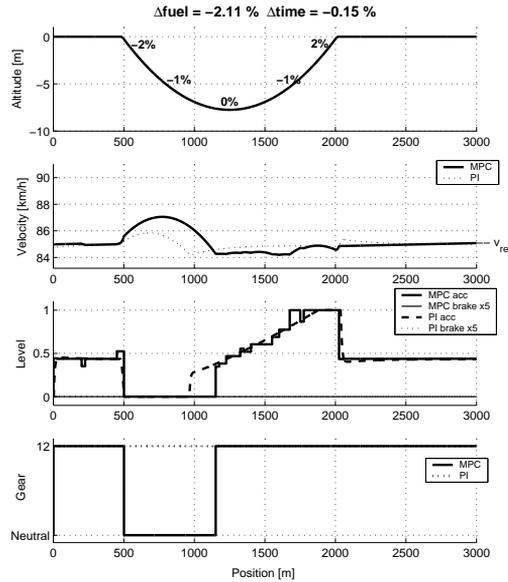


Figure 6.10: A 1500m depression. The MPC controller uses neutral gear to accelerate faster than if the twelfth gear was engaged.

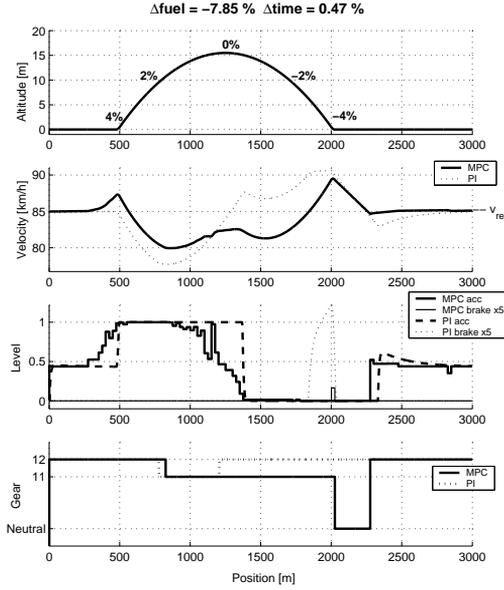


Figure 6.11: A 1500m crest. The MPC controller accelerates the vehicle before the crest and lets it slow down at the top.

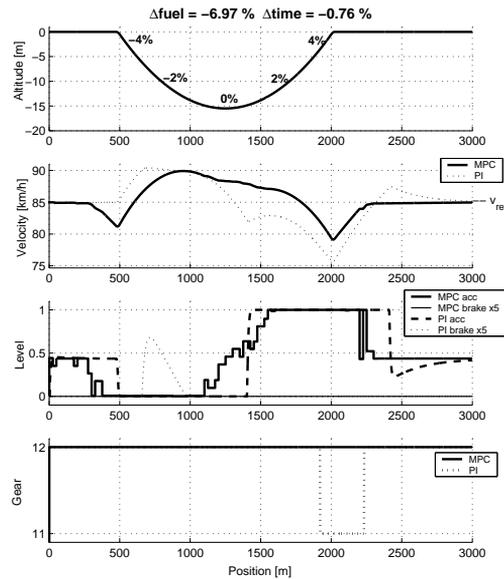


Figure 6.12: A 1500m depression. The MPC controller lets the vehicle slow down before the depression and accelerates it at the bottom.

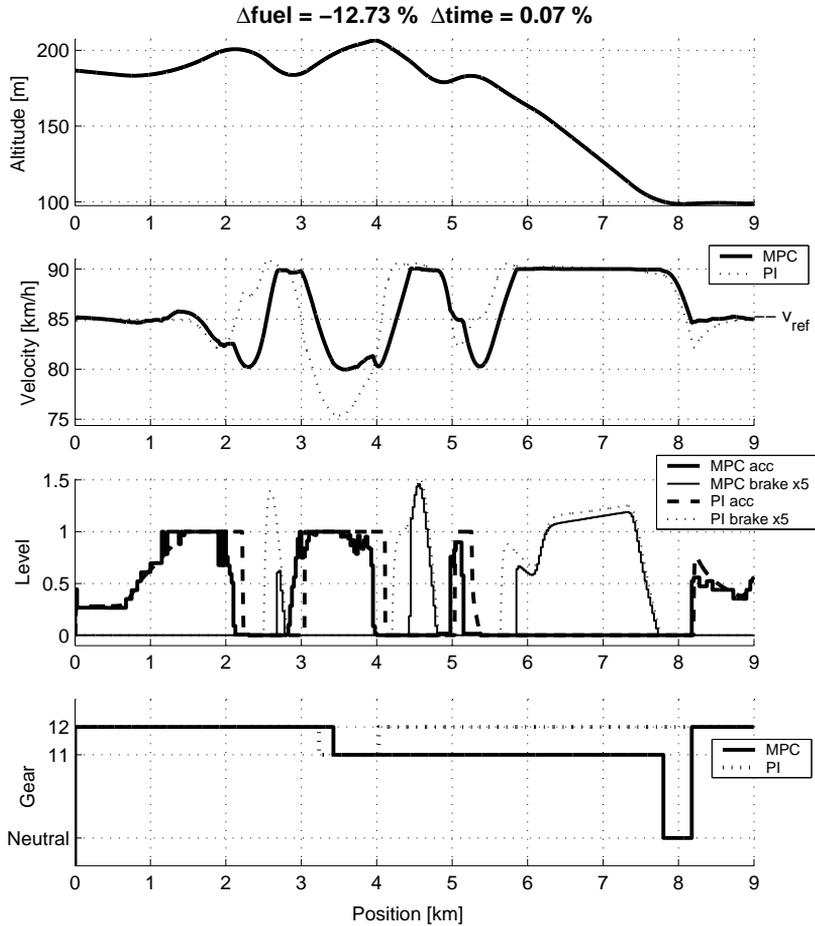


Figure 6.13: A section on the road from Linköping to Jönköping. The section begins 97km from Linköping and ends 21km from Jönköping. Acceleration prior a steep ascent is seen at 3km. Retardation prior a descent is seen at 2, 4 and 5km. A lower gear is used to reduce the load on the brake systems between about 6 and 8km. Neutral gear is finally used at about 8km to slow down to the reference velocity after the steep descent.

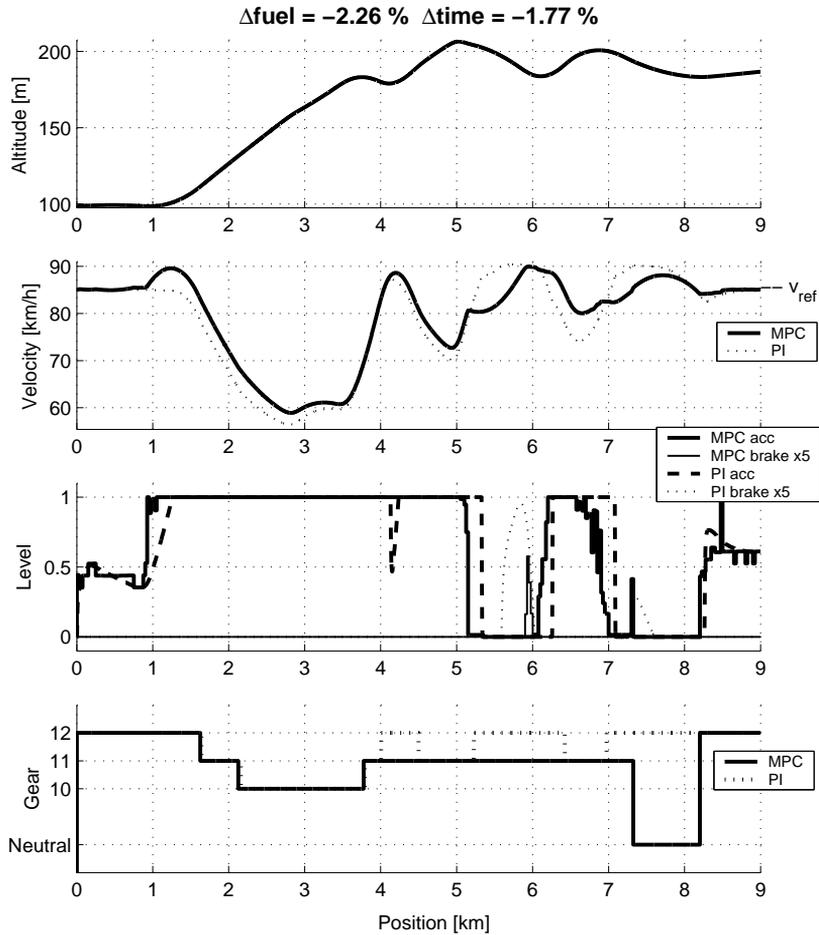


Figure 6.14: A section on the road from Jönköping to Linköping. The section begins 21km from Jönköping and ends 97km from Jönköping. Acceleration prior a steep ascent is seen at 1,4 and 6km. Retardation prior a descent is seen at 5 and 7km. A lower gear is used to reduce the load on the brake systems between about 5.5 and 6km. Between about 7 and 8km, the truck is let to accelerate on neutral gear. When the slope lessens and the truck retards, the twelfth gear is engaged when the reference velocity is reached.

6.4 Authentic road sections in detail

A portion of the road between Linköping and Jönköping is selected. Simulations are made with this section in both directions. The results are shown in figures 6.13 and 6.14. The effects discovered in the artificial road sections are also found in these simulations with authentic sections.

In figure 6.13, acceleration prior a steep ascent is seen at 3km. Retardation prior a descent is seen at 2, 4 and 5km. A lower gear is used to reduce the load on the brake systems between about 6 and 8km. Brake use is however mostly lowered by the retardation before the descent. The truck is let to slow down to the reference velocity on neutral gear after a steep descent at about 8km.

In figure 6.14, acceleration prior a steep ascent is seen at 1,4 and 6km. Retardation prior a descent is seen at 5 and 7km. A lower gear is used to reduce the load on the brake systems between about 5.5 and 6km. Brake use is however mostly lowered by the retardation before the descent. Between about 7 and 8km, neutral gear is used and the the truck accelerates. When the slope lessens and the truck retards, the twelfth gear is engaged when the reference velocity is reached.

The use of fuel is greatly reduced (-12.73%) in the first section shown in figure 6.13. The reduction is mainly made through the retardations before the downhill slopes at about 2, 4 and 5km. The acceleration prior the uphill at 3km leads to that the twelfth gear is used for about 200m longer and a higher velocity is maintained throughout the hill compared to the PI-case. The higher velocity in the uphill between 3 and 4km lessens the increase of the travel time that is a result of the retardations. The change in travel time for this section then becomes negligible (+0.07%). The magnitude of the fuel consumption reduction is of course dependent on the fact that the altitude above sea level decreases with almost 100m over the 9km in figure 6.13. However, when this section is used in the other direction, figure 6.14 shows that a decent reduction of the fuel consumption still can be achieved (-2.26%).

Figure 6.14 shows the section where the altitude increases with about 100m over 9km. Fuel is primarily saved through the retardations before the downhill slopes at 5 and 7km. In this case, the accelerations prior ascents evidently increases the mean velocity more than the retardations prior descents decreases it, resulting in a travel time reduction (-1.77%).

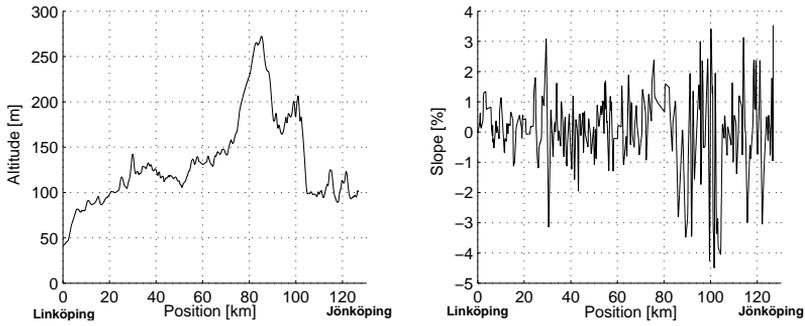
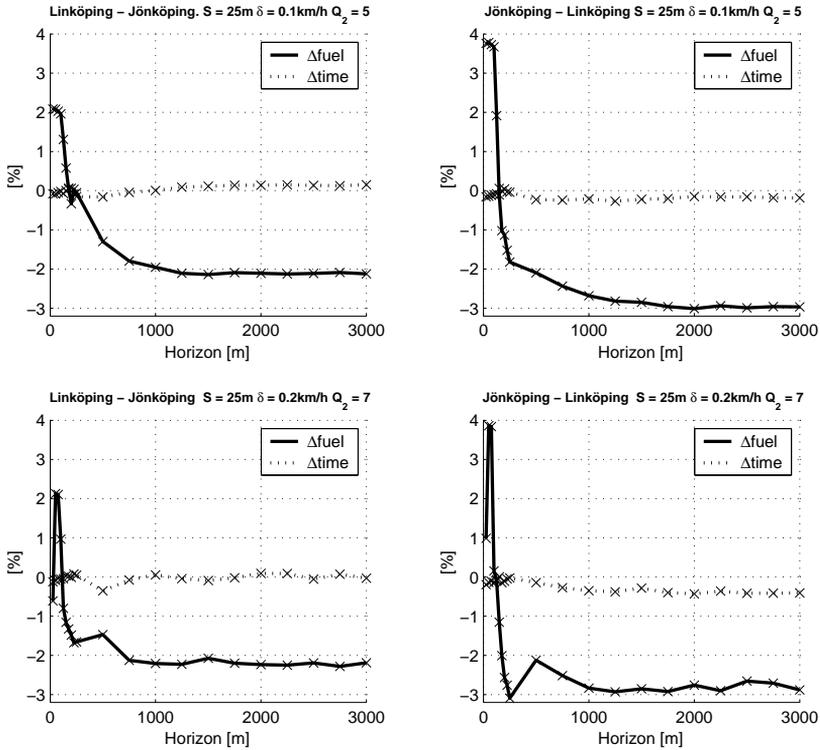


Figure 6.15: The road between Linköping and Jönköping

Figure 6.16: The effect on fuel consumption and travel time on the road Linköping to Jönköping. The stage grid S is 25m and the velocity discretization δ is 0.1 and 0.2km/h.

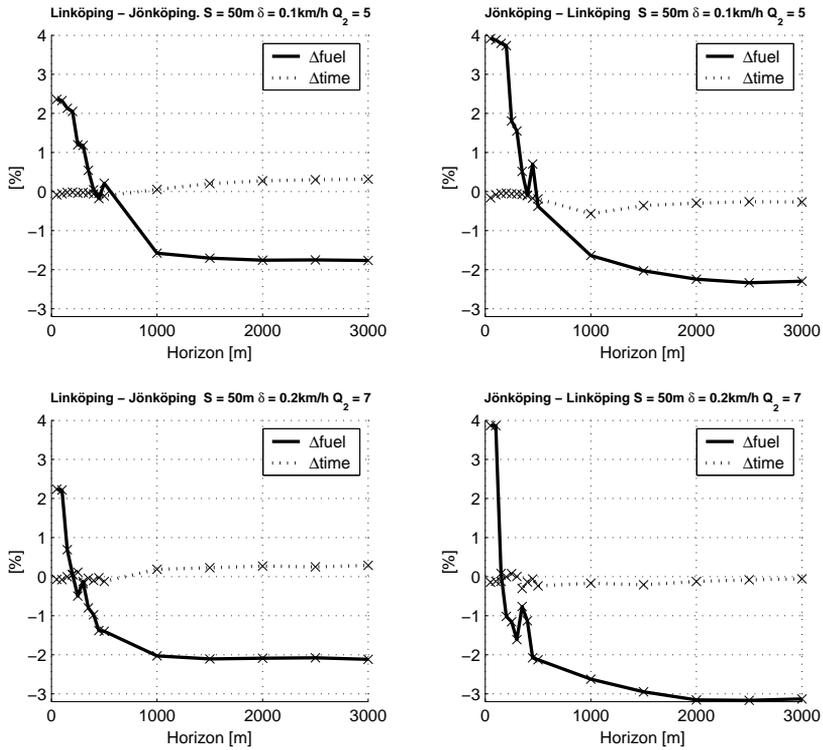


Figure 6.17: The effect on fuel consumption and travel time on the road Linköping to Jönköping. The stage grid S is 50m and the velocity discretization δ is 0.1 and 0.2km/h.

6.5 From Linköping to Jönköping and back

The altitudes above sea level and the road slopes are shown in figure 6.15. The entire distance is about 127km. The slope varies between about -4% and +3%. Simulations are made with the road in both directions. The prediction horizon is varied between 25m and 3000m. Stage grids S of 25 and 50m are used in conjunction with velocity discretizations δ of 0.1 and 0.2 km/h.

The effects on the fuel consumption and the travel time when $S = 25m$ and δ equals 0.1 and 0.2km/h are shown in figure 6.16. It is seen that there are only small differences between $\delta = 0.1$ and $\delta = 0.2$ when the horizon is more than 1000m.

The fuel consumption is at best reduced with about 2% in the direction toward Jönköping and about 3% toward Linköping. The altitude above sea level is around 60m higher in Jönköping than in Linköping, see figure 6.15. Going toward Jönköping thus in general means facing more uphill than downhill slope, which may explain most of this difference.

The travel time is moderately affected and the magnitude does not vary much with the horizon according to figure 6.16. The fuel use is however clearly dependent on the horizon. A horizon longer than 2000m seems absolutely superfluous and good results are achieved with about 1000m, at least for this road configuration.

The effects on the fuel consumption and the travel time when $S = 50m$ and δ equals 0.1 and 0.2km/h are shown in figure 6.17. The solutions with $\delta = 0.1$ are poorer with at least half a percent, with equal regard to fuel use and travel time, than when $\delta = 0.2$. It may be unexpected that the higher accuracy gives somewhat poorer performance. This indicates that a smaller δ will not always give better solutions. In appendix B the mean error in predicted velocities with $h = S = 50m$ and $N=20(1000m)$ on the road Linköping to Jönköping are estimated to be about 0.1km/h which is the size of the smaller δ . This may suggest that it is not beneficial to use a discretization too close to the mean error.

6.6 Neutral gear

The use of neutral gear adds another degree of freedom in the problem. To estimate the magnitude of the gain that is achieved through this, simulations are made where neutral gear is disallowed. The stage grid S is 25m, the velocity discretization δ is 0.2km/h and the penalization factor Q_2 is set to 7. A comparison between figures 6.16 and 6.18 reveals the effects. Traveling toward Jönköping, the fuel consumption reduction is only lessened by about half a percent. The travel time is however increased half a percent. This increase does of course contribute to the relatively small change in the fuel consumption. If fuel use and travel time are given the same weight, the solutions become about two times poorer compared to when neutral gear was

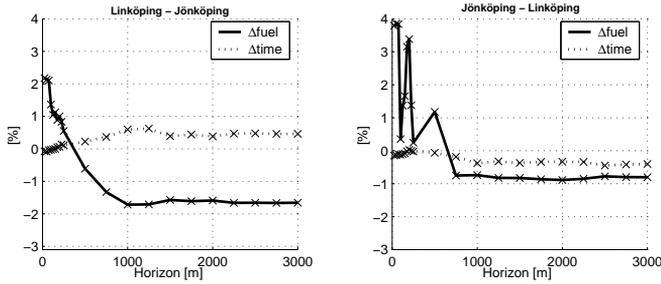


Figure 6.18: The effect on fuel use and travel time on the road Linköping to Jönköping. Use of neutral gear is disallowed. $S = 25$, $\delta = 0.2$ and $Q_2 = 7$.

used

The most evident difference appears in the direction toward Linköping. The change in travel time is similar but the fuel use is reduced about three times more when neutral gear is allowed than when it is disallowed. The altitude above sea level is around 60m higher in Jönköping than in Linköping, see figure 6.15. Neutral gear is as expected most useful when there is more downhill than uphill slope.

6.7 Complexity

The ratio q tells the required computer time to simulate the system one second. A ratio of one means that the simulation precisely can be run real time. A ratio greater than one means that the simulation demands more time than the actual time simulated and conversely.

In order to assess the computational complexity and to validate the approximation in (5.11) the ratio q is determined for varying step length N for a stage grid of 25 and 50m and velocity discretizations of 0.1 and 0.2km/h. The results are shown in figure 6.19. The test track Linköping to Jönköping is used in these simulations. The linear dependence of the horizon length appears clearly in all figures. With $S = 25$, $\delta = 0.1$ (upper left figure) a horizon of about 750m allows the simulation precisely to be run in real time. According to (5.12) the same computational complexity is achieved with a horizon four times as long if the discretization δ is doubled. With $S = 25$, $\delta = 0.2$ the simulation should thus run in real time up to a horizon length of about $750 \cdot 4 = 3000\text{m}$ which the upper right figure confirms. Doubling the stage grid S will lead to half as many algorithm runs on the same distance. The same horizon length is further achieved with half as many steps which halves the complexity. In total it should thus be four times as demanding using $S = 25\text{m}$ than $S = 50\text{m}$ which is confirmed by comparing the upper and lower figures.

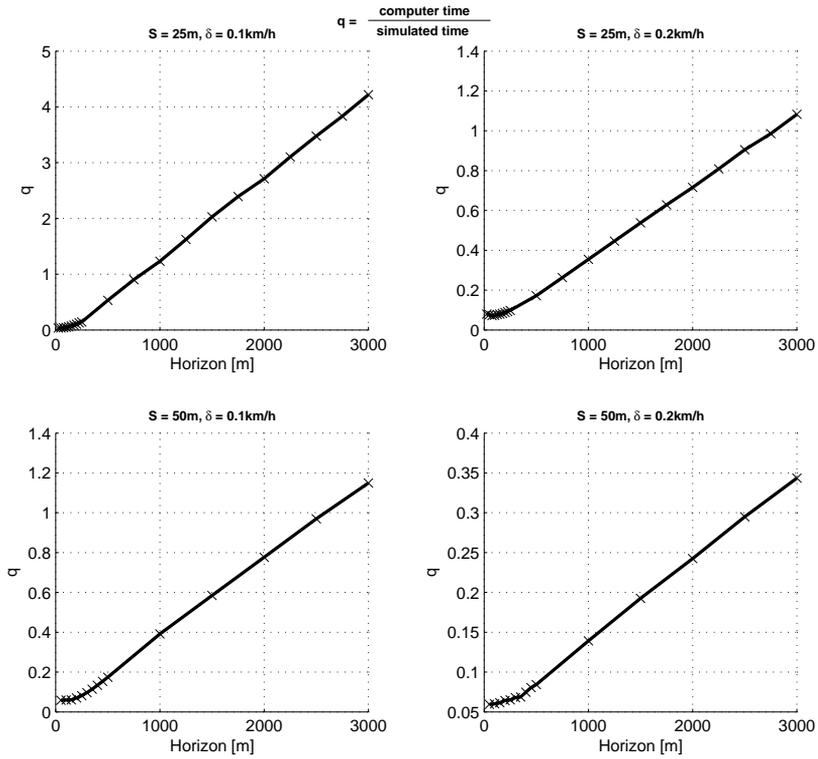


Figure 6.19: The ratio q tells the required computer time to simulate the system one second. Simulations are made on the road Linköping to Jönköping.

Chapter 7

Conclusions

The scenario in this thesis has been a heavy truck traveling on a road with a constant reference velocity and where small deviations from this reference are allowed. The aim was to reduce the fuel consumption with almost no increase in the travel time. The fuel use can effortlessly be lowered by a lowered mean velocity but the travel time will then obviously increase. In simulations a 5km/h positive and negative deviation from the reference velocity is allowed. Simulations on the test track Linköping to Jönköping and back with a load of 40 metric tons reduces the fuel consumption with 2.5%, 2% in the direction toward Jönköping and 3% going back toward Linköping, whereas the travel time only is insignificantly changed. These results are achieved provided that more than 1500m of the road ahead is taken into consideration (see figures 6.16 and 6.17). The reductions are achieved through rather intuitive actions.

Simulations show that the greatest fuel use reductions are made when a downhill with sufficient slope is ahead. The vehicle velocity is lowered before the downhill and the truck is then let to accelerate in the descent (see figure 6.6). Slowing down before a steep downhill will further in general lower the need for braking.

When there is a steep uphill ahead, it may be favorable to accelerate before the hill. A higher velocity reaching the uphill can lessen the need for lower gears (see figure 6.3). This action does not decrease the use of fuel of noticeable amounts, it may even slightly increase it. Simulations show that the travel time is, by the higher mean velocity, shortened of a greater magnitude than the change in fuel consumption. Considering a route, the time decrease will however counterbalance the time increases introduced where the vehicle velocity is lowered prior a descent.

The neutral gear can finally be used in order to gain kinetic energy from the torque otherwise used to pull the engine. On artificial road sections simulations shows that the fuel consumption can be fairly decreased whereas the travel time only is slightly affected (see figures 6.9 and 6.10). On a route with more downhill than uphill slope, the use of neutral gear seems to be the

most potent way of reducing fuel use. When going back this route and mostly going uphill, a clever choice of the pedal level appear to be just as important (compare figures 6.16 and 6.18). On the test track Linköping to Jönköping the solutions became two to three times better, with equal regard to fuel use and travel time, with the possibility of using neutral gear.

The limited range of allowed velocities results in the rather attractive property that the computational complexity of the algorithm is linear in the horizon length. The complexity however increases with the square of the number of possible states in a stage. Horizons of more than 1000m produce the best results amongst the simulations made. With a stage grid of 50m and a velocity discretization of 0.2km/h results close to the best ones is achieved with a horizon of 2000m (see figure 6.17). This simulation runs four times as fast as real time which means that simulating one seconds requires one fourth of a second of computer time (see figure 6.19). A stage grid of 25m and the same discretization gives about the same result with a horizon of 1500m (see figure 6.16). These simulation runs about twice as fast as real time.

Chapter 8

Future work and extensions

Any controller that make use of topographic information is dependent on the road maps being reasonably accurate and that the vehicle position is robustly determined. This is obviously of importance and a controller's robustness against the inevitable errors in the information must be investigated. In the case of a MPC controller, the performance further relies heavily on the model which is used.

The repeated computation in the control algorithm is rather demanding. This is a drawback which is not easy to circumvent. In the following approaches to this problem are presented.

An much simpler and more ad hoc algorithm could be developed by taking the discovered actions as starting points. At regular intervals calculations could be made deciding if it is favorable to slow down, accelerate or using neutral gear. The ordinary cruise controller is used if none of these actions are rated as beneficial. The calculations should probably take at least 1000m of the road ahead into account according to the results in chapter 6.

If a simple logic is established that determines which gear to use in a state and the use of neutral gear is disallowed, the complexity could be reduced up to sixteen times (set $k_g = 4$ and $m = -3$ in (5.13)). In case this logic resembles an existing automatic gear selection system, the MPC algorithm could for example be used on-line to generate reference trajectories that can be fed to a feedbacked cruise controller.

References

- Bansal, V., Sakizlis, V., Ross, R., Perkins, J., and Pistikopoulos, E. (2003). New algorithms for mixed-integer dynamic optimization. *Computers and Chemical Engineering*, 27(5):647–668.
- Bellman, R. (1957). *Dynamic Programming*. Princeton University Press, Princeton, N.J.
- Bellman, R. and Dreyfus, S. (1962). *Applied Dynamic Programming*. Princeton University Press, Princeton, N.J.
- Bemporad, A. and Morari, M. (1999). Control of systems integrating logic, dynamics and constraints. *Automatica*, 35(3):407–427.
- Bensoussan, A. and Menaldi, J. (1997). Hybrid control and dynamic programming. *Dynamics of Continuous, Discrete and Impulsive Systems*, 3:395–442.
- Bertsekas, D. (1995). *Dynamic Programming and Optimal Control*, volume I. Athena Scientific, Belmont, Massachusetts.
- Biegler, L. and Grossmann, I. (2004a). Part ii. future perspective on optimization. *Computers and Chemical Engineering*, 28(8):1193–1218.
- Biegler, L. and Grossmann, I. (2004b). Retrospective on optimization. *Computers and Chemical Engineering*, 28(8):1169–1192.
- Branicky, M. and Mitter, S. (1995). Algorithms for optimal hybrid control. In *Proceedings of the 34th Conference on Decision and Control*, New Orleans, LA USA.
- Camacho, E. and Bordons, C. (2004). *Model Predictive Control*. Springer, London.
- Denardo, E. (1982). *Dynamic Programming: models and applications*. Prentice-Hall, Englewood Cliffs, N.J.
- Eldén, L. and Wittmeyer-Koch, L. (1996). *Numerisk analys*. Studentlitteratur, Lund.

- Eriksson, L., Fredriksson, J., Fröberg, A., Krus, P., Larsson, J., Nielsen, L., and Sjöberg, J. (2004). Center for automotive propulsion simulation. <http://www.capsim.se/>.
- Hedlund, S. (2003). *Computational Methods for Optimal Control of Hybrid Systems*. Phd thesis, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- Heemels, W., Schutter, N. D., and Bemporad, A. (2001). Equivalence of hybrid dynamical models. *Automatica*, 37(7):1085–1091.
- Ljung, L. and Glad, T. (2003). *Reglerteori*. Studentlitteratur, Lund.
- Ljung, L. and Glad, T. (2004). *Modellbygge och simulering*. Studentlitteratur, Lund.
- Nielsen, L. and Eriksson, L. (2003). *Course material Vehicular Systems*. Linköping Institute of Technology, Vehicular Systems, ISY, Linköping.
- Nielsen, L. and Kiencke, U. (2000). *Automotive control systems*. Springer, Berlin.
- Peña, M., Camacho, E., and Piñón, S. (2003). Hybrid systems for solving model predictive control of piecewise affine systems. In *Proceedings IFAC Conference on Analysis and Design of Hybrid Systems ADHS'03*, pages 76–81, Saint-Malo, France.
- Piccoli, B. (1999). Necessary conditions for hybrid optimization. In *Proceedings of the 38th IEEE Conference on Decision and Control*, volume 1, pages 410–415, Phoenix, Arizona USA.
- Sussmann, H. (1999). A maximum principle for hybrid optimal control problems. In *Proceedings of the 38th IEEE Conference on Decision and Control*, volume 1, pages 425–430, Phoenix, Arizona USA.

| Parameter | Value | Unit |
|--|-----------------------|-----------------------------------|
| Engine | | |
| Inertia J_e | 3.5 | kgm^2 |
| Number of cylinders n_{cyl} | 6 | - |
| Number of rev. per stroke n_r | 2 | - |
| Engine map \hat{T}_{map} | | |
| a_e | $-3.30 \cdot 10^{-3}$ | $\frac{Nm}{rpm}$ |
| b_e | 9.16 | $\frac{Nm \text{ stroke}}{mg}$ |
| c_e | -82.53 | Nm |
| Fuel flow limit $\hat{\delta}_{max}$ | | |
| a_δ | $-1.59 \cdot 10^{-4}$ | $\frac{mg}{rpm^2 \text{ stroke}}$ |
| b_δ | 0.42 | $\frac{mg}{rpm \text{ stroke}}$ |
| c_δ | -57.05 | $\frac{mg}{\text{stroke}}$ |
| Drag torque \hat{T}_{drag} | | |
| a_d | $-8.80 \cdot 10^{-2}$ | $\frac{Nm}{rpm}$ |
| b_d | -51.51 | Nm |
| Idle values | | |
| Engine idle speed N_{idle} | 600 | rpm |
| Idle fueling δ_{idle} | 10.44 | $\frac{mg}{\text{stroke}}$ |
| Final drive | | |
| Ratio i_f | 3.27 | - |
| Efficiency n_f | 0.97 | - |
| Longitudinal forces | | |
| Wheel inertia J_w | 32.9 | kgm^2 |
| Rolling resistance coefficient c_r | $7 \cdot 10^{-3}$ | - |
| Gravity constant g | 9.81 | $\frac{m}{s^2}$ |
| Air drag coefficient c_w | 0.6 | - |
| Max. cross section area A_a | 10 | m^2 |
| Air density ρ_a | 1.29 | $\frac{kg}{m^3}$ |
| Wheel radius r_w | 0.52 | m |
| Brakes | | |
| Maximum torque k_b | $20 \cdot 10^3$ | Nm |

Table A.1: Vehicle parameters

The ratios i_t and efficiencies η_t for the gearbox are stated in table A.2. The ratios are chosen on the pattern of the Scania GRS900R gearbox. The efficiencies are however only made up and do not come from measurements. These have been chosen so that the efficiencies increases as the ratio decreases.

Gear selection is made as described in section 2.5. The thresholds are shown in table A.3. The cruise controller parameters are finally presented in table A.4.

| Gear | Ratio | Efficiency | Gear | Ratio | Efficiency |
|------|-------|------------|------|-------|------------|
| 1 | 11.27 | 0.88 | 7 | 3.01 | 0.94 |
| 2 | 9.14 | 0.89 | 8 | 2.44 | 0.95 |
| 3 | 7.17 | 0.90 | 9 | 1.91 | 0.96 |
| 4 | 5.81 | 0.91 | 10 | 1.55 | 0.97 |
| 5 | 4.62 | 0.92 | 11 | 1.23 | 0.98 |
| 6 | 3.75 | 0.93 | 12 | 1 | 0.99 |

Table A.2: Gearbox parameters

| Gear | Down | Up | Gear | Down | Up |
|------|------|------|------|------|------|
| 1 | - | 1800 | 7 | 1300 | 1800 |
| 2 | 1300 | 1800 | 8 | 1300 | 1800 |
| 3 | 1300 | 1800 | 9 | 1300 | 1800 |
| 4 | 1300 | 1800 | 10 | 1300 | 1800 |
| 5 | 1300 | 1800 | 11 | 1300 | 1700 |
| 6 | 1300 | 1800 | 12 | 1300 | - |

Table A.3: Engine speed thresholds for gear shifting [rpm].

| | |
|-----------------------------|---------------|
| Pedal controller | |
| Proportional constant K_P | 1.845 |
| Integrator constant I_P | 0.2 |
| Saturation value | 30 |
| Brake controller | |
| Proportional constant K_B | 0.3 |
| Integrator constant I_B | 0.02 |
| Saturation value | 30 |
| Threshold v_{max} [km/h] | $v_{ref} + 5$ |

Table A.4: PI controller parameters

Appendix B

Calculations

In this appendix various calculations are performed. The vehicle model is first given a more compact description by the introduction of a set of parameters. It is further shown how the pedal and brake levels used in the control algorithm are computed. Finally, the numerical stability and accuracy of Euler's method applied to the vehicle model is investigated.

B.1 Vehicle model

The complete driveline model is stated in (2.10). With the identity

$$c_r \cos \alpha + \sin \alpha = \sqrt{1 + c_r^2} \sin(\alpha + \arctan c_r)$$

the equation becomes

$$\dot{v} = \frac{r_w}{J_w + mr_w^2 + \eta_f i_f^2 \eta_t i_t^2 J_e} \left(\eta_t i_t \eta_f i_f T(v, P, G) - k_b B - \frac{1}{2} c_w A_a \rho_a r_w v^2 - mgr_w \sqrt{1 + c_r^2} \sin(\alpha + \arctan c_r) \right).$$

Introduce the parameters

$$\begin{aligned}
 c_1 &= \frac{r_w}{J_w + mr_w^2 + \eta_t i_t^2 \eta_f i_f^2 J_e} \\
 c_2 &= \frac{1}{2} c_w A_a \rho_a r_w \\
 c_3 &= \eta_t i_t \eta_f i_f \\
 c_4 &= \frac{30 i_t i_f}{\pi r_w} \\
 c_5 &= \frac{1}{6 \cdot 10^4} \frac{n_{cyl}}{n_r} \\
 c_6 &= mgr_w \sqrt{1 + c_r^2} \\
 c_7 &= \arctan c_r
 \end{aligned}$$

and the driveline model can be written as

$$\dot{v} = c_1 (c_3 T(v, P, G) - k_b B - c_2 v^2 - c_6 \sin(\alpha + c_7)) \quad (\text{B.1})$$

The engine speed is

$$N = \begin{cases} c_4 v & G \neq 0 \\ N_{idle} & G = 0 \end{cases}$$

The mass flow of fuel in (2.13) becomes

$$\dot{m}_f(N, P, G) = \begin{cases} c_5 N P \hat{\delta}_{max}(N) & G \neq 0 \\ c_5 N_{idle} \delta_{idle} & G = 0 \end{cases} \quad (\text{B.2})$$

The engine torque T is approximated according to (2.8) and the upper fueling bound δ_{max} according to (2.6). The equations are repeated below for convenience.

$$\begin{aligned}
 \hat{T}_e(N, P, G) &= \begin{cases} a_e N + b_e P \hat{\delta}_{max}(N) + c_e & P > 0, G \neq 0 \\ a_d N + b_d & P \leq 0, G \neq 0 \\ 0 & G = 0 \end{cases} \\
 \hat{\delta}_{max}(N) &= a_\delta N^2 + b_\delta N + c_\delta
 \end{aligned}$$

B.2 Pedal and brake level

The problem faced is to calculate a pedal or brake level that transforms the system from a velocity v_0 to another velocity v_1 using the gear g_1 when the road slope is α . We assume that another gear than neutral is active ($G \neq 0$).

The problem is represented according to (5.4) with $M = 1$, which means that the step length h used for integration is set equal to the stage grid S . The equation then becomes

$$v_1 = v_0 + \frac{h}{v_0} f_1(v_0, \underline{u}, \alpha) \quad v_k > 0 \quad \forall k \quad (\text{B.3})$$

where $\underline{u} = [P \ B \ G]^T$ and $f_1 = \dot{v}$ using (B.1).

When calculating a pedal level P it is assumed that $P > 0$ which means that the engine torque is

$$\hat{T}_e(N, P, G) = a_e N + b_e P \hat{\delta}_{max}(N) + c_e \quad (\text{B.4})$$

If then

$$-\epsilon \leq P \leq 1 + \epsilon, \quad \epsilon > 0$$

holds, a feasible pedal level is found. The parameter ϵ will thus determine the values of the pedal level that are treated as minimum ($P = 0$) and maximum fueling ($P = 1$) respectively. The brake level B is assumed to be zero. Insert (B.4) in (B.3) with $N = c_4 v$ and $\underline{u} = [P \ 0 \ G]^T$. Solving for P yields

$$P = \frac{-c_1 c_3 c_e h - a_e c_1 c_3 c_4 h v_0 - v_0^2 + c_1 c_2 h v_0^2 + v_0 v_1 + c_1 c_6 h \sin(\alpha + c_7)}{b_e c_1 c_3 h (c_\delta + b_\delta c_4 v_0 + c_4^2 a_\delta v_0^2)} \quad (\text{B.5})$$

When calculating a brake level B it is assumed that $P = 0$ which means that the engine torque is the drag torque

$$\hat{T}_e(N, P, C) = a_d N + b_d \quad (\text{B.6})$$

If then

$$0 \leq B \leq 1$$

holds, a feasible brake level is found. Insert (B.6) in (B.3) with $N = c_4 v$ and $\underline{u} = [0 \ B \ G]^T$. Solving for B yields

$$B = \frac{b_d c_1 c_3 h + a_d c_1 c_3 c_4 h v_0 + v_0^2 - c_1 c_2 h v_0^2 - v_0 v_1 - c_1 c_6 h \sin(\alpha + c_7)}{c_1 k_b h} \quad (\text{B.7})$$

B.3 Numerical stability of Euler's method

The system equations are integrated with Euler's method, see (5.4) and (5.5). The velocity function is linearized and then stability will be investigated using the linearized system. The fuel flow function (B.2) is only a polynomial of the velocity. If the calculation of the velocity is stable, the computation of the flow will also be stable.

B.3.1 Linearization

The rewrite (5.3) is used to replace the time dependence with position dependence. With (B.1) the system becomes

$$f(v, \underline{u}, \alpha) = \frac{dv}{ds} = \frac{c_1}{v} (c_3 T(v, P, G) - k_b B - c_2 v^2 - c_6 \sin(\alpha + c_7))$$

The system can be linearized around a singular point where $f(v_0, \underline{u}_0, \alpha_0) = 0$ and written on the form

$$\dot{z} = az + Bw, \quad z = v - v_0, \quad w = \underline{u} - \underline{u}_0$$

where $a = \left. \frac{\partial f}{\partial v} \right|_{v_0, \underline{u}_0}$ and the the row vector B are given by $b_i = \left. \frac{\partial f}{\partial u_i} \right|_{v_0, \underline{u}_0}$

(Ljung and Glad, 2003).

The eigenvalue of the system matrix is a and it is interesting when investigating system properties. In tables B.1 and B.2, the eigenvalues for three different stationary points and two vehicle masses are shown. The vehicle

| v_0 | P_0 | B_0 | α_0 | λ |
|-------|-------|--------|------------|------------------------|
| 85 | 0.443 | 0 | 0.00% | $-0.238 \cdot 10^{-3}$ |
| 90 | 0 | 0.0490 | -4.00% | $-0.224 \cdot 10^{-3}$ |
| 85 | 0 | 0 | -1.25% | $-0.192 \cdot 10^{-3}$ |

Table B.1: Eigenvalues λ at different stationary points. The velocity v_0 is in km/h. The mass is 40 metric tons.

| v_0 | P_0 | B_0 | α_0 | λ |
|-------|-------|--------|------------|------------------------|
| 85 | 0.331 | 0 | 0.00% | $-0.451 \cdot 10^{-3}$ |
| 90 | 0 | 0.0153 | -4.00% | $-0.445 \cdot 10^{-3}$ |
| 85 | 0 | 0 | -1.80% | $-0.382 \cdot 10^{-3}$ |

Table B.2: Eigenvalues λ at different stationary points. The velocity v_0 is in km/h. The mass is 20 metric tons.

parameters besides the mass are found in appendix A.

B.3.2 Test problem

To study the stability for the numerical method, the following test problem can be used (Eldén and Wittmeyer-Koch, 1996; Ljung and Glad, 2004)

$$\dot{x} = \lambda x, \quad x(0) = x_0$$

Applying Euler's method yields

$$x_{n+1} = x_n + h\lambda x_n = (1 + h\lambda)x_n$$

The solution to the difference equation is

$$x_n = x_0(1 + h\lambda)^n$$

The number sequence must be decreasing to regard the method as stable. It is easy to see that the solution above is decreasing if

$$|1 + h\lambda| < 1 \Leftrightarrow -2 < h\lambda < 0$$

holds. A stable differential equation has $\lambda < 0$ and because of $h > 0$, the step length must be chosen as

$$h < \frac{2}{|\lambda|}$$

to receive a decreasing solution.

B.3.3 Conclusions

The reasoning about the test problem can directly be used on the linearized system. The absolute values of the eigenvalues for the different points are all less than $\frac{1}{2}10^{-3}$, see tables B.1 and B.2. This indicates a rather slow system. Comparing the two tables also show that the eigenvalues roughly double when the mass halves. The Euler method is thus stable for step lengths more than $\frac{2}{\frac{1}{2}10^{-3}} = 4000m$. Such long step lengths are not reasonable considering the truncation errors. The global truncation error in Euler's method is, as well known, $O(h)$ (Eldén and Wittmeyer-Koch, 1996). Accuracy is thus a greater concern when choosing the step length. See the following section for a discussion about the truncation error.

B.4 Accuracy in Euler's method

The error sources when integrating differential equations numerically are truncation and rounding errors.

A floating point representation with base β and t fraction numbers leads to a relative rounding error less than $\frac{1}{2}\beta^{-t}$ (Eldén and Wittmeyer-Koch, 1996). MatLab on a PC uses binary representation $\beta = 2$ and double precision (64 bits) $t = 52$ according to the IEEE floating point standard. The relative rounding error is thus less than

$$\frac{1}{2}2^{-52} \approx 1.1 \cdot 10^{-16}$$

The pedal and brake levels that are used in the algorithm presented in chapter 5 are calculated as described in detail in section B.2 above. The system equations are solved with aid of Euler's method with a step length h equal to the stage grid S . Levels are further kept constant during S m. If the relatively small rounding errors are neglected the errors introduced by these calculations come from truncation errors in Euler's method. To investigate the magnitude of these errors computed pedal and brake levels are used for open-loop control. First are the levels calculated according to (B.5) and (B.7). The levels are then applied to the system equations which are solved with a higher accuracy than the accuracy used in the calculation. The predicted velocity which was used in the level calculation can then be compared to the velocity obtained when solving the system with the higher accuracy. This is done in the following using the road Linköping to Jönköping and comparing predicted velocities to actual velocities obtained when applying Euler's method with a step length of $h = 2m$. A step of 2m corresponds to a sample time of $\frac{2}{85/3.6} \approx 0.08s$ when traveling in 85km/h. In the calculations the aim has been to calculate a pedal level that maintains the reference velocity (85km/h) or a brake level such that the maximum velocity (90km/h) is kept. Gear selection is made based upon current engine speed as described in section 2.5.

In figure B.1 the errors made in one step are shown for different step lengths h . The errors are all rather small. The greatest errors occur when the altitude is changing quickly. When the slope lies between about -2 and 2% the errors are as smallest.

It is also interesting to compare the errors using different step lengths h and a number of steps N such that $h \cdot N$ equals the same length. In figure B.2 the errors made when $h \cdot N$ equals 1000m are shown.

B.4.1 Conclusions

The errors made with Euler's method are naturally decreased when shorter step lengths are used. The magnitude of the errors on one road section can be seen in figure B.1. It is however also worthwhile to compare step lengths when predicting the velocity over a fixed total length as in figure B.2.

According to figure B.2 the maximal error of using N steps of hm roughly increases with 50% when h doubles and N halves. The available measurements give one slope value each 25m. Using a step length greater than 25m will skip some slope measurements whereas a step length less than or equal to 25m will ensure that every measurement is taken into account. Longer step lengths will therefore render into additional errors. With $h = 50m$ it seems however that this effect is negligible owing to the fact that the maximal errors do not noticeably increase with another proportion compared to the step lengths 10 and 25m. The main source of error is then still the inaccuracy in Euler's method.

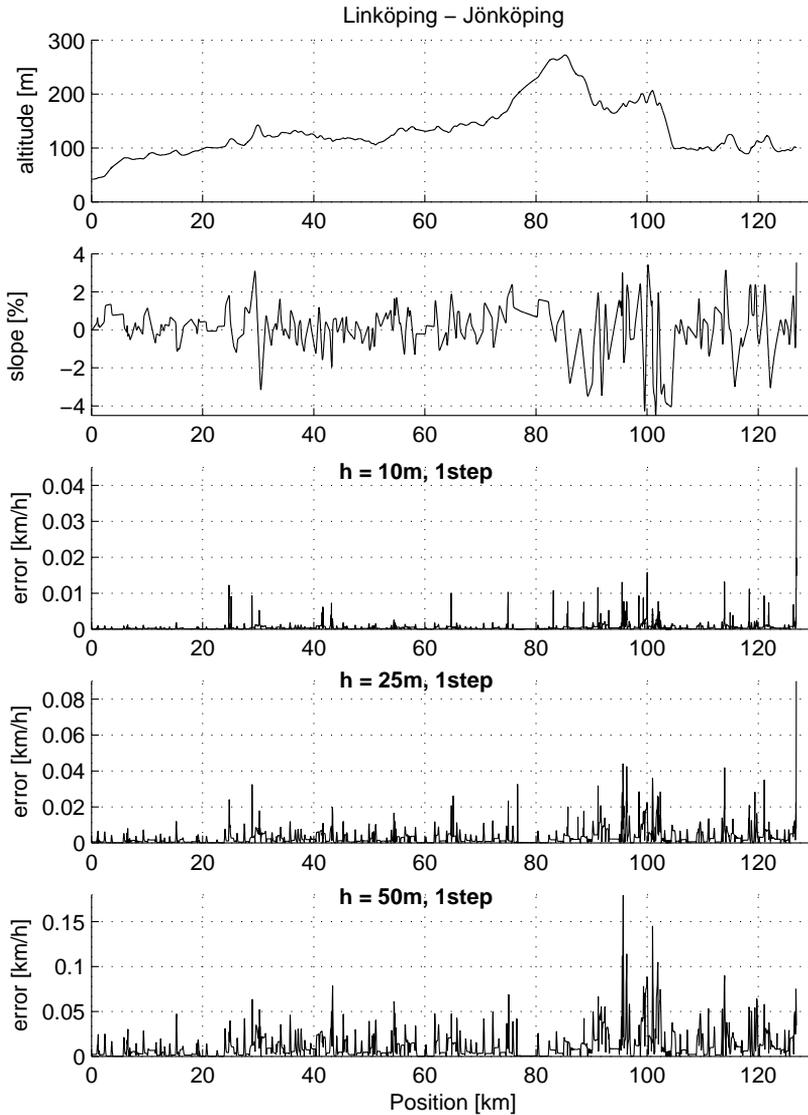


Figure B.1: The error made in one step of length h when using calculated pedal and brake levels which are constant during the length h . A pedal or brake level is repeatedly computed and applied to the system during the length h . The predicted end velocity is then compared to a more accurate simulated velocity of the vehicle.

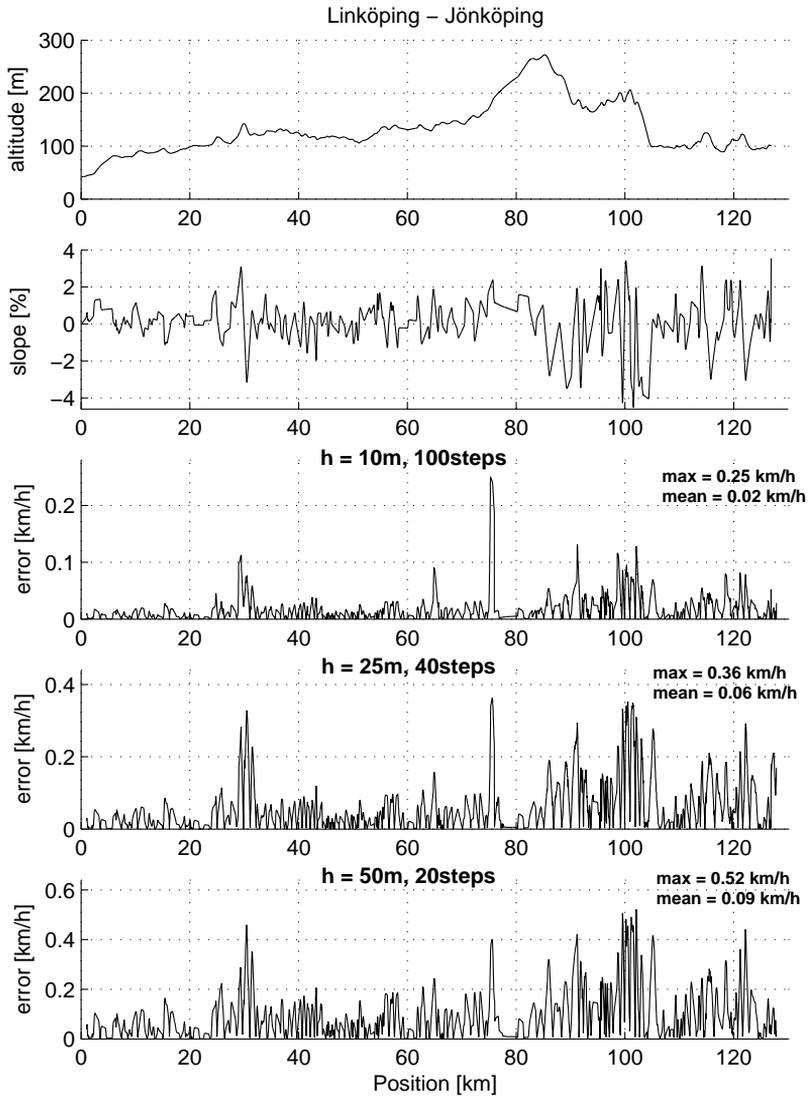


Figure B.2: The error made with N steps of length h when using calculated pedal and brake levels which are constant during the length h . The error is the difference between the velocity that is predicted through N steps of length h and a more accurate simulated velocity.

It is beneficial to compare the magnitude of the errors in figure B.2 with the velocity discretization δ used in the DP algorithm, see chapter 5. If the errors are smaller than δ along the horizon the vehicle will reach the predicted states and the optimal path will not change due to these errors. The optimal path can however change in subsequent samples with consideration of the additional slope values at the end of the horizon.

Studying figure B.2 thus give hints about what velocity discretization δ that are suitable to use in conjunction with a given step length h . It is unnecessary to increase the complexity by using a discretization of a much higher accuracy than the accuracy in the velocity predictions. Making δ twice as accurate will increase the computational complexity four times, according to section 5.7.

Sensor errors, imperfect slope measurements, model errors and other disturbances will make the difference between predicted and actual velocities greater. With additional errors such as these, the benefit of a shorter distance between computations of control signals increases as it reduces the time of open-loop control and increases the feedback. The reasoning in this section therefore sets a upper bound to the accuracy achievable in the velocity predictions.

Copyright

Svenska

Detta dokument hålls tillgängligt på Internet - eller dess framtida ersättare - under en längre tid från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns det lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida: <http://www.ep.liu.se/>

English

The publishers will keep this document online on the Internet - or its possible replacement - for a considerable time from the date of publication barring exceptional circumstances.

The online availability of the document implies a permanent permission for anyone to read, to download, to print out single copies for your own use and to use it unchanged for any non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional on the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its WWW home page: <http://www.ep.liu.se/>