

Development and Implementation of Stop and Go Operating Strategies in a Test Vehicle

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

by
Ann-Catrin Johansson

Reg nr: LiTH-ISY-EX-3632-2005

22nd June 2005

Development and Implementation of Stop and Go Operating Strategies in a Test Vehicle

Master's thesis

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

by **Ann-Catrin Johansson**

Reg nr: LiTH-ISY-EX-3632-2005

Supervisor: **Michael Back**
DaimlerChrysler
Anders Fröberg
Linköpings universitet

Examiner: **Professor Lars Nielsen**
Linköpings universitet

Linköping, 22nd June 2005

	Avdelning, Institution Division, Department Vehicular Systems, Dept. of Electrical Engineering 581 83 Linköping	Datum Date 22nd June 2005
	Språk Language <input type="checkbox"/> Svenska/Swedish <input checked="" type="checkbox"/> Engelska/English <input type="checkbox"/> _____	Rapporttyp Report category <input type="checkbox"/> Licentiatavhandling <input checked="" type="checkbox"/> Examensarbete <input type="checkbox"/> C-uppsats <input type="checkbox"/> D-uppsats <input type="checkbox"/> Övrig rapport <input type="checkbox"/> _____
URL för elektronisk version http://www.vehicular.isy.liu.se http://www.ep.liu.se/exjobb/isy/2005/3632/		
Titel Title Författare Author	Utveckling och implementering av Stop and Go strategier i ett testfordon Development and Implementation of Stop and Go Operating Strategies in a Test Vehicle Ann-Catrin Johansson	
Sammanfattning Abstract <p>The department REI/EP at DaimlerChrysler Research and Technology and the Laboratory for Efficient Energy Systems at Trier University of Applied Science, are developing control functions and fuel optimal strategies for low speed conditions. The goal of this thesis project was to further develop the fuel optimal operating strategies, and implement them into a test vehicle equipped with a dSPACE environment. This was accomplished by making optimal reference signals using dynamic programming. Optimal, in this case, means signals that results in low fuel consumption, comfortable driving, and a proper distance to the preceding vehicle. These reference signals for the velocity and distance are used by an MPC controller (Model Predictive Control) to control the car. In every situation a suitable reference path is chosen, depending on the velocities of both vehicles, and the distance. The controller was able to follow another vehicle in a proper way. The distance was kept, the driving was pleasant, and it also seems like it is possible to save fuel. When accepting some deviations in distance to the preceding car, a fuel reduction of 8 % compared to the car in front can be achieved.</p>		
Nyckelord Keywords	Stop and go, Dynamic programming, MPC, dSPACE, Fuel consumption	

Abstract

The department REI/EP at DaimlerChrysler Research and Technology and the Laboratory for Efficient Energy Systems at Trier University of Applied Science, are developing control functions and fuel optimal strategies for low speed conditions. The goal of this thesis project was to further develop the fuel optimal operating strategies, and implement them into a test vehicle equipped with a dSPACE environment. This was accomplished by making optimal reference signals using dynamic programming. Optimal, in this case, means signals that results in low fuel consumption, comfortable driving, and a proper distance to the preceding vehicle. These reference signals for the velocity and distance are used by an MPC controller (Model Predictive Control) to control the car. In every situation a suitable reference path is chosen, depending on the velocities of both vehicles, and the distance. The controller was able to follow another vehicle in a proper way. The distance was kept, the driving was pleasant, and it also seems like it is possible to save fuel. When accepting some deviations in distance to the preceding car, a fuel reduction of 8 % compared to the car in front can be achieved.

Keywords: Stop and go, Dynamic programming, MPC, dSPACE,
Fuel consumption

Preface

This thesis has been carried out at the DaimlerChrysler department for powertrain control, REI/EP, in Esslingen am Neckar, Germany.

Thesis outline

Chapter 2: The Vehicle Model

A brief account of the mathematical relations the model is based on.

Chapter 3: Dynamic Programming

An introduction to dynamic programming. A description on how it is used to create fuel optimized trajectories.

Chapter 4: MPC - Model Predictive Control

The MPC controller is described, and how it uses the trajectories.

Chapter 5: dSPACE

The equipment from dSPACE makes it possible to control the car with the controller developed in Simulink.

Chapter 6: Tests and Results

Test drives with the controller were performed. Here are the results.

Chapter 7: Conclusions

Conclusions made from the test results.

Chapter 8: Further Work

Further work that can be done within this project.

Acknowledgment

First of all I want to thank my supervisor at DaimlerChrysler Michael Back for giving me the opportunity to do this thesis project. I also want to thank him and all the people at REI/EP for making my stay in Germany a pleasant time. Thanks also to my supervisor at the university Anders Fröberg, and my opponent Johanna Töpel. Last, but not least, I want to thank Anders Burström for always being there for me.

Contents

Abstract	v
Preface and Acknowledgment	vi
1 Introduction	1
1.1 Background	1
1.2 Methods	1
2 The Vehicle Model	3
2.1 Dynamics	3
2.2 The Driveline	4
2.2.1 Basic Transmission Relations	4
2.2.2 The Subsystems	6
3 Dynamic Programming	9
3.1 The Recursive Formula	9
3.2 The Cost Function	10
3.3 Interpolation	11
3.4 Optimal Trajectories	12
4 MPC - Model Predictive Control	15
4.1 The Discrete-Time State-Space Model	15
4.2 Cost Function	16
4.3 Prediction of Future States	16
4.4 Analytical Solution	17
4.5 Linearization of the system model	18
4.6 Discretization	19
4.7 The Controller	20
5 dSPACE	23
5.1 From MATLAB To Reality	23
5.2 The Hardware	23

6	Tests and Results	25
6.1	Compensation of Time Delay	25
6.2	Reference Track	28
6.3	Fuel Consumption	28
6.3.1	Torque Limits	28
6.3.2	Brake Torque	30
6.3.3	Weighting Constants	30
7	Conclusions	33
8	Further Work	35
	References	37
	Notation	39
A	Reference Trajectories	41
A.1	Trajectory 1: Low velocity, acceptable distance	41
A.2	Trajectory 2: High velocity, acceptable distance	42
A.3	Trajectory 3: Acceptable velocity, short distance	42
A.4	Trajectory 4: Acceptable velocity, acceptable distance	43
A.5	Trajectory 5: Acceptable velocity, long distance	43
A.6	Trajectory 7: High velocity, short distance	44
A.7	Trajectory 8: High velocity, long distance	44
A.8	Trajectory 9: Low velocity, short distance	45
A.9	Trajectory 10: Low velocity, long distance	45
A.10	Short distance	46

Chapter 1

Introduction

The department REI/EP at DaimlerChrysler Research and Technology and the Laboratory for Efficient Energy Systems at Trier University of Applied Science are developing control functions and fuel optimal strategies for low speed conditions with stop and go situations, like in traffic jams. The goal of this thesis project is to further develop the fuel optimal operating strategies and implement them into a test vehicle equipped with a dSPACE environment.

1.1 Background

The public funded project "3D SIAM" is supported by the German Ministry for Education and Research and is a collaboration between several companies and organizations. In the project, sensors are developed that can give a three dimensional image of the traffic situation in front of a vehicle. An automatic driver for low speed situations is being developed, based on these sensors. This driver can be used for many purposes, for example keeping a safety distance to the car in front. Another benefit could be reduced fuel consumption in low speed conditions, like in traffic jams, and also to drive in a comfortable way. A previous thesis work in this area has shown, that it is possible to construct an automatic driver that follows a preceding vehicle and reduces the fuel consumption [3]. A controller that uses information about the preceding vehicle's velocity was created in Simulink. The optimization method used was dynamic programming. The criteria for the optimization was to get a low fuel consumption and to reach the velocity of the preceding car.

1.2 Methods

The first tests of the controller discussed in the previous section showed, that the optimization method was too slow to be used in real time applications.

Hence, a new approach to the problem was chosen. Instead of optimizing in real time, dynamic programming is used to calculate optimal "paths" in advance. These reference paths for the velocity and distance, describes how the car should behave to reach the requirements. Some new criteria were also added: The distance to the object in front should be kept, and the gear and the engine torque should have a smooth behavior, for a comfortable drive. A model of the preceding car is needed, and the simplest possible is chosen. The preceding car is assumed to keep its velocity during the optimization interval. When a number of different reference paths are made, an MPC (Model Predictive Control) controller is created to follow them. In every situation a suitable reference path is chosen, depending on velocities of both vehicles, and the distance. The MPC controller is a suitable controller, since it has the advantage of being able to make use of future reference values. As its name indicates, a model of the vehicle is needed for the controller. How the model is created is described in the next chapter.

Chapter 2

The Vehicle Model

To be able to make an MPC controller, a model of the vehicle is needed. This model will be used to predict future states. The model used comes from [3], and will be described briefly. The input to the model consists of three signals, the required driving torque, divided into M_{engine} and brake torque M_{brake} , and the gear g . When the engine torque is applied, it will be transferred to the wheels through subsystems. As output from the system, the velocity will be obtained. The dynamics of the model are based on Newton's second law, $F = ma$. Newton's second law for rotation can be used to describe the transmission of torque from one part of the driveline to another;

$$M(t) = J \frac{d}{dt} \omega(t) \quad \implies \quad \omega(t) = \frac{1}{J} \int_0^t M(\tau) d\tau \quad (2.1)$$

where $M(t)$ is the torque [Nm], $\omega(t)$ the angular velocity [rad/s] and J the moment of inertia [Nm/s²]. When no physical relation is available for a special part of the engine, maps have been used. The maps have been generated using testbed obtained data.

2.1 Dynamics

The dynamics are based on Newton's second law. The acceleration can be expressed like:

$$\dot{v} = \frac{1}{m} F_{\text{tot}} = \frac{1}{m} (F_{\text{df}} - F_{\text{res}}) \quad (2.2)$$

F_{df} is the drive force and F_{res} is the resistance force acting on the car. F_{df} corresponds to the torque that has been transmitted from the engine via the driveline to the wheels. To calculate this, a model of the driveline is needed. This is discussed in the next section. F_{res} consists of several different parts:

- Air resistance

$$F_{\text{air}} = \frac{1}{2} \rho_{\text{air}} A_f c_w v^2 \quad (2.3)$$

The air resistance depends on the air density ρ_{air} , the maximal vehicle cross area A_f , the air drag coefficient c_w , and the velocity v . The value of c_w depends on the design of the car.

- Rolling resistance

$$F_{\text{rollingres}} = K \cdot mg \cdot \cos(\gamma) = K \cdot mg \quad (2.4)$$

The rolling resistance is proportional to the mass. The constant K is actually depending on the velocity, but for low velocities it can be assumed to be constant. The rolling resistance is also dependent on the slope of the road, γ . But since that signal will not be accessible in the car, γ will always be set to zero.

- Gravitation

$$F_{\text{incline}} = mg \cdot \sin(\gamma) = 0 \quad (2.5)$$

Since the slope of the road, γ , is always set to zero (see above), the force from the gravitation will also always be zero.

2.2 The Driveline

The drive force, F_{df} , corresponds to the torque transmitted from the engine via the driveline to the wheels. To calculate this force, a model of the driveline is made. When creating a model of the driveline, the main parts to model are engine, clutch, transmission, propeller shaft, final drive, drive shafts and wheels, see figure 2.1. Some simplifications are made when making the model. In this case an automatic gearbox, with fixed gear ratio, is used, and no clutch has to be modeled. The propeller shaft and final drive will be consolidated to a differential gear. Another simplification is that all the wheels will be placed in the middle of the car, which leaves no need to model the drive shafts. Torsional effects will be neglected.

2.2.1 Basic Transmission Relations

If the transmission from the engine to the wheels were stiff, the model would look like figure 2.2. There is a fixed conversation ratio for the transmission,

$$i = \frac{\omega_{\text{engine}}}{\omega_{\text{wheels}}} \quad (2.6)$$

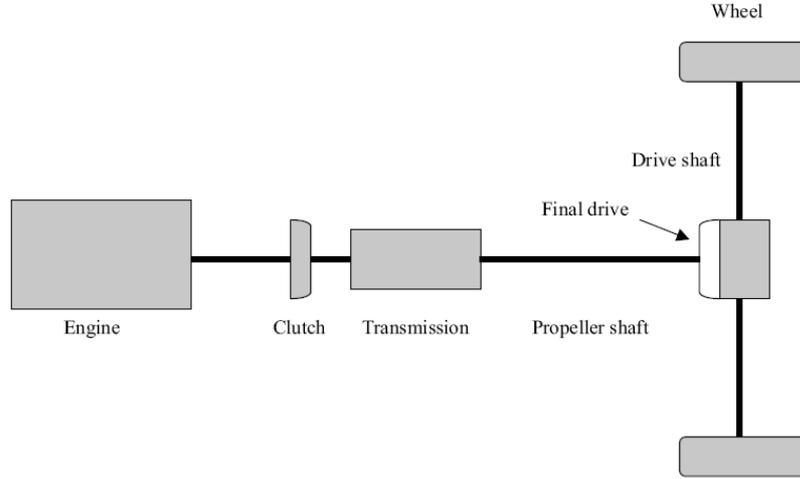


Figure 2.1: The main parts to model in the driveline of the vehicle. The figure comes from [3].

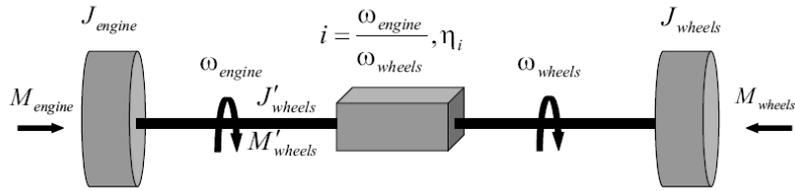


Figure 2.2: A stiff transmission between the engine and the wheels. The figure comes from [3].

and there is also an efficiency factor η . Using the conservation law of energy, the relation between the torques and the moments of inertia can be found [6].

$$P = M\omega \quad \Rightarrow \quad M'_{\text{wheels}}\omega_{\text{engine}} = \frac{1}{\eta}M_{\text{wheels}}\omega_{\text{wheels}} \quad \Leftrightarrow$$

$$M_{\text{wheels}} = \eta i M'_{\text{wheels}} \quad (2.7)$$

$$E_k = \frac{1}{2}J\omega^2 \quad \Rightarrow \quad \frac{J'_{\text{wheels}}\omega_{\text{engine}}^2}{2} = \frac{1}{\eta} \frac{J_{\text{wheels}}\omega_{\text{wheels}}^2}{2} \quad \Leftrightarrow$$

$$J_{\text{wheels}} = \eta i^2 J'_{\text{wheels}} \quad (2.8)$$

Fig 2.3 shows the model of the driveline with their respective torques and angular frequencies.

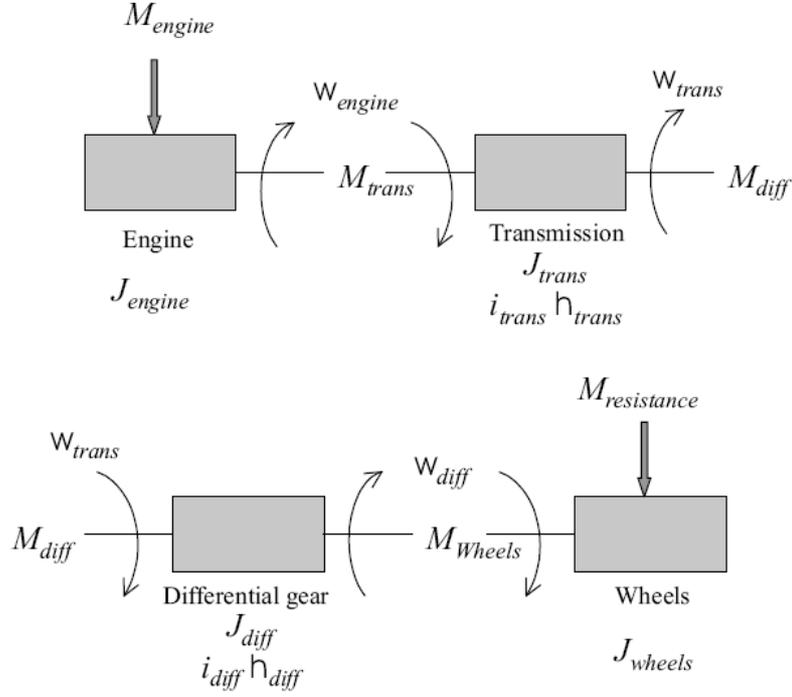


Figure 2.3: The model of the driveline with torques and angular frequencies. The figure comes from [3].

2.2.2 The Subsystems

The driveline contains a number of subsystems. In this section the modeling of the subsystems will be further explained.

- The combustion engine
The input to the combustion engine will be the desired engine torque. Two maps will be used to calculate the upper and lower boundaries for possible engine torque, with respect to the current rotational speed.

$$f_{\min}(\omega_{\text{engine}}) \leq M_{\text{engine}} \leq f_{\max}(\omega_{\text{engine}}) \quad (2.9)$$

- The transmission
For each gear there is a conversion ratio for the transmission i_{trans} , and a efficiency factor η_{trans} . They transform the torque and the moment of inertia from the engine according to equations (2.7) and (2.8).
- The differential gear
The differential gear is located between the transmission and the wheels.

Here are the torque and the moment of inertia for the wheels calculated. They are calculated in the same way as for the gearbox, except that the transmission ratio i_{diff} and the efficiency factor η_{diff} are constant and do not depend on the gear.

- The wheels

With help of equation (2.1), a new angular frequency can be calculated. The torque comes from the differential gear where the brake and resistance torque have been subtracted: $M_{\text{wheels}} = M_{\text{diff}} - M_{\text{brake}} - M_{\text{resistance}}$. The total moment of inertia is the sum of the moment of inertia from the differential gear and the total moment of inertia caused by the wheels. Since all four wheels are considered as one, the moment of inertia at the wheels will be

$$J = J_4 + m_{\text{tot}}r^2 \quad (2.10)$$

With a wheel radius of r_{wheel} and equation (2.2), the model can be expressed like:

$$\dot{s} = v$$

$$\dot{v} = \frac{1}{m} \left(\frac{1}{r_{\text{wheel}}} (M_{\text{Wheels}} - M_{\text{brake}}) - \frac{1}{2} \rho_{\text{air}} A_f C_w v^2 - K \cdot mg \right) \quad (2.11)$$

Chapter 3

Dynamic Programming

Dynamic programming, developed by Richard Bellman in the 1950s, is an optimization method that is suitable for problems that can be divided into a series of smaller problems which can be treated sequentially. It is especially useful solving multi stage problems where the possible alternatives in every stage is limited. More about dynamic programming can be read in [3] and [4].

3.1 The Recursive Formula

Consider a system with system equation $\bar{x}(k+1) = f(\bar{x}(k), \bar{u}(k))$ and terminal boundary conditions $\phi(\bar{x}(p)) = 0$. Every state $\bar{x}(k)$ is associated with a control function $\bar{u}(k)$, that brings the system into the next state. The optimal path between $\bar{x}(0)$ and the terminal hypersurface, $\bar{x}(p)$, is consequently associated with a set of optimal control functions. All points on the optimal path are possible starting points for the same optimal control functions.

It is a tedious task to calculate the cost for every possible way to find the optimal route. Instead, it is calculated backwards, by calculating the optimal way from every point to the terminal hypersurface. The cost of the route from a state $\bar{x}(0)$ to the terminal hypersurface $\bar{x}(p)$, can be formulated as

$$J = \sum_{k=0}^{p-1} L(\bar{x}(k), \bar{u}(k)) \quad (3.1)$$

L describes the cost of going from the state $\bar{x}(k)$ to the next applying $\bar{u}(k)$. Let $R(\bar{x}(k_1))$ represent the minimum cost going from $\bar{x}(k_1)$ to $\bar{x}(p)$. Thus $R(\bar{x}(k_1))$ can be expressed as the minimization of (3.1):

$$R(\bar{x}(k_1)) = \min_{\{\bar{u}(k_1), \dots, \bar{u}(p-1)\}} J = \min_{\{\bar{u}(k_1), \dots, \bar{u}(p-1)\}} \left\{ \sum_{k_1}^{p-1} L(\bar{x}(k), \bar{u}(k)) \right\} \quad (3.2)$$

Since the cost in the end state must be predetermined, a recursive formula can be set up. The cost in the final stage is

$$R(\bar{x}(p)) = \phi(\bar{x}(p)) \quad (3.3)$$

The remaining cost in time $p - 1$ can then be expressed as

$$R(\bar{x}(p-1)) = \min_{\bar{u}(p-1)} \{L(\bar{x}(p-1), \bar{u}(p-1)) + \phi(\bar{x}(p))\} \quad (3.4)$$

Inserting (3.3) into (3.4) leads to

$$R(\bar{x}(p-1)) = \min_{\bar{u}(p-1)} \{L(\bar{x}(p-1), \bar{u}(p-1)) + R(\bar{x}(p))\} \quad (3.5)$$

and the minimum cost for an arbitrary point $\bar{x}(k)$, can now be expressed as

$$R(\bar{x}(k)) = \min_{\bar{u}(k)} \{L(\bar{x}(k), \bar{u}(k)) + R(\bar{x}(k+1))\} \quad (3.6)$$

The minimum cost for every point can now be calculated recursively as (3.6).

After this backward calculation, a forward calculation is needed if the optimal route is sought, and not only the cost of it. Finding the optimal route is the same thing as finding the optimal control vectors $\bar{u}^*(k)$ for every point. The optimal $\bar{u}^j(k)$ for every stage k is the one that minimizes the sum of the cost, $L(\bar{x}(k), \bar{u}(k))$, and the remaining cost, $R(\bar{x}(k+1))$. Consequently, $\bar{u}^*(k)$ can be calculated as:

$$\bar{u}^*(k) = \arg \min_{\bar{u}(k)} \{L(\bar{x}(k), \bar{u}(k)) + R(\bar{x}(k+1))\} \quad (3.7)$$

3.2 The Cost Function

The cost function, $L(\bar{x}(k), \bar{u}(k))$, must be defined for the controller. Since the system is based on time, and not on distance like in [3], a new strategy has to be set up. A prediction horizon is chosen. In the end of that time period, the controlled vehicle should have retained or reached an acceptable distance to the preceding vehicle. It should also reach the same velocity as the preceding car. This demands a model of the preceding car. The simplest model possible was used; the preceding vehicle is assumed to keep a constant velocity v_p during the optimization period. This, together with the aim to save fuel, gives a cost function looking like

$$L = f_{\text{fuel}}(n, M_{\text{engine}}) + Q_1(v - v_p)^2 = f_{\text{fuel}}(C \cdot v, M_{\text{engine}}) + Q_1(v - v_p)^2 \quad (3.8)$$

The first part of the function, $f_{\text{fuel}}(C \cdot v, M_{\text{engine}})$, stands for the fuel consumption. It is given by a map with the rotational speed n , and the engine

torque M_{engine} as inputs, and the fuel flow as output. The present rotational speed, can be expressed in the present velocity v multiplied with

$$C = \frac{30 \cdot i_{\text{diff}} \cdot i_{\text{trans}}}{\pi \cdot r_{\text{wheel}}} \quad (3.9)$$

C is dependent on the present gear, since i_{trans} is gear dependent. The second part of the cost function, $Q_1(v - v_p)^2$, stands for the deviation of the velocity of the car, v , from the velocity of the preceding car, v_p . Q_1 is a weighting constant. The size of Q_1 determines the significance of this part.

The boundary conditions $\phi(\bar{x}(p))$ must also be defined. The velocity $v(p)$, and distance $d(p)$, that the vehicle will reach in the last prediction step, must be determined. The aim would be to be as close as possible to the velocity of the preceding vehicle, and the desired distance, d_p :

$$\phi(\bar{x}(p)) = Q_2(v(p) - v_p)^2 + Q_3(d(p) - d_p) \quad (3.10)$$

Where Q_2 and Q_3 are weighting constants.

3.3 Interpolation

The equation for the optimal route (3.7) assumes a time-discrete system. Most technical systems are time-continuous. A time-continuous system's state and input variables must be discretized with a quantization step. The size of the quantization step is a question of priorities. A shorter step will lead to a higher accuracy, and a longer step will give a shorter computation time. However, it is important to choose it small enough, so the next velocity can be reached with maximum torque during one time step. A problem that can occur when discretizing, is that the system can reach states which do not agree with the discrete states $x^i(k)$. For that reason, at time k , $\tilde{R}^{ij}(k+1)$ is used as an interpolation between the contiguous states' remaining cost. The equation (3.7) can then be expressed like

$$\bar{u}^*(k) = \arg \min_{\bar{u}(k)} \{L(\bar{x}(k), \bar{u}(k)) + \tilde{R}(\bar{x}(k+1))\} \quad (3.11)$$

The state-space model used in [3] had only one state; the velocity. That model was expanded, and the state-space model used for this optimization has two state variables; the velocity of the car, $x_1(k)$, and the distance to the car in front, $x_2(k)$. Both are discretized and saved as vectors, with the sizes i and j respectively. At the backwards calculation a two dimensional matrix for the remaining cost is created, for every point of time k , with the size i times j . The remaining cost at time k , $R(\bar{x}(k))$, is created by starting in every $x_1(i)$ and $x_2(j)$ and applying all possible torque and gear values. The torque values are also discretized. The new obtained values on velocity and distance, are used for interpolation in the remaining cost matrix for the next

time step, $R(x(k+1))$. This interpolated value is then added with the cost to get there. The cost of the cheapest way is then saved in the matrix $R(\bar{x}(k))$ at position (i, j) . These matrices are then used in the forward calculation, to find the cheapest way to the end. The starting point is the current velocity and distance. The torques and gears for the optimal path are saved in vectors.

3.4 Optimal Trajectories

Now it is possible to generate optimal trajectories; to get from one velocity to another and at the same time reach the desired distance between the cars. For each situation, a reference trajectory for the two states, velocity and distance, will be created. But the trajectories are not really satisfying, because the behaviour of the car will be spasmodic. Weighting constants that can help avoid this are added in the forward calculation. Weighting constants for big changes in gear and torque are included. A number of optimal trajectories can now be obtained for later use together with the MPC.

Optimizations were performed with different weighting constants, and the best ones were saved. The criteria were low consumption, no oscillating behaviour, and to reach the desired final values. An example of this can be seen in figure 3.1. Both simulations have a smooth behavior of the torque and the gear. The first case is represented by the dashed line. Here, the weighting constant of the velocity, Q_1 , in the cost function (3.8), is set to zero. That means that it is not important to be close to the velocity of the preceding car. In the second case, represented by the solid line, a value on the weighting constant is added, and the velocity of the controlled car is closer to the velocity of the preceding car. This lowers the peak of the velocity curve, and also keeps the distance in a better way.

During the optimizations, 150 Nm was used as maximum torque, and -400 Nm as minimum (braking). The velocity was also limited to 0–40 km/h and the distance between the cars was limited to 10–30 m. The quantization steps were 0.67 km/h and 1 m respectively.

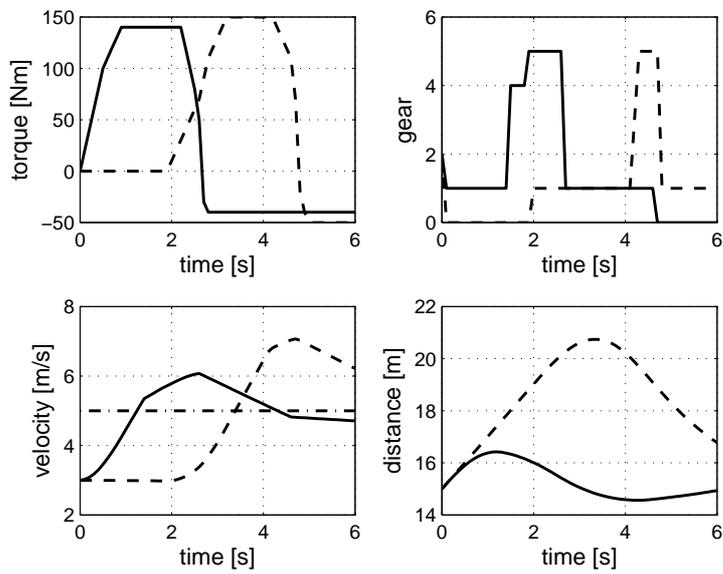


Figure 3.1: The dashed and the solid line represents two optimizations, with different weighting constants. The solid line represents the situation where some weighting on the velocity is added. The preceding car is driving at 5 m/s (the dashdot line).

Chapter 4

MPC - Model Predictive Control

Model Predictive Control (MPC) is an advanced control technique that has had a significant and widespread impact in industrial process control. The primary reason for its popularity is that it explicitly can handle equipment and safety constraints. Operations near such constraints are often necessary for the optimum control strategy. It can also handle future reference signals in a very effective way, which is useful in this case. The method is not much more complicated for multi variable systems than single variable. The fundamental idea of MPC is to formulate the control problem as an optimization problem, and solve this every time new data is received. Since this requires a big computational effort, MPC has formerly been used only in relatively slow processes, such as chemical industries. With modern computers MPC can now also be used in faster systems. More about MPC controllers can be found in [2] and [5].

4.1 The Discrete-Time State-Space Model

Since MPC has the advantage of treating multi variable systems the same way as single variable systems, the multi variable system briefly discussed in section 3.3 will be assumed in this chapter. The system has two states, the velocity and the distance, represented by $x(k)$. The input $u(k)$ is the engine torque. The discrete state-space system model will then be:

$$x(k+1) = Ax(k) + Bu(k) \quad (4.1)$$

$$y(k) = Cx(k) \quad (4.2)$$

where

$$x(k) = \begin{bmatrix} x_1(k) \\ \vdots \\ x_n(k) \end{bmatrix}, \quad u(k) = \begin{bmatrix} u_1(k) \\ \vdots \\ u_m(k) \end{bmatrix}, \quad y(k) = \begin{bmatrix} y_1(k) \\ \vdots \\ y_p(k) \end{bmatrix} \quad (4.3)$$

4.2 Cost Function

A common problem is to follow a predefined trajectory, r , like the trajectories made in chapter 3. The cost function J can then be expressed like

$$\min_{u_{\min} \leq u \leq u_{\max}} J = \sum_{j=0}^{N-1} \|y(k+j+1) - r(k+j+1)\|_{Q_1}^2 + \|u(k+j)\|_{Q_2}^2 \quad (4.4)$$

where $\|x\|_Q^2 = x^T Q x$. N is the prediction horizon and Q_1 and Q_2 weighting matrices (not to be mixed up with the weighting constants in the previous chapter). N should be selected to cover a settling time of the system. Choosing different values on the weighting matrices decides which part is most important to be small. If, for example, it is more important that the trajectory is followed closely than using small inputs, Q_1 should be chosen big and Q_2 small.

4.3 Prediction of Future States

If a signal $u(k)$ is applied to our system, equation (4.1) will be received. If this model is used recursively, a two step prediction becomes

$$x(k+2) = A^2 x(k) + ABu(k) + Bu(k+1) \quad (4.5)$$

and a N step prediction becomes

$$\begin{aligned} X &= Hx(k) + SU \\ Y &= CX \end{aligned} \quad (4.6)$$

where

$$\begin{aligned} X &= \begin{bmatrix} x(k+1) \\ \vdots \\ x(k+N) \end{bmatrix}, \quad U = \begin{bmatrix} u(k) \\ \vdots \\ u(k+N-1) \end{bmatrix} \\ H &= \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}, \quad S = \begin{bmatrix} B & 0 & \dots & 0 \\ AB & B & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \dots & B \end{bmatrix} \end{aligned}$$

$$C = \begin{bmatrix} C & & & \\ & C & & \\ & & \ddots & \\ & & & C \end{bmatrix}$$

4.4 Analytical Solution

A big advantage using MPC is that it can handle constraints, but it can still be useful in cases with no constraints. Known future reference signals can be used in a very effective way. With (4.6) the cost function (4.4) can be expressed in a more compact way, using the following weight matrices

$$Q_1 = \begin{bmatrix} Q_1 & & & \\ & Q_1 & & \\ & & \ddots & \\ & & & Q_1 \end{bmatrix}, \quad Q_2 = \begin{bmatrix} Q_2 & & & \\ & Q_2 & & \\ & & \ddots & \\ & & & Q_2 \end{bmatrix}$$

and the vector R with the future reference signals, derived in the previous chapter,

$$R = \begin{bmatrix} r(k+1) \\ r(k+2) \\ \vdots \\ r(k+N) \end{bmatrix}$$

By using these matrices, the cost function can be expressed like

$$\begin{aligned} J &= \sum_{j=0}^{N-1} \|y(k+j+1) - r(k+j+1)\|_{Q_1}^2 + \|u(k+j)\|_{Q_2}^2 \\ &= (Y - R)^T Q_1 (Y - R) + U^T Q_2 U \\ &= (C(Hx(k) + SU) - R)^T Q_1 (C(Hx(k) + SU) - R) \\ &\quad + U^T Q_2 U \end{aligned} \quad (4.7)$$

Since there are no constraints, the cost function can be minimized by finding the U that makes the gradient equal to zero. Using the rule

$$\frac{d(x^T P x)}{dx} = 2Px \quad (4.8)$$

and the chain rule, the gradient of J with respect to U is zero if

$$2S^T C^T Q_1 (C(Hx(k) + SU) - R) + 2Q_2 U = 0 \quad (4.9)$$

which is equal to

$$U = -(S^T C^T Q_1 C S + Q_2)^{-1} S^T C^T Q_1 (C H x(k) - R) \quad (4.10)$$

To get the control signal $u(k)$ only the first m rows in U are needed, as $u(k)$ is calculated again in the next timestep:

$$u(k) = [I \ 0 \ \dots \ 0] U \quad (4.11)$$

4.5 Linearization of the system model

Sometimes the system model is not linear, as in equation (4.1) in the previous chapter. Then the expression has to be linearized [5]. Assume the state vector x can be expressed like

$$\frac{dx}{dt} = f(x, u, t) \quad (4.12)$$

If the system is in state $x = x_0$ with input signals $u = u_0$, then the effect of small disturbances Δx and Δu will be:

$$\begin{aligned} \frac{dx}{dt} &= f(x_0 + \Delta x, u_0 + \Delta u, t) \\ &\approx f(x_0, u_0, t) + \left. \frac{\partial f}{\partial x} \right|_{(x_0, u_0, t)} \Delta x + \left. \frac{\partial f}{\partial u} \right|_{(x_0, u_0, t)} \Delta u \end{aligned} \quad (4.13)$$

where higher terms of Δx and Δu have been neglected. The expressions $\left. \frac{\partial f}{\partial x} \right|_{(x_0, u_0, t)}$ and $\left. \frac{\partial f}{\partial u} \right|_{(x_0, u_0, t)}$ denotes matrices with partial derivatives at (x_0, u_0, t) . Denote these matrices A and B , respectively. Since $x = x_0 + \Delta x$ and x_0 is a constant value of x , this gives $\frac{dx}{dt} = \frac{d\Delta x}{dt}$. The linearized model will then be

$$\frac{dx}{dt} = \frac{d\Delta x}{dt} = A \cdot \Delta x + B \cdot \Delta u + f(x_0, u_0, t)$$

If (x_0, u_0) is an equilibrium point at the time t , in other words $f(x_0, u_0, t) = 0$, the expression will look like the usual continuous-time linear state space-model

$$\frac{dx}{dt} = A \cdot \Delta x + B \cdot \Delta u = A(x - x_0) + B(u - u_0)$$

The model of the vehicle is, repeated from (2.11):

$$\dot{s} = v$$

$$\dot{v} = \frac{1}{m} \left(\frac{1}{r_{\text{wheel}}} (M_{\text{Wheels}} - M_{\text{brake}}) - \frac{1}{2} \rho_{\text{air}} A_f c_w v^2 - K \text{mg} \right) \quad (4.14)$$

Parts of the model are gear dependent. Since the gear is hard to handle in the linearization, one model for every gear will be made. The linearization of the model now becomes

$$\frac{ds}{dt} = \frac{dv}{dv} (v - v_0) + v_0 = v$$

$$\begin{aligned}
\frac{dv}{dt} &= \frac{\partial f}{\partial v}|_{v_0, M_0, g_0}(v - v_0) + \frac{\partial f}{\partial M}|_{v_0, M_0, g_0}(M - M_0) + f(v_0, M_0, g_0) = \\
&= \frac{1}{m} \left(\frac{1}{r_{\text{wheel}}} \frac{\partial M_{\text{diff}}}{\partial n} \cdot \frac{\partial n}{\partial v}|_{v_0, M_0, g_0} - \rho_{\text{air}} A_f C_w v_0 \right) (v - v_0) + \\
&+ \frac{1}{m} \frac{1}{r_{\text{wheel}}} \frac{\partial M_{\text{diff}}}{\partial M}|_{v_0, M_0, g_0} (M - M_0) + f(v_0, M_0, g_0) = \\
&= C_v (v - v_0) + C_M (M - M_0) + f(v_0, M_0, g_0) \quad (4.15)
\end{aligned}$$

To get a simpler expression, it would be desirable to linearize around an equilibrium point. To find such points, the equations in (4.14) should be set to 0. The first equation gives that v_0 has to be set to 0, and s_0 can be chosen as 0 for simplicity. By putting $v_0 = 0$ into the second equation, the value of M_0 for every gear will be given. Now the model can be written as

$$\begin{aligned}
\begin{bmatrix} ds/dt \\ dv/dt \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ 0 & C_v \end{bmatrix} \begin{bmatrix} s \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ C_M \end{bmatrix} [M - M_0] \\
y &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} s \\ v \end{bmatrix} \quad (4.16)
\end{aligned}$$

4.6 Discretization

To be able to use the linearized model in the MPC controller, it needs to be discretized. If the time-continuous system

$$\begin{aligned}
\dot{x}(t) &= A_c x(t) + B_c u(t) \\
y(t) &= C_c x(t) \quad (4.17)
\end{aligned}$$

is controlled with a partially constant signal $u(t)$ during the sample time T ; $u(t) = u(kT)$ when $kT \leq t < (k+1)T$. Then the corresponding time-discrete system will be

$$\begin{aligned}
x(kT + T) &= Ax(kT) + Bu(kT) \\
y(kT) &= Cx(kT) \quad (4.18)
\end{aligned}$$

where

$$A = e^{A_c T}, \quad B = \int_0^T e^{A_c t} B_c dt, \quad C = C_c \quad (4.19)$$

according to [2]. Here it is useful to know that

$$e^{A_c t} = \mathcal{L}^{-1}(sI - A_c)^{-1} \quad (4.20)$$

This will give

$$\begin{aligned}
A &= \begin{pmatrix} 1 & (e^{C_v T} - 1)/C_v \\ 0 & e^{C_v T} \end{pmatrix} \\
B &= \frac{C_M}{C_v} \begin{bmatrix} (e^{C_v T} - 1)/C_v - T \\ e^{C_v T} - 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (4.21)
\end{aligned}$$

4.7 The Controller

In the MPC controller, equation (4.10) will be computed every time step, using the precalculated discretization (4.21) and the chosen reference path. 10 reference signals are calculated for different situations, see figure 4.1. The

	$v - v_p < -1$	$ v - v_p < 1$	$v - v_p > 1$
$\Delta s < 10$	9	3	7
$10 < \Delta s < 20$	1	4	2
$20 < \Delta s$	10	5	8

Figure 4.1: The corresponding reference signals for every situation. Each number corresponds to a certain reference signal. Δs stands for the distance between the cars, v for the velocity of the car, and v_p for the velocity of the preceding car.

10th signal is not in the figure, but is used for hard braking when the distance to the preceding car is getting smaller than 7 m. The reference signals can be seen in appendix A. Each reference signal consists of a trajectory, for both states. The reference signals are chosen every time step, depending on current velocities and distance. If the same reference path is chosen several times in a row, the controller will continue on the same reference track. Otherwise, it will start from the beginning on a new one. An illustration of how the controller works can be seen in figure 4.2. As long as the distance is between 10–20 m, only the trajectories 1, 2 and 4 are used.

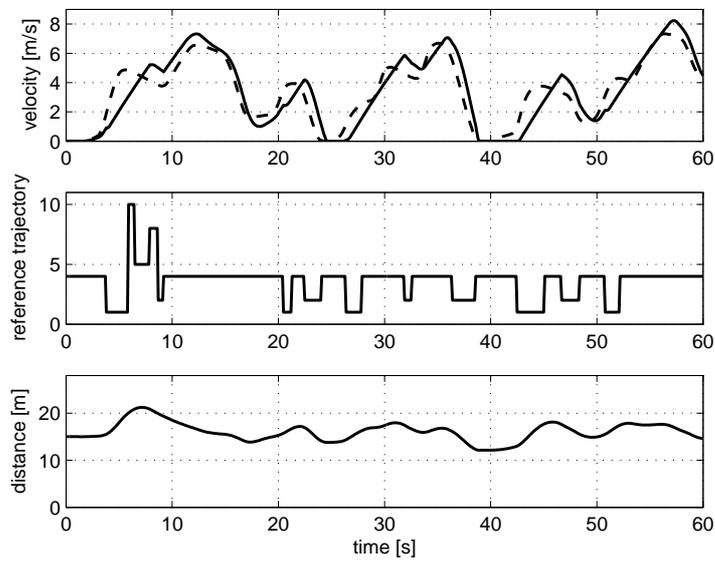


Figure 4.2: The controller is switching between the trajectories. This is from a simulation in Simulink. The dashed line in the figure on top is the preceding car, and the solid line represents the controlled vehicle.

Chapter 5

dSPACE

The idea behind the equipment from dSPACE is to offer a rapid prototyping technology. Using this makes it possible to use MATLAB and Simulink models in real time applications. Special soft- and hardware are developed to make this procedure as simple as possible. More information about the system can be found in [1] or [7].

5.1 From MATLAB To Reality

Figure 5.1 shows the structure of the combined soft- and hardware that are used when developing real time applications. The regulator is created in Simulink. A MATLAB toolbox, Real-Time Workshop, generates the corresponding C code for the model. Via the Real-Time Interface, dSPACE uses this code in different ways. The C code is compiled and downloaded to the MicroAutoBox. A number of files are also created, that are used to get information about the simulation and for changing parameters. The biggest advantage is that most of this happens automatically.

When the code is downloaded, a number of things can be done. For example, the parameters can be changed. There are two ways to do that. One way is to change the parameters in the Simulink model, and compile again, or ControlDesk can be used. ControlDesk is a tool for making a virtual instrument panel. ControlDesk can also be used to study the signals, and download them to MATLAB.

5.2 The Hardware

The MicroAutoBox can be connected between a laptop, equipped with the Simulink model and ControlDesk, and the car. See figure 5.2. The controller is downloaded to the MicroAutoBox, and can be controlled via ControlDesk.

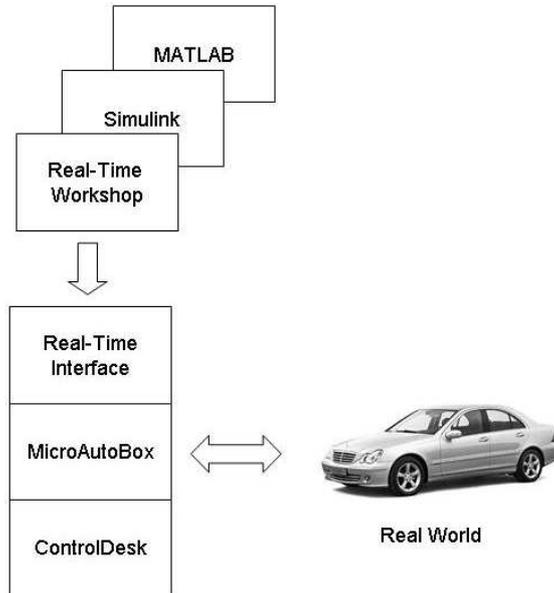


Figure 5.1: An overview of the way from model in Simulink to real time application.

MicroAutoBox is especially made for in-vehicle tests, and is connected to the CAN-bus system in the car. It can both give inputs to the car, and receive signals about the current status. The application program is stored on a non-volatile flash memory, that makes it possible for the MicroAutoBox to be restarted and not having to compile the Simulink-model again.



Figure 5.2: A laptop connected to the MicroAutoBox.

Chapter 6

Tests and Results

All tests were performed in a Mercedes-Benz C240 together with dSPACE MicroAutoBox. The car was adjusted for winter conditions. One consequence of that is that the car always starts on the second gear. Since the main working area is low speeds, second gear will be used almost all the time. The gear is chosen by the automatic gear box, and the controller will use the model for the present gear. The sensors were not ready and therefore not used in the tests, the preceding car is simulated. Either manually via ControlDesk, or by using a preprogrammed route. The prediction horizon N for the MPC controller was set to 20, which corresponds to 2 seconds.

6.1 Compensation of Time Delay

When first implementing the controller in the car, a time delay was identified. The car did not increase the velocity immediately when the controller required an acceleration, see figure 6.1. This caused the controller to start oscillating, as can be seen in figure 6.2. This behaviour was reproducible in the simulations when a time delay was added. There is a delay of n time steps, until the desired torque is adapted to the engine. That means, the calculated control signal will be applied in n time steps. So, the signal $u(k + n)$ should be used as input to the engine instead of $u(k)$. This was made by choosing another control signal from (4.11). Using this control signal, a much more stable velocity was received, as shown in figure 6.3.

Tests in the car showed that $n = 7$ made the system stable. The delay is consequently around 700 ms, since the sample time is set to 100 ms. A delay of 500 ms was detected between giving the desired torque as input to the engine, and until it is applied. But the case $n = 5$ was still oscillating. Trying $n = 7$ gave a better result. An additional delay of 200 ms must obviously arise from the controller.

When this adjustment was made, the controller followed the preceding

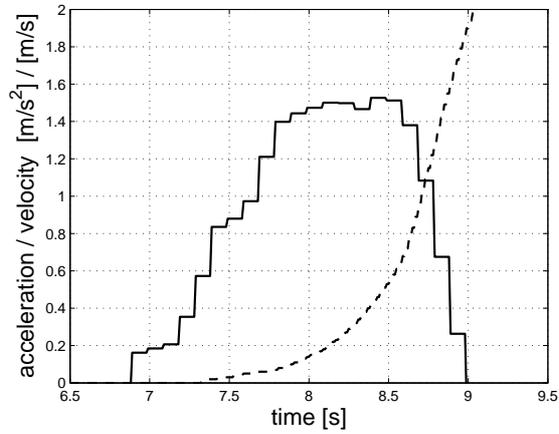


Figure 6.1: There is a delay in the system until the vehicle starts to accelerate. The solid line represents the desired acceleration, and the dashed line the velocity of the car.

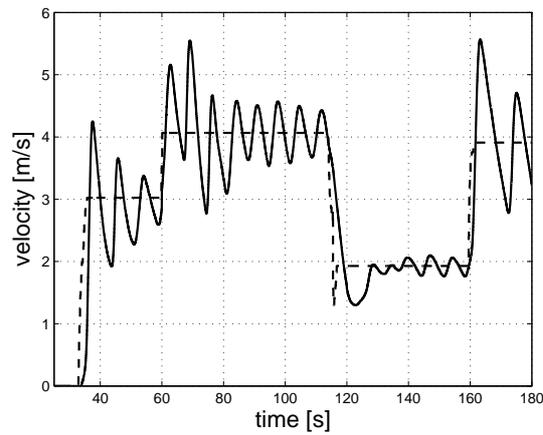


Figure 6.2: Results from a test drive. There was a delay in the system that caused the system to oscillate. The solid line represents the velocity of the controlled vehicle, and the dashed the preceding vehicle.

car in a better way. An example of the behaviour of the controller can be seen in figure 6.4. The preceding car may seem to drive in a strange way, with very quick jumps between velocities. This is due to the fact that the preceding car was simulated by hand in ControlDesk during the test.

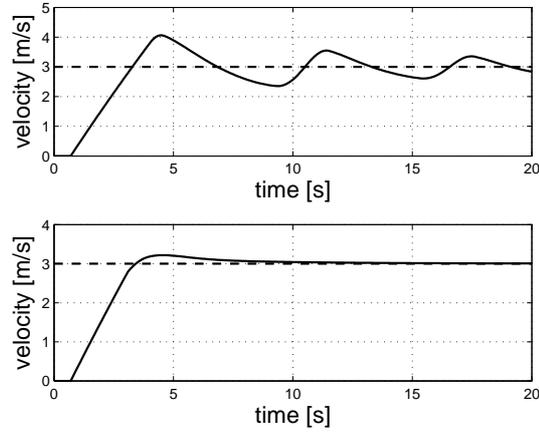


Figure 6.3: This is a simulation in Simulink. The top figure shows that the controller starts to oscillate when a delay of 0.7 seconds of the output signal is included in the model. The second figure shows that with the use of another control signal, the oscillations are reduced. The solid lines represents the velocity of the car, and the dashed lines the velocity of the preceding car.

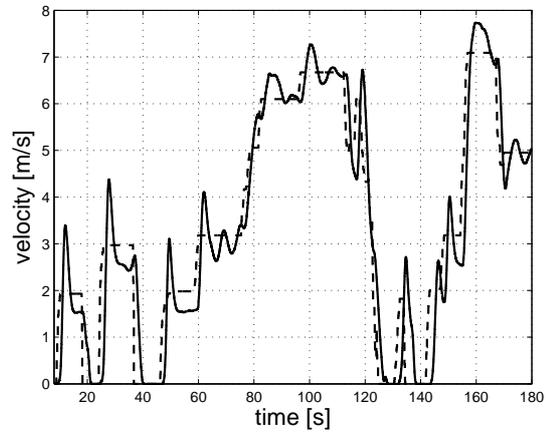


Figure 6.4: After adjusting the brake torque and compensating for the time delay, the controller is able to follow another car. This is the result from a test drive. The solid line represents the velocity of the car, and the dashed line the velocity of the preceding car.

6.2 Reference Track

As a reference track, measurements from the test track in the production plant of Untertürkheim have been used. A three minutes drive with stop and go behaviour was recorded, according to figure 6.5. The goal is to follow this

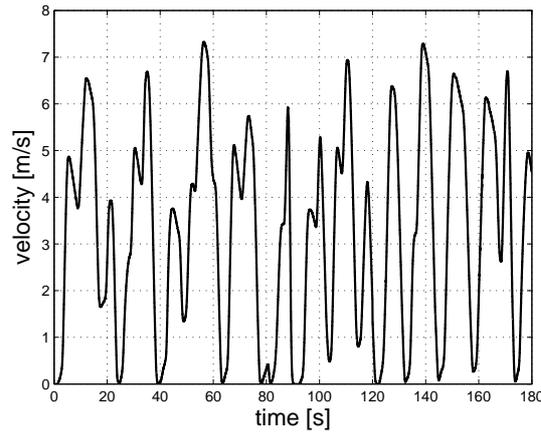


Figure 6.5: The reference track used in the tests.

reference track in a good way. The main tasks are to keep a proper distance to the preceding car, drive in a comfortable way, and at the same time minimize the fuel consumption.

6.3 Fuel Consumption

The dSpace hardware, with the reference track, starts running immediately when starting the car. The driver must give the controller permission to control the car manually. Permission was not given at the exact same time in every test. To make a fair comparison between the different configurations of the controller, a given period of the reference track is picked out. Some part of the beginning and of the end was cut away. The chosen period was 14,48–154,48 s in figure 6.5. The fuel consumption for the preceding car during this selected period was 34.40 l/100 km, and the distance traveled was about 440 m.

6.3.1 Torque Limits

In the first set of tests, different torque limits in the controller are tested; 150 Nm, 100 Nm, and 50 Nm. Their different behaviors can be seen in figure 6.6. The test made with the lowest limit seems to have some difficulties following

the track. The torque is not big enough to accelerate the car as fast as needed to be able to reach the desired velocities in time. The fuel consumption can

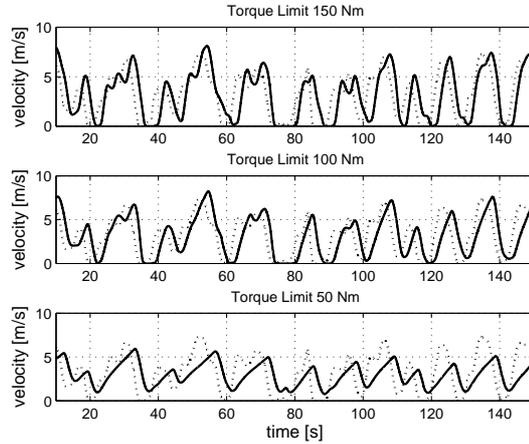


Figure 6.6: The behaviour of the controlled vehicle (solid line) following the reference track (dashed line) from section 6.2.

be seen in figure 6.7. All of them consume less fuel than the preceding car (the reference track). The fuel consumption decreases with a lower engine

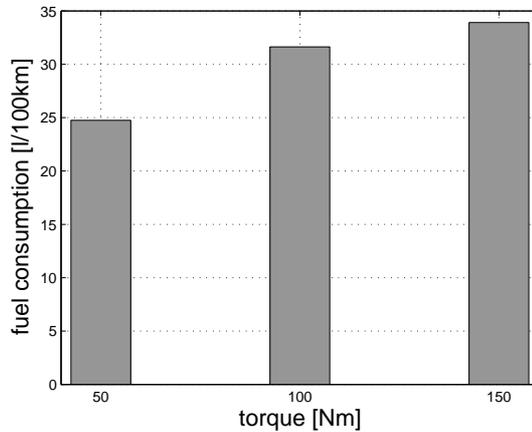


Figure 6.7: The fuel consumption for different engine torque limits. The consumption of the preceding car during the same time period was 34.40 l/100 km.

torque limit. But, as can be seen in figure 6.8, there will be a problem if the

value is too low. The limited acceleration possibility makes it hard to keep the distance and the controlled car will lose track of the car in front.

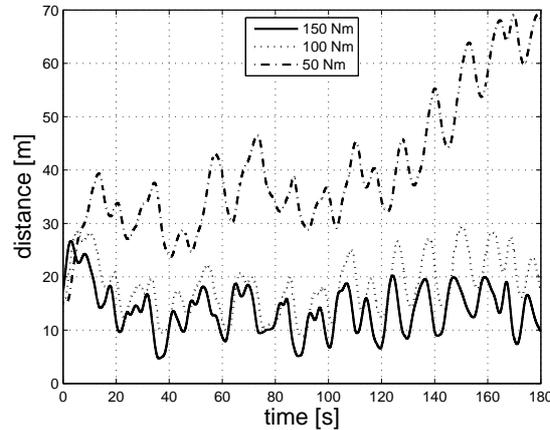


Figure 6.8: The distances for different engine torque limits.

6.3.2 Brake Torque

The brake torque must be big enough to brake the car, but not too big, since it is also important to drive in a comfortable way. The tests made in section 6.3.1 were made with a brake torque in a range of 0–2700 Nm. As mentioned earlier, test persons approved that as an comfortable brake force. It would be interesting to test if a smaller maximum brake torque reduces the fuel consumption even further. In this section all tests are made with a maximum engine torque of 100 Nm. Tests were made with the ranges 0–2400 Nm and 0–1700 Nm. Figure 6.9 shows how well they follow another vehicle. They all seems to follow pretty well. The resulting fuel consumption is shown in figure 6.10. A smaller brake torque limit results in a lower consumption. The distance to the preceding car is showed in figure 6.11. The distance is getting very short at some occasions, with a maximum brake torque of 1700 Nm. The braking does not seem to be strong enough. A bigger brake torque maximum should be used for safety reasons.

6.3.3 Weighting Constants

Another possibility to influence the fuel consumption could be to change the weighting constants. The weighting constants, Q_1 and Q_2 , are introduced in equation (4.4). Q_1 affects the deviations from the reference distance and velocity, and Q_2 the size of the engine torque. Some tests with changed

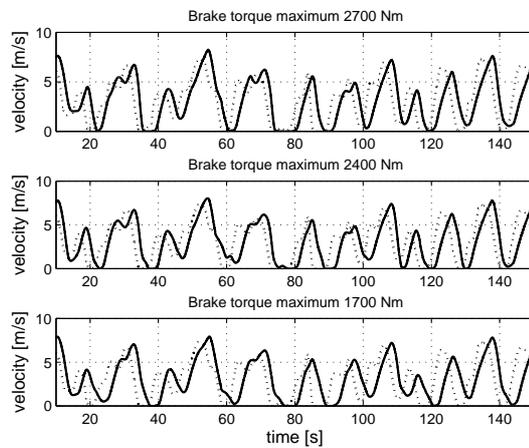


Figure 6.9: The behaviour of the controlled vehicle (solid line) following the reference track (dashed line) from section 6.2.

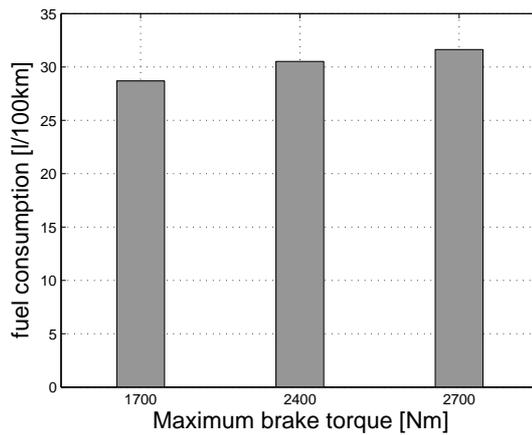


Figure 6.10: The fuel consumption for different brake torque limits. The consumption of the preceding car during the same time period was 34.40 l/100 km.

values of the weighting constants were made. But the fuel consumption did not differ much, and the distance was kept. Maybe the changes were not big enough, or the constants are not so important for the consumption.

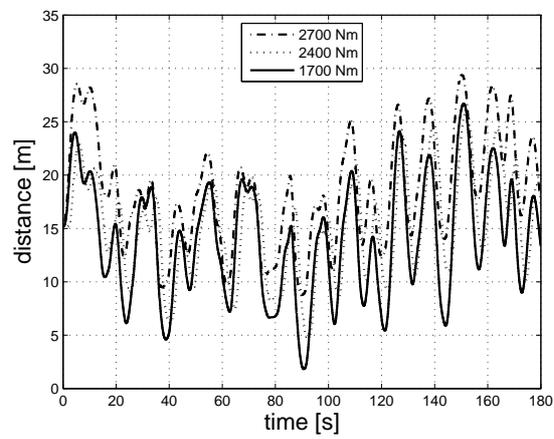


Figure 6.11: The distance to the preceding car, with different brake torque maximum.

Chapter 7

Conclusions

The goal of this thesis project was to further develop fuel optimal operating strategies for stop and go situations, and to implement them into a test vehicle. Based on a controller in Simulink using dynamic programming, a new MPC controller was created. The new controller is fast enough to run in real time.

The controller is able to follow another vehicle in a proper way. Test persons thought the braking and acceleration did not differ much from a normal manual driving. The distance to the preceding car was also kept.

It also seems like it is possible to save fuel. A limitation of the engine torque to 100 Nm gives a fuel reduction of 8 % compared to the car in front. But comparing with the car in front may not be the best way to evaluate if the car saves fuel or not.

Chapter 8

Further Work

The next step in developing an automatic driver, would be to connect the controller to the sensors. Based on sensor information and additional test scenarios an improvement of the controller could be done.

The trajectories could be improved too, and more of them could be calculated, for more situations.

The behaviour of the vehicle in front could be modeled in a better way. Right now the controller is always assuming that the car in front will keep it's current velocity during the prediction horizon.

The fuel consumption should be compared in a better way. A method that may be used, is to record a period of low speed, dense traffic, using the sensors. The velocity of the preceding car from that sequence can be used to evaluate, how a real driver would drive. And how much fuel that is used.

It may be possible to remove some of the criteria in the cost function. Right now, there are demands on both the distance and the velocity. It may be enough to only try to keep a proper distance. This could be investigated in test drives.

References

- [1] dSPACE. Catalog 2005. internet, 2005. www.dspace.com.
- [2] T. Glad, S. Gunnarsson, L. Ljung, T. McKelvey, A. Stenman, and J. Löfberg. *Digital Styrning Kurskompendium*. Linus & Linnea AB, 2002.
- [3] J. Jonsson. Fuel optimized predictive following in low speed conditions. Master's thesis LiTH-ISY-EX-3436, Department of Vehicular Systems, Linköpings Universitet, Linköping, Sweden, June 2003.
- [4] Arthur E. Bryson Jr and Yu-Chi Ho. *Applied Optimal Control*. Taylor and Francis, 1975.
- [5] J. M. Maciejowski. *Predictive Control with Constraints*. Prentice Hall, 2002.
- [6] C. Nordling and J. Österman. *Physics Handbook for Science and Engineering*. Studentlitteratur, 1999.
- [7] E. Wernholt. Minimanual för dSPACE, 2002.

Notation

Symbols used in the report.

Variables and parameters

γ	Angle of incline
η	Efficiency factor
η_{diff}	Efficiency factor of the differential gear
η_{trans}	Efficiency factor of the transmission
ρ_{air}	Air density
ω	Angle velocity
ω_{engine}	Angle velocity of the engine
ω_{wheels}	Angle velocity of the wheels
A_f	Maximal vehicle cross area
c_w	Air resistance coefficient
E_k	Kinetic energy
F_{air}	Air resistance
F_{df}	Driveforce
$F_{incline}$	Gravitational force caused by incline
F_{res}	Resistance force acting on the vehicle
$F_{rollingres}$	Rolling resistance
F_{tot}	Total force acting on the vehicle
g	Acceleration of gravity or gear
i	Conversion ratio
i_{diff}	Fixed conversion ratio for the differential gear
i_{trans}	Gear depending conversion ratio for the transmission
J	Moment of inertia
J_4	Moment of inertia for all four wheels
J_{wheels}	Moment of inertia for the wheels
K	Rolling resistance coefficient
m	Vehicle mass
M	Torque
M_{brake}	Brake torque
M_{engine}	Engine torque

$M_{resistance}$	Resistance torque
M_{wheels}	Torque acting at the wheels
n	Rotational speed
N	Prediction horizon
P	Power
Q	Weighting constant or matrix
r_{wheel}	Wheel radius
v	Velocity
v_p	Velocity of the preceding vehicle

Abbreviations

MPC Model Predictive Control

Appendix A

Reference Trajectories

The reference trajectories for the velocity and the distance used during the tests. These signals are added to the current velocity and distance, to get a reference signal suitable for the present situation.

A.1 Trajectory 1: Low velocity, acceptable distance

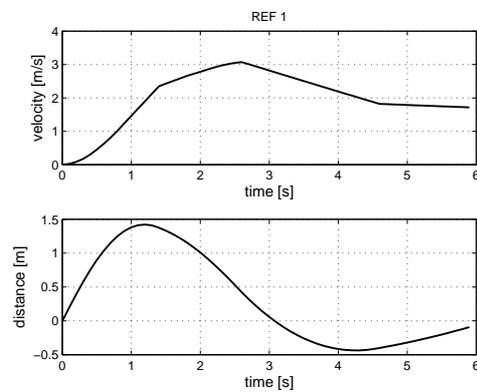


Figure A.1: In situations when the preceding car is driving more than 1 m/s faster than the controlled car this trajectory will be followed. It will raise the velocity with 2 m/s. The distance is acceptable, and will be kept.

A.2 Trajectory 2: High velocity, acceptable distance

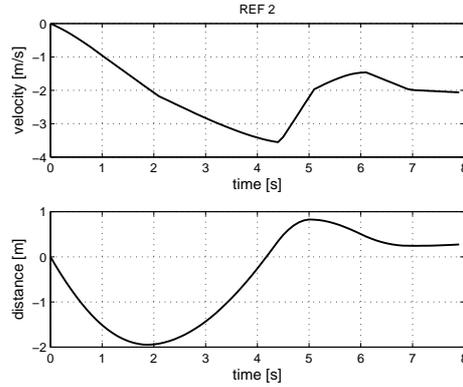


Figure A.2: In situations when the controlled car is driving more than 1 m/s faster than the preceding car this trajectory will be followed. It will lower the velocity with 2 m/s. The distance is acceptable, and will be kept.

A.3 Trajectory 3: Acceptable velocity, short distance

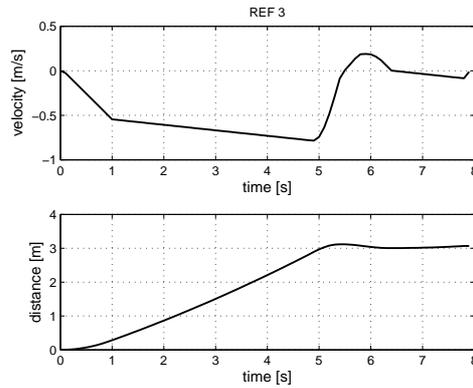


Figure A.3: If the difference between the velocities is small, but the distance is shorter than 10 m, this trajectory will be followed. It will keep the velocity and increase the distance 3 m.

A.4 Trajectory 4: Acceptable velocity, acceptable distance

When both velocity and distance is acceptable, trajectory number 4 is chosen. The reference velocity is then set to the velocity of the preceding car, and the reference distance to 15 m.

A.5 Trajectory 5: Acceptable velocity, long distance

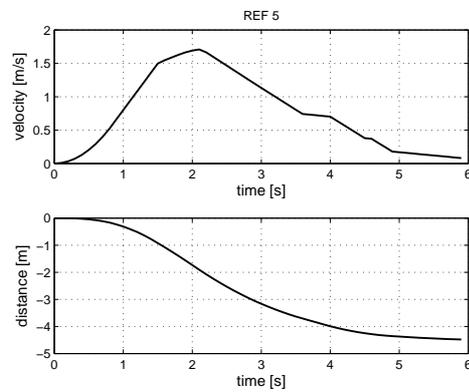


Figure A.4: If the difference between the velocities is small, but the distance is longer than 20 m, this trajectory will be followed. It will keep the velocity and decrease the distance about 5 m.

A.6 Trajectory 7: High velocity, short distance

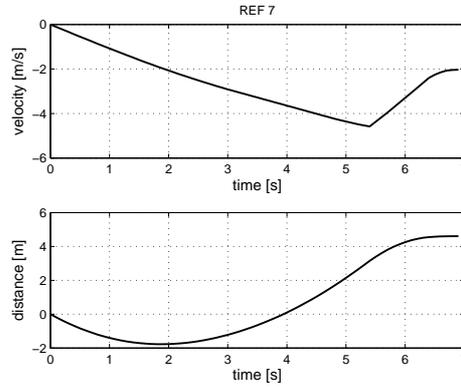


Figure A.5: In situations when the controlled car is driving more than 1 m/s faster than the preceding car, and the distance is shorter than 10 m, this trajectory will be followed. It lowers the velocity about 2 m/s and increases the distance 5 m.

A.7 Trajectory 8: High velocity, long distance

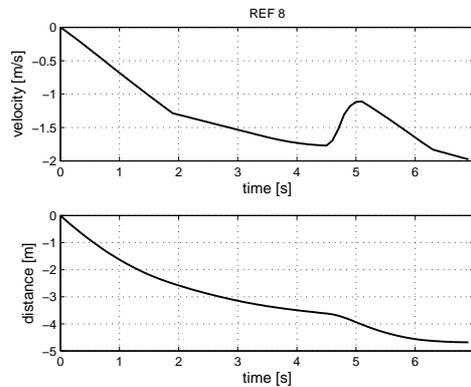


Figure A.6: In situations when the controlled car is driving more than 1 m/s faster than the preceding car, and the distance is longer than 20 m this trajectory will be followed. It lowers the velocity about 2 m/s and decreases the distance 5 m.

A.8 Trajectory 9: Low velocity, short distance

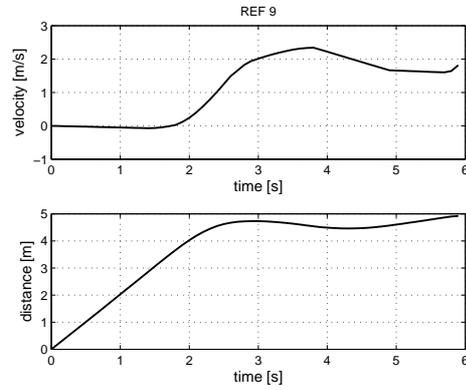


Figure A.7: In situations when the controlled car is driving more than 1 m/s slower than the preceding car, and the distance is shorter than 10 m, this trajectory will be followed. It increases the velocity about 2 m/s and increases the distance 5 m.

A.9 Trajectory 10: Low velocity, long distance

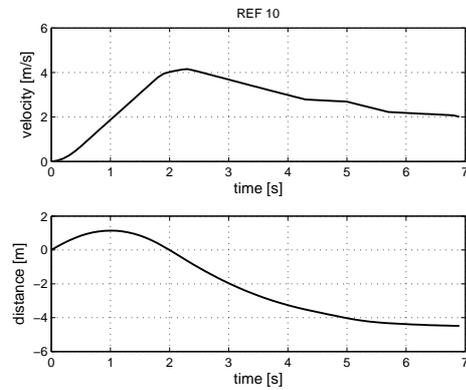


Figure A.8: In situations when the controlled car is driving more than 1 m/s slower than the preceding car, and the distance is longer than 20 m, this trajectory will be followed. It increases the velocity about 2 m/s and decreases the distance about 5 m.

A.10 Short distance

When the distance is shorter than 7 m this trajectory is chosen. Here is the reference velocity put to zero, and the reference distance to 15 m, to get the car to slow down quickly.

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/>