Model Predictive Control of a drivetrain with a Starter/Generator

Zandra Jansson

Model Predictive Control of a drivetrain with a Starter/Generator

Examensarbete utfört i Fordonssystem
vid Tekniska Högskolan i Linköping
av

Zandra Jansson

Reg nr: LiTH-ISY-EX-3082

Supervisor: **Michael Back, DaimlerChrysler**

Examiner: **Prof. Lars Nielsen**

Linköping, March 26, 2001.

Författare: Zandra Jansson
Author:

Sammanfattning
Abstract

The aim of this Master thesis is to control a vehicle with a Starter/Generator included in the drivetrain with the help of an on-line control method called Model Predictive Control (MPC). The controller is to be used as a cruise controller and attempts are made to lower the fuel consumption with the help of MPC. The control signals are the throttle angle and a current. The reference signal is velocity.

The control work is divided into two parts, linear and nonlinear MPC. For the linear MPC the vehicle model is firstly linearized and the obtained linear model is used in the making of the linear controllers. The controllers are derived for different criteria and around different working points, and tested on the nonlinear model. The following of the reference signal is good, but it is not possible to directly influence the fuel consumption. To be able to cover a larger reference velocity range the linear controllers are combined in two different switching strategies, were a supervisor decides what controller to use in a particular instant.

To be able to more directly influence the fuel consumption, nonlinear MPC is tried. A complete numerical solution would require substantial calculation time so a approximative method is used. The approximative method is tested for two different criteria, were one directly includes the fuel consumption. The approximative method does not give as good a result as the linear controllers with regards to the reference following and with regards to the fuel consumption the result is hard to interpret since the control values assume unrealistic values.

Abstract

The aim of this Master thesis is to control a vehicle with a Starter/Generator included in the drivetrain with the help of an on-line control method called Model Predictive Control (MPC). The controller is to be used as a cruise controller and attempts are made to lower the fuel consumption with the help of MPC. The control signals are the throttle angle and a current associated with the torque that can be obtained from the Starter/Generator. The reference signal is velocity.

The control work is divided into two parts, linear and nonlinear MPC. For the linear MPC the vehicle model is firstly linearized and the obtained linear model is used in the making of the linear controllers. The controllers are derived for different criteria and around different working points, and tested on the nonlinear model. The following of the reference velocity is good, but it is not possible to directly influence the fuel consumption. To be able to cover a larger reference velocity range the linear controllers are combined in two different switching strategies, were a supervisor decides what controller to use in a particular instant.

To be able to more directly influence the fuel consumption, nonlinear MPC is tried. A complete numerical solution would require substantial calculation time so a approximative method is used. The approximative method is tested for two different criteria, were one directly includes the fuel consumption. The approximative method does not give as good a result as the linear controllers with regards to the reference following. With regards to the fuel consumption the result is hard to interpret since the control values assume unrealistic values.

Acknowledgments

Notation

Symbols

A_v	Cross-sectional area of vehicle
α	Throttle angle
c_r	Rolling friction coefficient
c_W	Air resistance constant
F_{airres}	Force on vehicle caused by air resistance
F_B	Force on vehicle caused by breaking
$F_{incline}$	Force on vehicle caused by the incline of the road
$F_{rollingres}$	Force on vehicle caused by the rolling resistance
F_R	Sum of the resisting forces acting on the vehicle
g	Gravitational acceleration
i	Gear ratio
m_{eff}	Effective mass
M	Output signal horizon
M_{engine}	Engine torque
M_{SA}	Torque given by the starter/generator
N	control signal horizon
η_G	efficiency of the gear
n_{engine}	rpm in engine
n_{wheel}	rpm of the wheel
r	reference signal, reference velocity
r_{wheel}	wheel radius
ρ_{air}	air density
s	distance
v	velocity
$\overline{x}$	a vector
X	a vector in time
γ	angle of incline

Abbreviations

Consump.	Fuel Consumption
GPC	General Predictive Control
ICE	Internal Combustion Engine
Lin.	Linearisation
MPC	Model Predictive Control
Ref.	Reference velocity

RHC	Receding Horizon Control
RPM	Revolutions Per Minute
SOC	State Of Charge
Working p.	Working point

Contents

List of Figures

List of Tables

1 Introduction

This master thesis was made for and with the help of the DaimlerChrysler research and development department FT2/EA, powertrain control, in Esslingen am Neckar, Germany. The aim of the thesis is to control a nonlinear model of a vehicle, including a Starter/Generator in the drivetrain, with the help of a regulation method called Model Predictive Control (MPC). The controller is to be used as a cruise controller. Attempts will also be made to lower the fuel consumption of the car with the help of MPC. A presentation of linear and nonlinear Model Predictive Control theory is also included.

1.1 Background

Today we can all be said to be part of the information society. With the help of recent developments, such as wireless technology, the information is also readily available in new places, for example in cars and trucks. It is possible to get information about such important things for driving as the speed limit and incline of the road, and based on these facts make decisions about the driving strategy. It would be interesting to start taking more advantage of these known facts.

Another recent development can be seen when looking at the price of fuel. It has become expensive at the same time as people are becoming more and more aware of the effect of car exhausts on the environment. A car that has a low fuel consumption is attractive for consumers not only because it saves its owners money but also because it saves the environment. This is where an additional electrical source of power, the Starter/Generator, comes into the picture. The normal car of today can spare about 14 volts for loading of the car battery and the running of the air conditioning , the electrical seat adjustments, the on-board computer and other electrical appliances. This number would be raised to about 42 volts with the new Starter/Generator. Not only does this give new opportunities for including comforts in the car, it also gives a new dimension of freedom when optimizing the fuel consumption. The reason for this is that the torque required for reaching a certain speed can be obtained in part from the Starter/Generator and in part from the normal combustion engine.

Model predictive control is a control method which takes advantage of the information available and where in theory any criterion, such as a lower fuel consumption, can be implemented. This is why this particular control

method is chosen for this study.

1.2   Method

Firstly in the work of this thesis a literature study of Model Predictive Control methods is made. A selection of the methods are then chosen and applied to the nonlinear model $W203_{E20ML}$, a vehicle containing a Starter/Generator in the drivetrain. The model is implemented in Simulink and the data used for the model is provided by DaimlerChrysler. For the purpose of obtaining linear MPC controllers the model is linearized. The controllers obtained with the help of linear MPC and this linearized model are applied to the nonlinear plant. Then the linear controllers are combined in order to cover a larger velocity range with the help of so called supervisors. Finally nonlinear MPC is tried. The linearization, the linear controllers and the nonlinear controller are calculated with the help of Matlab and implemented in Matlab/Simulink.

1.3   Thesis Outline

The work done during the thesis and the concepts it contains are described in the following chapters.

Chapter 2 The Vehicle Model The model of the vehicle and its engine is presented. The assumptions made about what information is known beforehand and the physical limits to the system signals are also included in this chapter.

Chapter 3 MPC theory An introduction to Model Predictive Control, linear and nonlinear, is given. Also the history of MPC is presented.

Chapter 4 Control The nonlinear model is linearized around several working points and linearized controllers are calculated for different minimization criteria and applied to the nonlinear system. A nonlinear MPC controller is also tested.

Chapter 5 Tabular Results from simulations made for the different control strategies are presented here.

Chapter 6 Future work and extensions Extensions to the work done in the thesis can be found here.

Chapter 7 Conclusions The conclusions drawn from this work are presented and discussed.

2 The Vehicle

To control a drivetrain with a Starter/Generator with the help of Model
Predictive Control is the aim of this thesis. To be able to get a satisfactory
result when working with Model Predictive Control it is, as the name of the
method implies, crucial to have a good working model of the plant that is to
be controlled. A model needs to describe the characteristics of the vehicle
and engine well and at the same time not be unnecessarily complex.

The engine that is the base for the model used here is a M111 inserted in
the car model $W203_{E20ML}$, which is a new Mercedes C-class from Daim-
lerChrysler. The engine is an internal combustion engine (ICE) with spark
ignition, also called an Otto engine. The reason for looking at an engine
of the Otto type, as opposed to studying the other type of ICE engine, the
diesel engine, is that they are worse than the diesel engine from a fuel con-
sumption aspect. Thus the Otto engine has a higher optimization potential.
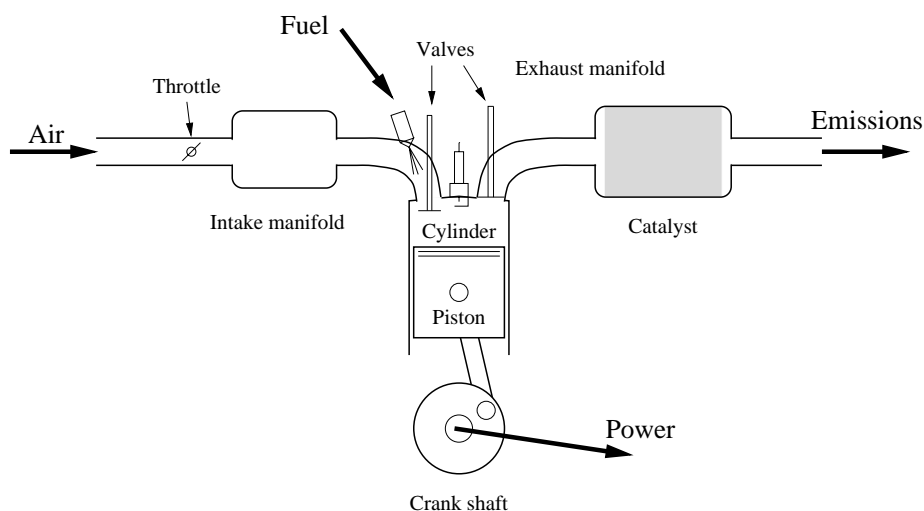
Figure 2.1: An internal combustion engine

A vehicle with an Otto engine works on the following basic principle: The
drive force for moving the vehicle is obtained when fuel mixed with air is
ignited. The amount of air that is used is regulated by the throttle. The
combustion moves a piston that turns the crank shaft. This results in a
rotational speed n, and a torque M_{engine}, of the crankshaft (figure 2.1). The

torque and the rotational speed are transformed into motion of the vehicle via the gear-box. The Starter/Generator (section 2.1) can, depending on n and a current I from the battery, add a torque to the one given from the engine.

For the control of the vehicle in this thesis the throttle angle α and the current I are considered to be the control signals, the wished for speed is the reference signal and the speed given by the model and the distance traveled are the outputs. Additional inputs in the model is the gear, the breaking force and the incline of the road (figure 2.2).
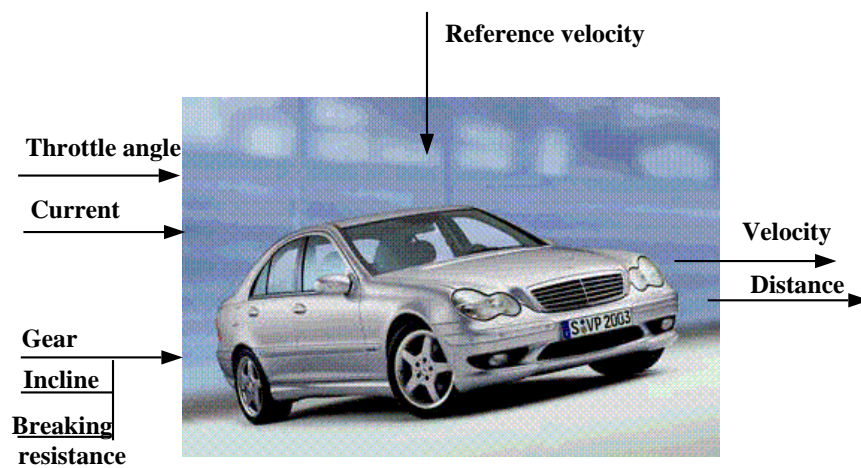
Figure 2.2: The reference, input and output signals of the modelled vehicle

These three additional inputs and the reference velocity are all considered to be known to the system. The speed limit, the incline and the curvation of the road are known beforehand from for example an ordinary map. The speed limit can be a reasonable reference velocity.

All physical actuators have limits, and this is true also for the ones included in this system. The throttle angle can only take a value between zero and almost ninety degrees and the current is also limited. It has its maximum around 150 A. Here all work is done with models, and this means that the result of a throttle angle of $600°$ and a current of 1000 A is a possible result. It must then be remembered that this is only mathematical results, which could not be reached in real life.

When modeling the engine of the vehicle there are some particularities that

need to be described that put up a problem. One of those is the strongly nonlinear connection between the rotational speed of the crankshaft n and the throttle angle α on the one hand and the delivered torque of the engine, M_{engine} one the other. It is also necessary for the purpose of this thesis to get information from the model about the amount of fuel used. This too is expressed by a nonlinear relationship between the two torques given from the engine and the Starter/Generator, M_{engine} and M_{SA}, the revolutions per minute of the crank shaft, n, and the fuel consumption. Because the mathematical foundation for describing these effects are very hard to obtain, these relationships will be expressed with measure-based maps.

Another function of the car that is modeled is the switching of the gears. If the focus of the study would be on dynamical changes it would be especially important to have a good model of gear switching. In this case, however, the main issue is the longterm behavior of the car, and each gear-shifting lasts only a very short period of time of the entire driving session. Therefore the assumption is made that the gears shift in steps, going instantly from one constant value to the other. The clutch signal can be totally disregarded for the same reason.

2.1 Engine with a Starter/Generator

At present, a car with a normal Otto engine is also equipped with a small electrical engine for starting, and a separate generator to load the battery that provides the start engine with power. The generator also supplies the electrical appliances in the car with power. It is run over a belt, and this fact sets a limit to the amount of power that can be obtained efficiently from the generator, to around 2 to 3 kW. It is possible to go above these limits, but the losses in the generator then turn out to be bigger than what can be considered acceptable. At the same time, the amount of applications in a car that use electrical power has not reached its maximum, but is instead steadily increasing. For example there are small electrical motors for seat adjustments, on-board computers, navigation systems and air conditioning. One solution to the problem of an increased demand for electrical power is the Starter/Generator. The Starter/Generator takes the role of both the start engine and the generator. It is located directly on the crank shaft and uses magnetic field to affect the shaft with a torque, M_{SA}. This torque can both work with the torque from the engine and against it. Energy can both be used to help the Otto engine and be stored in a battery.

The amount of energy loaded in the battery is measured with the state of

charge (SOC). This energy can, as mentioned above, be used to help the engine impel the car and thereby save gasoline or be saved in the battery for future needs. The SOC can of course not be allowed to go under a lower limit, since enough energy must be stored for starting the vehicle.

2.2   Vehicle model

The model of the vehicle containing an engine with a Starter/Generator is expressed in the state space form and has Newtons second law as its base:

$$F = m_{eff} \cdot a \tag{2.1}$$

The effective mass, m_{eff}, of the vehicle serves as an energy storage. The effective mass is the total vehicle weight plus an extra weight of ten percent of the vehicle mass. This is to compensate for the effects of the rotating masses of the drivetrain [1]. To reach the state space formulation that is wanted, the acceleration a, is two times integrated. The second energy storage is the battery, with its state of charge, SOC. It is modeled as an integration over the current, multiplied with a current-depending efficiency. This results in the following state equations:

$$\begin{aligned} \dot{s} &= v \\ \dot{v} &= \frac{1}{m_{eff}}\left[F_D - F_R - F_B\right] \\ \dot{SOC} &= \eta I \end{aligned} \tag{2.2}$$

The third state, the state of charge, will not be used in the deriving of the control laws, and will therefore be omitted in the following presentations of the state space description of the vehicle model.

The acceleration in equation (2.2) is obtained from a sum over the applied forces divided with the effective mass of the vehicle system. The forces are the driving force F_D, the resistance force F_R and the breaking force F_B. They are expressed with the relations expressed with equations (2.3).

$$
\begin{aligned}
F_D &= \tfrac{1}{r_{rad}} i \eta_G M_{tot}(\alpha, n, I) \\
F_R &= F_{airres} + F_{rollingres} + F_{incline} \\
F_{airres} &= \tfrac{1}{2} \rho_{air} A_v c_W v^2 \\
F_{rollingres} &= m_{eff} g c_r \cos\gamma \\
F_{incline} &= m_{eff} g \sin\gamma \\
M_{tot} &= M_{engine}(\alpha, n, I) + M_{SA}(n, I)
\end{aligned}
\tag{2.3}
$$

where the rotational speed of the engine is given from the following expression:

$$
\begin{aligned}
n &= \tfrac{v}{r_{wheel}} i \tfrac{30}{\pi} \\
i &= \tfrac{n}{n_{wheel}}
\end{aligned}
\tag{2.4}
$$

The complete model of the vehicle, including a drivetrain with a Starter/Generator is expressed with the help of equation (2.2) to (2.4) in equation (2.5).

$$
\begin{aligned}
\dot{s} &= v \\
\dot{v} &= f(\alpha, n, v) = \frac{1}{m_{eff}} \Big[\frac{1}{r_{wheel}} i \eta_G M_{tot}(\alpha, n, I) - \frac{1}{2} \rho_{air} A_v c_W v^2 - \\
&\quad - m_{eff} g c_r \cos\gamma - m_{eff} g \sin\gamma - F_B \Big]
\end{aligned}
\tag{2.5}
$$

It was previously mentioned (section 2) that there are strong nonlinearities in the vehicle that are hard to express as mathematical functions. To deal with these maps are used. One map describes the relationship between the rotational speed of the engine n, the throttle angle α and the delivered torque of the engine M_{engine} (figure 2.3). The other map shows how much fuel is used in relationship to the torque M_{engine} and the revolutions per minute of the crank shaft (figure 2.4).

The vehicle model is implemented in Matlab/Simulink with the help of the state space expressions and maps presented. The structure of the model can be seen in figure 2.5. For the exact Simulink model see appendix A.
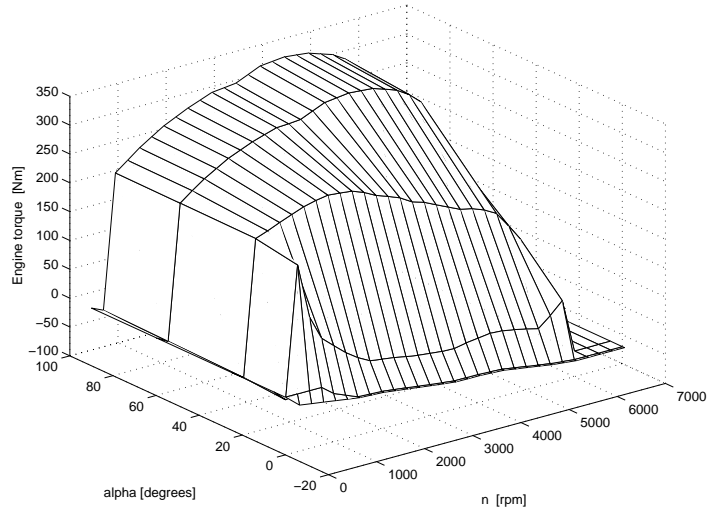
Figure 2.3: The engine torque depending on the throttle angle and the engine speed
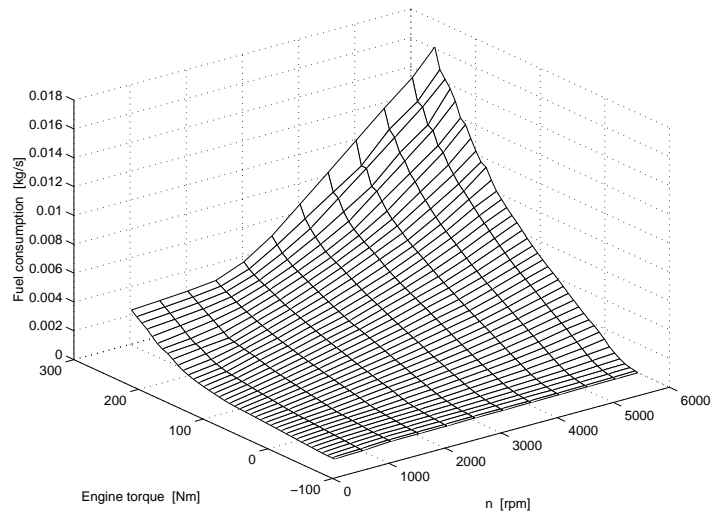
Figure 2.4: The fuel consumption depending on the engine torque and the engine speed
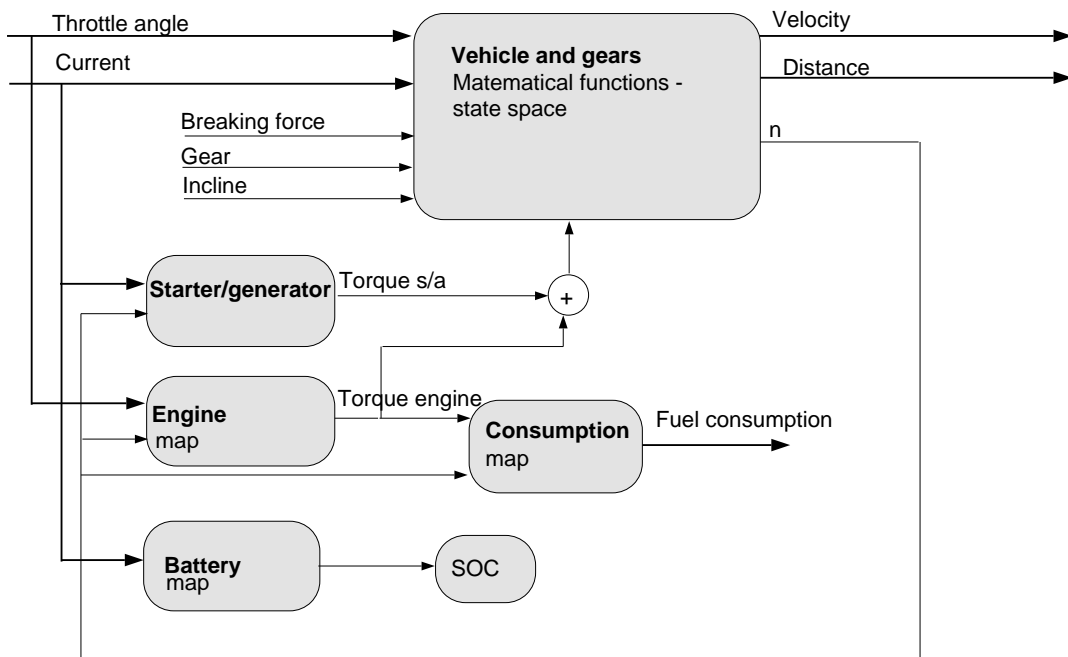
Figure 2.5: The vehicle and drivetrain model

3   MPC theory

3.1   Introduction to Model Predictive Control

The fundamental idea when working with Model Predictive Control is to express the output signals, $\overline{y}$, in terms of past and future control signals, $\overline{u}$, and obtain a forecast of process behavior with the help of a process model. The next step is then to find the future control strategy which will optimize this forecast according to some criterion.

For a short introduction to MPC, see [2], and for a more complete presentation see [3].

The MPC strategy is not only one fixed method. Instead there are a number of slightly different methods, all based on the same underlying ideas. They all use a model of the process explicitly to obtain the control signal by minimizing an objective function. There are some steps that can be said to be the core of model predictive control and at least some of these steps are included in each different variant of MPC. These steps are:

1. At present time t, calculate or predict the output $\hat{y}(t+k|t)$ from the process, over a time horizon M. These outputs will depend on future control signals $\overline{u}(t+j), j = 0, \ldots, N$, and past outputs and control signals. These are known. The output prediction is obtained by using a process model , hence Model Predictive Control.

2. Choose a criterion based on the variable $\hat{y}$ and the states $\hat{x}$ and optimize it with respect to $\overline{u}(t+j), j = 0, \ldots, N$ under the chosen constraints. The current state of the plant is given as the initial state.

3. Apply the first control signal $\overline{u}(t)$ given by the optimization to the plant.

4. At time $t+1$ go to 1 and repeat.

The constant moving of the time horizon one step forward leads to a so called receding horizon. MPC can be called a receding strategy. One could say that this strategy is similar to driving a car. The driver knows the desired reference trajectory for a finite control horizon. With this knowledge and the knowledge of the car characteristics he or she decides which control actions, such as accelerating or breaking, are necessary. Only the first control action is taken at each instant.

It should be noted that the first three steps above represent an open-loop
control strategy. If all the obtained control signals were to be applied to the
system and there were some sort of error, for example a imperfect model
or disturbances, these errors would never be fed back into the system. The
vital thing that turn the entire MPC strategy into a closed-loop control lies
in the fact that only the first control signal, $\overline{u}(t)$, is applied. The process is
done all over again, and as can be seen in point 2, the current state of the
plant is taken as initial value. This means that errors in the system will be
taken into account, and the control strategy is a closed-loop strategy.

There are several tuning parameters in the MPC algorithm, and they should
be chosen with care to get the desired result. They are:

- M. M is the time horizon for $\overline{y}$, the outputs. M should be chosen so
  that it covers the settling time for the system, thus making it possible
  to look beyond to the more stable behavior. It should at the same time
  be avoided to chose a to big value on M, since the size of the output
  horizon influences the calculation time of the optimization problem.

- N. N is the time horizon for $\overline{u}$, the control signals. It is often a smaller
  number than M, since it more or less decides the size of the optimizing
  problem.

- V. V is the criterion to be optimized. This is maybe the most impor-
  tant parameter. If the criterion is an unreasonable one that the system
  can not fulfill, this will not show in any way except by the result from
  the controlled plant. It is up to the designer of the control system to
  chose reasonable, sensible criterion.

The list is based on information found in [2, 3].

As a more practical explanation of how to derive a control law with the
help of MPC an example will be shown. The number of states, the control
signals and the values on M and N will be exactly the same as the ones
corresponding to the linear MPC of the vehicle system in this thesis.

First the model is stated in discrete state space form:

$$\overline{x}[n+1] = A_d\overline{x}[n] + B_d\overline{u}[n]$$
$$y[n] = C_d\overline{x}[n]$$

(3.1)

*$\overline{x}$ is the state space vector containing the two states $x_1 = s$ and $x_2 = v$.
$\overline{u}$ is in the same way the vector containing the control signals $u_1 = \alpha$ and*

$u_2 = I$. *Big letters symbolize a vector in time. In this thesis the output and the reference are both scalar. In theory, there is no problem with the output and reference being vectors containing several signals.*

The next step is to chose a criterion to be minimized. The simplest one is perhaps to use only the difference between the output and the reference value, in this case shown with a quadratic norm and a weighting matrix Q_y:

$$
\begin{aligned}
V_{ex} &= \sum_{i=1}^{M}(y_i - r_i)^T Q_y(y_i - r_i) = \\
&= |Y - R|_{Q_y}
\end{aligned}
\tag{3.2}
$$

The criterion expressed with the help of the linear, discrete state space model takes the form seen in (3.3).

$$
\begin{aligned}
V_{ex} &= \left| \begin{bmatrix} y(t+M) \\ \vdots \\ y(t+1) \end{bmatrix} - \begin{bmatrix} r(t+M) \\ \vdots \\ r(t+1) \end{bmatrix} \right|_{Q_y} = \\
&= \left| \begin{bmatrix} C_d A_d^M \\ C_d A_d^{M-1} \\ \vdots \\ C_d A_d \end{bmatrix} \overline{x}(t) + \begin{bmatrix} C_d A_d^{M-2}B & C_d A_d^{M-1}B \\ C_d A_d^{M-3}B & C_d A_d^{M-2}B \\ \vdots & \vdots \\ 0 & C_d B_d \end{bmatrix} \begin{bmatrix} \overline{u}(t+N) \\ \vdots \\ \overline{u}(t) \end{bmatrix} - R(t) \right|_{Q_y} = \\
&= \left| H\overline{x}(t) + S\overline{U}(t) - R(t) \right|_{Q_y}
\end{aligned}
\tag{3.3}
$$

Here, and in the rest of the control work done in this thesis the change in the control signals is considered to be zero after the point in time N. Q represents the weights put on different variables in the criterion to be minimized. Only the relationship between the weights is important, so in the example here weighting does not make sense. In other criteria however, weights are an important way to influence the control structure, by for example giving a heavy weight on a variable important to keep small. Thus the variable is forced to a small value when the criterion with variables times the weights is minimized.
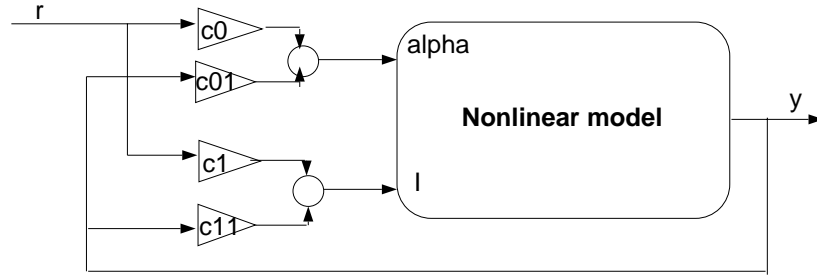
Figure 3.1: The obtained control structure when using a unconstrained, quadratic criterion

The most common form of constraints are $u_{min} \leq u \leq u_{max}$ but for this simple example no constraints will be taken in to account. This turns the optimization of V_{ex} into a straightforward derivation problem:

$$\frac{\partial V_{ex}}{\partial \overline{U}} = 2 S^T Q_y (H\overline{X} + S\overline{U} - R) \tag{3.4}$$

The optimal control signal, $\overline{U}^*$, is given by $\frac{\partial V_{ex}}{\partial \overline{U}} = 0$. The solution is:

$$\overline{U}^* = -(S^T Q_y S)^{-1} S^T Q_y (H\overline{x} - r) \tag{3.5}$$

From this solution only $\overline{u(t)}$ is applied to the model and then the process starts over again.

If the model is linear and the criterion chosen is in a quadratic form as in the example, and unconstrained, it is possible to find an analytical solution to the optimization of V. The solution to the control problem will give a $\overline{u}(t)$ were each control signal u_i is on the form:

$$u_i = c_0 y + c_1 r \tag{3.6}$$

With this result the constants c_0 and c_1 only need to be calculated once, off-line, and can then be applied to control the process. It is exceptionally easy to implement (figure 3.1).

If an analytical solution is unobtainable the minimization has to be done in each time step, be applied as a controller and calculated again in the next time step, as described above. The controller is said to be calculated on-line.

The strength of MPC lies in the four steps listed above. It is easy to take constraints of different kinds into account, they only need to be included

in the criterion to be optimized. A multi-variable case can be dealt with. MPC can also be used to control a wide variety of processes, both linear and nonlinear (section 3.2). The method of course also has weaknesses. One is that the derivation of the control law can be very complicated, even more so if the criterion has constraints. Another drawback is the need for a good model of the process to be controlled. These models can be very hard to obtain, especially for a nonlinear system.

3.2   Nonlinear Model Predictive Control

When a system has a mild nonlinearity, it might be possible to control it with a linear controller if this controller is carefully tuned. For a system with strong nonlinearities it is more interesting to make use of a nonlinear controller. The fundamentals of MPC theory is such that it can be applied to linear and nonlinear systems alike. In analogy with equation (3.1) a general nonlinear problem with the criterion to be minimized can be defined as:

$$
\begin{aligned}
&min_{\overline{u}([k|k],\dots,\overline{u}[k+N-1])} V = \sum_{j=1}^{M} (l(\overline{x}_j, u_j)) \\
&\overline{x}[n+1] = f(\overline{x}[n], \overline{u}[n]) \\
&y = g(\overline{x}, \overline{u}) \\
&\overline{u}[k+j] \in U, j = 0, \dots, N-1 \\
&\Delta \overline{u}[k+j|k] = 0, j \geq N
\end{aligned}
\tag{3.7}
$$

In the expression above the model and the criterion are nonlinear. Both these properties will lead to a nonlinear controller, as will constraints on the criterion. The problems with obtaining the nonlinear controller are mainly of a practical nature. It can be very hard to optimize a nonlinear problem mathematically. An analytical solution most often does not exist. A complete numerical solution can require a substantial computational time, most often longer than what is acceptable in an on-line controller.

The whish for short calculation time combined with the whish for a nonlinear controller has lead to several methods that mix linear and nonlinear solutions to different degrees. One method that bases itself fully on linear MPC is to linearize the system around several working points and with the help of those models calculate linear controllers. The controllers are then combined, and each controller is used to regulate the nonlinear plant in a area around the working point it was created for.

Another method that tries to take the nonlinearity more into account is described in [4]. This method is the one used for the nonlinear control in

section 5. The idea is to compute the control sequence by calculating the criterion at time t completely numerically and approximate the criterion in the following time steps, $t+1$ to $t+N$, with a linearization. The optimization problem this leads to can be expressed as follows:

$$
\begin{aligned}
&min_{\overline{u}[t|t]} V \\
&\overline{x}[t+1] = f(\overline{x}[t], \overline{u}[t]) \\
&\overline{y} = g(\overline{x}, \overline{u}) \\
&\overline{u}[t|t] \in U \\
&\Delta\overline{u}[t+i|t] = linearcontrol(\overline{x}[t+j|t]), j = 1, \ldots, M-1
\end{aligned}
\tag{3.8}
$$

If one compares expression (3.7), the complete numerical solution of the nonlinear problem, with expression (3.8), the approximative solution, there are two major differences. The first one is that in (3.7) the number of variables to be optimized are the number of inputs times the control horizon but in (3.8) the optimization variables are only the number of inputs, regardless of the control horizon. It is good for stability to choose a rather large control horizon, but this is in normal MPC hindered in some degree by the fact that the number of optimization variables are the control horizon times the number of outputs. When using expression (3.8) the optimization is done with regard to the present time step of the control signals only. N does not affect the number of optimization variables and can therefore be chosen big.

The second difference is that the control horizon for (3.7) is the chosen control horizon N. In (3.8) the control horizon instead equals the output horizon M. The reason for this is that the on-line computation of the controller in (3.8) is not affected by N. To get an overview the differences between a complete nonlinear solution and one of the many approximative methods for solving nonlinear problems, see table 3.2.

	Controller (3.7)	Controller (3.8)
Optimization Variables	number outputs* N	number outputs
Control Horizon	N	M

Table 3.1: Two controllers for nonlinear systems

3.3   History of predictive control

The history of Model Predictive Control can be said to start in the 1970s
when several articles concerning predictive control appeared. The processing
industry also began to take notice of this control method. Several branches
of predictive control developed more or less independently. One was what
is now called Model Predictive Control (MPC). Two of the first varia-
tions within this branch where called Model Predictive Heuristic Control
(MPHC), later known as Model Algorithmic Control (MAC) and Dynamic
Matrix Control (DMC). Both these algorithms explicitly use a dynamic pro-
cess model but where MAC uses the impulse response, DMC uses the step
response. They took advantage of the increasing potential of digital comput-
ers at the time [3]. A second branch of predictive control, General Predictive
Control (GPC), was developed in the field of self tuning [5]. A third branch,
called Receding Horizon Control (RHC) was first developed as a variation of
Linear Quadratic Gaussian Control (LQG). All these variations of predictive
controllers can be seen as relatives to controllers from optimal control and
to linear programming [3].

MPC became popular especially in the chemical process industries. This
was in part due to the simplicity of the algorithm and in due to the use of
impulse or step response model, which was often preferred to input/output
or state space models as being more intuitive. In time however, MPC was
also formulated in the state space context, and that is the way it will be
used in this thesis.

4 Linear control

4.1 Linearization of the vehicle model

The complete model of a vehicle, with a Starter/Generator included in the engine model, can be linearized around any number of working points (operating points). Each operating point consists of a velocity v_0, a gear i_0, a throttle angle α_0 and a current I_0. The linearization of a state space system is done by derivation of the system with respect to the states and the input signals around the chosen working point [6]. The vehicle model described by equation (2.5) thus gives the linearized model:

$$
\dot{s} = 1(v - v_0)
$$
$$
\dot{v} = \frac{\partial f}{\partial v}(v - v_0)|_{\alpha_0,n_0,v_0,I_0} + \frac{\partial f}{\partial \alpha}(\alpha - \alpha_0)|_{\alpha_0,n_0,v_0,I_0} + \frac{\partial f}{\partial I}|_{\alpha_0,n_0,v_0,I_0}(I - I_0) =
$$
$$
= \frac{1}{m_{eff}}[\frac{1}{r_{wheel}}i\eta_G\frac{\partial M_{engine}}{\partial n}\frac{\partial n}{\partial v}|_{\alpha_0,n_0,v_0,I_0} - \rho_{air}A_v c_W v](v - v_0) +
$$
$$
+ \frac{1}{m_{eff}}[\frac{1}{r_{wheel}}i\eta_G\frac{\partial M_{engine}}{\partial \alpha}]|_{\alpha_0,n_0,v_0,I_0}(\alpha - \alpha_0) \qquad (4.1)
$$

n_0 is given by equation (2.4). The two derivates of the engine torque, M_{engine}, must also be known to be able to make the linearization. $M_{engine}(\alpha, n)$ is expressed by a map (figure 2.4) and the derivates can be obtained by a interpolation of the map values, a known n and a known α (Appendix B, section B.1.1). But the throttle angle is not known, so before the torque derivates can be found the throttle angle must be approximated. The approximation of α is given by observing what throttle angle value the nonlinear model assumes when stabilized at the current working point velocity, v_0, by a rough PID regulator and using this value as the known throttle angle value.

To each operating regime a linear model as seen in (4.1), called local model, is associated. The local models are represented by a continuous state space formulation, where $\dot{\overline{x}}(t)$ represents the system in (4.1) and $\overline{y}(t)$ the system output, as seen in (4.2).

$$
\dot{\overline{x}}(t) = A\overline{x} + B\overline{u}
$$
$$
\overline{y}(t) = C\overline{x} \qquad (4.2)
$$

The differential equation (4.2) describes the system to be controlled, but when implementing the control signals it is common that they are piecewise constant. They are held at the optimal value calculated over one time step [7]. Therefore the continuous representation is transformed to a time discrete state space form and the result is:

$$\overline{x}[t+1] = A_d \overline{x}[t] + B_d \overline{u}[t]$$
$$\overline{y}[t] = C_d \overline{x}[t] \tag{4.3}$$

For each of the local models obtained a controller, called local controller, can be derived with the help of linear MPC theory [4]. When a linearized model and a quadratic criterion, V, are used in the MPC algorithm the exact solution to the MPC optimization problem can be found. As previously explained (section 3.1) this means that one gets the sought controller expressed on the form $\overline{u}[t] = c_0 \overline{y} + c_1 \overline{r}$, which is easy to implement (section 3.1, figure 3.1). Because of this it would be very desirable if the local controllers also could be used for the nonlinear vehicle system.

When one compares the response from the unregulated nonlinear model with that from the unregulated local models they of course vary from each other. As seen in figure 4.1 the output velocities from the nonlinear and the linearized models reach different end values and assume different shapes. Still the similarities are such that there can be good hope of being able to control the nonlinear model with the controllers obtained by using the linear models.

For all the work done with linear MPC in this thesis the same output horizon and control horizon are chosen. To look beyond the initial state of the system a big enough value must be chosen for M. Also, the output horizon must be long enough so that it is really possible for the system to fulfill the criterion. However, M also affects the computational time of the optimization, and should for this reason not be to big. The control horizon N should be chosen as small as possible while still being big enough so that it is possible for the system to fulfill the criterion, that is reach the desired velocity, and still not be to big because of its affect on the size of the optimization problem. Several different values are tested. For example with an output horizon of ten and a control horizon of two, the system is unable to reach the value of the reference signal. Finally the values of ten seconds for the output horizon M and four for the control horizon N are found to be good compromises. These values are used throughout the entire work with linear MPC here.
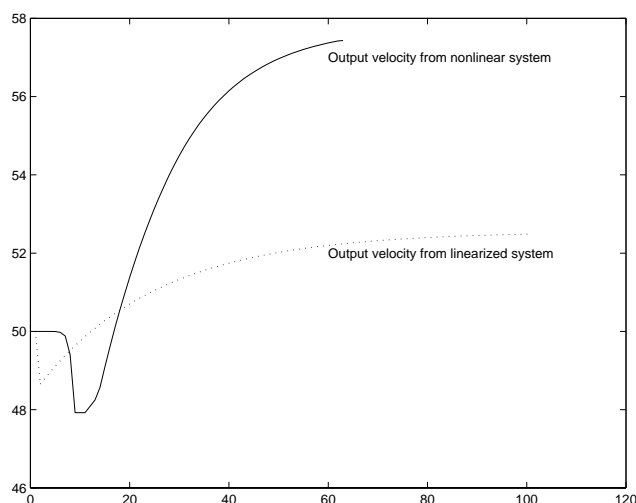
Figure 4.1: Output velocities from the nonlinear model and one linearized model, when excited with a throttle angle of seven°and a current of zero A. The x-axis is the time and the y-axis the velocity in km/h

4.2 Control with criterion one

The first attempt to control the nonlinear model of the vehicle is made with the quadratic criterion V_1.

$$V_1 = (Y - R)^T Q_y (Y - R) + \overline{U}^T Q_u \overline{U} \tag{4.4}$$

This criterion takes into account the error between the reference value, r, and the output of the system, y, and it weights the control signals, $\overline{u}$, over the output horizon M and the control horizon N. This should lead to the reference velocity being followed and to moderate values on the control values. If u_1, that is α, is kept rather small this should also have a positive effect on the fuel consumption, since it is indirectly influenced by the throttle angle (figure 2.3 and 2.4).

Q_y and Q_u are weighting matrixes. If there for example is a output horizon of three and a control horizon of two the weighting matrices will be:

$$Q_y = \begin{pmatrix} k_0 & 0 & 0 \\ 0 & k_1 & 0 \\ 0 & 0 & k_3 \end{pmatrix}$$

$$Q_u = \begin{pmatrix} k_4 & 0 \\ 0 & k_5 \end{pmatrix}$$

By giving for example Q_y a heavier weight than Q_u using small input values is prioritized. Using only a little current and a small perturbation in the throttle angle is then considered more important than that the reference velocity should be closely followed. This happens because the minimization of V forces the signals with a heavy weight to be small, so that the total value of V will be small. In the criteria described in this thesis the foremost concern is that the vehicle reaches and holds the velocity given as a reference and the weights are adjusted to fulfill this concern. When weights are considered good they make the system reach the desired velocity (not over, also not under) and hold it. First when this requirement is fulfilled is it meaningful to try to minimize the fuel consumption at each speed by distributing the desired torque needed for this speed between the ICE and the Starter/Generator.

There is one decision to make when one tries to find the best weights for the system, and that is if the weights should be tuned so that the local controller is as good as possible for its corresponding local model, or if the controller instead should be tuned to fit the nonlinear system. Here the second choice is made, the tuning is made with the nonlinear system as a reference. If the weighting leads to making the controller better in one area and worse in another, it is prioritized that the controller gives a good result in the area around its working point.

The attempt to control the nonlinear $W203_{E20ML}$ model with linear controllers is for criterion (4.4) tried around four different working points. They are presented in table 4.1.

linearization point	speed v_0 km/h	gear i_0	throttle angle α_0
1	40	3	5.6
2	50	3	6.5
3	70	4	7.4
4	100	4	9.5

Table 4.1: Linearization points for V_1 and V_2

Around each of these points the system is linearized and a control strategy is derived with linear MPC theory (for the exact Matlab implementation see

Appendix B, B.1.1 and B.2.1). After that each local controller is tested for a range of velocities on the non-linear plant. The resulting output velocities and the fuel consumption from the plant when controlled with one of the four controllers over a hundred time steps can be seen in section 6, table 6.1. A study of the absolute deviation of the output velocity compared to the reference, r, shows that the resulting velocities follow the reference value tolerably well, but there is room for improvement (table 4.2). As could be expected, the controllers made for the higher working points, 3 and 4, give the best result for the higher speeds. For the lower speeds it is not as obvious that the controllers 1 and 2 are better. They also give a higher fuel consumption (section 6, table 6.1). This is mainly because the lower gear, i_0, used when running local controllers 1 and 2 results in a higher value of RPM, which brings the fuel consumption up (figure 2.4).

reference speed km/h	20	30	40	50	60	70	80	100
linearization point								
1	0.61	1.95	0.49	0.18	0.42	1.41	2	3
2	0.49	1.75	0.2	0.24	0.93	2.03	2.71	4
3	1.27	0.67	0.3	0.37	0.7	0.17	0.24	0.4
4	1.3	0.46	0.04	0.69	1.15	0.68	0.56	0.2

Table 4.2: The absolute deviation from the reference velocity for the local controller corresponding to working point 1,2,3 and 4 under V_1

When continuing the control work in this thesis it will be realized that the results regarding control signals and such for criterion V_1 are almost identical, only slightly worse, than the results for the following criterion V_2. For this reason the next criterion is studied more carefully with regard to the behavior of the control signals and can be considered representative of both linear criterion one and two.

4.3   Control with criterion two

To get a smooth change of the throttle angle a term ΔU can be included in the optimization criterion V. $\Delta \overline{U}$ describes the changes in the inputs α and I between two time steps. The new criterion is expressed by equation (4.5).

$$V_2 = (Y - R)^T Q_y (Y - R) + \overline{U}^T Q_u \overline{U} + \Delta \overline{U}^T Q_{\Delta u} \Delta \overline{U} \qquad (4.5)$$

Again the same four linearization points as for V_1 are chosen (table 4.1) Around these the model of the vehicle is linearized and for each resulting local model a controller is obtained (Appendix B, section B.1.1 and B.2.1). The different controllers each has areas in which they follow the desired velocity, r, close and others where r is not so closely followed (section 6, table 6.2).

reference speed km/h	20	30	40	50	60	70	80	100
linearization point								
1	0.25	0.15	0.0	0.10	0.25	0.41	0.57	1.1
2	0.33	0.13	0.14	0.01	0.20	0.40	0.62	1.3
3	0.88	0.49	0.26	0.19	0.38	0.01	0.27	0.7
4	1.12	0.44	0.0	0.73	1.17	0.75	0.41	0.1

Table 4.3: The absolute deviation from the reference velocity for the local controller corresponding to working point 1, 2, 3 and 4 under V_2

This new criterion gives a better following of the reference speed than V_1 for all controllers, and for each individual speed also, with one or two exceptions. The pattern of the controllers made for low working point functioning well for low speeds and vice versa is clearer here (table 4.3).

When looking at the input signals, that is α and I, it is interesting to see if the inclusion of them in the criterion is enough to keep them at a reasonable level, even though the optimization is done without constraints, or if they turn out to violate the physical constraints of the system. In a study of figure 4.2 it can be seen that the inclusion of the control signals in the criterion is indeed enough in this case. As could be expected the throttle angle peeks with the changing of velocities, and lessens again when the system is back in a steady state and has reached the desired speed.

The control signal I shows much the same behavior as the throttle angle. It is kept within its limits and peeks when the speed changes are taking place, giving the system an extra boost (figure 4.2).

The torque affecting the vehicle is divided between the torque provided by the engine, M_{engine}, and the one from the Starter/Generator, M_{SA}. The predominate part of the total torque is provided by M_{engine}. It is interesting to note that the controllers derived with the higher working points use more torque from the Starter/Generator than the ones that were derived from the lower working points. However, regarding the engine torque the situation is
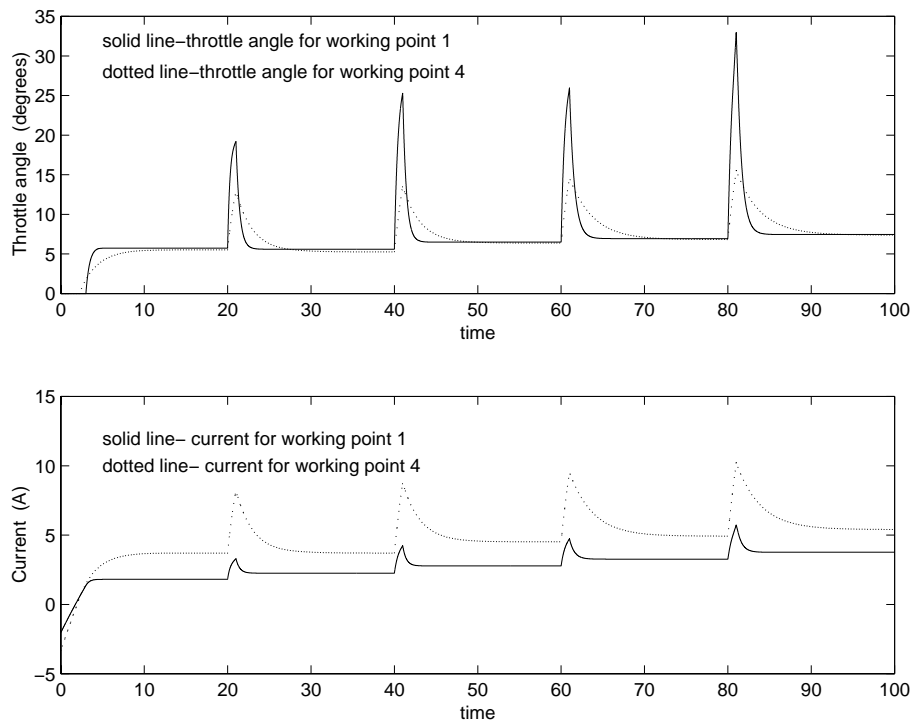
Figure 4.2: The throttle angle control signal and the current control signal given to the nonlinear velocity model when controlled with linear MPC controller for working point 1 and 4

reversed (figures 4.3).

When comparing the fuel consumption using the two different criteria V_1 and V_2 it can be seen V_1 gives lower fuel consumption compared to V_2 in the lower velocity range, and V_2 gives a lower consumption compared to V_1 in the higher velocity range. These differences are very small, to small to say definitely if one or the other criterion is better for the fuel consumption aspect. Criterion two is still to be preferred, since it gives a good, smooth following of the reference velocity.

The next step is to investigate if it is preferable to use one of the local controllers obtained from V_2 to control an entire velocity range, as was done in section 4.2 and 4.3 or if the best control result is obtained by using a supervisor, a switching control.
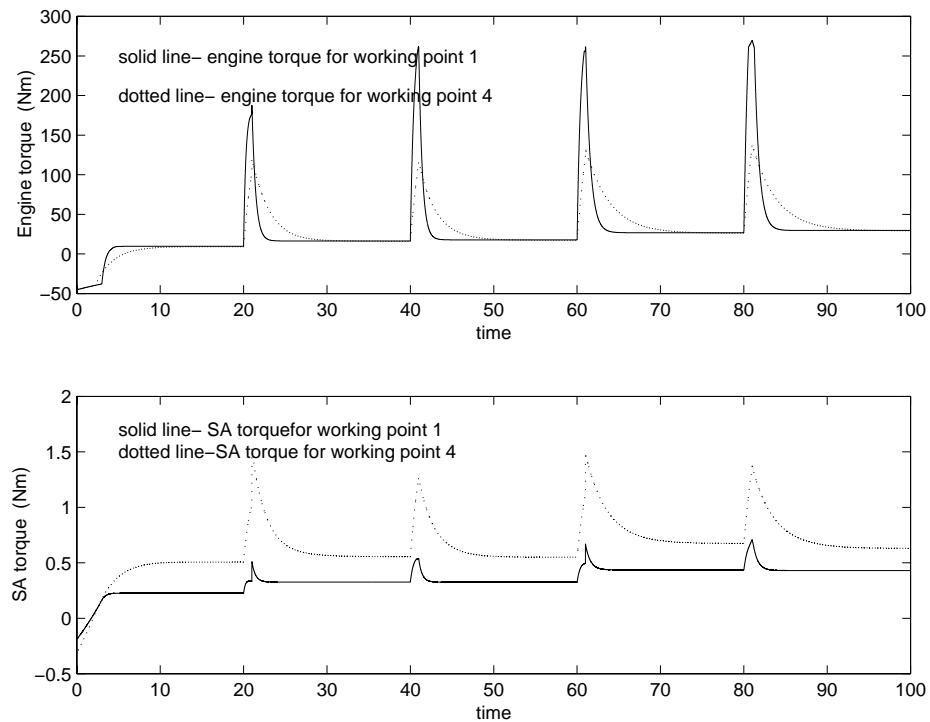
Figure 4.3: The engine torques and Starter/Generator torques when using controllers 1 and 4

4.4   Switching control

Switching control is a way to combine several local controllers so that they together can cover a entire velocity range. The most appropriate local controller is picked for each speed and this picking is handled by a so called supervisor.

The first supervisor tested is based on the idea that the output from the local models decide which local controller to use on the nonlinear model, as described in [4]. The supervisor receives the same inputs as the nonlinear system and applies these to each of the several local models. The outputs are then compared to the reference velocity. The controller corresponding to the local model whos output differs the least from the given reference is chosen as controller for the nonlinear system (figure 4.4).

This strategy has one flaw when used in this context. The decision made
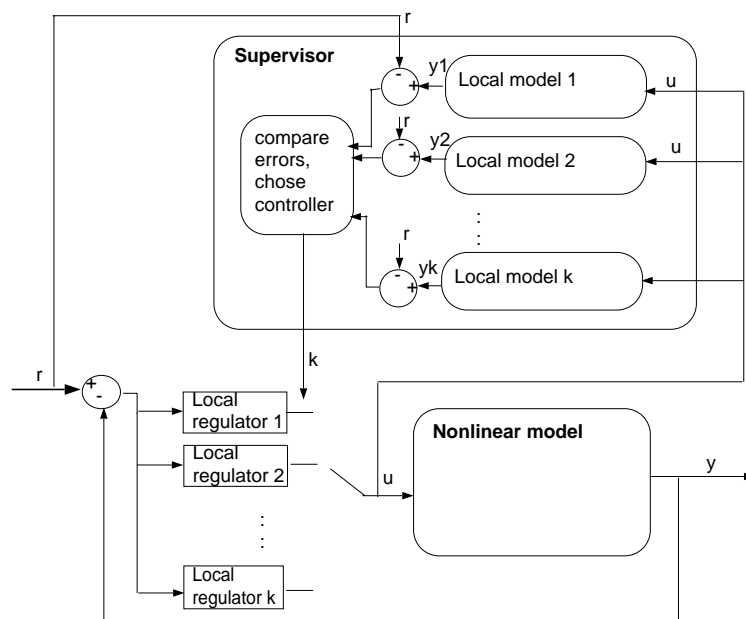
Figure 4.4: Supervisor according to strategy 1

regarding which controller to use to control the nonlinear model in the current operating regime is based on the error between the reference and the controlled linear model output. As was mentioned in section 4.1 the decision was made in this thesis to adjust the weights in the minimization criterion used for the local controllers to suit the nonlinear model best. If for example the smallest error between reference value and linear model output is found to be given from local model one, the local controller one will be picked to control the nonlinear plant. But because of the choice made regarding the weighting matrices it is possible that in fact using local controller two will lead to a closer following of the reference for the nonlinear model. The choice made has the positive effect that when actually controlling the nonlinear plant the MPC controllers follow the reference as well as possible, but the negative effect that the controller picked by the supervisor might not be the optimal one at the current speed for the nonlinear controller. Because of this it can be stated that the above standing supervisor strategy is not a good choice for this particular control problem.

After the realization that the choice made in 4.1 about weights makes the first supervisor inappropriate a second, more basic supervisor structure is

tested. This supervisor has a preset choice of local controllers and the choice is based only on the reference velocity of the car. The switching between controllers is completely rigid. Since the weighting matrixes were chosen in the way explained above, this will ensure that the controller producing the smallest error between reference velocity and output velocity is chosen every time. Such a rigid switching strategy can produce spikes. This is sudden peaks in the input as a result of the abrupt switching from one controller to another. It can result in a strange behaviour of the output signal. As can be seen in figure 4.6 such spikes are not a problem in this case, but if they appear it is not a problem to avoid it. For example one can search for the points were the different controllers produce the same control signal and let the switching take place only in these points.
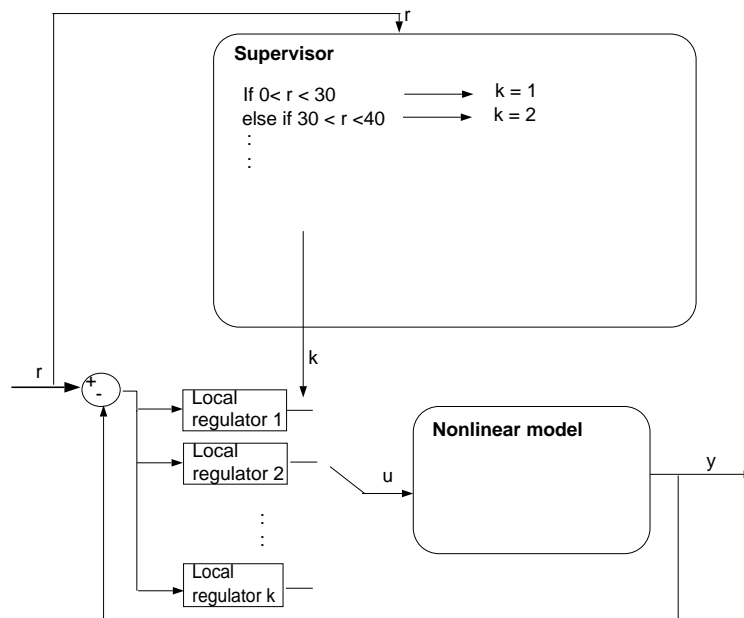
Figure 4.5: Supervisor according to strategy 2

Both supervisors give satisfactory results, but the rigid switcher follows the reference velocity closer. It reaches a better end value and also follows the steps of the reference velocity closer and quicker (figure 4.6).

The conclusion drawn is that switching strategy using Supervisor 2 is best for the purpose at hand. It is now interesting to see if it is in fact best to use a switcher control on the nonlinear model compared to the results from

solid – reference velocity
dashed – rigid switcher output
dotted – supervisor switcher output
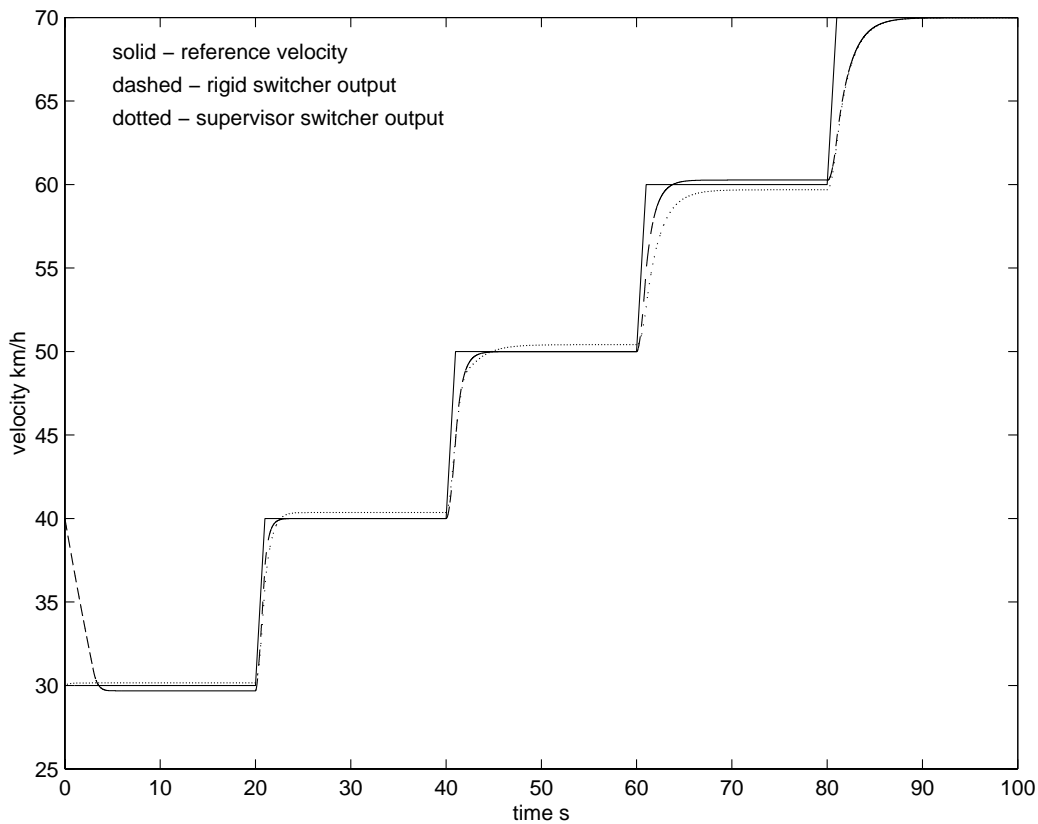
Figure 4.6: Output velocity from the two different switching strategies

using one single local controller. The comparison is made using the local controllers around working point 1, 2, 3 and 4 with criterion V_2. In other words the best of the supervisors is compared with the best of the local controllers.

Because of the structure of supervisor 2 it is given beforehand that it will give the best following of the reference velocity since it picks the most appropriate local controller in each instant. This result can be seen in the deviation from the reference presented in table 4.4.

reference speed km/h	30	40	50	60	70
supervisor 2	0.32	0.00	0.01	0.27	0.00
working point 1	0.32	0.00	0.10	0.31	0.49
working point 2	0.62	0.14	0.01	0.27	0.53
working point 3	1.40	0.70	0.65	0.31	0.00
working point 4	2.86	1.55	1.73	1.15	0.77

Table 4.4: Comparison of output velocity from nonlinear plant controlled with supervisor 2 and local controllers derived with criterion V_2

4.5   Linear control with fuel consumption in the criterion

One way to lower the amount of fuel used by the car at a certain speed could be to put a heavy weight on the throttle angle and a low weight on the current in the criterion to be minimized. This is true because fuel consumption is lower for a lower value on the engine torque, and the torque in its turn is low for small values on α, the throttle angle (figures 2.3, 2.4). Even though the fuel consumption might be brought down by a low value on alpha, this does not mean that efficiency of the engine, regarding the fuel consumption, is good for a low throttle angle value, and this can not be controlled with this linear criterion.

When experiments with the weighting matrices are made, it becomes clear that with a heavy weight on the throttle angle and a low weight on the current, I, the car does not reach the desired speed. When attempts are made to balance the weighting matrices so that the fuel consumption will be kept low but the reference speed will be reached the gain in fuel consumption is minimal or more often none at all compared to the case of V_1 and V_2.

An important question to ask at this point is if it is at all useful to have a MPC controller. Why not use ordinary PID controllers for the throttle angle and the current instead. With the control structure of equation (4.6) it is possible to give "more weight" to the current by using a large constant, K_i, for the integrating term for I, exactly that which did not succeed with weighting matrices.

$$y(s) = G \cdot u(s) = (K_p + \tfrac{K_i}{s} + K_d \cdot s) \cdot u(s) \qquad (4.6)$$

To use mainly the current and just a very small throttle angle brings the fuel consumption down and still the desired velocity is reached. The reason one
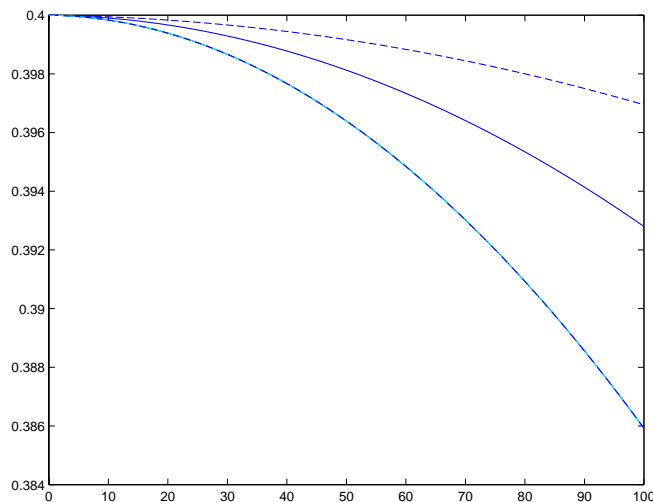
Figure 4.7: State of charge for the nonlinear system controlled with an MPC controller and two PID controllers using a bigger current as input. The x-axis is the time and the y-axis the state of charge, were one corresponds to a charge of 100 percent

gets this result is that the current takes on unrealistically big values. The current has a limit around 150 Ampere. To stay well below this limit but still draw a big enough current to make the fuel consumption go down is possible. But then the problem of the state of charge (SOC) in the electrical battery appears. The PID controllers does not take into account that the charge in the battery keeps decreasing when a too big current is used. In the long run, the battery is simply emptied. In figure 4.7 the up-most line responds to a fuel consumption that matches the consumption reached with MPC, and the two lower ones represent cases of PID control that use a bigger current, thus giving a lower fuel consumption as result. The two PID regulation examples unfortunately also give the result of emptying the battery quick. This leads to the conclusion that lowering the fuel consumption by putting a heavy weight on control signal α is not a good method.

The minimization of V is so far only a minimization of the control input and the difference between the wanted output (the cruise controller) and the actual speed. In an attempt to lower the fuel consumption more, a term that represents this consumption needs to be included in the criterion. The

consumption in this case is dependent of the rotational speed of the engine, n, and the engine torque, M_{engine}. The relationship between n, M_{engine} and the fuel consumption is expressed with a map (figure 2.4). This is a nonlinear relationship and if it was included in V it would not be possible to solve the minimization of the criterion explicitly. An alternative way to include the fuel consumption in the criterion that is to be minimized needs to be found. One can observe that the RPM is constant in a linearization around a certain speed and gear. The engine torque in its turn depends on α and this constant RPM. This lead to the idea to use a polynomial of the second order fitted to the values of $M_{engine}(\alpha)$ at a fixed value for n, instead of the consumption map. The resulting criterion is then:

$$V_3 = (Y - R)^T Q_y (Y - R) + \overline{U}^T Q_u \overline{U} + \Delta \overline{U}^T Q_{\Delta u} \Delta \overline{U} + M_{poly}(\alpha)$$
$$= (Y - R)^T Q_y (Y - R) + \overline{U}^T Q_u \overline{U} + \Delta \overline{U}^T Q_{\Delta u} \Delta \overline{U} + k_0 \alpha^2 + k_1 \alpha + k_2$$

$$(4.7)$$

The minimization of this criterion is possible to solve exactly, just as the criteria (4.4) and (4.5), but because of the constant terms in V_3 the solution also gets a slightly different form. Whereas (4.4) and (4.5) results in a controller that expresses the control signals in constants times the reference value and output value, (4.7) also include constant terms in the controller. These constant terms dominate the controller so much that is it not possible to even reach several desired velocities, even with extremely low weights on the input signals. So to include a term representing the fuel consumption in this form turned out to give unacceptable result, not reaching the desired velocity. So far the use of linear controllers to follow a reference speed has been successful but it seems to be difficult to use these controllers when also taking the fuel consumption into account. The next step is to move on to nonlinear controllers.

5 Nonlinear control

5.1 Complete nonlinear control

Nonlinear control using MPC in this case means using a criterion V that might not be possible to solve analytically. The first attempt is to simply solve the MPC algorithm with the criterion (5.8).

$$V_{1nonl} = (Y - R)^T Q_y (Y - R) \qquad (5.8)$$

The nonlinear part is the output Y, which is the output from the nonlinear vehicle model. The tuning parameters are chosen in accordance with the linear control section to the output horizon of ten, the control horizon of four and Q_y as the unit matrix. With the help of a Matlab function the attempt is made to optimize the throttle angle and the current under (5.8). It is soon realized that this is not feasible. The Matlab functions can not handle a non-linear optimization problem of this scale, and even if it could, the time required for the task is unreasonably large.

5.2 Approximative nonlinear control

A new method must be found for the problem. There are a number of approximate methods that can be used and here the one presented in section 3.2, equation (3.8) is chosen. The tuning parameters are chosen slightly different compared to the ones in section 4, with a smaller output horizon of four and the control horizon of two. The result of choosing these tuning parameters are shown for a simple criterion in equation (5.9).

$$V_{nonlin} = (Y - R)^T Q_y (Y - R) =$$
$$= \left| \left(\begin{pmatrix} y(t+3) \\ y(t+2) \\ y(t+1) \\ y(t) \end{pmatrix} - \begin{pmatrix} r(t+3) \\ r(t+2) \\ r(t+1) \\ r(t) \end{pmatrix} \right) \right|_{Q_y} \qquad (5.9)$$

After these necessary choices have been made the method is implemented in the following way: The first action that is taken is that a local controller is run for one time step. Which local controller to be used depends on the
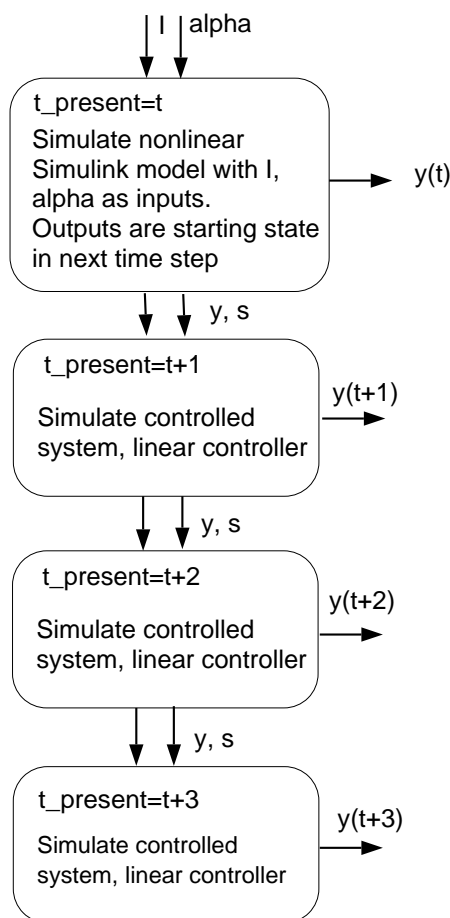
Figure 5.8: Approximative nonlinear method

reference velocity in the present time step. The choice is made in the same way that controllers are picked in switching strategy two.

The resulting values for the throttle angle and the current are used as starting values in the next step, the first time step in the MPC algorithm. In this step the nonlinear, unregulated system is run to with the control signals α and I as variables. After that the regulated system, controlled with a linear controller, is used to approximate output y for the following time steps in the output horizon. All steps are affected by the optimization (and control) variables, α and I, because they affect the first, nonlinear step. The output velocity and distance from this step are of the starting values for the next,

controlled step, so that the current state of the model is the initial values for the following step (figure 5.8).

The result is a typical on-line controller and all the steps above have to be run for each time step of the entire vehicle system, producing a control law for each step.

5.2.1   Control with nonlinear criterion one

The work with the new method starts using criterion V_{1nonl}.

$$V_{1nonl} = (Y - R)^T (Y - R) \tag{5.10}$$

This criterion concentrate only on reaching the given reference velocity and does not take the input signal or the fuel consumption into account at all.

The resulting output and fuel consumption when the system is simulated with a velocity profile going from thirty to seventy in steps, each step twenty time units long and a gear profile using third gear for velocities 30-50 and fourth gear for velocities 60-70 can be seen in section 6, table 6.4. The deviation for the reference velocity is presented in table 5.5. The resulting output velocities are acceptable over the entire velocity range, using this one controller, but the fuel consumption is higher than when using the Supervisor 2 as control, and also higher than when using each of the local controller derived from criterion V_2 separately.

reference speed km/h	30	40	50	60	70	consump. l
V_{1nonl}	0.23	0.11	0.29	0.35	0.54	0.1175

Table 5.5: Absolute deviation from reference velocity when using nonlinear controller with criterion V_{1nonl}

When studying the control signals, that are not included in the criterion, it can be seen that the throttle angle reaches values totally unrealistic (figure 5.9). Since the control signals are not included in expression (5.10) and the optimization problem is unbounded there is nothing to suppress the signals. The values assumed by α makes the result of this simulation hard to set in relation to a real life control of a vehicle.

Figure 5.9: Control signals for approximative nonlinear control using crite-
rion V_{1nonl} with R as a profile from 30 to 70 km/h

5.2.2   Control with nonlinear criterion two

The main reason for trying nonlinear MPC is because it is possible to use
a nonlinear optimization criterion and include the fuel consumption in the
criterion. This leads to expression (5.11).

$$V_{2nonl} = (Y - R)^T Q_y (Y - R) + \text{(fuel consumption)} \tag{5.11}$$

Again the focus is on the output reaching and keeping the reference velocity,
with the added wish to keep the fuel consumption low. As in criterion (5.10)
no explicit consideration is taken to the input signals, $\overline{U}$. Since the criterion
now has two terms the weighting becomes interesting, and the control is
tested with several different weights. A pattern emerges that shows that
for very low values on Q_y the total fuel consumption is lowered when using
criterion (5.11). At the same time this leads to the reference velocity not
being kept, but instead heavily overshot (figure 5.10).

This is of course unacceptable. The weight Q_y is therefore steadily increased.
This results in a better following of the reference (section 6, table 6.5). The
increasing of the weight is kept up until the reference velocity is followed
in a satisfactory way and the fuel consumption is as low as possible given
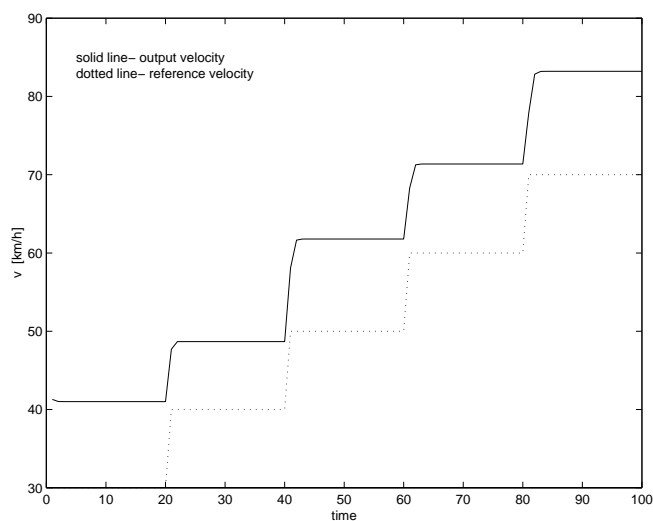
Figure 5.10: The output velocity from the nonlinear model when controlled with approximative nonlinear method, criterion 2, weight $Q_y = 0.000001$

that the velocity output is satisfactory (figure 5.11). The fuel consumption, with a good following of r, then reach a minimum at the exact same value that was given by (5.10).

Including the fuel consumption in the criterion did not have the desired effect of bringing the fuel consumption down to a low level. It could not improve the fuel economy and also did not dampen the values that the control signals assume. Again the throttle angle take on values that are impossible when considering its physical limit of 90°. Depending on what weights are used the current can also assume very big values (figure 5.12).

The unsatisfactory result when using this method can be ascribed to the fact that the method is only approximative. The approximations are made using linear control strategies derived from a linearization of the vehicle system. The linearization, in its turn, is done around working points well within the area of states were it is physically possible for the system to be, and in this linearization maps are used. When using the criteria V_{1nonl} and V_{2nonl} the control signals assume values that are far beyond the values covered by the map, values the system can not reach in real life.
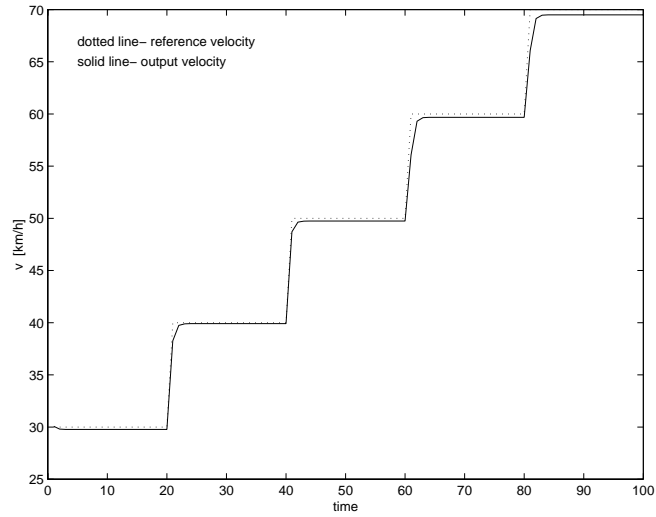
Figure 5.11: The output velocity from the nonlinear model when controlled with approximative nonlinear method, criterion 2, weight $Q_y = 0.001$
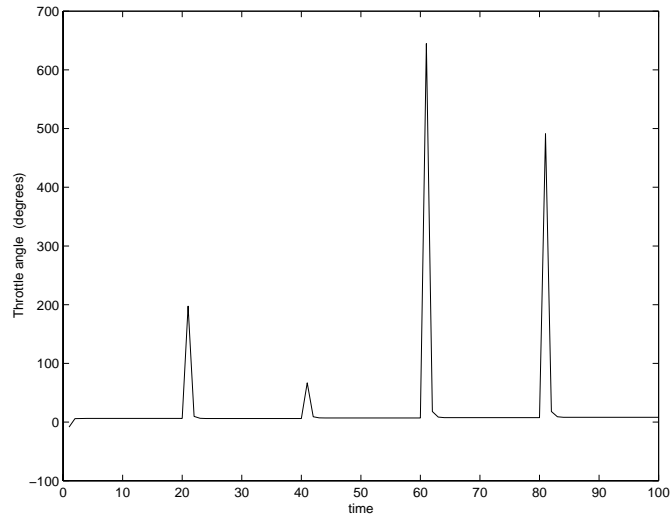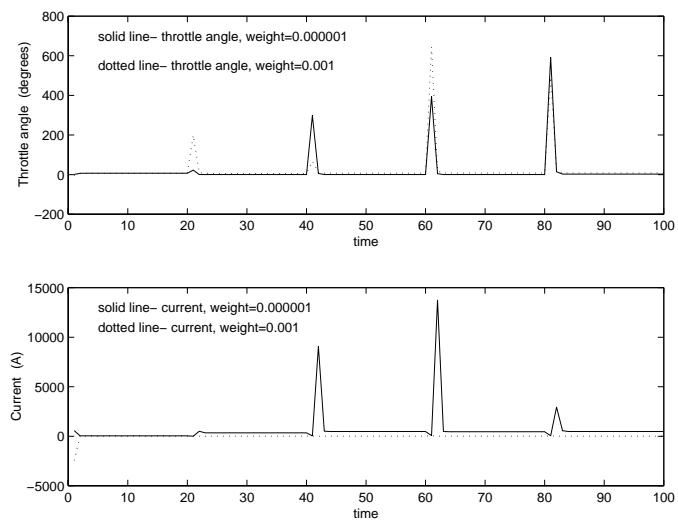

Figure 5.12: Control signals for approximative nonlinear control using criterion V_{2nonl} with R as a profile from 30 to 70 km/h

6 Tabular

6.1 Tables for linear control

The following two tables show the fuel consumption and the output velocity when the nonlinear vehicle model is controlled with the local controllers that were the result of MPC with criterion V_1 and V_2 recpectivly (section 4).

ref. speed km/h	20	30	40	50	60	70	80	100
lin. point								
working p.1	19.39	28.05	39.51	49.82	60.42	71.41	82.0	103
consump. l	0.053	0.069	0.074	0.101	0.132	0.146	0.178	0.260
working p.2	19.51	28.25	39.8	50.24	60.93	72.03	82.71	104
consump. l	0.054	0.070	0.075	0.102	0.134	0.148	0.180	0.264
working p.3	21.27	30.67	40.3	49.63	59.3	69.83	80.24	100.4
consump. l	0.036	0.050	0.063	0.082	0.109	0.140	0.173	0.251
working p.4	21.3	30.46	40.04	49.31	58.85	69.32	79.66	100.2
consump. l	0.036	0.050	0.062	0.081	0.107	0.139	0.171	0.247

Table 6.1: Output velocity and consumption from nonlinear model when using controllers derived with V_1

ref. speed km/h	20	30	40	50	60	70	80	100
lin. point								
working p.1	20.25	30.15	40	50.1	60.25	70.41	80.57	101.1
consump. l	0.042	0.057	0.067	0.106	0.137	0.172	0.216	0.344
working p.2	20.33	30.13	39.86	49.99	60.2	70.4	80.62	101.3
consump. l	0.042	0.057	0.077	0.105	0.135	0.170	0.213	0.338
working p.3	20.88	30.49	40.26	49.81	59.62	69.99	80.27	100.7
consump. l	0.036	0.050	0.064	0.086	0.114	0.145	0.177	0.256
working p.4	21.12	30.44	40	49.27	58.83	69.25	79.59	100.1
consump. l	0.036	0.050	0.062	0.081	0.108	0.139	0.171	0.247

Table 6.2: Output velocity and consumption from nonlinear model when using controllers derived with V_2

The weighting matrixes have been chosen to give the best possible result (with regard to reaching and holding the reference velocity) in the nonlinear

model at the linearization speed. Each reference speed is held constant one
hundred time steps. the consumption is the total consumption after this
time and the velocity is the end value of the output velocity.

6.2 Tables for switching control

The nonlinear model is controlled with Supervisor two and the local con-
trollers for criterion V_2. The reference velocity is a profile one hundred time
steps long, starting on thirty and ending one seventy. Each velocity is held
for twenty time steps and shifted in steps. The gear follows this shifting,
starting on two, shifting to three for speeds 40-50 km/h and the shifting to
gear four.

ref. profile km/h	30	40	50	60	70	consump. l
supervisor 2	29.68	40.00	49.99	60.27	70.00	0.1082
working p.1	29.68	40.00	50.10	60.31	70.49	0.1115
working p.2	29.38	39.86	49.99	60.27	70.53	0.1094
working p.3	28.60	39.30	49.35	59.69	70.00	0.1062
working p.4	27.14	38.45	48.27	58.82	69.23	0.1015

Table 6.3: The resulting velocity profile for the model with supervisor 2 and
controllers from criterion V_2

6.3 Tables for nonlinear control

Nonlinear control with the criterion $V_{1nonl} = (Y - R)^T Q_y (Y - R)$ is made
with the reference velocity as a profile one hundred time steps long, starting
on thirty and ending on seventy. Each velocity is held for twenty time steps
and shifted in steps. The gear follows this shifting, starting on two, shifting
to three for speeds 40-50 km/h and the shifting to gear four.

ref. profile km/h	30	40	50	60	70	consump. l
reached speed	29.77	39.89	49.71	59.65	69.46	0.1175

Table 6.4: The resulting velocity profile for the model controlled with non-
linear controller, criterion V_{1nonl}

Nonlinear control with the criterion $V_{2nonl} = (Y-R)^T Q_y (Y-R) + fuel consumption$

was made with the same reference velocity and gear shifting as for the criterion V_{1nonl}. Several different weights were tried.

ref. profile km/h	30	40	50	60	70	consump. l
reached speed $Q_y = 10^{-6}$	41.00	48.60	61.80	71.40	83.20	0.081
reached speed $Q_y = 5 \cdot 10^{-6}$	31.11	43.16	55.36	65.76	77.90	0.136
reached speed $Q_y = 10^{-4}$	30.43	41.78	52.29	63.07	72.93	0.126
reached speed $Q_y = 5 \cdot 10^{-5}$	29.90	40.22	50.22	60.24	70.21	0.119
reached speed $Q_y = 5 \cdot 10^{-4}$	29.78	39.92	49.76	59.71	69.54	0.1178
reached speed $Q_y = 10^{-3}$	29.77	39.91	49.73	59.68	69.50	0.1175
reached speed $Q_y = 2$	29.77	39.89	49.71	59.65	69.47	0.1175

Table 6.5: The resulting velocity profile for the model controlled with non-linear controller, criterion V_{2nonl}

7 Future work and extensions

The work done in this thesis can in no way be regarded as finished. There exist any number of ways to proceed.

The very first step should be to make full use of the Model Predictive Control algorithm by including constraints in the optimization problem. As mentioned before, all physical systems have limits and this holds true for the vehicle which is of current interest here. To be able to include the constraint of the throttle angle values (between zero and ninety degrees) would be of great interest as would the constraints on the current and the torques. It would result in a picture of the control problem at hand that would be more true to reality.

Another way available when going further with this work is to try other approximative nonlinear solutions. An interesting solution is presented in [3] where the prediction of the plant output is done in two parts. The free response, the future response obtained if the input is maintained at a constant level for the duration of output and control horizons, is obtained from a nonlinear model of the plant. The forced response, the response due to future control moves, is obtained using a linear model. The two responses are then added to make up the sought after prediction.

Finally, one should turn ones eye to the big field of predictive control, where MPC is just a small part, and test other predictive control methods.

In the future the energy a vehicle consumes should be optimized in all parts, from the fuel consumption of the engine to the electrical power needed for the air conditioning.

8 Conclusions

The aim of this thesis was to regulate a nonlinear vehicle model, including a Starter/Generator in the drivetrain, with the control strategy Model Predictive Control. An attempt to lower the fuel consumption of the vehicle with the help of this control strategy was also to be made. To do this the model and its nonlinearities was first presented and explained.

Both linear and nonlinear MPC was used. To be able to derive linear control laws it was necessary to linearize the vehicle model around several working points.

The first criterion chosen for the linear MPC was V_1 (equation (4.4), section 4.2). To make the output follow the reference and at the same time prevent the control signals from assuming big numerical values, which is impossible in reality, the criterion includes the difference between the reference and output, and the control signals.

The linear control of the model with criterion V_1 resulted in a good following of the reference velocity, for all the local controllers. The expected result that each controller would function at its best in an area around the working point for which it was made, could be seen in part. However, the result was not unambiguous. The control signals were kept at reasonable levels. The result was very similar, but slightly worse, than the one obtained when using linear criterion V_2. The behavior of the model when regulated with the local controllers corresponding to V_2 was therefore studied more carefully.

The inclusion of the change in control signals, $\Delta \overline{U}$, in V_2 (equation (4.5)) was made so that the throttle could be affected not to make big changes fast and not show a fluttering behavior. The new criterion led to a slightly smoother and closer following of the reference signal and the following of the reference velocity was very good in general, especially so in an area around each controllers working point. This pattern is an expected result, and can be seen more clearly when studying the output from control with V_2 than from outputs from control with criterion V_1 (tables 4.2 and 4.3).

When studying the two control signals, the throttle angle and the current, it can be seen that the each peak rather quickly for a velocity change, thus giving extra power. They then return to a steady, lower value to keep the speed. The values of the control signals keep within the physical boundaries of the system, even though the optimization is unbounded. The inclusion and weighting of α and I in the criteria leads to this result (figure 4.2).

The total torque is distributed between the torque from the engine, M_{engine}, and the torque from the Starter/Generator, M_{SA}. The main part of the torque is taken from the engine. It both remains at a higher steady level when driving at a constant speed and reaches higher peaking values when the reference shifts to a higher value. The torque from the Starter/Generator has a low constant velocity level, close to zero. It also peaks when the reference velocity increases, to help give extra power, but the peaks are much smaller than the ones from M_{engine} (figure 4.3).

Observing the good results when using local controllers on the nonlinear system, especially when operating in an area around their respective working point and using criterion V_2, leads to the idea of a switching strategy. In a switching strategy several controllers are combined in a control structure, where each controller is used when it is most appropriate. The choosing of which linear controller to use in a particular instant is done by a supervisor.

Two different supervisors were tested. Supervisor 1 gave an unsatisfactory result because it sometimes chose the wrong controller, that is a controller that lead to the difference between the reference and the output being greater than it could have been, had another controller been chosen. The reason for this was that the structure of the supervisor was based on the fact that the local controllers would be optimally tuned for their corresponding local models. However, in this thesis the choice was made to tune the linear controllers to perform at their best for the nonlinear system instead. Thus the nonlinear system follows the reference better when controlled with these linear controllers, but Supervisor 1 can be deemed inappropriate to use in this context.

Supervisor 2 was of a completely rigid structure, and based itself on previous information about which local controller would give the best output behavior when applied to the nonlinear model with a certain reference signal. The drawback of this method is that the information must be gathered in advance, but when this information is available, as in this case, the structure will guarantee that the best local controller always is chosen. Supervisor 2 has all the positive effects of the best local controllers, the ones corresponding to linear criterion two, with the added quality that the positive effects are over the entire velocity range, not just in the area around each controllers working point.

Since the vehicle model is strongly nonlinear it is interesting to try to control it with nonlinear MPC, even though the linear results are satisfactory with regards to the following of the reference velocity. It was not possible to affect

the efficiency of the engine with respect to the fuel consumption with linear control. Only with nonlinear control is it possible to take the nonlinear fuel consumption map directly into consideration. A criterion with only the output error, $Y - R$, is the first one tried. The criterion is nonlinear because the output no longer is calculated from the state space matrices from the linearized model, but directly from the nonlinear model. The problem is to complicated to solve with the help of Matlab calculations, and a possible solution would have a computational time far, far beyond what could be considered acceptable. The complete nonlinear solution is abandoned, and a approximate method is used.

The approximate nonlinear method is tested with to different criteria. The first one, V_{1nonl}, is again simply the difference $Y - R$. The criterion, containing only one term, is not subject to the tuning of weight matrices, since only the difference in weights between two terms in the criterion is important. The strength of the method is that the reference velocity is followed well over the entire velocity range, using just one controller. However, not only is the fuel consumption higher than the results from both control with local controllers and from the switching strategies. The control signals also assume unrealistic values, that make it hard to evaluate the results at all, since they could never be reached by a physical system.

Finally a nonlinear controller is made using the approximative method described in section 3.2 and 5.2, with a criterion that includes the fuel consumption as a term to be minimized. Now that two terms are present in the new criterion V_{2nonl}, the weighting again becomes an important issue. It is possible to weight the criterion in such a way that the fuel consumption is considerably lowered, but the the reference is not followed well at all. Different weights are tested, but under the condition of a good following of the reference value there is a limit to how low the fuel consumption can get. The end value is the same one as when using nonlinear control with criterion V_{1nonl}.

When studying the control signals they again show the tendency to assume very large values. This bring the vehicle system into a state were the linear controllers, the linearization and the maps used as parts of the approximation are put out of the running. To be able to say anything conclusive about the use of the nonlinear approximative Model Predictive Control it must be tested either with the control signals in the criterion, properly weighted to be kept within certain limits, or the optimization problem must be turned into a bounded one.

The conclusions to be drawn from the control work done here is firstly that MPC is indeed a usable method when one wants a cruise control affect. Using linear MPC is straight forward and good for the understanding of how the changing of the control parameters affect the behavior of the model and its output. The results obtained from linearizing the model and creating local linear controllers are amazingly good, considering the heavily nonlinear system the controller regulate. The best result is obtained when the local controllers are combined in the Supervisor 2 structure. This method of building a switching strategy is also very uncomplicated, but demand a knowledge of how the local controllers affect the nonlinear plant, so it must be possible to test this. If that is not the case, it might be wise to reconsider the choice made with regards to the weighting strategy in section 4.4. If the tuning is instead made with the local, linear, model in focus, supervisor strategy one might be more appropriate.

References

[1] L. Guzzella. Fahrzeugsantriebsysteme- Vorlesungsunterlagen Wintersemester 1999/2000. *Labratorium fur Motorsysteme ETH-Zurich*, 1999.

[2] T. Glad and L.Ljung. *Reglerteori- Flervariabla och olinjära metoder*. Studentlitteratur, 1997.

[3] E.F. Camacho and C. Bordons. *Model Predictive Control*. Springer, 1999.

[4] F. Allgöwer and A. Zheng, editors. *Nonlinear Model Predictive Control*, chapter Integrating Predictive and Swtiching Control. Birkhäuser Verlag, 1991.

[5] W.H. Kwon. Advances in Predictive Control: Theory and Application. *Automatica*, 1994.

[6] T. Glad and L.Ljung. *Reglerteknik Grundläggande teori*. Studentlitteratur, 1989.

[7] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Model Predictive Control: a review. *Automatica*, 1998.

A Appendix

A.1 Simulink model
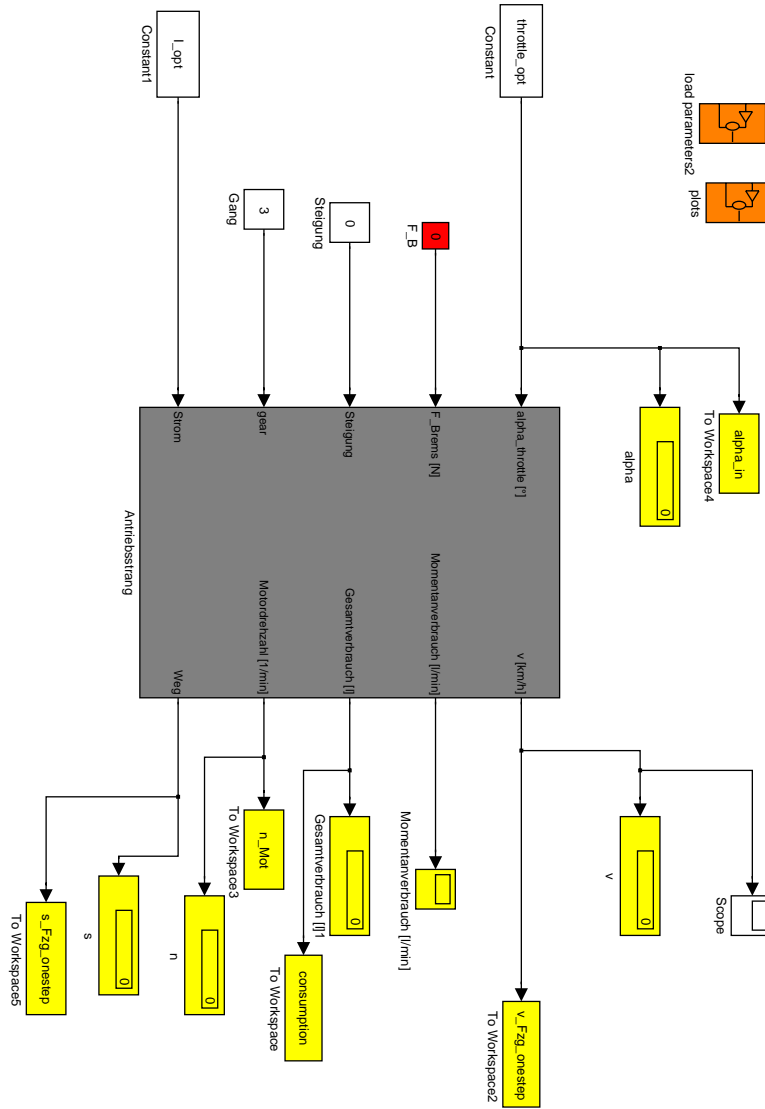
Figure A.1: The Simulink model of the vehicle w203, representing the entire vehicle system
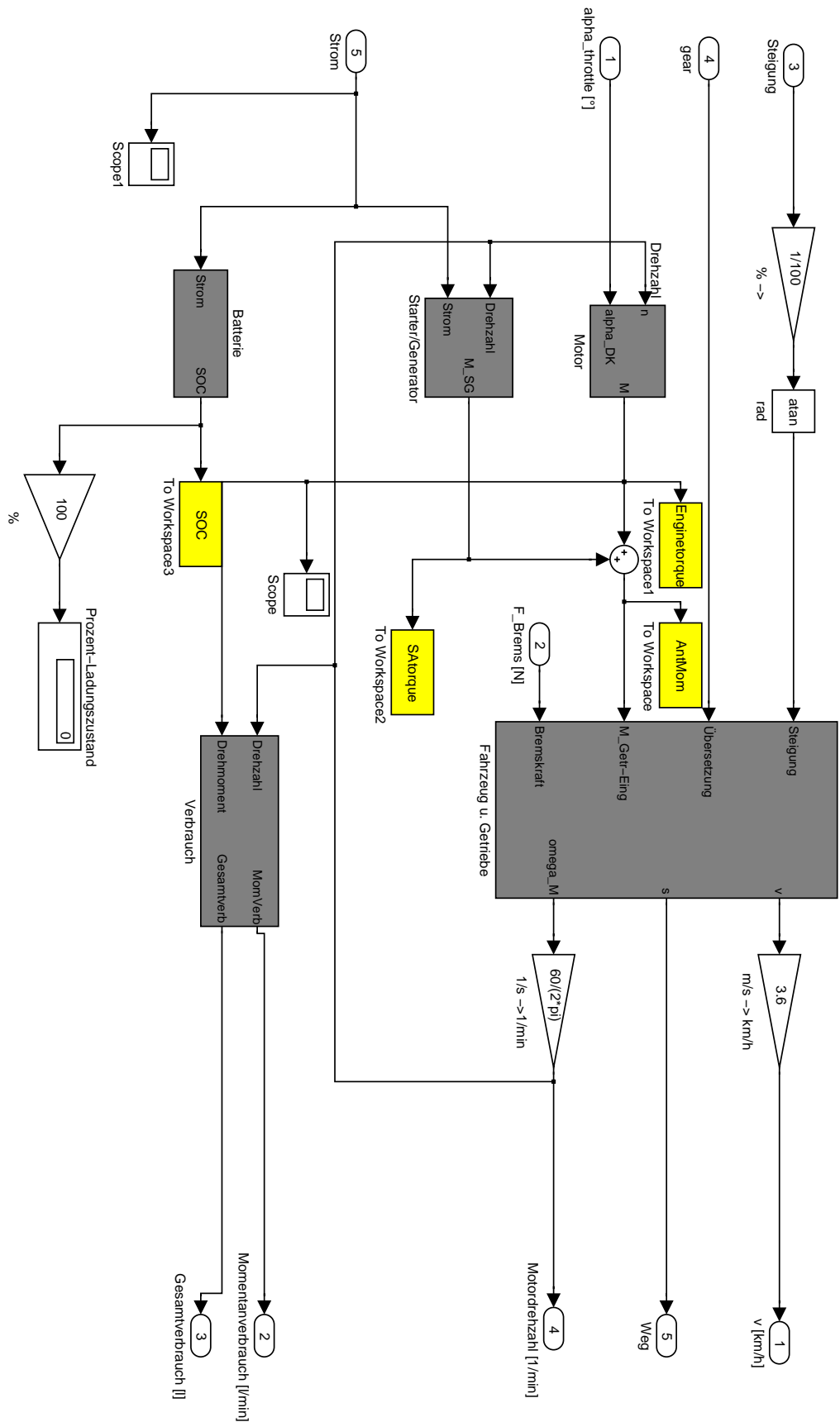
Figure A.2: The Simulink block Antriebstrang, representing the drive train
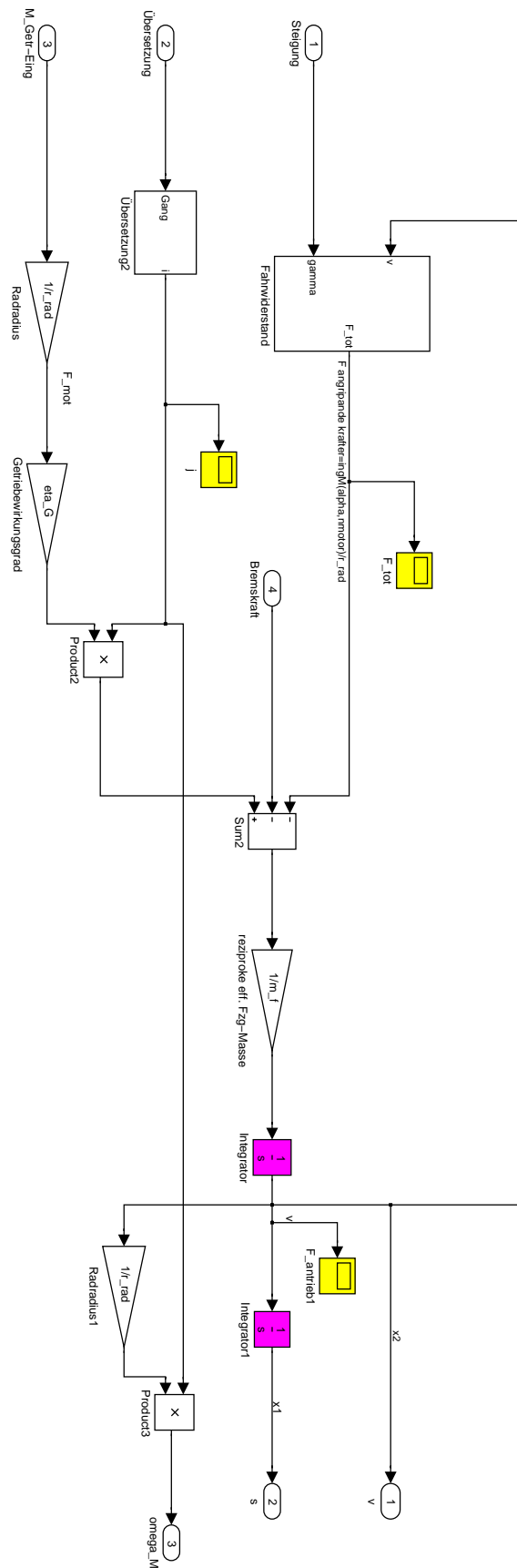
Figure A.3: The Simulink block Antribsstriebstrang/Fahrzeug u. Getriebe, representing vehicle and gearbox

Figure A.4: The Simulink block Antribsstriebstrang/.../Fahrwiederstand, representing the driving resistance
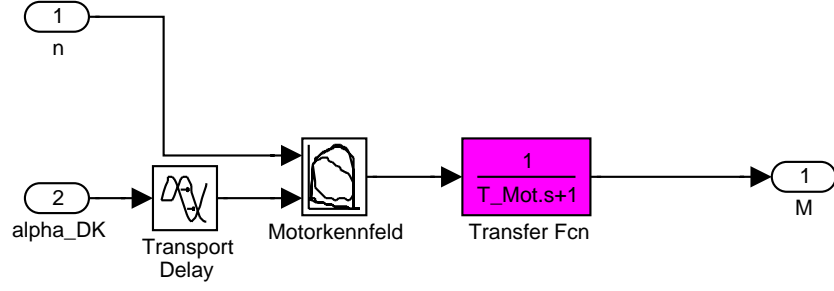
Figure A.5: The Simulink block Antribsstriebstrang/Motor, representing the engine
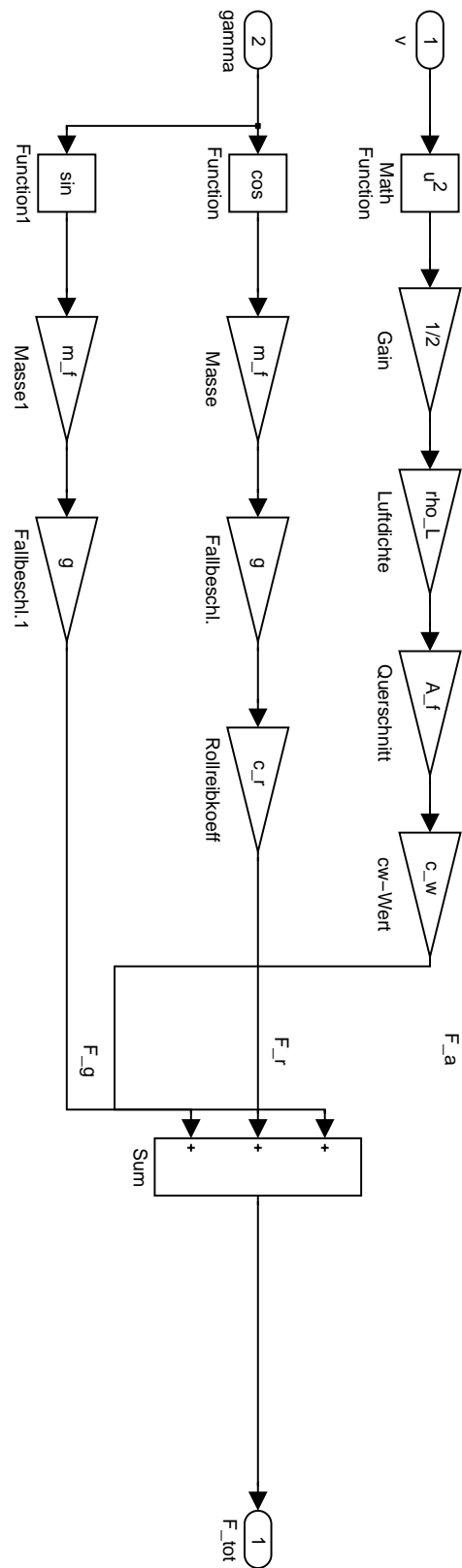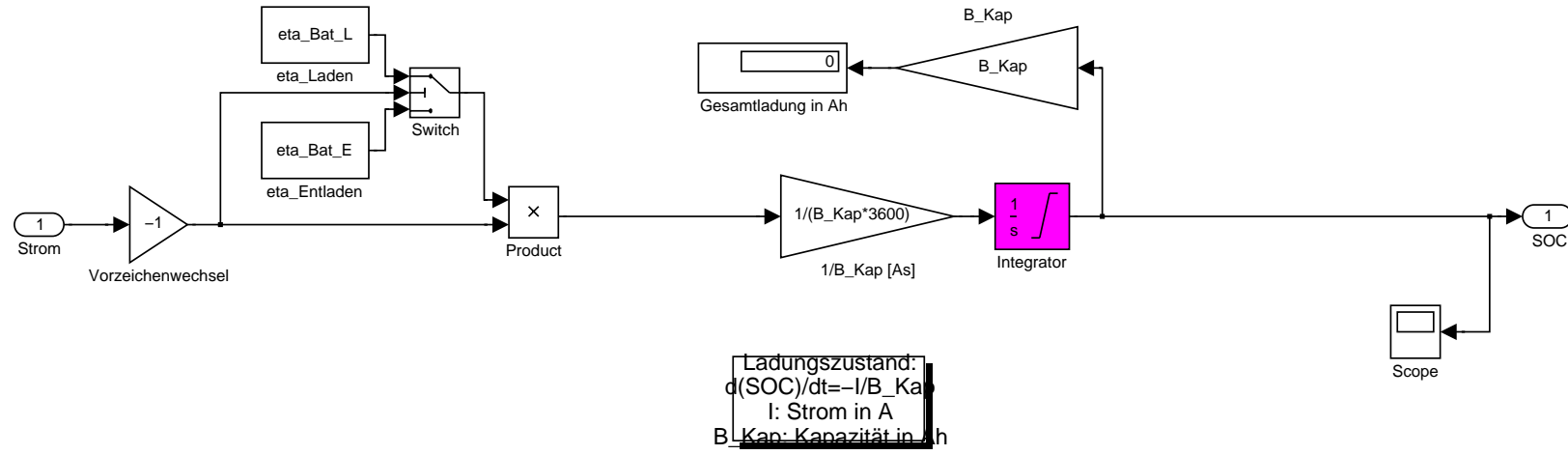
Figure A.6: The Simulink block Antribsstriebstrang/Batterie, representing the battery
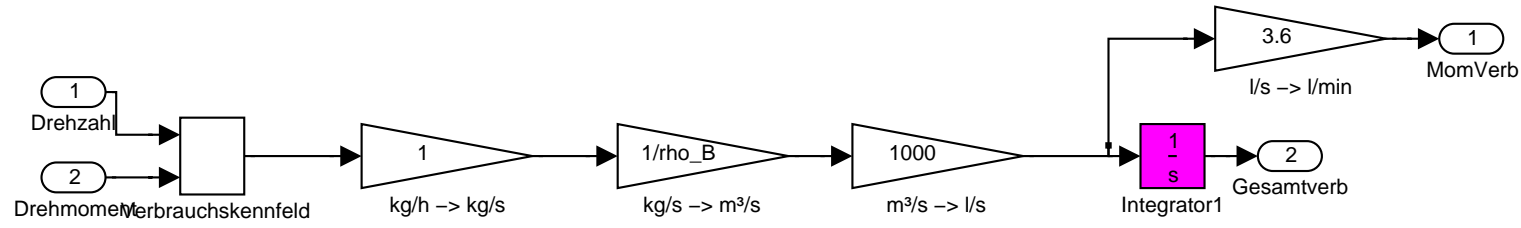
Figure A.7: The Simulink block Antribsstriebstrang/Verbrauch, representing the fuel consumption

B Appendix

B.1 Linear control Matlab files

The linear control Matlab files give as their final output a off-line control strategy consisting of four constants on for the vehicle system output y and one for the reference value r, for both the control signals, the throttle angle and the current.

B.1.1 Linearization.m

This Matlab function linearizes the nonlinear system, given the working point (a0, i0,v0), the vectors which make the engine torque map (a is alpha, b is n and M is M_mot), the step size to take when linearizing the map (chosen as delta_a is 0.5 and delta_b is 200) at the chosen point and finally the sampling time T (chosen as 1 s).

The outputs are the discrete state space matrices, the RPM (n0) responding to the working point, the sampling time and the throttle angle and velocity given in the working point.

```
function [Ad,Bd,Cd,Dd,initial_throttle,initial_speed,T_sample,n0]=
linearisation(i0,v0,a,b,M,a0,delta_a,delta_b,T)

\%*********Constants****************************************
r_rad = 0.308;                  \% dynamischer Radradius [m]
J_4rad = 4.03;                  \% Trägheitsmasse der 4 Räder [kg*m^2]
m_Fzg = 1470;                   \% Fahrzeugmasse [kg]
m_Zul = 100;                    \% Zuladung [kg]
c_r = 0.015;                    \% Rollreibungskoeffizient
g = 9.81;                        \% Erdbeschleunigung [m/s^2]
rho_L = 1.1688;                 \% Luftdichte [kg/m^3]
c_w = 0.27;                     \% Luftwiderstandsbeiwert
A_f = 2.07;                     \% Frontfläche [m^2]
eta_G = .93;                    \% Getriebewirkungsgrad
m_f = (m_Fzg + m_Zul);
dndv=(30*i0)/(r_rad*pi);

v0=v0*1000/3600;
b0=(v0*i0*30)/(r_rad*pi);                \%räkna om i m/s
```

```
\%----------Calculation of the derivates used in the linearization
\%----------by using the map  M_mot, n, alpha
value1a = interp2(a,b,M,a0,b0);
value2a = interp2(a,b,M,a0+delta_a,b0);
dmdalpha = (value2a - value1a)/delta_a;
value1b = interp2(a,b,M,a0,b0);
value2b = interp2(a,b,M,a0,b0+delta_b);
dmdn = (value2b - value1b)/delta_b;

\%*********Calculate the constants in the linearisations with the***
\%*********already calculated derivates

\%constant for (v-v0)
Konst_v=(1/m_f)*(i0*eta_G*(dmdn)*(dndv)/r_rad - rho_L*A_f*c_w*v0);

\%constant (alpha- alpha0)
Konst_alpha=(1/m_f)*(i0*eta_G*(dmdalpha)/r_rad);

\%constant (I-I0)
Konst_current=42/(0.85*(30*i0*v0/pi*r_rad));

\%**********calculating the state space matrices in the*************
\%**********continous version of the model

A=[0 1;
   0 Konst_v];
B=[ 0 0;
   Konst_alpha Konst_current];
C=[0 1];
D=[0 0];

\%**********calculating the discrete state space matrices**********
sys_cont=ss(A,B,C,D);
sys_discrete=c2d(sys_cont,T);

Ad=sys_discrete.a;
Bd=sys_discrete.b;
Cd=sys_discrete.c;
Dd=sys_discrete.d;
```

```
initial_throttle=a0;
initial_speed=v0*3.6;
T_sample=T;
n0=b0;
```

B.2 Mpc_matrix.m

This Matlab function takes the outputs from the linearization above and as its inputs and gives as outputs the control constants to be used in controlling the vehicle system via the throttle angle and the current with the help of the vehicle system output y and the reference value r. It was used for the linear criterion V_1.

```
function [U_opt_alpha,U_opt_I] =mpc_matrix_delta(Ad,Bd,Cd,Qu,Qy,Qdeltau,...
M,n_const,n_vector,throttle_a)

\%*********************************************************************
\%[U_opt_alpha,U_opt_I] = mpc_matrix_delta_u(Ad,Bd,Cd,Qu,Qy,Qdeltau,
\%M_mot,n0,n,alpha).
\%n_const=n0 throttle_a=alpha
\%Gives the control constants for alpha and I
\%*********************************************************************
R=ones(10,1);

H=[Cd*Ad^10;
   Cd*Ad^9;
   Cd*Ad^8;
   Cd*Ad^7;
   Cd*Ad^6;
   Cd*Ad^5;
   Cd*Ad^4;
   Cd*Ad^3;
   Cd*Ad^2;
   Cd*Ad];

S=[Cd*Ad^6*Bd Cd*Ad^7*Bd Cd*Ad^8*Bd Cd*Ad^9*Bd;
   Cd*Ad^5*Bd Cd*Ad^6*Bd Cd*Ad^7*Bd Cd*Ad^8*Bd;
   Cd*Ad^4*Bd Cd*Ad^5*Bd Cd*Ad^6*Bd Cd*Ad^7*Bd;
   Cd*Ad^3*Bd Cd*Ad^4*Bd Cd*Ad^5*Bd Cd*Ad^6*Bd;
```

```
   Cd*Ad^2*Bd Cd*Ad^3*Bd Cd*Ad^4*Bd Cd*Ad^5*Bd;
   Cd*Ad*Bd   Cd*Ad^2*Bd Cd*Ad^3*Bd Cd*Ad^4*Bd;
   Cd*Bd      Cd*Ad*Bd   Cd*Ad^2*Bd Cd*Ad^3*Bd;
   0 0        Cd*Bd      Cd*Ad*Bd   Cd*Ad^2*Bd;
   0 0        0 0        Cd*Bd      Cd*Ad*Bd;
   0  0       0  0       0  0       Cd*Bd  ];

syms x1 x2
X=[x1;x2];

syms a3 I3 a2 I2 a1 I1 a I r
U=[a3; I3; a2; I2; a1; I1; a; I];
deltaU=[a3-a2; I3-I2; a2-a1; I2-I1; a1-a; I1-I; a; I];

\%*****************************************************************
\%The criterium V=(Y-r*R)'*Qy*(Y-r*R)+ deltaU'*Qdeltau*deltaU =
\%(H*X+S*U-r*R)'*Qy*(H*X+S*U-r*R) +  deltaU'*Qdeltau*deltaU
\%the derivate 0.5* dVdU = S'Qy(HX+SU-r*R) +Qdeltau*deltaU-
\%[specialmatrix]*deltaU

dVdU=S'*Qy*(H*X + S*U-r*R) + Qdeltau*deltaU...
   -[0 0 0 0 0 0 0 0;0 0 0 0 0 0 0 0; Qdeltau(1,:);Qdeltau(2,:);
Qdeltau(3,:);Qdeltau(4,:); Qdeltau(5,:); Qdeltau(6,:)]*deltaU;

\%pretty(vpa(dVdU,2));
\%*****************************************************************
\%calculate the symbolic solution to dVdU=0. U*=U_opt
S= solve(dVdU(1,1),dVdU(2,1), dVdU(3,1),dVdU(4,1),dVdU(5,1),dVdU(6,1),...
dVdU(7,1),dVdU(8,1),'a','a1','a2','a3','I','I1','I2','I3');
U_opt=[vpa(S.a3,4);vpa(S.I3,4);vpa(S.a2,4);vpa(S.I2,4);vpa(S.a1,4);...
vpa(S.I1,4);vpa(S.a,4);vpa(S.I,4)];

U_opt_alpha=U_opt(7,1);
U_opt_I=U_opt(8,1);
```

B.2.1   Mpc_matrix_delta_u.m

This Matlab function takes the outputs from the linearization above and as
its inputs and gives as outputs the control constants to be used in controlling

the vehicle system via the throttle angle and the current with the help of
the vehicle system output y and the reference value r. It was used for the
linear criterion V_2.

```
function [U_opt_alpha_r,U_opt_alpha_x2,U_opt_I_r,U_opt_I_x2] =
mpc_matrix_delta_u(Ad,Bd,Cd,Qu,Qy,Qdeltau,M,n_const,n_vector,throttle_a)

\%[U_opt_alpha_r,U_opt_alpha_x2,U_opt_I_r,U_opt_I_x2]=
\%  mpc_matrix_delta_u(Ad,Bd,Cd,Qu,Qy,Qdeltau,M_mot,n0,n,alpha)
\%n_const=n0 throttle_a=alpha
\%Gives the control constants for alpha and I

R=ones(10,1);
H=[Cd*Ad^10;
   Cd*Ad^9;
   Cd*Ad^8;
   Cd*Ad^7;
   Cd*Ad^6;
   Cd*Ad^5;
   Cd*Ad^4;
   Cd*Ad^3;
   Cd*Ad^2;
   Cd*Ad];

S=[Cd*Ad^6*Bd Cd*Ad^7*Bd Cd*Ad^8*Bd Cd*Ad^9*Bd;
   Cd*Ad^5*Bd Cd*Ad^6*Bd Cd*Ad^7*Bd Cd*Ad^8*Bd;
   Cd*Ad^4*Bd Cd*Ad^5*Bd Cd*Ad^6*Bd Cd*Ad^7*Bd;
   Cd*Ad^3*Bd Cd*Ad^4*Bd Cd*Ad^5*Bd Cd*Ad^6*Bd;
   Cd*Ad^2*Bd Cd*Ad^3*Bd Cd*Ad^4*Bd Cd*Ad^5*Bd;
   Cd*Ad*Bd    Cd*Ad^2*Bd Cd*Ad^3*Bd Cd*Ad^4*Bd;
   Cd*Bd       Cd*Ad*Bd   Cd*Ad^2*Bd Cd*Ad^3*Bd;
   0 0         Cd*Bd      Cd*Ad*Bd   Cd*Ad^2*Bd;
   0 0         0 0        Cd*Bd      Cd*Ad*Bd;
   0  0        0  0       0  0       Cd*Bd  ];

syms x1 x2
X=[x1;x2];
syms a3 I3 a2 I2 a1 I1 a I r
U=[a3; I3; a2; I2; a1; I1; a; I];
deltaU=[a3-a2; I3-I2; a2-a1; I2-I1; a1-a; I1-I; a; I];
```

```
\%****************************************************************
\%The criterium V=(Y-r*R)'*Qy*(Y-r*R) + U'*Qu*U + deltaU'*Qdeltau*deltaU =
\%(H*X+S*U-r*R)'*Qy*(H*X+S*U-r*R) +  U'*Qu*U + deltaU'*Qdeltau*deltaU
\%the derivate 0.5* dVdU = S'Qy(HX+SU-r*R)+QuU +Qdeltau*deltaU-...
[specialmatrix]*deltaU

dVdU=S'*Qy*(H*X + S*U-r*R) + Qu*U + Qdeltau*deltaU...
   -[0 0 0 0 0 0 0 0;0 0 0 0 0 0 0 0; Qdeltau(1,:);Qdeltau(2,:);
    Qdeltau(3,:);Qdeltau(4,:); Qdeltau(5,:);Qdeltau(6,:)]*deltaU;

\%pretty(vpa(dVdU,2));
\%****************************************************************
\%calculate the symbolic solution to dVdU=0. U*=U_opt
S= solve(dVdU(1,1),dVdU(2,1),
dVdU(3,1),dVdU(4,1),dVdU(5,1),dVdU(6,1),dVdU(7,1)...
,dVdU(8,1),'a','a1','a2','a3','I','I1','I2','I3');

U_opt=[vpa(S.a3,4);vpa(S.I3,4);vpa(S.a2,4);vpa(S.I2,4);vpa(S.a1,4);...
       vpa(S.I1,4);vpa(S.a,4);vpa(S.I,4)]

U_opt_alpha=U_opt(7,1);
U_opt_I=U_opt(8,1);

U_opt_alpha_r=subs(U_opt_alpha,{'r' 'x2'}, [1 0]);
U_opt_alpha_x2=subs(U_opt_alpha,{'r' 'x2'}, [0 1]);
U_opt_I_r=subs(U_opt_I,{'r' 'x2'}, [1 0]);
U_opt_I_x2=subs(U_opt_I,{'r' 'x2'}, [0 1]);
```

B.3   Nonlinear control Matlab files

The nonlinear Matlab files give as their output a on-line control strategy
with two values for the control signals, the throttle angle and the current,
at present time t.

B.3.1   nonl_onestep_controlfuel_gear.m

This Matlab file calculates the values of the control signals at present time
t, for the criterion given in Ny_krit_fuel_gear.m. For solving the nonlinear

optimization problem it also calls upon Ny_solvefuel_gear.m. The method
used is the approximative method descriebed in section 3.2 and 5.2.

```
function[v_velocity,throttle_angle,Current,consumption_tot,...
consumption_tot_vector,satorque,enginetorque]=...
nonl_onestep_controlfuel_gear(velocity,gear)

\%----------------------------------------------------------------
\%[v_velocity,throttle_angle,Current,consumption_tot,...
\%SAtorque,enginetorque]=nonl_onestep_controlfuel_gear(velocity,gear)
\% this program are the same as nonl_onestep_control but trying to
\%include gear as a input.
\%----------------------------------------------------------------
global g_v
global r
global throttle_opt
global I_opt
global v_Fzg_onestep0
global position_onestep0
global x_result
global consumption
global SAtorque
global Enginetorque

consumption=0;
consumption_tot=0;
consumption_tot_vector(1)=0;
\%-------------------------------------------------------------
\%load all the necesary things as globals in the workspace by
\%running  load_onestep.m!
\%*********************************************************************
R=velocity;
v_Fzg_onestep0=40;
position_onestep0=0;
enginetorque=0*ones(length(R),1);
satorque=0*ones(length(R),1);
consumption_tot_vector=0*ones(length(R),1);

 for k=1:(length(R))
    r=R(k,1)
```

```
 k
 g_v=gear(k,1)
\%solve the criterion with one nonlin step followed by some lin.
\%This gives optimal throttle angle and current, x_result.
ny_solvefuel_gear

v_Fzg_onestep0;
position_onestep0;

\%simulate nonlinear model
throttle_opt=x_result(1,1);
I_opt=x_result(1,2);
throttle_angle(k)=throttle_opt;
Current(k)=I_opt;

sim('w203_onestep_gear')

\%do an update on the states
v_Fzg_onestep0=...
v_Fzg_onestep.signals.values(length(v_Fzg_onestep.signals.values));
position_onestep0=...
s_Fzg_onestep.signals.values(length(s_Fzg_onestep.signals.values));
v_velocity(k)=v_Fzg_onestep0;

enginetorque(k)=Enginetorque.signals.values(length...
(Enginetorque.signals.values));
satorque(k)=SAtorque.signals.values(length(SAtorque.signals.values));
consumption_tot=...
consumption_tot+consumption.signals.values(length...
(consumption.signals.values));
consumption_tot_vector(k+1)=...
consumption_tot_vector(k)+...
consumption.signals.values(length(consumption.signals.values));
end
```

B.3.2   Ny_solvefuel_gear.m

```
\%the input has to be declared global in the function f=output and in
\%the matlab window. This happens here. Also the data needed for the
```

```
\%simulation of w203 is loaded.
global g_v
global r
global v_Fzg_onestep0
global position_onestep0
global x_result
global U_opt_alpha_r
global U_opt_alpha_x2
global U_opt_I_r
global U_opt_I_x2
global alpha_in
global I

w203_data_optim2
w203_data_optim

\%********************************************************************
qu1=[1 0 0 0 0 0 0 0];qu2=[0 1 0 0 0 0 0 0];
qu3=[0 0 1 0 0 0 0 0];qu4=[0 0 0 1 0 0 0 0];
qu5=[0 0 0 0 1 0 0 0];qu6=[0 0 0 0 0 1 0 0];
qu7=[0 0 0 0 0 0 1 0];qu8=[0 0 0 0 0 0 0 1]

\%********************************************************************
\%calculate the U=[u(t+10)..u(t)] with a linear controller
\%for ex with mpc_matrix_delta_u

if ( r<=40)
   [Ad,Bd,Cd,Dd,initial_throttle,initial_speed,T_sample,n0]= ...
   linearisation2(3,40,alpha,n,M_mot,5.6,0.5,200,1);
   Quu=0.0058*eye(8,8);
   Qyy=eye(10,10);
   Qdeltadelta=[0.001*qu1;qu2;0.001*qu3;qu4;0.001*qu5;qu6;0.001*qu7;qu8];
   U_opt=mpc_matrix_onestep(Ad,Bd,Cd,Quu,Qyy,Qdeltadelta,...
   M_mot,n0,n,alpha);

elseif (r > 40)& (r <= 60)
   [Ad,Bd,Cd,Dd,initial_throttle,initial_speed,T_sample,n0]= ...
   linearisation2(3,50,alpha,n,M_mot,6.5,0.5,200,1);

   Quu=0.0105*eye(8,8);
```

```
    Qyy=eye(10,10);
    Qdeltadelta=[0.005*qu1;qu2;0.005*qu3;qu4;0.005*qu5;qu6;0.005*qu7;qu8];
    U_opt=mpc_matrix_onestep(Ad,Bd,Cd,Quu,Qyy,Qdeltadelta,...
    M_mot,n0,n,alpha);

elseif (r > 60) & (r <=80)
    [Ad,Bd,Cd,Dd,initial_throttle,initial_speed,T_sample,n0]= ...
    linearisation2(4,70,alpha,n,M_mot,7.42,0.5,200,1);
    Quu=0.02*eye(8,8);
    Qyy=eye(10,10);
    Qdeltadelta=[0.05*qu1;qu2;0.05*qu3;qu4;0.05*qu5;qu6;0.05*qu7;qu8];
    U_opt=mpc_matrix_onestep(Ad,Bd,Cd,Quu,Qyy,Qdeltadelta,...
    M_mot,n0,n,alpha);

elseif (r > 80)
    [Ad,Bd,Cd,Dd,initial_throttle,initial_speed,T_sample,n0]= ...
    linearisation2(4,100,alpha,n,M_mot,9.5,0.5,200,1);
    Quu=0.2*eye(8,8);
    Qyy=eye(10,10);
    Qdeltadelta=[0.08*qu1;qu2;0.08*qu3;qu4;0.08*qu5;qu6;0.08*qu7;qu8];
    U_opt=mpc_matrix_onestep(Ad,Bd,Cd,Quu,Qyy,Qdeltadelta,M_mot,n0,n,alpha);
end

U_opt_alpha=U_opt(5,1);
U_opt_I=U_opt(6,1);

U_opt_alpha_r=subs(U_opt_alpha,{'r' 'x2'}, [1 0]);
U_opt_alpha_x2=subs(U_opt_alpha,{'r' 'x2'}, [0 1]);
U_opt_I_r=subs(U_opt_I,{'r' 'x2'}, [1 0]);
U_opt_I_x2=subs(U_opt_I,{'r' 'x2'}, [0 1]);
\%*********************************************************************

\%This simulation is run to get good starting values for fminsearch

sim('w203_call_controlled_gear')
alpha0=alpha_in.signals.values(length(alpha_in.signals.values));
I0=I.signals.values(length(I.signals.values));

optimset('TolFun',1e-1);
```

```
[x eval] = fminsearch('ny_kritfuel_gear',[alpha0 I0],[],r, U\_opt);
x;
x_result=x;
eval;
```

B.3.3   Ny_krit_fuel_gear.m

```
function out = ny_kritfuel_gear(u,r,U_opt)
global g_v
global in_alpha
global in_current
global U_opt_alpha_r
global U_opt_alpha_x2
global U_opt_I_r
global U_opt_I_x2
global I_opt
global throttle_opt
global v_Fzg_onestep0
global position_onestep0
global v_Fzg0
global position0
global rot_speed
global T_engine
global n_v
global M
global Verbrauch

Qy=0.000001*eye(2,2);

\%***********************************************************************
\%Initial values for w203_call should be the most recent from
\%nonl_onstep_control, that is equal to v_Fzg_onestep0, position_onestep0
\%-------------- t ----------------------------------------------------

v_Fzg0=v_Fzg_onestep0;
position0=position_onestep0;

in_alpha = u(1);
in_current = u(2);
```

```
sim('w203_call_gear')
y1=v_Fzg.signals.values(length(v_Fzg.signals.values));

\%-------------- t+1,(3 sec)  -----------------------------------------

v_Fzg0=y1;
postion0=s_Fzg.signals.values(length(s_Fzg.signals.values));
\%call_controlled _ny har simuleringstid 3*t_sim och data_optim

sim('w203_call_controlled_ny_gear')
y2=v_Fzg.signals.values(length(v_Fzg.signals.values));

\%*******************************************************************
\% the criterion to be minimized

rot_speed=n_Mot.signals.values(length(n_Mot.signals.values));
T_engine=MotMom.signals.values(length(MotMom.signals.values));
fuel_consump=interp2(M,n_v,Verbrauch,T_engine,rot_speed)

R_reference = r*[1;1];
Y=[y2;y1];

out=(Y-R_reference)'*Qy*(Y-R_reference)+fuel_consump
```

B.3.4   load_onestepfuel_gear.m

```
w203_data_optim
w203_data_optim2
global g_v
global in_alpha
global in_current
global U_opt_alpha_r
global U_opt_alpha_x2
global U_opt_I_r
global U_opt_I_x2
global r
global throttle_opt
global I_opt
```

```
global v_Fzg_onestep0
global position_onestep0
global v_Fzg0
global position0

global x_result
global rot_speed
global T_engine
global n_v
global M
global Verbrauch
```