CHALMERS

Hybrid Powertrain Control

A Predictive Real-time Energy Management System for a Parallel Hybrid Electric Vehicle

Master of Science Thesis

JOAKIM PERSSON
THOMAS LUNDBERG

Department of Signals and Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden, 2007
Report No. EX029/2008

ABSTRACT

The purpose of this diploma work has been to see whether it is possible to develop a rule-based controller that mimics the behavior of an optimal control strategy for a hybrid city bus. This control strategy improves fuel efficiency by use of preview information about the road ahead. The rule based controller has been designed for easy implementation into the ISAM engine management system.

Dynamic programming is used to find an optimal solution which in turn is used as a blueprint for a rule-based controller. The transition from optimal control to rule-based control is carried out using fuzzy logic.

Preview information will in reality be given from a topographic map combined with a GPS. This information together with a speed curve will give information about the future power demand. To simulate a system like this we use parts of the drive cycle in front of the vehicle, which gives both the road incline and the desired speed.

Our simulations show that the fuel reduction on a city bus route is about 3.5% when using optimal control, compared with the ISAM control system which is used as a reference system. When the rule-based control is used, the fuel reduction is about 2%. These results have been obtained by controlling the torque split between the internal combustion engine and the electrical machine, without optimizing gear selection.

We also carried out simulations including optimization of gear shifting. This resulted in a fuel reduction of about 12%. However, these results are based on somewhat unrealistic presumptions, i.e. gear shifting occurs instantaneously. They are therefore not considered in the rule-based controller.

Keywords: Powertrain control, Dynamic programming, Optimal control, Fuzzy logic, Preview information, Hybrid electrical vehicle, HEV, Electrical horizon.

PREFACE

This project is a diploma thesis for Master of Science degree from the Electric Engineering program at Chalmers University of Technology in Gothenburg. The project has been carried out at Volvo Technology Corporation.

The problem we have worked with is complex and standard control strategies are insufficient. Model-based controllers could be an option, but since the system is of such complexity and the time of a thesis work is limited we have tried a different approach. The method we have chosen is called Fuzzy Logic, and has been adopted extensively in Asia and other parts of the world. The technique is well suited to mimic the behavior of a tutor, or optimal solution.

It has been interesting to both work full time in an industrial project and to learn new methods. All the work has been done using computer simulation, which is an effective and fast way to see the potential of different approaches.

We would like to express our gratitude towards our department at Volvo Technology and our supervisors Jonas Edvardsson and Charlotte Holmen for making this work possible. Outside our department, we would like to thank Lisa Ehrlich, Stefan Pettersson, Lars Johannesson and our examiner Jonas Sjöberg at Chalmers University.

Thank you for valuable input, your time and all productive discussions.

Joakim & Thomas
Göteborg, June, 2007

Table of Contents

Abbreviations

BAT	Battery
BSFC	Break Specific Fuel Consumption
CBR85	City Bus Route 85
CL	Clutch
CVT	Continuously Variable Transmission
DC	Drive Cycle
DP	Dynamic Programming
EAUX	Electrical Auxiliaries
EM	Electric Machine
FC	Fuel Consumption
FM	Fuzzy Matrix
GB	Gear Box
GPS	Global Positioning System
HEV	Hybrid Electrical Vehicle
ICE	Internal Combustion Engine
ISAM	Integrated Starter Alternator Motor
MAUX	Mechanical Auxiliaries
MB	Mechanical Break
MF	Membership function
MSD	Mean Standard Deviation
N_{EM}	Speed of the Electric Machine
N_{ICE}	Speed of the Internal Combustion Engine
NOC	Near Optimality Controller
PE	Power Electronics
P_{EAUX}	Power of the Electrical Auxiliaries
P_{EM}	Power of the Electric Machine
P_{EMLOSS}	Power loss in the Electrical Machine
PHEV	Parallel Hybrid Electrical Vehicle
PI	Preview Information
P_{ICE}	Power of the Internal Combustion Engine
PID	Proportional, Integral, Derivative Regulator
SD	Standard Deviation
SDTL	Standard Deviation Threshold Limit
SM	Simplified Model
SoC	State of Charge
SoC_{MIN}	State of Charge minimum limit
SoC_{MAX}	State of Charge maximum limit
T_{BREAK}	Break Torque
T_{DEM}	Torque demand
$T_{DEM,ref}$	Reference Torque demand
T_{EM}	Torque of the Electric Machine
$T_{EM,\ NOC}$	NOC T_{EM} for a specific drive cycle
$T_{EM,\ BACKGROUND\ NOC}$	Background control T_{EM} for a specific drive cycle
$T_{EM,\ OPTIMAL}$	Optimal T_{EM} for a specific drive cycle
$T_{EM,ref}$	Reference T_{EM}
T_{EXTRA}	Extra Torque
T_{ICE}	Torque of the Internal Combustion Engine
$T_{ICE,ref}$	Reference T_{EM}
TRM	Transmission
VEH	Vehicle
VTEC	Volvo Technology

1 Introduction

1.1 Background

In society of today, concern is growing regarding the environmental impact of transportation that depends on fossil fuels. Both people in general and governments are becoming more aware about the recent environmental changes e.g. global warming. This has led to harsher limits on how much emissions a both light and heavy duty vehicles are allowed to produce, Euro 4 and Euro 5 legislations [1]. These restrictions and the fact that our easy accessible oil deposits are almost depleted have led to increasing fuel prices. The cost of gasoline in Sweden has for example been doubled in the last 20 years [2]. All this have forced the transport industry to take different measures to both lower emissions as well as the fuel consumption.

One such approach is the Hybrid Electric Vehicle (HEV), which combines an internal combustion engine (ICE) with an electric machine (EM) and an energy storage. The EM, combined with the battery, then becomes a torque-producing unit that can propel the vehicle, as well as an energy buffer that can absorb energy by using the EM as a generator. This can be done when the kinetic energy of the vehicle is to be lowered, for example when stopping, or when the velocity should be kept constant during downhill driving.

The control problems consists in making the best use of the battery while preserving its life expectancy, and at the same time lower the fuel consumption as much as possible. The problem with the strategies of today is that they do not know how much energy the system can be expected to gain from regenerative breaking, or how much energy the vehicle will use in the near future. These limitations lead to restrictive control algorithms that do not take full advantage of the battery. Therefore the present control algorithms do not lower the fuel consumption as much as would be possible if information about the future was available to the control system.

Advances in communication and on-board computing have made it possible to create increasingly advanced navigation systems. Map data and a GPS system enable construction of an electronic horizon on-board a vehicle. This preview information can be used for driver assistance systems but could also be used for optimization of drive-line components e.g. to save fuel and/or improve vehicle performance. Such a system will be available for the European market around year 2008-2010.

Earlier work [3] has shown that by using optimal control, through dynamic programming, the fuel consumption could be lowered by approximately 10% for a city bus, when including gear shifting in the control algorithm. Unfortunately, dynamic programming is a computationally demanding task which is not suited for implementation in a vehicle. Therefore this master thesis investigates the possibility to implement a nearly optimal control strategy that efforts to mimic the behavior of the optimal one.

1.2 Thesis description

The purpose of this master thesis work is to derive a control algorithm that mimics the behavior of an optimal controller, using preview information about future driving conditions. We call this type of control algorithm a near optimality controller (NOC). The goal is to reduce the fuel consumption by utilizing the battery and the electrical machine in the same way that the optimal controller does. The algorithm should work in such a way that it easily can be tuned for a specific route, while at the same time giving a good result for an unknown route.

The optimal solutions are calculated using dynamic programming (DP). These solutions will serve as a lower boundary for fuel consumption for a specific route. However the solutions will also give vital information about how the control strategy should be designed to achieve a minimal fuel consumption strategy.

The preview information that will be available to the controller is the topography, the speed curve for a specific route and the vehicle's attributes. Combined, they provide information about the expected power demand of the vehicle.

The implementation will be done in Matlab Simulink, using Volvo Technology's (VTEC's) current model of a city bus, in such a way that it easily can be implemented in a vehicle in the future.

1.3 Disposition

In chapter 2 a review of the models, the drive cycles, preview information estimation, and a figurative way of looking at stored energy are presented. This is to give the reader a background of the theory so that all the different mathematical approaches in the following chapters will be understood. Chapter 3**Error! Reference source not found.** explains the optimization methods as well as the approaches that have been evaluated to achieve a NOC. In chapter 4, results from both the optimal solutions and the NOC are presented. The optimal control results are divided in to two sections with different constraints on the optimization problem this is followed by the results from the NOC. Conclusions and ideas for future work are presented in chapter 5 and 6.

2 Models

During this thesis work, two models of a HEV have been used for the calculations and evaluations needed. Both the models are implemented in the Matlab Simulink environment. One model is VTEC's currently used model, named ISAM, and the other one is a simplified model of a HEV.

2.1 A Parallel Hybrid Electrical Vehicle overview in General

Generally a parallel hybrid electrical vehicle (PHEV) is a hybrid vehicle with an internal combustion engine (ICE), an electric machine (EM) and a battery (BAT). The PHEV could be built up by other energy converters and other energy storages but in general they consist of one ICE, one EM, one fuel tank and a battery. As can be seen in Figure 2-1, the ICE and the EM are mounted on the same axle which means that they run at the same speed when they are not separated by the clutch. Both the ICE and the EM can produce torque to propel the vehicle, and the EM can also regenerate energy to the battery. The total torque delivered to the vehicle (VEH) is the sum of the two torque producers, excluding some minor frictions. For more detailed information about PHEVs, see for example 'Hybrid Drive Systems for Vehicles' by Mats Alaküla [4].

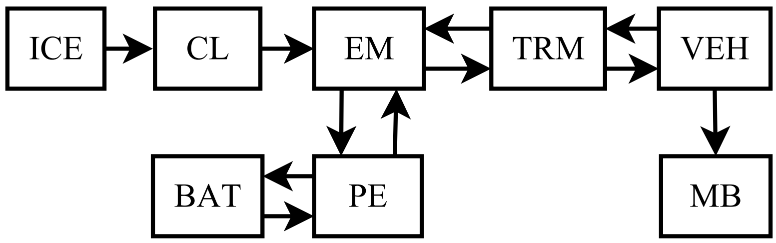

Figure 2-1
Parallel Hybrid Power Train. ICE: Internal Combustion Engine, CL: Clutch, EM: Electric Machine, TRM; Transmission, VEH: Vehicle, BAT: Battery, PE: Power Electronics, MB: Mechanical Brake. The vectors indicate possible energy flows in the Powertrain.

2.2 ISAM

VTEC's current model of a hybrid city bus is named Integrated Starter Alternator Motor (ISAM) and it is both a HEV model as well as VTEC's current control system for a HEV. The model is rather complex and it is difficult to give an overall picture of the control systems that are implemented in the model. The details of the model are confidential, and parts of the control algorithm will therefore only be described in general terms.

2.2.1 Overview of the ISAM model

An overview of the HEV part of ISAM can be explained in the following way. The model is module based and constructed of several sub-blocks, describing certain functional parts of the HEV. The parts that are modeled and how they interact can be seen in Figure 2-2.

Figure 2-2
The HEV part of the ISAM model. ICE: Internal Combustion Engine, MAUX: Mechanical Auxiliaries, CL: Clutch, EM: Electric Machine, EAUX: Electrical Auxiliaries, BAT: Battery, GB: Gearbox, RA: Rear axle, VEH: Vehicle.

The way that this model works is that the drive cycle is fed to a standard PID-driver. For more information about PID regulators see for example 'Reglerteknikens grunder' by Bengt Lennartsson [5], which tries to follow the desired velocity by changing the angle of the speed pedal. This in turn gives an indication of the torque needed to fulfill the drive cycle. The control system part of ISAM changes all the other variables i.e. gear choice and the amount of torque that the ICE and the EM individually should produce etc. The part of ISAM that is shown in Figure 2-2 is only the HEV which contains the physical models of each sub-system as well as actuators and sensors from and to ISAM's control systems.

2.2.2 Transition from time to distance

Since the ISAM model is time-based, and preview information most likely will be distance-based, the model has to be changed to incorporate these differences. To construct preview information we have chosen to use a small part of the drive-cycle. This means that the drive cycle also has to be distance based. To achieve this all, drive-cycle variables where converted to be functions of distance instead of functions of time.

The drive-cycle model was changed so that the look-up maps used vehicle distance on the x-axis instead of simulation time. This created a problem with the driver model, which basically is a standard PID- regulator that controls the acceleration of the vehicle to minimize the error between the vehicle speed and the desired speed. If the velocity at a certain distance is zero the driver gets stuck, since the distance does not change if the desired speed of the drive cycle is zero.

This problem was solved by adding a new drive cycle parameter to the drive cycle. The new parameter was the time the vehicle should stand still at each stop. This parameter was then used in the ISAM model in the following way.

1. If the vehicle reached a point where the vehicle's speed was zero a timer started.
2. When the timer reached the stop time parameter the vehicle's position was shifted forward to the next position where the vehicles speed differed from zero.

In this way the vehicle stands still at each stop and starts when the stop-time is reached.

The PID-driver in the ISAM model in general gives a somewhat unsatisfactory result in following a certain velocity profile. When the velocity profile of the drive cycle saturates or reaches a peak, the PID-driver overshoots the target velocity almost every time. This problem was corrected by complementing the PID-driver with an anti windup structure. This regulator structure made the overshoots disappear instantly. For information about anti windup regulators see for example 'Reglerteknikens grunder' by Bengt Lennartsson [5]. Therefore this modified version of the PID driver has been used throughout the project.

2.3 Simplified Model

The simplified model (SM) that is used during the optimization is a simplified version of ISAM's HEV model. The simplicity of the SM keeps the calculation time of the DP within acceptable limits. The SM is built up by a battery model, an ICE model and an EM model. ISAM contains a lot of different states that are of no or little use to the fuel consumption calculation. Therefore the SM contains only one state which is the state-of-charge (SoC), which is updated according to Equation 2-1. The rest of the model is based on static look-up tables.

$$\frac{dSoC}{dt} = f(T_{EM}, N_{EM}, SoC)$$

Equation 2-1

Inputs to the simplified model are:

- T_{ICE}, torque from the ICE
- N_{ICE}, speed of the ICE
- T_{EM}, torque from the EM

- N_{EM}, speed of the EM

The simplified model then calculates the fuel consumption and the new SoC for these specific torques and speeds. A schematic figure of the model can be seen in Figure 2-3. Equation 2-1 is used it the BAT-block to calculate the new SoC, while the fuel consumption is calculated in the ICE-block by using a look-up table. These signals are then passed as output from the model.

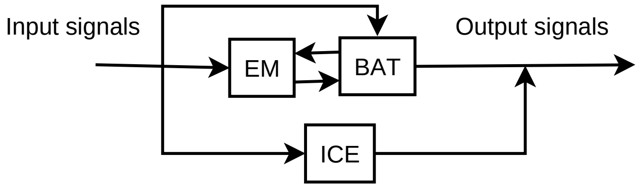

Figure 2-3
Schematic view of the simplified model. The signals that the vectors in the figure contain are described in the sections below.

2.3.1 Battery Model

The modeling of the battery in the SM is mostly done by look-up tables, which give the open circuit voltage and the internal resistance the battery is operating at. The battery open circuit voltage is fed to the EM and is used when the EM needs to produce torque. The dynamic state of the battery, SoC, is changed according to Equation 2-1. Inputs to the battery model are the temperature of the battery, the load from electrical auxiliaries and the power that EM consumes or generates. The outputs of the battery model are the open circuit voltage and the SoC. During simulation the temperature of the battery is held constant at 20 degrees Celsius and the load from electrical auxiliaries is set to 6 kW.

2.3.2 Combustion Engine Model

The internal combustion engine model in the SM is a 2 dimensional look-up table, where the input variables are T_{ICE} and N_{ICE}. The inputs are validated with respect to their min and max levels before the fuel consumption is interpolated. The output of the ICE model is the fuel consumption.

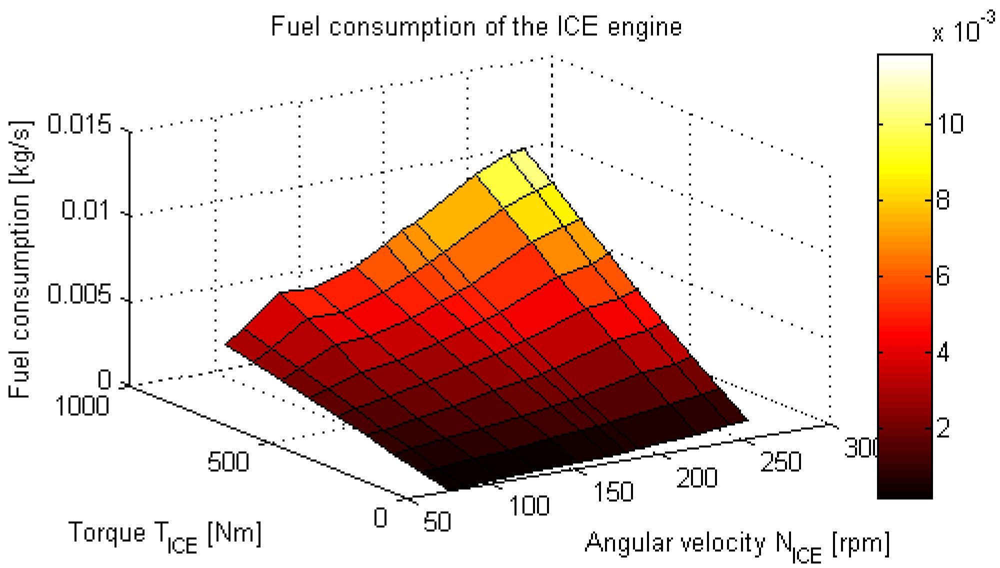

Figure 2-4
The Internal combustion engines fuel consumption lookup table.

2.3.2.1 Starting cost for the ICE

The cost of starting the engine has been calculated from the sum of the energy cost converted to kg fuel for the EM and the amount of fuel the ICE uses during the cranking process. Figure 2-5 show a cranking of the ICE, which occurs in the interval [458.9,460.3].

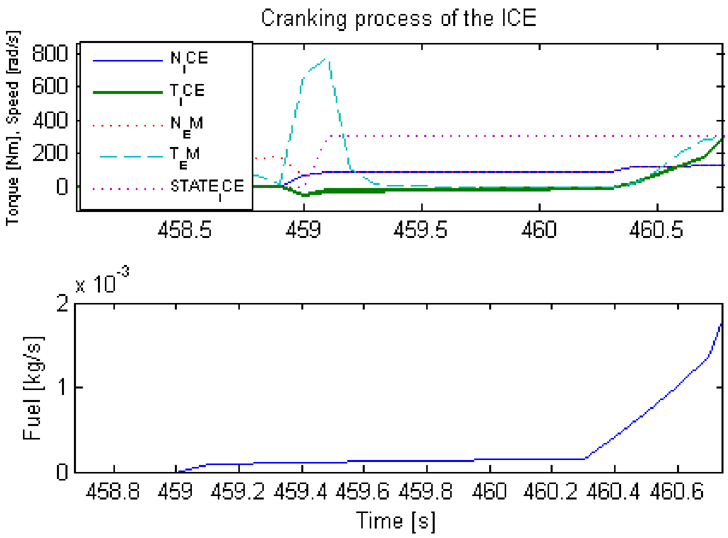

Figure 2-5
Cranking process of the ICE.

2.3.3 Electric Machine Model

The EM model is a static model, which means that the torque that is requested is instantaneously produced. It has a maximum torque that is a function of speed which can be studied in Figure 2-6. The regenerative section of the EM capabilities is simply a mirrored function of the maximum torque, i.e. the EM can consume the same amount of power that it can produce.

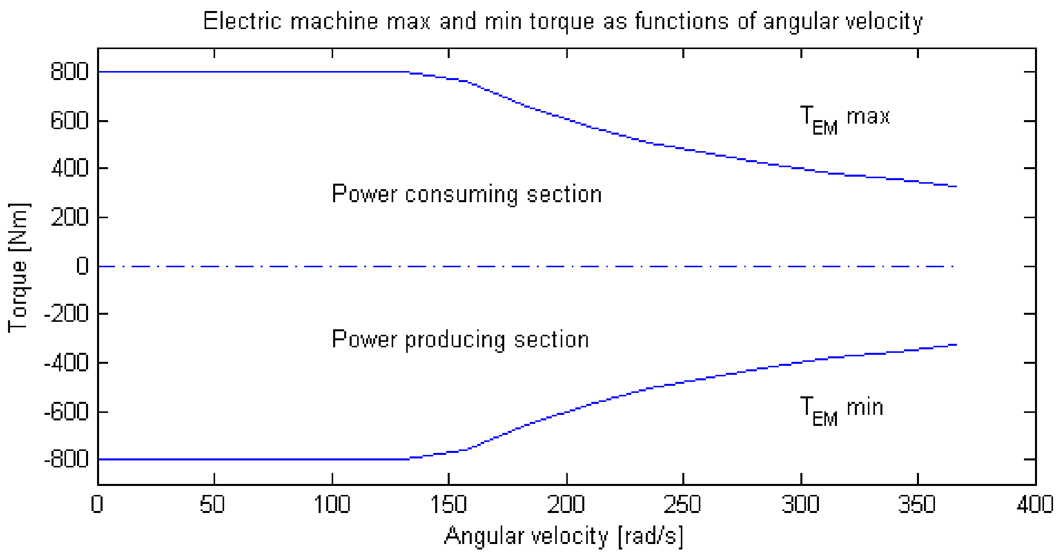

Figure 2-6
The EM model working area.

Inputs to the EM model are T_{EM} and N_{EM}. The model then calculates the power needed to create this torque or the power generated at this certain angular speed according to Equation 2-2. The output of the EM model is the useful Power produced, which is calculated by taking the difference between the total power produced and the power losses in the EM.

$$P_{EM} = T_{EM} \cdot N_{EM} + P_{EMLOSS}$$

Equation 2-2

The P_{EMLOSS} term in Equation 2-2 is the built in losses in the EM, which are a sum of frictional forces and field losses. The power losses are interpolated from a lookup table in the EM model which can be seen in Figure 2-7.

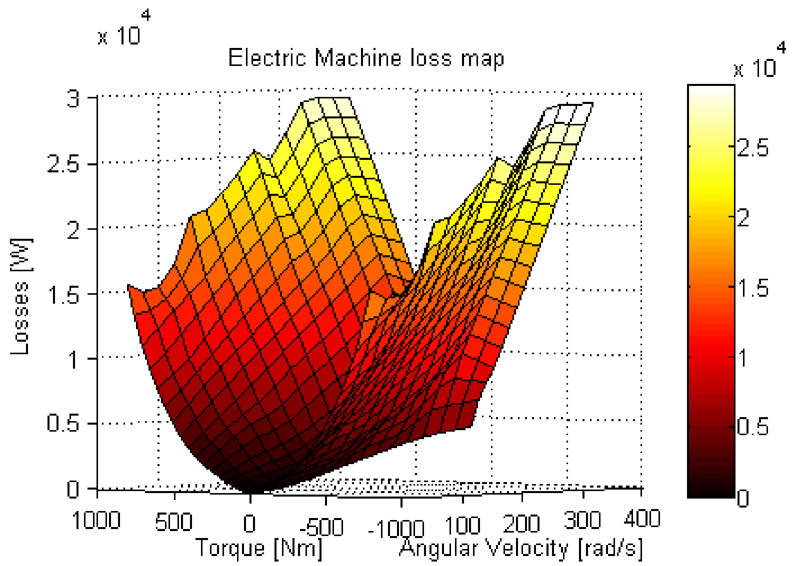

Figure 2-7
The Electric Machine loss map.

2.4 Drive Cycles

A number of drive cycles has been used to study the behavior of the optimal solutions versus ISAM's control signals. The majority of all simulations and evaluations have been done on a drive cycle called CBR85. A random drive-cycle generator has been constructed to be able to design tailor-made drive cycles. One cycle produced by this generator has also been evaluated in this report.

2.4.1 CBR85

The CBR85 drive cycle is a cycle taken from City Bus Route 85 in Göteborg, the route no longer exists, but it has been used for many years for simulations by Volvo. The route stems from Körkarlens gata and goes by Masthugget before it returns to Körkarlens gata in Göteborg. Velocity and altitude trajectories can be seen in Figure 2-8 and Figure 2-9 below. Note that the velocity and altitude trajectories in the figures in chapter 2 are functions of distance and not functions of time.

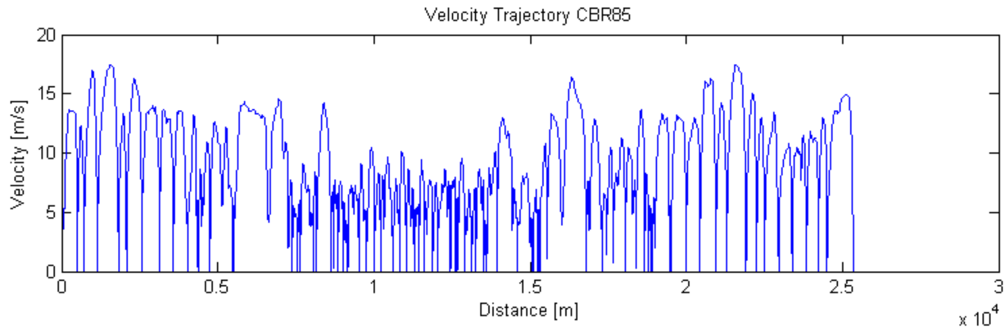

Figure 2-8
Velocity trajectory for the CBR85 drive cycle.

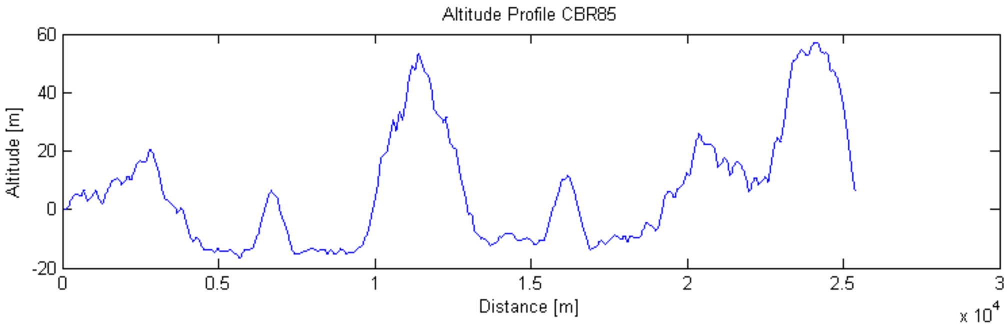

Figure 2-9
Altitude profile for the CBR85 drive cycle.

2.4.2 Random Drive Cycle Generator

The random drive-cycle generator is a tool that constructs drive cycles that can be used for simulations on both the simplified model and the ISAM model. The generator can be used to design drive cycles that are tailor made for specific routes or scenarios. Input data to the generator consists of *route distance, maximum angle of the road, maximum angle change of the road, stop distribution, speed limits along the route, deceleration, acceleration, resolution, time sample frequency and the length of a low pass filter*. With these variables, the generator constructs a drive cycle that fulfills the desired constraints and returns four vectors containing, distance, velocity, acceleration, and stop time. The script also returns an approximation of the total simulation time. An example of a randomly generated drive cycle that is used in our simulations can be seen in Figure 2-10.

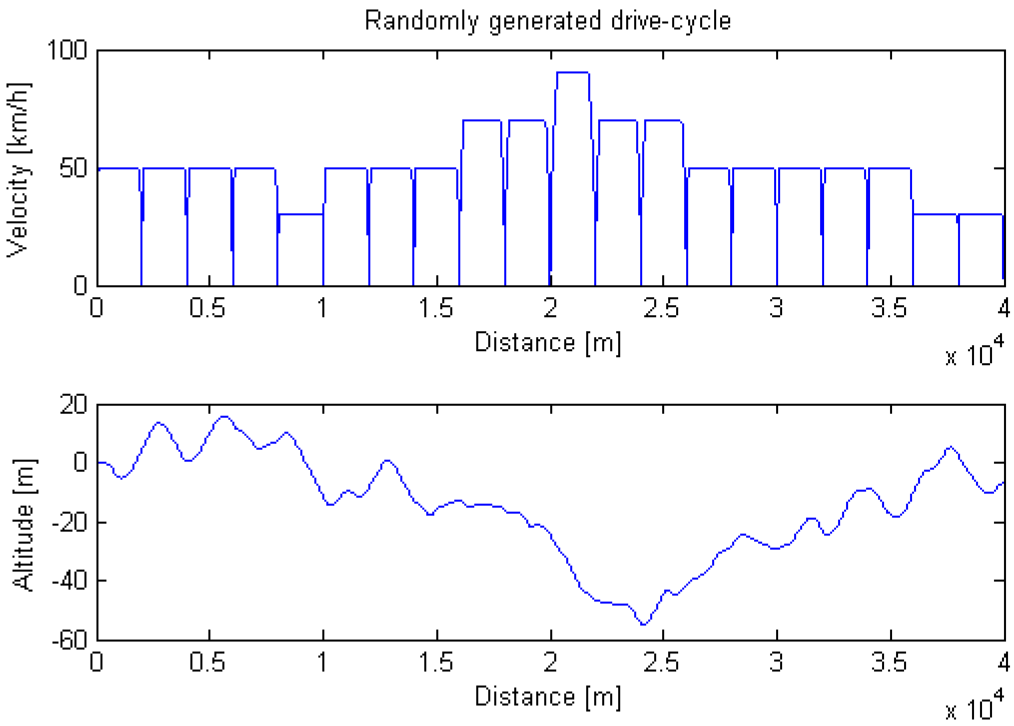

Figure 2-10
Randomly generated drive cycle.

2.5 Preview Information

Preview Information (PI) describes what is expected to happened in the near future. In our case PI consists of map-data describing the path ahead. By using a Global Positioning System (GPS), we can determine our position on a topographic map. When the desired route is specified, a road profile can be calculated, and by using a database of current speed limits, we can also estimate the speed of the vehicle in the near future. This data is then connected to a distance vector describing at what distance in front of the vehicle this information applies. Figure 2-11 shows an illustration of a preview window.

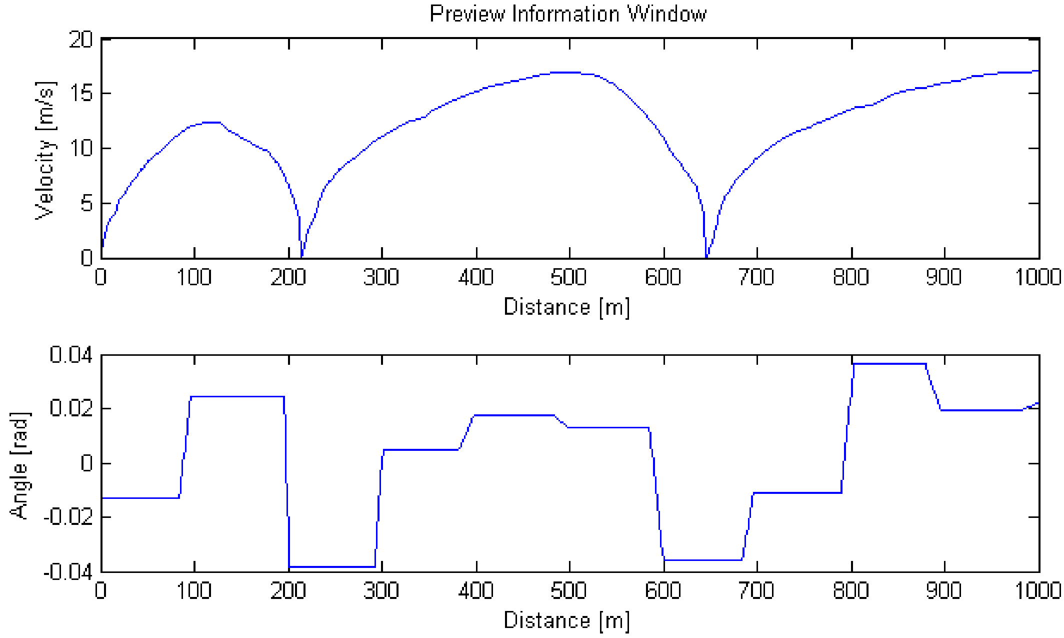

Figure 2-11
Preview window example

2.5.1 Power Estimation

To estimate the power needed to fulfill these preview demands, the required torque on the wheels must first be calculated. This is done according to Equation 2-3.

$$T_{acc} = \frac{a_{veh} \cdot J}{r_{wheel}} + F_{loss} \cdot r_{wheel}$$

Equation 2-3

The acceleration a_{veh} can be approximated from the preview information. By assuming constant acceleration between two sample points, the time between two sample-points can be calculated using Figure 2-12, Equation 2-4 and Equation 2-5.

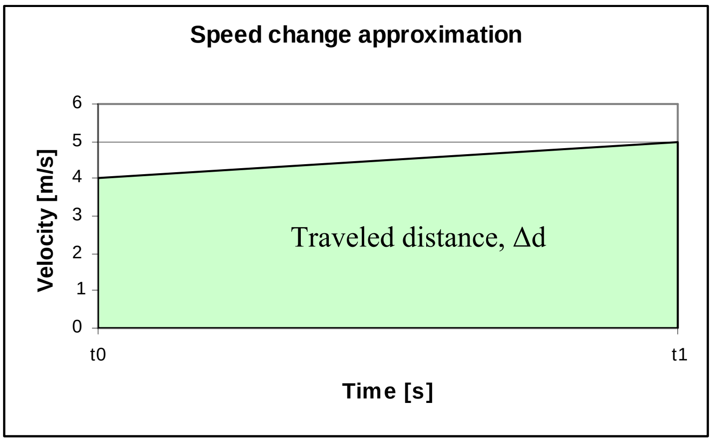

Figure 2-12
Speed change approximation between two sample-points

$$\Delta d = (t_1 - t_0) \cdot v_0 + \frac{(v_1 - v_0) \cdot (t_1 - t_0)}{2} \quad where\, v_0 = 4 \; and \; v_1 = 5$$

Equation 2-4

The distance is also given from the PI, and therefore the time difference t_1-t_0 can be calculated

$$\Delta t = t_1 - t_0 = \frac{v_1 + v_0}{2 \cdot \Delta d}$$

Equation 2-5

The acceleration can now be estimated according to Equation 2-6.

$$a_{veh} \approx \frac{\Delta v}{\Delta t} = \frac{v_1 - v_0}{\Delta t} = \frac{v_1^2 - v_0^2}{2 \cdot \Delta d}$$

Equation 2-6

The losses come from drag, roll resistance, grade forces and mechanical friction. The losses are calculated according to Equation 2-7 - Equation 2-11. Lord Rayleigh's formula [6] is used to calculate the drag in Equation 2-8.

$$F_{loss} = F_{air} + F_{roll} + F_{grade} + F_{mech}$$

Equation 2-7

$$F_{air} = \frac{\rho_{air} \cdot C_d \cdot A \cdot (v_0 + \frac{\Delta v}{2})^2}{2}$$

Equation 2-8

$$F_{roll} = m_{veh} \cdot g \cdot \mu_{roll} \cdot \cos(\alpha) \quad where\ \alpha\ is\ the\ road\ angle$$

Equation 2-9

$$F_{grade} = m_{veh} \cdot g \cdot \sin(\alpha)$$

Equation 2-10

$$F_{mech} = \text{Constant}$$

Equation 2-11

By multiplying the torque given from Equation 2-3 with the angular speed of the wheels, and summing these values inside a preview window, the total power requirement is given according to Equation 2-12.

$$P_{window} = \sum_{window} T_{acc} \cdot N_{veh}$$

Equation 2-12

Driving along the part of a CBR85 shown in Figure 2-11, using a 10 second preview window gives a power estimation according to Figure 2-13. Each point along the curve represents the sum of the power need in a 10 second window ahead of the vehicle. The top part of the figure shows the total power in each window. The middle part shows the positive power, or power needed to propel the vehicle, in each window. The bottom part shows the negative power, or power that could be regenerated, in each window.

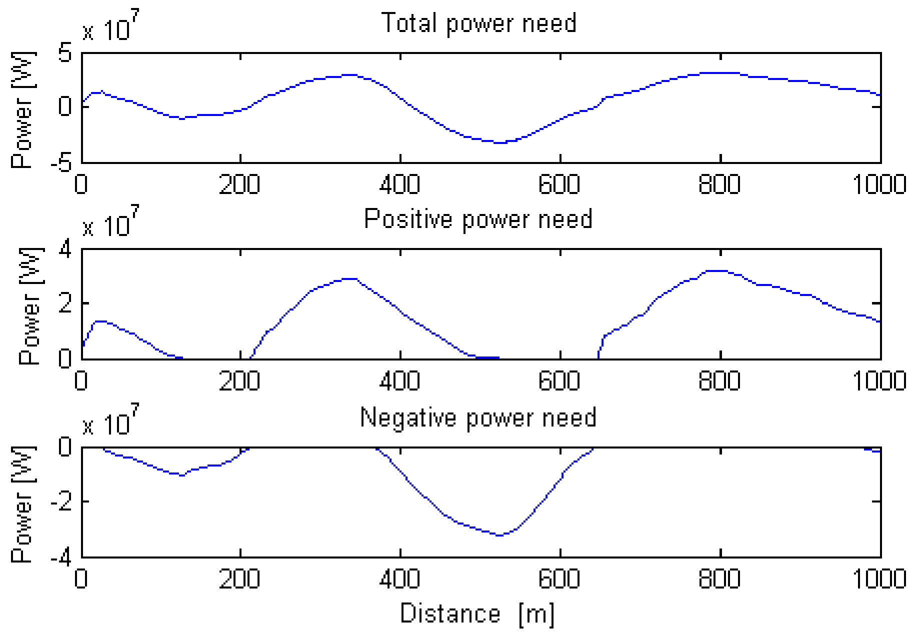

Figure 2-13
Power estimation using a 10 second preview window. Total power is the sum of positive and negative power. Each point in the plot represents the energy in a 10 second window. Positive power must be produced by the vehicle and negative power can be regenerated by the electric machine.

2.6 Battery Evaluation

To understand how much energy the battery contains, and to compare this energy buffer to the energy that is needed to propel the vehicle a test was made. This was to evaluate how far the vehicle could travel by means of only using the EM. The amount of energy that the vehicle could use during this test was the energy in the desired SoC interval, given by SoC_{MIN} and SoC_{MAX}. The battery in total contains 11kWh and the energy between the limits is 10% of the total energy which equals 1.1kWh. The reason to only use this amount of the total energy in the battery is to reduce the ware and increase the lifetime of the battery.
The test was made in such a way that the vehicle was started from stand still and accelerated up to a certain velocity, while the inclination of the road was held constant. During this drive, the power produced is calculated by using Equation 2-3 multiplied with the angular speed of the wheels. When the integral of the power reached 1.1kWh the distance was recorded. The test was made for several inclinations and top speeds and the result can be seen in Figure 2-14. This is a slightly overestimated distance, since no losses have been included in the calculations but it gives an approximation of the distance at different velocities and inclinations.

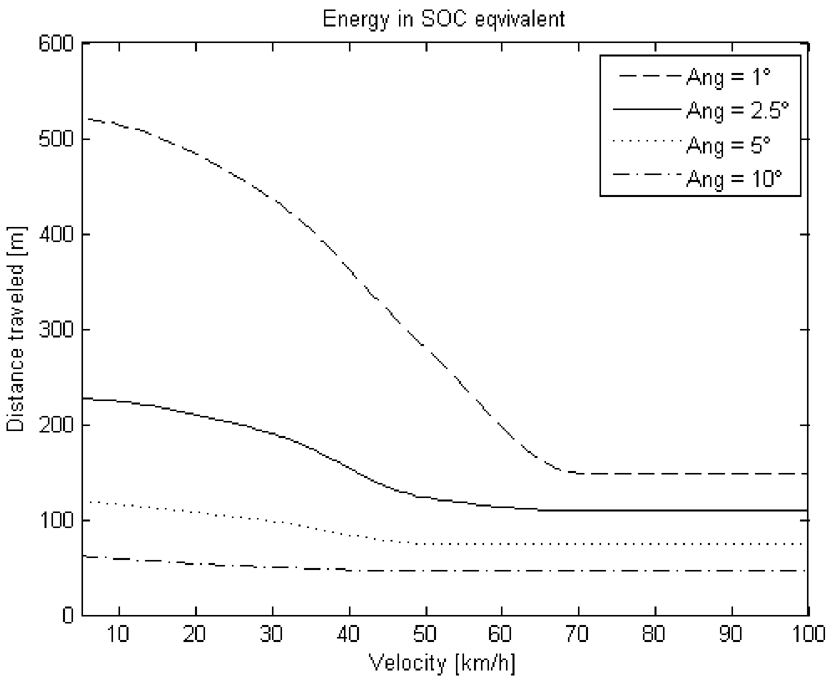

Figure 2-14
Utilization of the energy in the battery buffer.

As can be seen in the right side of Figure 2-14 when the desired top speed is relatively high the vehicle is accelerating during the whole test and the traveled distance becomes constant.

The energy in the SoC buffer can also be transformed directly to an amount of diesel by using the conversion that 1 liter of diesel fuel (0.850 g) contains 40.9 M Joule † and that 1 Joule corresponds to 27.778 µ kWh. The efficiency of the diesel engine can roughly be estimated as 0.4 for all its working points. This estimation results in that the amount of fuel that 1.1kWh holds is approximately 0.24 liter or 0.206 kg of diesel fuel.

† Diesel fuel energy content varies depending on what type of diesel one looks at.

3 Problem description and solution methods

This chapter will describe the optimization problem in general, the technique that is used to find the optimal control and the technique that is used to mimic and utilize the optimal control behavior.

3.1 Describing the optimization problem

The optimization problem should contain a cost function that should either be minimized or maximized, and also describe the dynamics of the problem. If there are active constraints that limit the space in which the optimal solution should be contained, they should also be specified. A general optimization problem is given by Equation 3-1, which is a minimization problem.

$$\min \int_{t_{start}}^{t_{end}} L(x(t),u(t))dt + \phi(x_{t_{end}})$$

$$\dot{x}(t) = f(x(t),u(t))$$
$$u(t) \in U, 0 \le t \le t_{end}$$
$$x(t_{start}) = x_{start}\ ,\ \psi(x(t_{end})) = 0$$

Equation 3-1

Where x(t) and u(t) are vectors with n and m elements, ψ(t) is a vector with r elements. The control signal u(t) is limited by the constraint u(t) ∈ U where U is a set in R^m. If u is a scalar, then U often is an interval and the constraint is in the form $u_{min} \le u \le u_{max}$. The criteria that is to be minimized is given by the cost functions L and ϕ. There are also constraints on the final time instance that are set by ψ. A more thorough description of the optimization problem and its solution can be studied in 'Reglerteori Flervariabla och olinjära metoder' by Torkel Glad and Lennart Ljung [7].

Optimization problems like these can be solved in numerous ways, a few examples are:

- Pontryagin's min max method
- Calculus of variations
- Sequential quadratic programming
- Dynamic programming

We choose to use dynamic programming as working method since it is relatively easy to implement and to understand.

3.2 Dynamic Programming

Dynamic programming (DP) is a technique to find a solution to an optimization problem like the one in Equation 3-1. In our case, DP is used to find a sequence of control signals that minimize the fuel consumption of a HEV, while fulfilling a set of constraints. Dynamic Programming was invented by Bellman and Dreyfus 1962 [8] and is based on what is referred to as Bellman's principal of optimality which states "*The optimum path between two given points is also optimum between any points lying on the path*". An example of an optimal path can be seen in Figure 3-1, if path AB is optimal then path AC is also optimal no matter how the curve continues from point C.

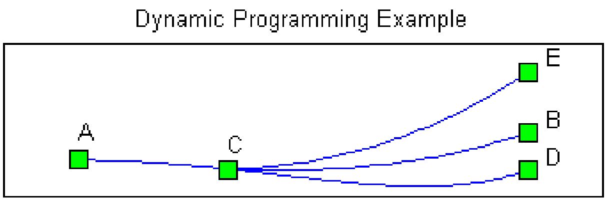

Figure 3-1
Dynamic Programming Example

3.2.1 Finding the Optimal Solution

DP is implemented in the following way. The SoC space between SoC_{MIN} and SoC_{MAX} is divided into a discrete space with 100 points between. This value of a 100 has been evaluated in a previous thesis work [3].

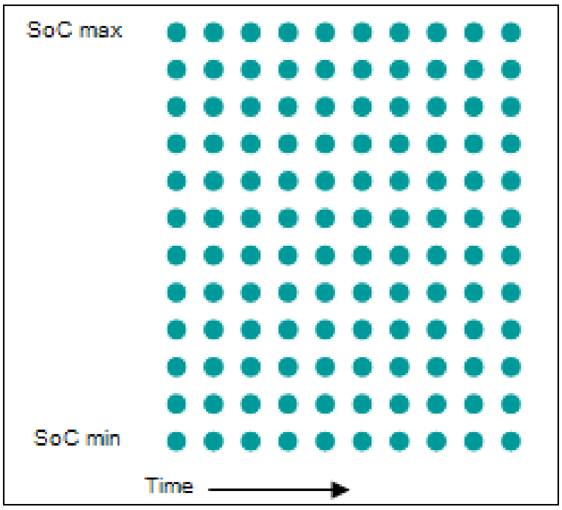

Figure 3-2
The SoC space descritization, in reality there are 100 points between SoC_{MAX} and SoC_{MIN}.

At every time instance, every SoC level is evaluated by looking at the fuel cost for every torque split between the EM and the ICE. The torque produced by the EM is varied between $T_{EM,min}$ and $T_{EM, max}$, which is divided into an interval of 100 points. T_{EM} min and max can be seen in Figure 2-6. $T_{EM} + T_{ICE}$ has to fulfill the T_{DEM} from the forward simulation done by ISAM. The fuel cost for every control signal is evaluated and the minimum cost with its corresponding control signals is saved. The globally optimal solution is then calculated by minimizing the path cost over the total time interval. An example of the minimization algorithm can be seen in Figure 3-3, where the fuel costs have been replaced by integers for easier understanding.

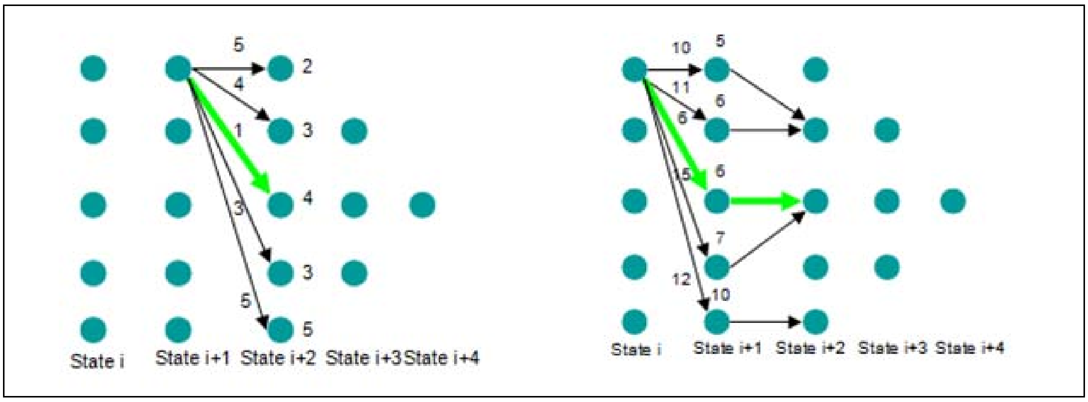

Figure 3-3
Example of Dynamic Programming algorithm procedure.

The minimum cost to go from State i+1 to State i+4 at the highest SoC level (left-hand side of Figure 3-3) is 5. 1 in cost for the path between State i+1 to State i+2, and 4 in cost for the optimal path between State i+2 to State i+4. This evaluation is done for all the SoC levels at State i+1. Then one moves to State i (right-hand side of Figure 3-3) and evaluates all the path costs to go to State i+1 for all the SoC levels. The minimum cost to go from State i to State i+4 is then the optimal path between State i to State i+4.

3.3 Fuzzy Logic

A major part of this thesis work has been to find a way to categorize preview information, and current vehicle state, in such a way that it can be used to control the vehicle in a nearly optimal way. In the approach we have chosen, the optimal control signals from the dynamic programming solution, are connected to the corresponding vehicle states and preview information at each time instance. By organizing the optimal torque split in an N-dimensional matrix where each dimension represents a vehicle state or a preview window, the optimal control signals can be represented by positions in this N-dimensional space. To accomplice this, the dimension has to be discretised, and the number of dimensions kept low in order to keep the complexity within acceptable limits. In Figure 3-4 this is illustrated for a 2-dimensional case.

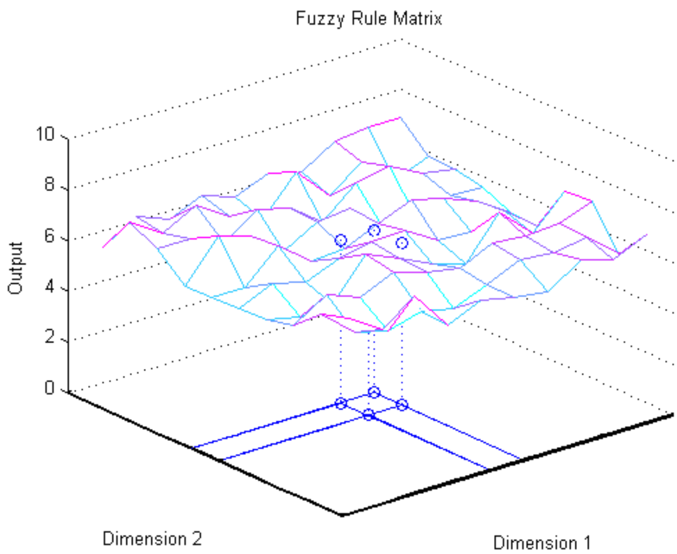

Figure 3-4
Illustration of a 2-dimensional matrix. The surface is given from our method to generalize the optimal solution for each combination of the two dimensions.

This method brings a couple of other questions to light. For example, what happens in the transition between two quantized steps, and what makes one control signal better than another, when we have combined a set of states to represent a single control signal? Is there a way to weigh these control signal in such a way that we can pick the best representative for a specific combination of states? One way to solve this problem, which is the approach we have chosen, is to use Fuzzy logic. Fuzzy logic makes the transition between quantized steps smooth, and also weighs the control signals in such a way that the best representative, according to certain rules, is picked. In Figure 3-5, the smooth quantification of a dimension is illustrated.

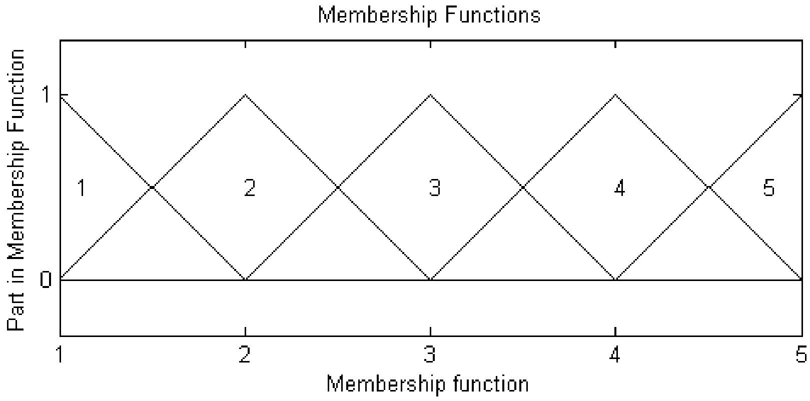

Figure 3-5
Membership functions in one dimension.

Each quantification step in Fuzzy Logic is called a Member Function (MF). The sum of all MFs is always one, and as the figure shows, it is possible to be a part of two MFs at the same time. There are different shapes for MFs, and in our case, they are triangular, because of the rapid calculation time on linear curves. This means that two MFs are active at all times. This also means that for every set of states, there will be two active MFs for each dimension, and hence 2^M combinations of MFs for an M-dimensional space. By doing this, we create a finite number of possible combinations which we arrange in an M-dimensional matrix. We call this matrix a Fuzzy matrix (FM) since it contains the fuzzy control signals connected to each set of states. This leads us to the question of how to combine multiple control signals into a single value.

3.3.1 Training a Fuzzy Matrix

We assume that the optimal T_{EM} can be described as a function of the vehicle states and PI. By doing this, the optimal solution is presumed to act alike for every identical set of states and PI. Since the size of the FM grows with every dimension, a few key states that are presumed to be most significant to the optimal T_{EM} are chosen. The states we use in this thesis work can be seen in chapter 4.5. The T_{EM} can then be described by a multi-dimensional function, which can be seen in Equation 3-2.

$$T_{EM}(t_i) = f(State_1(t_i), State_2(t_i), ..., State_n(t_i))$$

Equation 3-2

To train a FM, an optimal control solution is stepped through and the optimal T_{EM} along with the significant states values are collected at each time instance see Equation 3-3 and Figure 3-6. The optimal $T_{EM}(t_i)$ is placed in its corresponding position in the FM, which has the significant states as its dimensions. An example can be seen in Equation 3-3 and Figure 3-6 where the significant states are SoC, T_{DEM} and two preview windows.

$$T_{EM}(t_i) = f(SoC(t_i), T_{DEM}(t_i), PI_1(t_i), PI_2(t_i))$$

Equation 3-3

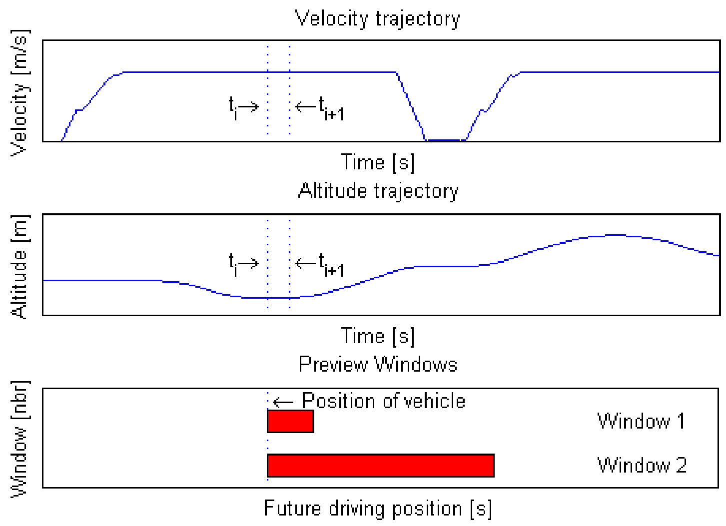

Figure 3-6
Data collection from an optimal control path.

Since the degree of membership for the active MFs in each dimension vary, this can be used to weigh a mean of all hits at a single point in the FM. By using the minimum value of all MFs as a weight for the corresponding control signal a weighted mean can be calculated according to Equation 3-4.

$$\frac{\sum_{n=1}^{N} T_{EM,n} \cdot \min((1-x_{1,n}),(1-x_{2,n}),\ldots(1-x_{M,n}))}{\sum_{n=1}^{N} \min((1-x_{1,n}),(1-x_{2,n}),\ldots(1-x_{M,n}))}$$

Equation 3-4

Where n is the specific hit from the optimal control, $(1\text{-}x_{M,n})$ is the amount of membership in the membership function at dimension M associated with the n'th hit.

This is done for all 2^N combinations of MF. If we do this for a whole optimized drive cycle (DC), or preferably many different DCs, we hopefully get a good coverage of our FM, as well as many hits at each point in the matrix. To evaluate if a certain value is good in the FM we also calculate the Standard Deviation (SD) for every point in the matrix. This value tells us how much the optimized control signals has varied around our weighted mean, and hence gives us a hint if the value is a generally good control signal. This part can be called the training part of the FM, since we try to learn it good behavior from an optimal solution.

3.3.2 Evaluation of a FM

The work that has been done on the creation process of the FM is summed up in the steps bellow and the chapters they are explained.

- Choosing the dimensions of the FM (chapter 3.3.2.1).
- Setting the number of MF of each dimension in the FM, (chapter 3.3.2.2).
- Number of preview dimensions in the FM, (chapter 3.3.2.3).
- Choosing how long time the preview windows should look ahead, (chapter 3.3.2.4).

All of the above parameters might not be altered individually without affecting the optimal choice of the other parameters. This means that if one alters the number of membership functions for a certain dimension in the FM, it is not given that the window length of a preview vector still is the optimal choice. This chapter will therefore discuss some approaches to evaluate the performance of a FM.

3.3.2.1 Choosing the dimensions of the FM

The number of dimensions and the number of MFs determines the size of the FM *i.e.* if one uses five dimensions and each of the dimensions should contain ten membership functions then the total number of positions that are available is ten to the power of five. This equals a hundred thousand different positions. It is then very likely that the FM obtains a very poor hit rate if it is trained on only one optimal solution for a certain drive cycle. Our definition of hit rate is the number of trained operating points divided by the total size of the matrix. A lot of positions in the matrix will probably not be trained with a value from the optimal solution. To investigate the quality of a FM after training, we look at the hit rate and the mean standard deviation (MSD) of the FM, see chapter 3.3.2.4 for MSD. An example of a significant state space can be seen in Figure 3-7.

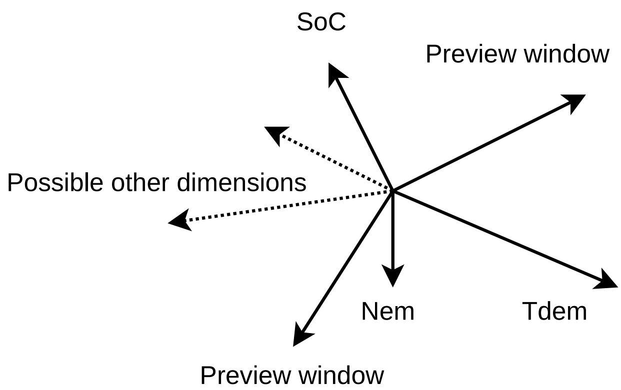

Figure 3-7
The vector room containing all the operation points of the FM, the length of the vectors indicates how many membership functions each vector contains.

Our choices of dimensions are the SoC-level, the T_{DEM} and two preview windows. This is due to that the SoC-level tells something about how much energy that can be withdrawn or stored in the battery. The torque demand gives how much energy that needs to be produced to propel the vehicle according to the drive cycle in the specific time instance. The other two dimensions which are the preview windows show how much energy that needs to be produced or how much energy can be regenerated during their time length. Another dimension that could be a strong candidate is N_{EM}.

3.3.2.2 Setting the number of MF of each dimension in the FM

Setting the number of MF that each dimension should contain is not a straightforward procedure. It is not obvious how much one can quantize a dimension before loosing too much information. We used a trial-and-error method to find good values for the MF count on the different dimensions. The goal was to get a high hit rate while keeping the SD as low as possible.

3.3.2.3 Number of preview dimensions in the FM

The choice of having two preview dimensions in the FM is due to that two windows should be able to give a good understanding of how the future looks. Let's say that a specific road chapter looks like the road inclination in Figure 3-8 and that the SoC level is relatively low and the vehicle is positioned at the arrow head.

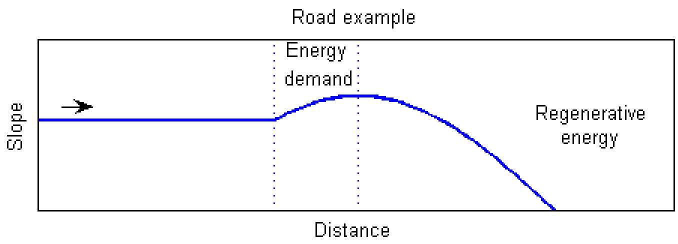

Figure 3-8
Road example.

If only one preview dimension had been present and its window length would end somewhere inside the energy demand region, then the best thing for the vehicle to do is to produce extra torque with the ICE, so the EM can regenerate energy. This is to increase the SoC level so that the EM can be used to create torque while in the upward slope. If two preview windows had been present with one ending somewhere in the energy demand region and the other ending at the end of the downward slope, then the best thing to do would be to utilize all the energy in the buffer to create torque for the upward slope since there is a lot of regenerative energy behind the peak of the slope, that can be used later to charge the battery. One could also think of other similar situations when the slope of the road is inverted to the one in Figure 3-8 and the SoC level is high and other examples as well. This is one reason for using two preview windows. Another reason is that the size of the FM, which has been reasoned about in chapter 3.3.2.1, grows very fast with its dimensions.

3.3.2.4 Choosing the length the preview windows

The length of two preview vectors were evaluated from a test where a lot of different lengths where tested. All combinations of having preview vector 1 ranging in the interval [1,100] and preview vector 2 ranging in the interval [101,400] where tested. The FMs created in this test were then evaluated by calculating their hit rate and their mean standard deviation. The mean standard deviation was calculated in the following way. First the standard deviation of all the operating points was calculated according to Equation 3-5. Standard deviation can be read about in any standard calculus book i.e. 'Statistical Digital Signal Processing and Modeling' by M. H. Hayes [9].

$$\sigma = \sqrt{\frac{1}{N}(\sum_{i=1}^{N} x_i^2) - \bar{x}^2}$$

Equation 3-5

$\bar{x}$ is the mean of all the training points in that specific operating point. The mean standard deviation of the matrix is the sum of all the standard deviations in the matrix divided by the number of trained data points. This value gives some sort of indication on how different the training data is in every point: If the mean standard deviation of a matrix is high, it is very likely that the operating points have been given a lot of different control signals and they may not give a good result. An example of such a test can be seen in Figure 3-9.

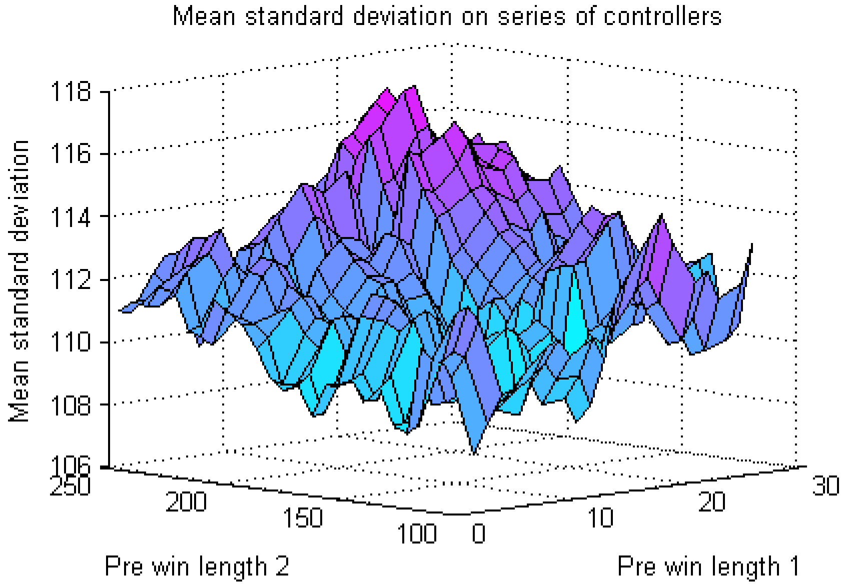

Figure 3-9
Mean standard deviation for a series of controllers with different window lengths. Pre win length 1 and 2 is the preview window lengths in seconds.

Two different settings involving the preview windows configuration have been tested. Configuration one, is when the windows have the same starting point. The starting point in the test is the vehicle's position. The second configuration that has been tested has the windows following each other, i.e. the second window starts where the first is ending. These configurations can be seen in Figure 3-10.

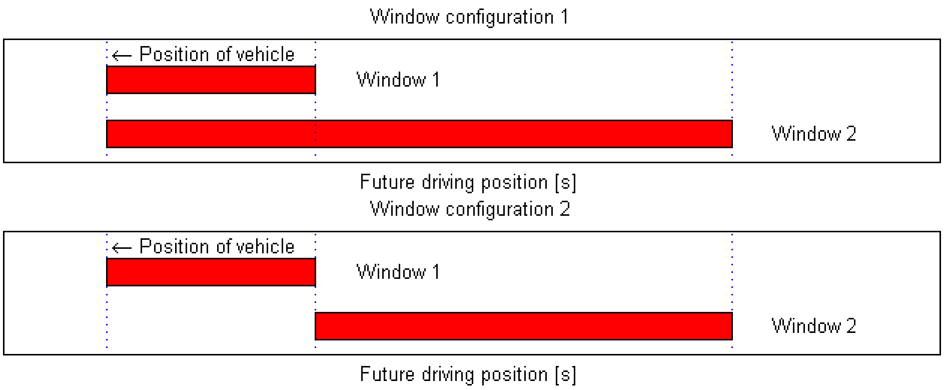

Figure 3-10
Different window configurations in the preview information gathering.

When comparing the test results, configuration one gave a slightly lower mean standard deviation and a slightly higher hit rate, so configuration one has been used in our simulations.

The matrix that is used in our control strategy, is the matrix with the lowest mean standard deviation. The problem with using this certain matrix is that the hit rate is closely coupled to the mean standard deviation. High hit rate turned out to give a high mean standard deviation, this is expected because if one shrinks the size of the FM more hits will end up in a certain position of the FM. This means that a lot of different torque splits will end up in the same position which will cause a higher standard deviation in that position.

3.3.3 Utilizing the Fuzzy Matrix

Now that we have created the FM to mimic an optimal solution, we want to utilize it in a controller. This is done by looking at the same states that were used to create the FM. By taking these values and using their degree of member ship in each MF, we produce 2^M values from the M-dimensional FM. These values are then weighted together according to Equation 3-6, where the weights come from the degree of membership, for each value.

$$\frac{\sum_{k=1}^{2^M} T_{EM,k} \cdot \min((1-x_{1,k}),(1-x_{2,k}),...(1-x_{M,k}))}{\sum_{k=1}^{2^M} \min((1-x_{1,k}),(1-x_{2,k}),...(1-x_{M,k}))}$$

Equation 3-6

This final value can then be used as a control signal. To determine if the value is any good we look at the mean of the SD, for every active FM point. If the MSD of all active points is lower than a certain threshold-value, it can be used as a control signal.

3.3.4 Torque Split Control

When a FM has been trained on an optimal solution for a given drive cycle, it is filled with different control signals in different positions and some positions are not altered at all. Two questions that arise immediately are: When should the control system use a value proposed by the FM and what control signals should be used when the FM does have a suggestion? Some of the values proposed by the FM has a SD that is about 30% or higher of the maximum torque, which is not good since the optimal solution has used a lot of different control signals for this specific position. Some sort of upper limit on the SD has to be incorporated in the control algorithm. If a position in the FM, on the other hand, is trained by only one value it has a SD that equals zero. This is also a value that can not be used as a control signal, since if a position in the FM is only trained by one value, how can one know that it is a good control value for that specific situation.

As a lower bond on the SD value, the control algorithm looks at the number of hits in that specific position in the FM. If a position has been trained with two values or more it is a potential control signal. As a higher bond on the SD, some tests have been done to evaluate what happens if the values under a specific limit are used. If the limit is relatively high, the percentage of usage of the FM increases but the final result can become unstable. The SoC trajectory also becomes more or less unstable i.e. the minimum value of the SoC trajectory can go below SoC_{MIN}, which is not acceptable. When choosing a limit, one has to look both at the FC reduction and at the SoC trajectory.

In Figure 3-11, the FC reduction on different SD threshold limits (SDTL) can be seen. Everything looks good and well and one could think that a value on the SDTL around 160 can be a possible choice. However when looking at the SoC trajectory at this SDTL, one realizes that this limit is unthinkable. The minimum value of the SoC trajectory is 0.46, which is way below the SoC_{MIN} and hence the limit is too high.

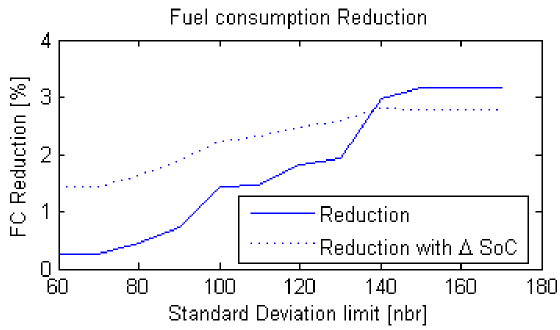

Figure 3-11
FC reduction with different SD threshold limits.

When the NOC is used in the SM, the ISAM control system can not be used to act as a background controller. To compensate for this, a charge sustaining control strategy according to Equation 3-7 - Equation 3-12 is used. When using the NOC implementation in ISAM, the existing control strategy can be used as a background controller.

$$T_{DEM} = T_{DEM,ref} - T_{EXTRA}$$

Equation 3-7

$$T_{DEM} = T_{ICE} + T_{EM} + T_{BREAK}$$

Equation 3-8

$$T_{ICE} = \max(\min(T_{DEM}, T_{ICE,MAX}), T_{ICE,MIN})$$

Equation 3-9

$$T_{EM} = \max(\min((T_{DEM,ref} - T_{ICE}), T_{EM,MAX}), T_{EM,MIN})$$

Equation 3-10

$$T_{BREAK} = (-T_{DEM,ref} + T_{ICE} + T_{EM}) \cdot \Theta(-T_{DEM,ref} + T_{ICE} + T_{EM})$$

Equation 3-11

Where

$$T_{DEM,ref} = T_{EM,ref} + T_{ICE,ref}$$

Equation 3-12

$T_{EM,ref}$ and $T_{ICE,ref}$ are the torques that the forward ISAM simulation produced. Θ is Heaviside's step function. $T_{ICE, MIN}$, $T_{ICE, MIN}$, $T_{EM, MIN}$ and $T_{EM, MAX}$ are the minimum and maximum torques that the ICE and the EM can produce at that specific speed, see Figure 2-6 and Figure 3-13.

T_{EXTRA} is extra torque that is to be produced as a function of the current SoC level. If the SoC level is below the SoC buffer midpoint, the T_{EXTRA} term decreases linearly with the distance from the midpoint. This is to produce more torque with the ICE to recharge the battery. If the SoC level is above the SoC midpoint, the T_{EXTRA} term grows according to a fourth degree function. This is to utilize the energy in the battery so the battery doesn't get overloaded. The T_{EXTRA} function can be seen in Figure 3-12.

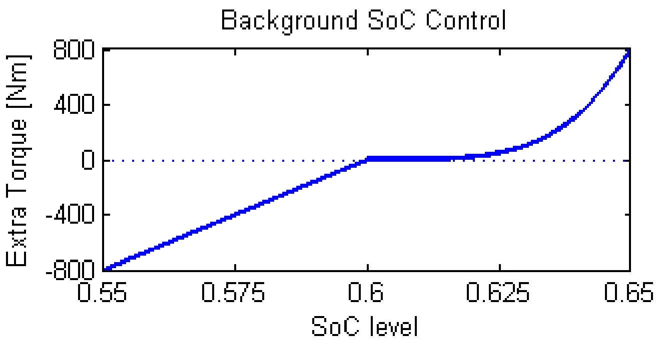

Figure 3-12
The T_{EXTRA} function in the background control of NOC.

3.4 Engine Operating Point

The operation point of the ICE is crucial for its fuel consumption. The operation point can be moved by changing the angular speed of the ICE through a gear change, or changing the requested torque by letting the EM add or remove torque from the ICE. To measure how efficiently the ICE converts the chemically bound energy in the fuel to useful power, Break Specific Fuel Consumption (BSFC) is used. BSFC is the fuel consumption at a specific operation point, divides with the produced power. Equation 3-13 shows how BSFC is calculated, and Figure 3-13 shows the BSFC map for the ICE used in the HEV.

$$BSFC(N_{ICE}, T_{ICE}) = \frac{FC(N_{ICE}, T_{ICE})}{N_{ICE} \cdot T_{ICE}}$$

Equation 3-13

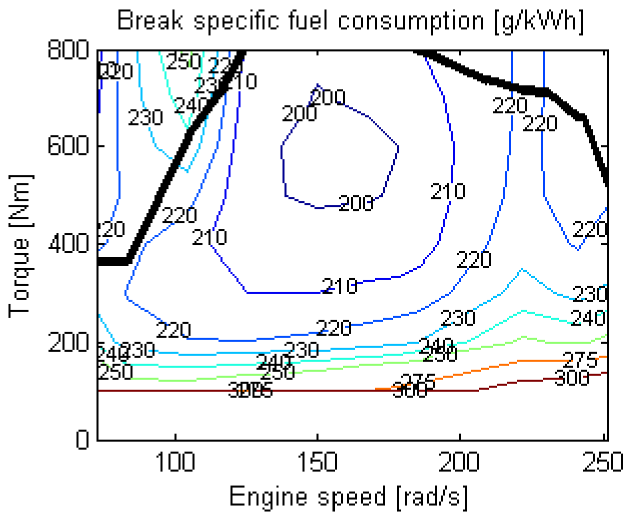

Figure 3-13
Break specific fuel consumption map with no operation points plotted. The solid thick line is the ICE's maximum torque at different engine speeds.

By using this information, it is possible to evaluate an area around the requested torque from the ICE to find a better operation point. When a better point is found, the EM is used to move along the torque axis of the map, and the gearbox is used to move in the angular speed dimension. To make this strategy charge sustaining, the area can be changed depending on current SoC and/or preview information. For example, if the SoC level is low, the limits on the torque axis are moved so the ICE has to produce more torque then T_{DEM}.

3.5 Implementation of Near-Optimal-Controller

One of the goals for this thesis work has been to implement our control strategy in the existing HEV Simulink model. This should be done in an efficient way without slowing down the simulation time of the model too much. At the same time, the implementation has to work in the simplified model, which is based on Matlab scripts. To accomplish this, algorithms making up our controller where implemented in C-code. To make the code useful in Matlab and Simulink, two mex wrapper functions were created. In Simulink, the controller is implemented as an S-function, while in Matlab, the controller is called like any other function.

The controller is divided into three parts.

- **Pre-processing** calculates the energy in preview windows according to chapter 2.5.1.
- **Control** utilizes fuzzy-logic together with a base controller or alternatively the ISAM control system to find a near optimal control signal.
- **Post-processing** takes the proposed control signal and evaluates it according to physical limitations. This part also has the possibility to improve the ICE operation point using the BSFC map.

By doing this the controller becomes very flexible, and can be configured in many ways. Since each part is independent of the other parts they can be used separately or combined depending of the desired behavior.

3.5.1 Implementation in ISAM

The implementation in the ISAM Simulink model is mainly done using S-functions. The S-function block calls the mex complied C-code, described in chapter 3.5. The S-function also has a mask to simplify things for the end user. The Simulink blocks are divided into four different types, Actuators, Sensors, Processors and Gates. These blocks can then be combined to a controller structure. Figure 3-14 shows the collection of available blocks.

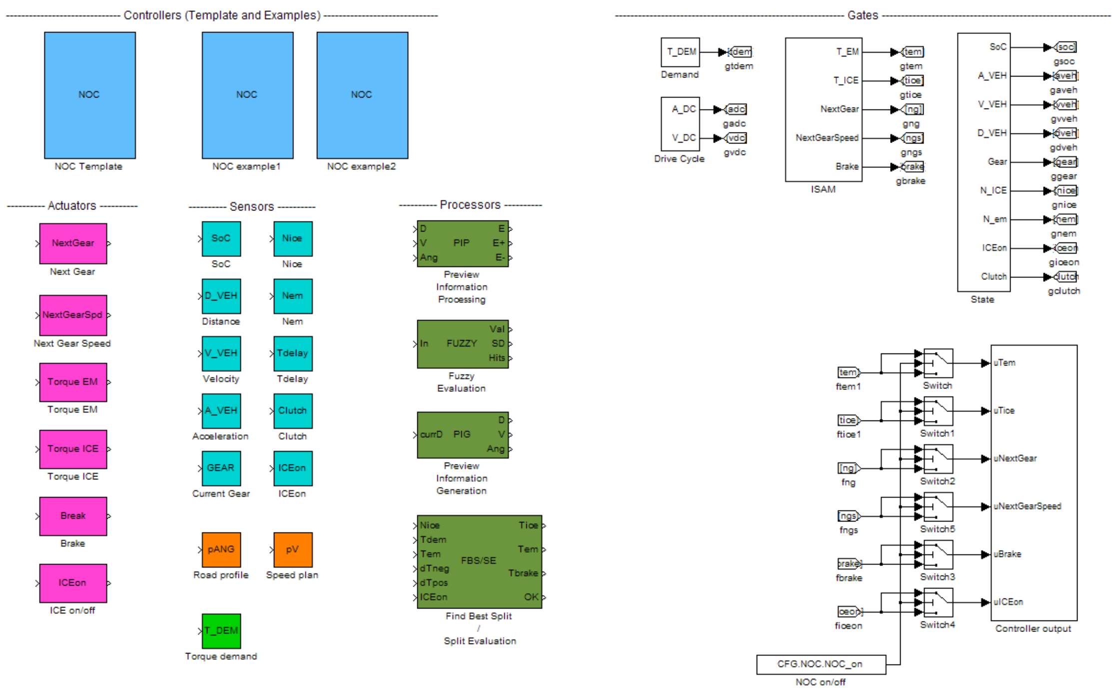

Figure 3-14
Available blocks (Actuators, Sensors, Processors and Gates) in the Simulink NOC implementation.

The sensors are divided into three sub groups:

- State Sensors (Cyan), capture the current state of the vehicle.
- Preview Sensor (Orange), capture information about the road ahead.
- Demand Sensor (Green), captures the driver demand.

The four processors can be described as:

- **PIP** (Preview Information Processing), is used to calculate the energy in preview windows according to chapter 2.5.1.
- **FUZZY** (Fuzzy logic evaluator), is used to calculate a value from the FM, according to chapter 3.3.3.
- **PIG** (Preview Information Generation), is used to construct preview information from a given drive-cycle. This is done to simulate the data which in the future will be available from the onboard navigation system.
- **FBS/SE** (Find Best Split / Split Evaluation) is used to evaluate a given split and torque demand. It can also be used to find a better operation point for the ICE.

3.5.2 Implementation in Simplified Model

The implementation for the simplified model is done much in the same way as in Simulink. Since all signals are available in the simplified model there is no use for sensors, actuators and gates, the only thing that are used is the processors. To use the processors an ordinary function call is used, for example if the Preview-Information-Processing processor should be used one uses the call:

```
[Etot Epos Eneg] = pip(DIST, VEL, ANG, winlen, vehParam, mode);
```

Where DIST, VEL and ANG are vectors with preview information, winlen is a vector with the length of the preview windows, vehParam is a vector with vehicle parameters and mode tells the function if the preview windows should follow after each other or start at the same place. Etot, Epos and Eneg then become vectors which return the total, positive and negative energy in each window.

The only processor that is not implemented in the simplified model is the Preview Information Generator. This is because the simplified model can use part of the forward simulation data for preview information.

4 Results and discussion

In this chapter a comparison between the two models used in this thesis will be made. The results from two optimal solutions, with and with out gear shifting, will be presented and compared to ISAM's control. The results from the near optimal control, with different sets of significant states, will be presented and compared to the optimal and ISAM's solution.

4.1 Comparison between ISAM and the Simplified Model

Since we use the simplified model to calculate the optimal control signals using DP, it would be desirable if these solutions also were representable for ISAM. To compare the results from these two models, we first ran a forward simulation using the drive cycle CBR85 in ISAM. The torque demand, split, ICE speed and EM speed from this simulation where then ran through the simplified model. To evaluate the result, we compared the SoC trajectory for the two models, as well as the cumulative fuel consumption. The results can be seen in Figure 4-3 and Figure 4-4.

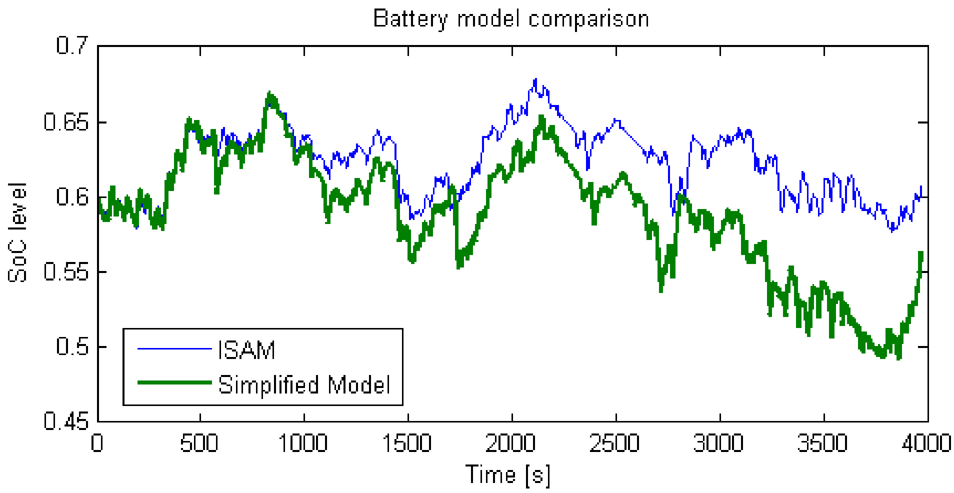

Figure 4-1
Comparison between battery models in ISAM and SM.

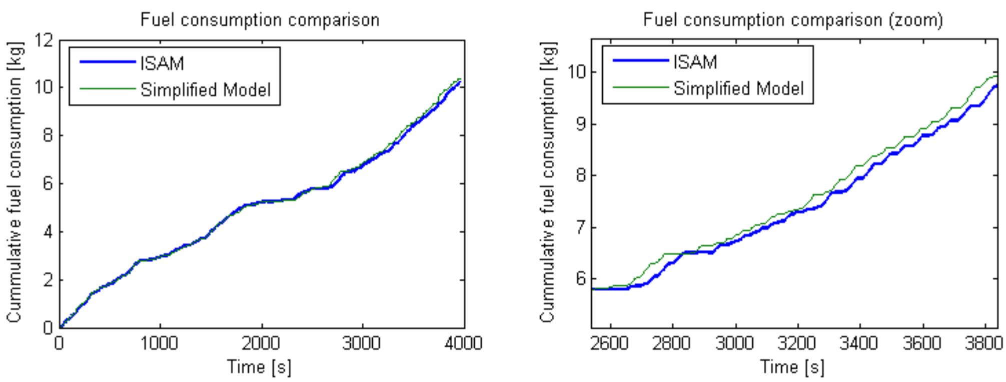

Figure 4-2
Comparison of fuel consumption between ISAM and SM.

From the SoC trajectory, we see that the battery model in SM clearly is missing some dynamics, since it diverges with time. This implies that the optimal control signals will not give an optimal result for ISAM. If we look at the fuel consumption the result is better. The SM tends to use a little more fuel, but if we look at the shape of the curves they are very similar. This indicates that fuel consumption results from the SM can be compared with the result from ISAM. However including the SoC in this comparison will not give an accurate result, because of the differences in the battery models.

Since the goal of this thesis work has been to find a control algorithm that mimics the result of an optimal solution the differences between the SM and ISAM isn't too much of a problem. The results can still be evaluated in the SM using the torque demand from ISAM. The FC from ISAM's control signals ran through the SM can also be used for comparison with the FC for the NOC and optimal solution. The problem occurs when the controller is to be implemented in ISAM. Since it has been trained on an optimal solution from another model, it is not certain that the controller will lower the fuel consumption, or in other way improve the results in ISAM. To be able to evaluate the performance of the controller in this way, an improved version of the SM has to be used.

4.2 Optimal Solution

The results from the optimal solution, calculated on the SM, will be compared to the result from ISAM's control signals put through the SM. The results will be presented in a break specific fuel (BSFC) map, a SoC trajectory and a cumulative-fuel-consumption trajectory. The BSFC map is to emphasis that the operation points that ISAM uses diverge from the optimal solution. A clean BSFC map can be seen in Figure 3-13 in chapter 3.4, to show what the cost is for different operation points. The SoC trajectory gives an idea of where the optimal solution diverges from ISAM's solution. The cumulative fuel-consumption-trajectory is presented due to that one of the main goals of the thesis is to minimize the FC.

4.2.1 Torque split and ICE on/off optimization

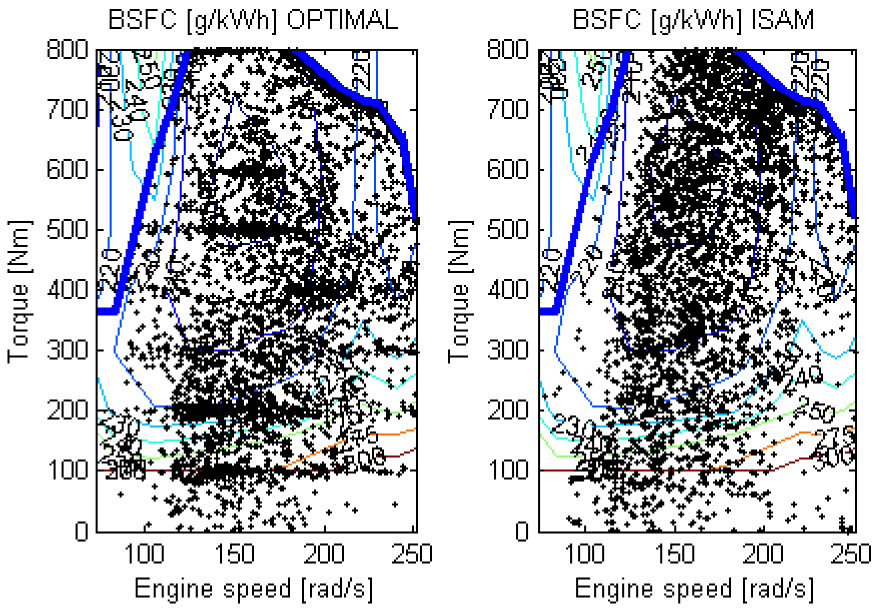

Figure 4-3
BSFC for both the optimal solution and ISAM's control when the torque split between the ICE and the EM is optimized.

In Figure 4-3, one can see that the optimal solution clusters the operation points at certain torque levels. This is a consequence of that the optimal solution utilizes more efficient operation points, in terms of BSFC. The reason way the operation points gather in horizontal lines is that the gear choice in the optimization is the same as ISAM's control. To fulfill the torque demand, (T_{DEM}) the optimization chooses a less expensive point for the ICE, which means that the T_{ICE} is either increased or decreased, and the EM compensates for this by changing its torque. To move the operation points vertically, a gear change has to be made, which is not implemented in this optimization.

The optimal solution has been calculated with $SoC_{MIN} = 0.52$ and $SoC_{MAX} = 0.65$. Figure 4-4 shows that the optimal solution doesn't utilize the whole SoC buffer, which is a result of the constraint that SoC must be above 0.6 at the end of the DC.

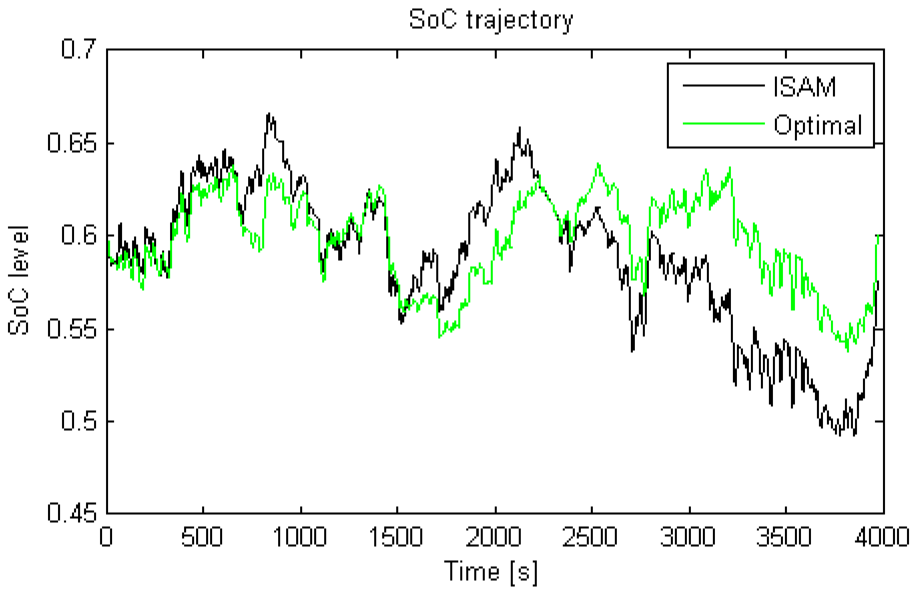

Figure 4-4
SoC trajectory for both the optimal and ISAM's solution.

ISAM's control strategy tends to use a bigger part of the SoC buffer because of the differences in the battery models, described in chapter 4.1. The energy flow through the battery is still a valid comparison and is reduced in the optimal solution. While ISAM transfers 125.6 MJ, the optimal solution transfers 94.3 MJ through the battery, which is a reduction by 24.9 %. The fact that less energy is driven trough the battery is beneficial, because it indicates that the battery is used more efficiently and it also reduces the wear on the battery.

The cumulative fuel plot in Figure 4-5 show that both the optimal and ISAM control strategy consume almost the same amount of diesel. The optimal solution consumes 10.06 kg and ISAM consumes 10.37 kg of diesel fuel. This is a reduction by 2.98 % in fuel consumption for the optimal solution.

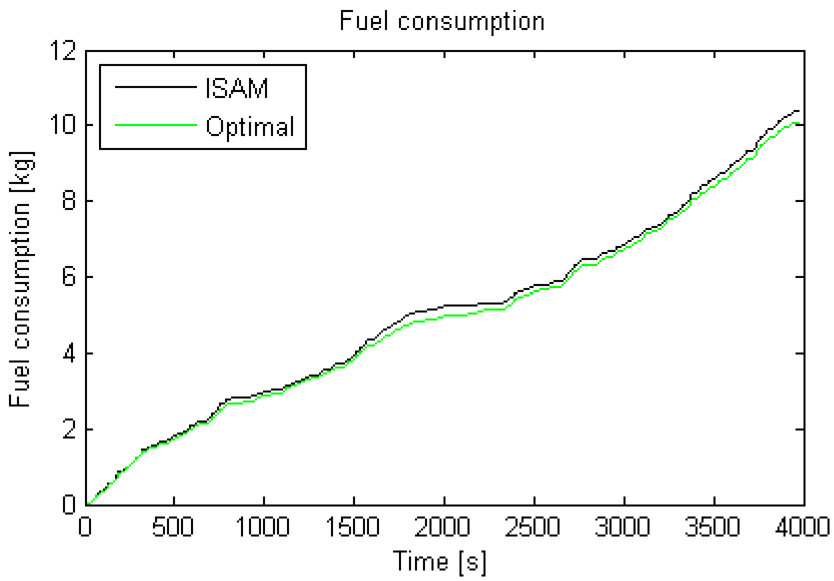

Figure 4-5
Cumulative fuel consumption plot for the CBR85 drive cycle.

If one recalls the SoC buffer evaluation in chapter 2.6, a difference of 0.1 in SoC level corresponds to 1.1kWh and is approximately equal to 0.206 kg of diesel fuel. This means that the SoC difference of 0.06, in Figure 4-4, equals 0.124 kg of fuel, which results in a fuel consumption reduction by 3.47% for the optimal solution. However, since the battery models differ, including the end SoC in this comparison is not fair. The optimal solution also has a less amount of energy flowing through the battery which is a further advantage for the optimal solution.

Figure 4-6 shows that the optimal solution chooses to turn on and off the ICE more frequently than ISAM, this is partly due to the fact that in the optimization algorithm, the ICE has no cranking process, i.e. if the ICE is turned on it can directly start to produce torque.

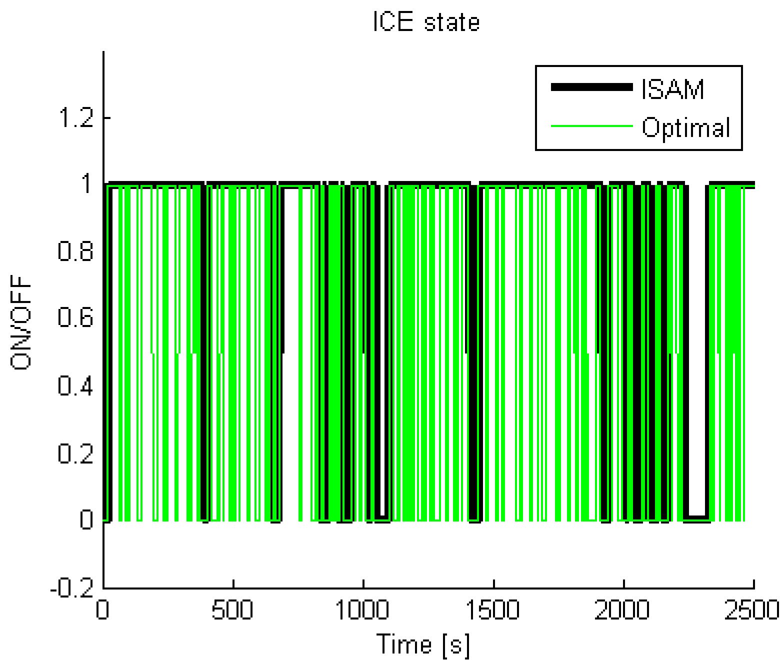

Figure 4-6
ICE state for both ISAM and the optimal solution. Value one on the y-axis means that the ICE is on and value zero means that the ICE is off.

4.2.2 Torque split with Gear Shifting and ICE on/off optimization

As seen in a Figure 4-3 the optimal solutions operation points (OP) clustered in horizontal lines. To be able to move the OP vertically, gear change has to be incorporated in the DP. The results from this addition can be seen in Figure 4-7.

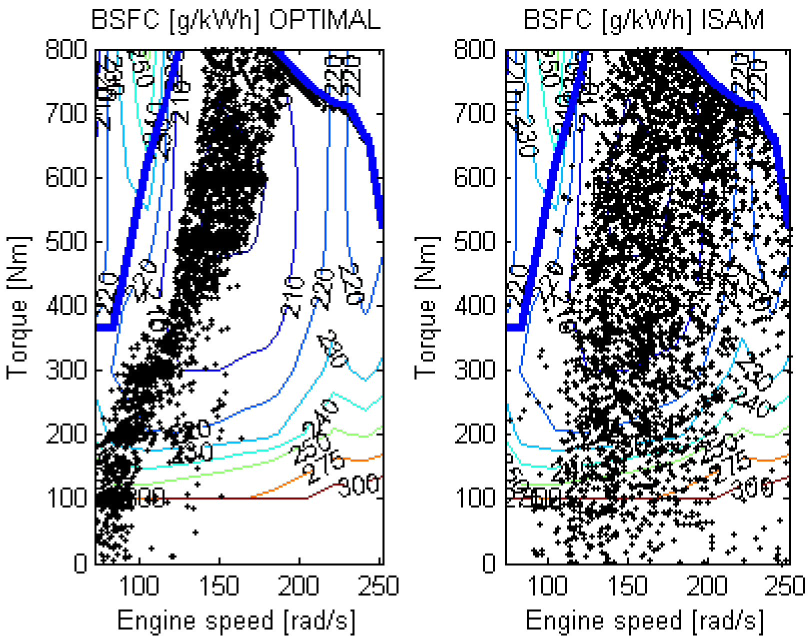

Figure 4-7
BSFC map for both the optimal and ISAM's control strategy when gear choice, ICE on/off and the torque split is optimized.

Figure 4-7 clearly shows that the optimal solution generally use the ICE in more economical OPs. Since the gear box isn't continuously variable, the solution still contains horizontal lines. The problem with this optimization is that the gear shifting sequence is not physically possible. The optimization algorithm is searching for the operation point with the minimal FC and does not consider which gear was used the previous time instance, or how long time that gear has been used. This results in a gear sequence that can make a lot of shifts during a short time interval. In the right picture of Figure 4-8, the optimal solution changes gear nine times over a time span of five seconds, which is of course impossible. Another cost, that has been discarded, is that each gear shift cost a certain amount of fuel if the ICE is on and the clutch is disengaged. In this case, the ICE produces torque that is not forwarded through the powertrain.

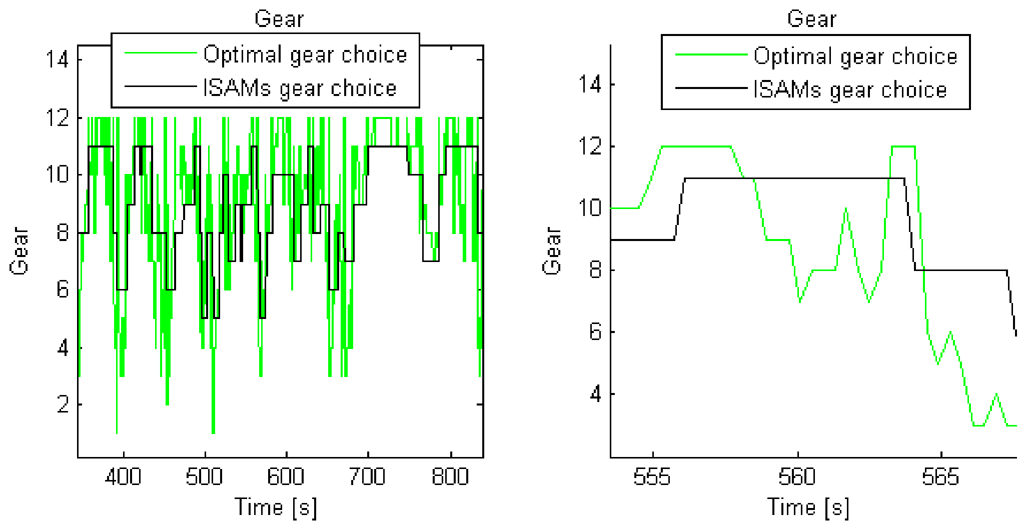

Figure 4-8
Gear shift sequence for ISAM and the optimal gear shift sequence.

The FC reduction by the optimal solution is 11.66 % without considering the difference at the final SoC value. With the SoC difference included, the reduction is 12.25 %. The energy flow trough the battery is reduced by 26.6 %.

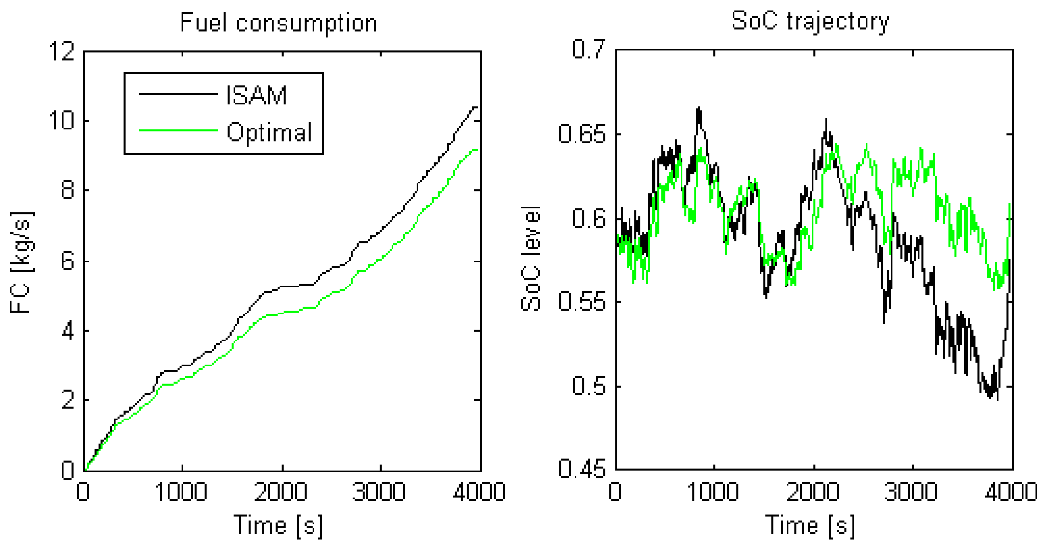

Figure 4-9
Soc and FC trajectories for both ISAM and the optimal solution.

Even though the optimal solution is impossible to achieve, it gives a clear indication that today's control system, can be improved to achieve a less expensive control strategy. By using a continuously variable transmission (CVT) and a better control system the scattered operating points of ISAM's control strategy, in Figure 4-7 , could be brought together to behave similarly to the optimal solution.

4.2.3 Comparison between the optimization algorithms

With the results from chapter 4.2.1 and 4.2.2, this can also be seen in Table 4-1, one can draw some conclusions. The ability to move the operation points in the BSFC map is crucial to be able to construct a controller that minimizes the FC. By studying Figure 3-13 one realizes that both horizontal and vertical movement is of great importance since the contour lines in center of the BSFC map are circular.

CBR85	FC reduction	FC with SoC end difference
Torque split	2.98 %	3.47 %
Torque split with gear shift	11.66 %	12.25 %

Table 4-1. FC reduction compared to ISAM on the CBR85 drive cycle for both the optimization algorithms.

4.3 Near Optimal Controller

The NOC that has been developed controls the torque split between the EM and the ICE and consists of a FM described in chapter 3.3 with a background controller that is described in chapter 3.3.4. The controller described in chapter 3.3.4 is only used in the SM, since ISAMs original controller can be used as a background controller in ISAM.

4.4 Comparison between the optimal control and NOC

The NOC result is compared with the optimal solution and the result from ISAM's control signals run through the SM. A comparison of the fuel consumption, the SoC trajectory and T_{EM} will be reviewed. Since the goal is to come as close to the optimal solution as possible, the optimal solution will be the reference. The NOC that has been used in this comparison is NOC uses a 5 dimensional FM, which has SoC, $_{TDEM}$, N_{EM} and two preview windows of length 8 and 132 seconds as its dimensions. The MF quantization for the respective dimensions are 7, 7, 7, 10 and 10.

4.4.1 Fuel consumption comparison

The FC is reduced compared to ISAM's control system when utilizing the NOC. In Figure 4-10, the last 500 seconds of the cumulative fuel consumption trajectory for the CBR85 drive cycle can be seen. As the figure shows, the trajectory from the NOC is very similar to the one from the optimal solution.

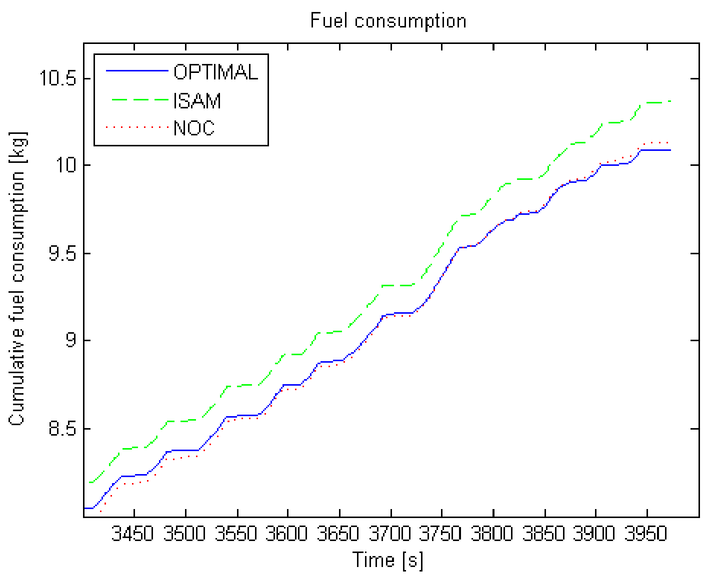

Figure 4-10
Cumulative fuel consumption for ISAM's control, the optimal solution and the NOC on CBR85.

In Figure 4-11, the background control system's FC can be seen. The background control system's FC is slightly higher then when the FM in the NOC is used. This shows that the FM actually is helping to reduce the total FC, and that preview information can be used to obtain a more fuel-economic vehicle.

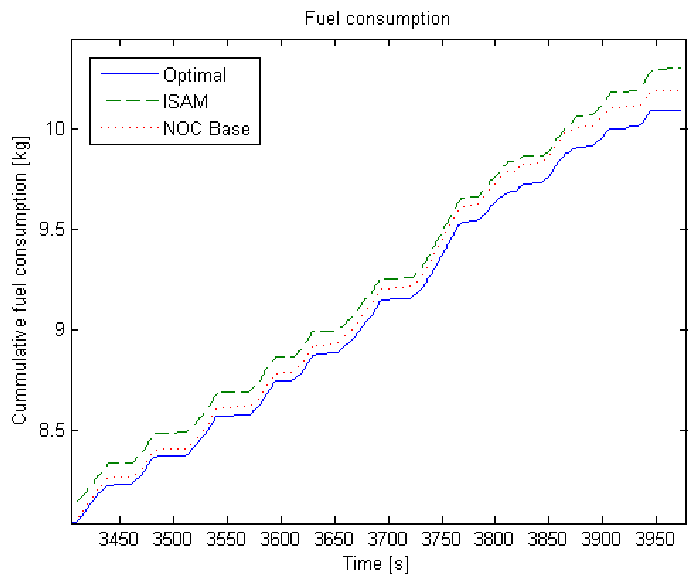

Figure 4-11
Cumulative fuel consumption for ISAM's control, the optimal solution and the background control system in the NOC without utilizing the FM in the NOC on CBR85.

In chapter 4.5, one can see the differences between different controller configurations and how it influences the fuel-consumption.

4.4.2 State of Charge comparison

Comparing the SoC trajectories is not very easy since a small change is SoC early in the simulation influences the rest of the behavior of the SoC. One can look at how the curves changes at each time instance and try to compare the similarities between the curves. One can see that the trends for both the optimal solution and the NOC solution are quite similar. The curve from ISAM control signals can not be taken too seriously, because of the differences in the battery model discussed in chapter 4.1.

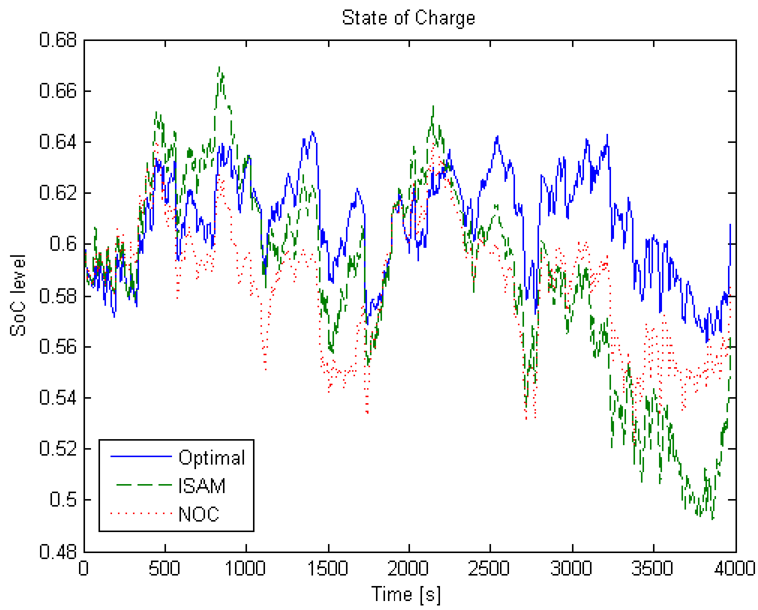

Figure 4-12
SoC trajectories for ISAM, the optimal solution and the NOC on CBR85.

4.4.3 Study of the electric machine torque differences

The T_{EM} that is produced by the NOC can be seen in Figure 4-13, the time interval from [537,550] seconds is particularly interesting. A close study of this $T_{EM, NOC}$ in comparison with the $T_{EM, BACKGROUND NOC}$ Figure 4-13 shows that the NOC follows the shape of the $T_{EM, OPTIMAL}$ more accurately than the $T_{EM, BACKGROUND NOC}$. The offset between $T_{EM, NOC}$ and $T_{EM, OPTIMAL}$ in this interval could be due to wrong dimensions in the FM, wrong scaling on the dimensions or insufficient training data.

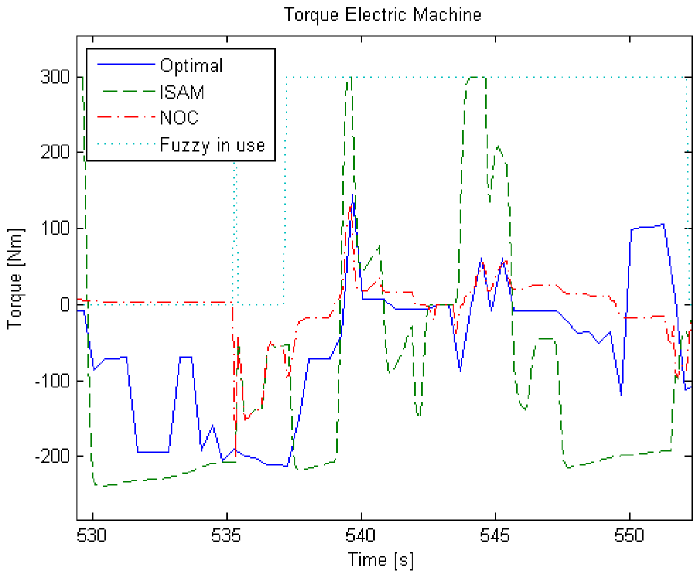

Figure 4-13
T_{EM} trajectories for ISAM's control, optimal control and NOC on CBR85. The dotted line is one when the SD of a specific work point in the FM is below the SD threshold in the NOC.

4.4.4 Utilization of the FM in the NOC

The utilization of the control values in the FM in the NOC depends heavily on the SDTL, the lower limit and the number of MF in each dimension. If a value in the interval of [100,130] is used as the SDTL, the typical use of the FM's control values lies about 40-60 % of the total control values. A plot of the FM's control values utilization for the first 500 seconds can be seen in Figure 4-14.

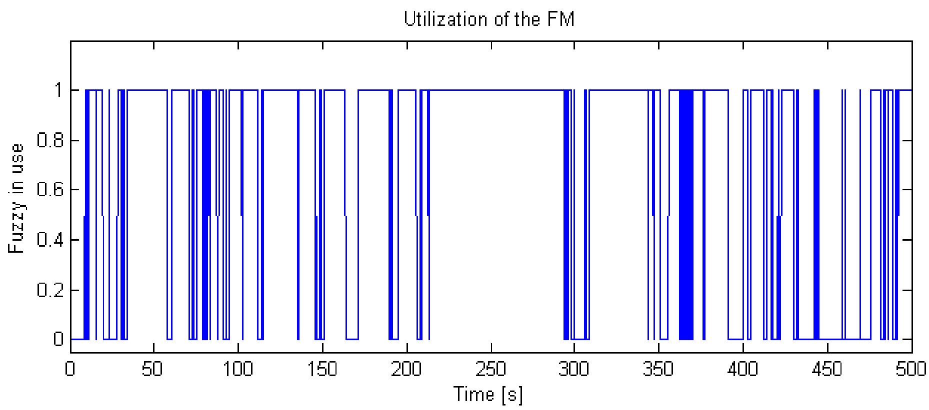

Figure 4-14
Utilization of the FM's control values as a function of time for CBR85 at SDTL 110.

4.5 Evaluation with multiple drive-cycles and NOC configurations

To get a good overview of how different controller types perform, simulations where made on two different drive-cycles, CBR85 and a 40 km long randomly generated cycle. The controllers used are described below

- Optimal uses the control signals calculated by DP. This controller is used as a reference.
- ISAM through SM uses the control signals generated by ISAM and runs them through the SM.
- NOC Background is the charge sustaining controller used in the SM when there is no valid output from the FM.
- NOC 2dim uses a 2 dimensional FM, which has SoC and T_{DEM} as its dimensions. The MF quantization for each dimension is 7 for SoC and 7 for $T_{DEM.}$
- NOC 4dim uses a 4 dimensional FM , which has SoC, T_{DEM} and two preview windows of length 8 and 132 seconds as its dimension. The MF quantization for the respective dimensions are 7, 7, 10 and 10.
- NOC 5dim uses a 5 dimensional FM, which has SoC, $_{TDEM}$, N_{EM} and two preview windows of length 8 and 132 seconds as its dimensions. The MF quantization for the respective dimensions are 7, 7, 7, 10 and 10.

All the FMs have been trained on CBR85 for three different initial SoC levels. Using the same controllers for the randomly generated drive-cycle will give an indication of how well a FM that has been trained on one cycle performs on a different one. The SD threshold was set to 110 for every NOC, and not tuned to give the best result for each controller. This was done to reduce the complexity of the test. Table 4-2shows a comparison of the following results from the simulation.

- Fuel consumption (FC)
- Fuel consumption, including difference in end SoC from start SoC (FC with ΔSoC)
- Energy flow through the battery (Energy flow)
- Usage of the fuzzy matrix in the NOC (FM usage)

Controller	**FC (%)**	**FC with ΔSoC (%)**	**Energy flow (MJ)**	**FM usage (%)**
		CBR85		
Optimal (ref)	10.0923 kg	10.0758 kg	97.29	0
ISAM through SM	-2.68	-3.59	124.29	0
NOC Background	-0.97	-0.54	97.43	0
NOC 2dim	-1.10	-0.86	102.58	66.23
NOC 4dim	-0.54	-0.99	98.86	41.14
NOC 5dim	-0.42	-0.37	98.57	60.37
		Random 40km		
Optimal (ref)	8.6704 kg	8.67	82.61	0
ISAM through SM	-2.90	-2.65	85.62	0
NOC Background	-2.38	-2.47	78.40	0

NOC 2dim	-2.70	-3.60	80.01	78.09
NOC 4dim	-3.83	-3.95	80.85	46.66
NOC 5dim	-4.42	-4.52	80.82	61.49

Table 4-2. Comparison of different drive cycles and controllers where the optimal is used as a reference.

The results from CBR85 show that the NOC improves the result compared with the ISAM control. It can also be seen that the choice of dimensions for the FM, and the number of MFs for each dimension is very important to the performance of the controller. The hit-rate is also affected by the choices made when designing the FM.

The results from the randomly generated cycle show that the FM trained on CBR85 performs really badly. Since the two drive-cycles are very different, this can be an indication of missing dimensions or that the preview information is interpreted in the wrong way. Here more work must be performed to evaluate whether a general control structure can be created that performs well on any given drive-cycle.

5 Conclusion

In this chapter, we presented the conclusions that we have drawn from this thesis work.

5.1 Dynamic Programming

DP is an easily understandable and implementable optimization algorithm. One disadvantage with DP is that its computational time is quit large. The time grows exponentially with the number of states that are included in the optimization problem. Since the optimization problem has to be discrete when using DP, the solution is not a continuously optimal solution. The more finely partitioned the states in the DP are, the closer the discrete optimal solution comes to the continuously optimal solution.

DP could also be used to change the control systems behavior. If the cost function in the DP algorithm is changed, one could for example develop a control system that minimizes the sum of the fuel and the emissions, or just the emissions. Any behavior that is preferable is attainable since the FM mimics the behavior of the optimal control.

The optimal results have also shown that if one could change gear more rapidly, and preferably with a continuously variable transmission the fuel consumption can be lowered considerably. The gear change speed becomes especially important during regenerative breaking, since during gear change, the energy has to be removed by the conventional brakes instead of being stored in the battery.

5.2 Preview Information

It can not be shown that the PI helps the controller in lowering the fuel consumption when only the torque split is controlled. When comparing the different control strategies on CBR85, see Table 4-2, one can se that most of the fuel saving is done by the background controller. This indicates that PI only can represent a small part of the total fuel savings. By including gear shifting and ICE state control, the PI probably would have a bigger impact on the FC.

An important aspect that has not been simulated is what happens if the predicted velocity in the drive cycle changes over time. The length of the preview windows would probably become shorter, since the future information gets more and more uncertain the further away from the vehicle one looks.

5.3 Fuzzy Logic and NOC

Using fuzzy logic has been an easy way to extract data from an optimal solution to be used in the controller. This is because of the straight-forward way to go from an optimal solution to a rule-based controller. When you have your optimal solution and has selected the parameterization of the controller, just train, evaluate and utilize the controller. The main problem lies instead in choosing the right states and parameter settings, which in itself is a somewhat difficult task.

5.4 Training Data

Training data from more real-world drive cycles, and preferably repetitive data for each cycle would have been useful in the evaluation and training of different controllers. If a collection of different velocity profiles on the same driving route would have been present, better evaluations regarding PI window lengths, FM size and dimension quantization etc could have been done. With a data collection like this, one could also evaluate if the optimal control has the same solution for similar situations.

5.5 Implementation in ISAM

The implementation in ISAM has been a bit of a challenge, since the NOC should influence the ISAM control system only when a better value can be presented, i.e. when the SD and hit rate in the FM fulfills the thresholds set. The solution should also be easy implementable and straight-forward to use. Most of this has been done, and works well. Unfortunately, the optimal solution calculated from the SM is not valid for ISAM and therefore, neither is the FM created from this solution.

6 Future work

There are many things in this thesis work that can be developed further. This chapter describes the main issues that have to be fulfilled. The work that has been done so far can be seen as a framework for a control system that utilizes preview information to improve fuel efficiency on a HEV. There are still many things that can be improved which could improve the results significantly.

6.1 Model work

The behavior of the battery model in ISAM and the simplified battery model differs too much to be neglected. The simplified model needs to be changed so that its behavior better resembles ISAM's battery dynamics.

6.2 Optimization work

The optimal gear shifting algorithm needs to be updated. The version that has been used in this thesis is not physically possible, and this has to be remedied. Maybe another optimization algorithm than DP has to be implemented, since DP's computational demands are high. To make a realistic gear shift algorithm, one has to evaluate each gear level position going to every possible new gear position at every time instance. With these added states, the DP computational time would probably be extended beyond convenience.

6.3 Implementation work

A gear shifting algorithm has to be incorporated in the NOC to be able to acquire results that are in the vicinity of the optimal control, with the gear shift included in the optimization. This could be done by first updating the current optimal algorithm with the gear shifting. After this, the most significant parameters for a gear level has to be extracted to be able to construct a FM that controls the gear selection.

Another parameter that has to be implemented in the NOC is when the ICE should be turned on and off. This parameter could also be controlled by a FM but a different rule-based controller is probably a better choice.

6.4 Preview Information

The PI that has been used in the project is one approach out of an infinite set of different possible approaches. First of all, the energy parameter that was chosen might not be the optimal parameter to use. Secondly, the choice of two windows might not be enough to characterize the future in the required way. Thirdly, the length of the windows and the fact that they are overlapping might also not be the best configuration. A more thorough investigation definitely would be in place here.

6.5 Parameter evaluation work

Regarding the FM that has been used in this project, a lot of questions still remain. What are the most influential states for a specific control signal i.e. what states should be used as dimensions in the FM when controlling the torque split, the gear or any other parameter? What is the optimal quantization for the respective dimensions? Is the MSD a good value to look at when evaluating the performance of a FM, or are the better variants available? Is there some way to extract the SDTL directly from a trained FM or should the SDTL be a function of the number of hits, or are there other ways to see if a FM value is acceptable or not? Since all the parameters above are closely coupled to the total result some sort of evaluation method would be preferable to be able to see if the result converges or diverges from the optimal solution depending an a change in the parameter vector.

7 References

1 European Commission, http://ec.europa.eu/enterprise/automotive/index.htm, 2007-06-04

2 Svenska Petroleum Institutet, http://www.spi.se, 2007-06-04

3 Eric Manderstedt, Magnus Nilsson, ‘Potential fuel savings by using information enabled control in a hybrid disel-electric bus’ (2006), Master thesis work

4 Mats Alaküla, ’Hybrid Drive Systems for Vehicles’, (2006), Course literature at Lund’s University of Technology

5 Bengt Lennartsson, ‘Reglerteknikens grunder’, (2002) Studentlitteratur, ISBN: 9789144024165

6 Huntley, H. E., ‘Dimensional Analysis’, (1967) Dover, LOC 67-17978.

7 Torkel Glad, Lennart Ljung, ‘Reglerteori Flervariabla och olinjära metoder’, (2003) Studentlitteratur, ISBN: 9144030037.

8 Richard E. Bellman, Stuart E. Dreyfus, ‘Applied Dynamic Programming’, (1962) Princeton Univ. Press, ISBN: 0-691-07913-7

9 Monson H. Hayes, ‘Statistical Digital Signal Processing and Modeling’, (1996) John Wiley & Sons, ISBN: 9780471594314.