EXAMENSARBETE

Finding the optimal speed profile for an electric vehicle using a search algorithm

Jonas Medin

Civilingenjör, Teknisk fysik och elektroteknik
2018

Luleå tekniska universitet
Institutionen för system- och rymdteknik

LULEÅ
TEKNISKA
UNIVERSITET

Abstract

This master thesis presents a method to find the optimal speed profile for a dynamic system in the shape of an electric vehicle and any topography using a search algorithm. The search algorithm is capable of considering all the speed choices in a topography presented discretely, in order to find the most energy efficient one. How well the calculations made by the search algorithm represents the reality, depends on the speed and topography resolution and the vehicle energy model.

With the correct settings, up to 18.4% of energy can be saved for a given topography compared to having the lowest constant speed allowed. The speed is ranging between 85-95 km/h but the method presented is capable of having any set of speed options, even if the resolution varies from point to point on the road. How to use this method and its properties is explained in detail using text and step for step figures of how the search algorithm iterates.A comparison between allowing regenerative braking and not allowing it is shown in the results. It is clear that there is most energy saving potential where no regenerative braking is allowed.

Acknowledgements

I would like to thank my supervisors at NEVS, Mustafa Ali Arat and Eduardo Simões da Silva for their guidance and support throughout this thesis. I am grateful for their continuous belief in my work and for encouraging me to explore new ideas.

Thanks to my examiner Wolfgang Birk for keeping the goals realistic and keeping me on track and thanks to Khalid Atta for always questioning me on everything.

Also a special thanks to the Department of Computer Science, Electrical and Space Engineering at LTU for letting me borrow an office to work in, without it this level of progress would have been a lot more difficult.

Contents

List of Figures

Abbreviations

Notation	Description
EC	Electric car
ECM	Engine Control Module
MPC	Model Predictive Control
Point	A single spot in a topography described by its x and y position
Node	One of many speed options in a single point
c.g	Center of gravity

Chapter 1

Introduction

1.1 The description from NEVS

The master thesis started with a discussion with NEVS in Trollhättan. A suggestion of a master thesis read as followed:

"The main obstacle to electric vehicles available in the market today is the range. To address this issue, there is a very big effort to improve battery technology, but it is also of critical importance to create more efficient driving strategies that make the most out of the energy available. To improve the energy efficiency of the vehicle in its path from A to B it is necessary to first find the most energy efficient route and secondly to find the best way to drive the chosen route. Driver attitude can account for considerable differences in energy consumption, such as by providing an optimum speed profile that minimizes consumption and maximizes regeneration within certain boundaries it is possible to improve energy consumption and as a consequence range.
This study will investigate various path planning strategies from vehicle dynamics, road topology and vehicle efficiency perspectives, ultimately to improve the drive range of an electric vehicle.
The study will be initiated with the use of different search algorithms (e.g. A-Star, random tree, ant colony, etc.) based on which a selection of alternative routes will be given.
The next stage of the study will look into the selection of the optimal route and drive profile (desired vehicle motion) by the use of optimal control strategies such as Model Predictive Control (MPC). The optimization task will need to account for a range of constraints including path, elevation, disturbances, charging stations, state of-charge (SoC), and to rank the routes based on time, energy efficiency, point of interest, etc. Different optimization algorithms will be tested to find out the best performing one."

The project description from NEVS has been modified in the way described in Section 1.2.

1.2 Problem formulation

The description in Section 1.1 suggests that before knowing how much energy is spent on each road choice, calculations on the relevant available roads should be made. It makes more sense to integrate an algorithm estimating energy use of an option with a path finding algorithm when looking through different road options. That way as few roads as possible can be investigated while still finding the optimal solution without looking at all the options.

The problem that has been solved in this thesis is find the optimal speed profile for any topography with an accurate vehicle model as the main foundation. It is done by describing the topography in a discrete way using points and having a specific number of speed options for each point discrete, given a limited number of choices for each step.

1.3 Effects on people and the environment

When NEVS formulated the master thesis, its intention was to extend the range of their electric vehicles as much as possible. To be able to plot the destination on a GPS and then get a recommendation back of what road option is the most energy efficient one, or event if the trip is theoretically possible or not. A driver could use that information and save as much energy as possible as simple as a push of a button.

Today's GPS does not take energy saving into account and mainly focus on traveling time. Those two may not always show the same road option and thus it is important to investigate to save as much energy as possible.

1.4 Delimitations

This master thesis was originally intended for two people to work on since the scope of it is too big for one. this work uses a realistic electric energy consumption model to find the optimal path given topography and a speed resolution. However, it does not include the safety aspects such as slowing down before sharp turns or roadwork or the weather conditions in the energy spending model. The optimal path in terms of which of many roads is the most energy efficient is not found, but a suggestion of what method to use is presented in Section 5.6.

1.5 Topography

The topographies used are based on data collected when measuring the height differences on a given road. These measurements could in theory be represented without errors if a 100% accurate function is used to represent it. The way the topography is described in this work is with a two row matrix having distances in the first row and height differences in the second. The data used is assumed to be noisy and inaccurate while describing something continuous with only a few points. How it is filtered and how the gaps are filled is described more in Section 2.4.4.

There are two ways of using the given discrete topography. The first one is to translate the discrete points into piecewise functions, put those functions into a equation and then calculate a speed algebraically. That continuous speed will be seen as discrete values in the cars internal controller that tries to adapt the car to that set speed. The second option is to still be in that discrete domain the whole time, basing calculations of discrete values and having a discrete output that can be used directly. It is possible to use the same resolution as the cars internal controller to minimize as much data loss as possible in the calculations

A discussion of which one is best in general is not included in this thesis, what is important is that the second alternative is possible to use in a search algorithm. Each option can be investigated and evaluated, and after an evaluation in every iteration, decisions of how to proceed can be made.

1.6 Literature study

When studying for this master thesis, academic work in several different areas was studied in order to combine them in trying to come up with something new. The beginning of the research started out with four main areas to study and they were: vehicle model, search algorithms, controllers and previous works solving similar optimization problems. The main foundation of getting the algorithm to work in reality is getting an accurate vehicle model to get feasible results. Also, to not reinvent what already has been published and for inspiration previous works needs to be studied. The other two is simply tools used to solve the problem of optimization, where one, both or none can be used depending on approach.

1.6.1 Vehicle model

The vehicle model is based on the book [3], chapter 4 and measurements on the drivetrain conducted by NEVS. These two sources make the foundation when calculating energy usage depending on topography and speeds. More in depth explanation of the theory can be found in Section 2.1.

1.6.2 Previous work

There are several scientific papers solving the problem of saving energy through adapting the speed to the topography. Most of them used a MPC controller as the main method of solving the problem. Two typical examples of that is the thesis by Erik Hellström [4] and by Mathias Mattson and Rasmus Mehler [8]. A more trial an error approach to the problem can also be found, by Jaime Junell and Kagan Tumer [5].

An earlier approach to solve the optimizing problem by keeping the data discrete and calculate it with a search algorithm was not found.

1.6.3 Search algorithms

Going back in time, Dijkstra's algorithm was the fastest one when it came to non-negative numbers when A* was introduced in 1968 by Hart, P., Nilsson, N., & Raphael, B found in [1]. Under very specific conditions (if the heuristic function is admissible) it is guaranteed to find the fastest route between two points.

When trying to gather information about these two algorithms, the examples found in the library and online normally used it for finding the shortest path between two points on a map. In order to use them for finding the most energy efficient speed-profile, some mathematical assumptions had to be made to fit the properties of the algorithms.

The end result was a combination of Breadth-first search and Dijkstra's algorithm. The algorithm uses Breadth-first searches way of deciding the next node and Dijkstra's algorithm for evaluating what nodes can be kept and what options should be thrown away.

Chapter 2

Theory

In this chapter the background information that supports the later presented algorithms is presented. The Vehicle model in Section 2.1 is used as the basis for all the other calculations and estimations. Continuing in Section 2.3, the energy spending between the two points is explained. Next is proofing the search algorithm used, finishing of with how the topography is filtered shown in Section 2.4.4.

2.1 Vehicle model

The vehicle models purpose is to represent the reality well enough for the the rest of the work to be based upon it. Even though the algorithm can handle all vehicle models as long as one condition is fulfilled which is explained in Section 2.4.2.

2.1.1 Vehicle dynamics

Figure 2.1 from [3] on page 107 shows the basic setup of how the rolling resistance acts as a longitudinal friction on the vehicle. When doing a force balance along the longitudinal axis, it becomes

$$F_{zf}(l_f + l_r) + F_{aero}h_{aero} + m\ddot{x}h + mgh\sin(\theta) - mgl_r\cos(\theta) = 0. \tag{2.1}$$

A graphical representation of how the different components of the equation above affect the system can be seen in Figure 2.1.

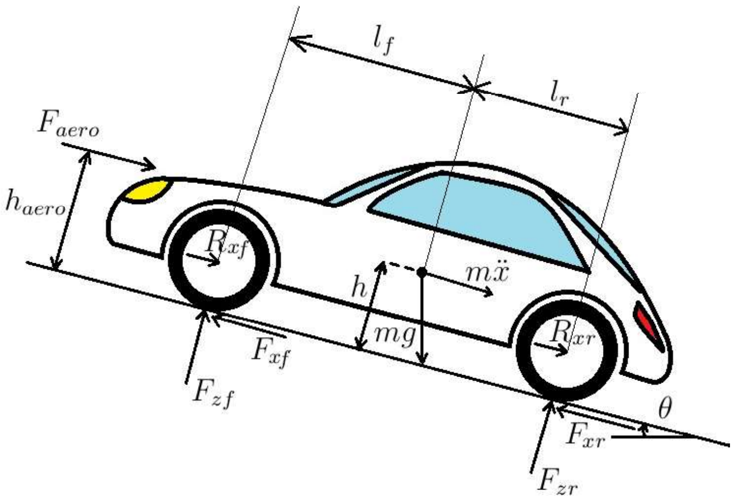

Figure 2.1: Longitude forces on the car.

Table 2.1 explains the different variables used in Equation 2.1.

Table 2.1: Components of Equation 2.1.

F_{zf}	is the rolling resistance
l_f	is the longitudinal distance of the front axle from the c.g. of the vehicle
l_r	is the longitudinal distance of the rear axle from the c.g. of the vehicle
F_{aero}	is the equivalent longitudinal aerodynamic drag force
h_{aero}	is the height of the location at which the equivalent aerodynamic force acts
m	is the mass of the vehicle
$\ddot{x}$	is the acceleration of the vehicle
h	is the height of the c.g of the vehicle
θ	it the angle of inclination of the road on which the vehicle is traveling

Of the components in the Equation 2.1, $F_{zf}(V_x)$ and $F_{aero}(V_x)$ is different disturbances depending on speed and $mg\sin(\theta)$ and $mgl_r\cos(\theta)$ on the angle of the hill.

2.1.2 Wheel friction

The longitudinal friction forces F_{xf} and F_{xr} act on the tires and are described in [3] on page 391-392 as

$$F_{xf} = C_{\sigma f}\sigma_{xf} \tag{2.2}$$

and

$$F_{xr} = C_{\sigma f}\sigma_{xr}. \tag{2.3}$$

Where F_{xf} is the friction force from the frontal tires and F_{xr} on the rear. $C_{\sigma f}$ and $C_{\sigma f}$ (the front and rear respectively) are described by Equation 2.4 and 2.5

$$\sigma = \frac{r_{eff}\omega_w - V_x}{V_x} \tag{2.4}$$

during braking and

$$\sigma = \frac{r_{eff}\omega_w - V_x}{r_{eff}\omega_w} \tag{2.5}$$

during acceleration. The friction of the wheels has to do with the fact that they are subjected to a deformation when rolling and doesn't return to its original form entirely. The deformation is proportional to the amount of weight put on each pair. Its equation becomes

$$F_{zf} = \frac{-F_{aero}h_{aero} - m\ddot{x}h - mgh\sin(\theta) + mgl_r\cos(\theta)}{l_f + l_r} \tag{2.6}$$

and

$$F_{zr} = \frac{F_{aero}h_{aero} + m\ddot{x}h + mgh\sin(\theta) + mgl_r\cos(\theta)}{l_f + l_r}. \tag{2.7}$$

Equation 2.6 and 2.7 can be used when optimizing and fine-tuning the control, although not used in this thesis. The wheel variables are defined in Table 2.2.

Table 2.2: Components affecting wheel friction.

$V_x(=\dot{x})$	is the longitudinal velocity of the vehicle
$C_{\sigma f}$	is the front longitudinal tire stiffness
$C_{\sigma r}$	is the rear longitudinal tire stiffness
r_{eff}	is the effective radius on a rotating tire
ω_w	is the rotation speed of the wheel

2.1.3 Aerodynamic drag

The drag force F_{aero} is described on page 97 in [3] as.

$$F_{aero} = \frac{1}{2}\rho C_d A_F (V_x + V_{wind})^2. \tag{2.8}$$

How to derive the aerodynamic drag coefficient C_d and the rolling resistances R_x is explained in [3] on pages 97−99. From [10] the parameters are defined as

$$C_d = \frac{2m\beta\tan^{-1}(\beta)}{V_0 T \rho A_F} \tag{2.9}$$

where β used for both C_d and R_x is defined as

$$\beta = V_0\sqrt{\frac{\rho A_F C_d}{2R_x}} \tag{2.10}$$

and the rolling resistance R_x as

$$R_x = \frac{V_0 m \tan^{-1}(\beta)}{\beta T}. \tag{2.11}$$

The range of 79-84% of the frontal area denoted A_F is calculated using the height and the width of passenger cars according to [9]. The reference also states that the relation in weight for cars weighing between 800-2000 kg in and its area can be calculated using Equation 2.12

$$A_F = 1.6 + 0.00056(m - 765). \tag{2.12}$$

The aerodynamic variables are defined in Table 2.3.

Table 2.3: Components affecting aerodynamic drag.

ρ	is the mass density of air
C_d	is the aerodynamic drag coefficient
A_F	is the frontal area of the vehicle
$V_x = \dot{x}$	is the longitudinal velocity of the vehicle
V_{wind}	is the wind velocity

All values of all the constants used in this chapter is described in Table 2.4.

Table 2.4: The chosen values in Section 2.1.

Variable	Value	Comment
m	2000.0 kg	Nice and even value
ρ	1.2526 kg/m^3	A value taken from Luleå in mid November
C_d 0.3	Same constant as a Saab 92 according to Wikipedia	
A_F	2.2916	calculated using Eq. 2.12

2.2 Explaining nodes

The term *nodes* comes from dealing with search algorithms and is often a point on a map and in this thesis a specific speed in a point on a map. Each node contains a lot of information used in the search algorithms decision making process. Very important remembering is that every point on the topography has many different nodes to choose from, not only a single one. An example of a node used in the Matlab code can be found in Table 2.5.

The nodes are a mix of information for calculating the optimal path, information making debugging errors easier and values without any purpose that can be adapted to a new feature without changes to the rest of the code. Having non-reserved values might slow down the process a bit but is very useful when in the development stage and when new ideas are tested.

Table 2.5: Explanation of what information is used in each node.

$1.7020e+03$	What number the node has, 1702 in this case
$2.6389e+01$	Speed in this particular node
$3.4049e+04$	Value in meter on the x-axis
$7.6857e+01$	Height in meter on the y-axis
$1.2876e+07$	Total energy use so far
$-1.1414e+06$	Energy comparison value (used in the A-star Algorithm)
$5.0521e+04$	Engine energy output to get between the two nodes
$1.0000e+00$	level (used in the A-star Algorithm)
$1.7010e+03$	Parent number, 1701 in this case
$2.5556e+01$	Parent speed used as an initial speed
$3.4029e+04$	Parent value in meter on the x-axis
$7.6976e+01$	Parent height in meter on the y-axis
$1.2826e+07$	Parent Total energy use
$-1.1861e+06$	Parent Energy comparison value (used in the A-star Algorithm)
$5.0057e+04$	Parent engine energy output to get between the two nodes
$1.0000e+00$	level (used in the A-star Algorithm)

2.3 From one point to another

To better understand what happens when making calculations using points and their corresponding energies, a derivation of the formulas used is needed. As is true with handling forces, the different energies are added upon each other and the resulting energy is simply equivalent to what the engine has to produce. The energies added are the potential energy from height difference, change in kinetic energy, air resistance, wheel friction and drive-train energy losses. The very simple case of two points with a height difference denoted h and different speeds v_1 and v_2 is shown in Figure 2.2 as

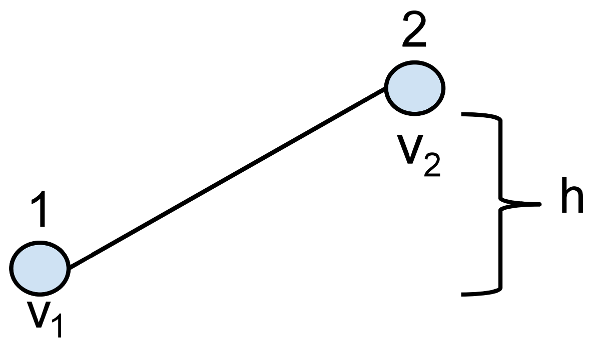

Figure 2.2: Two points in a topography given a constant distance between "1" and "2", and varying the height.

$$E_p = mgh \tag{2.13}$$

for the potential energy,

$$E_{k1} = \frac{mv_1^2}{2} \tag{2.14}$$

and

$$E_{k2} = \frac{mv_2^2}{2} \tag{2.15}$$

for the two kinetic energies the vehicle in point 1 and point 2. The difference in kinetic energy between the points is

$$E_k\Delta = E_{k1} - E_{k2} = \frac{m(v_1^2 - v_2^2)}{2}. \tag{2.16}$$

The energy consumed by the air resistance can be defined by the work made by that force in Equation 2.8 is

$$E_{aero} = d_{real} \cdot F_{aero} \tag{2.17}$$

where d_{real} (in meter) is the real distance between nodes using Pythagorean theorem. V_x in Equation 2.8 becomes $(v_1 + v_2)/2$ which represents an average between the two nodes shown in Figure 2.2. Weather factors such as the wind speed V_{wind} found in Equation 2.8 is assumed to be zero.

The wheel friction is kept constant at about 5-10% (put at 7.5% in this case) of what the aerodynamic energy consumes at about 90 km/h and simply is defined as

$$E_w = 0.075 \cdot E_{aero} \tag{2.18}$$

at that specific speed. The most complex energy loss is from the drive-train model called E_{DT} where a given lookup table is used to get the energy loss depending on speed and torque. The total amount of energy the engine (E_{eng}) has produce is

$$E_{eng} = E_p + E_k + E_{aero} + E_w + E_{DT}. \tag{2.19}$$

Replacing the energy variables, an expression of how what speed the energy produces can be derived. Equation 2.19 becomes

$$\begin{aligned} E_{eng} =& mgh + \frac{m(v_1^2 - v_2^2)}{2} + d_{real} \cdot \frac{1}{2}\rho C_d A_F \left(\frac{v_1 + v_2}{2}\right)^2 + \\ & 0.075 \cdot d_{real} \cdot \frac{1}{2}\rho C_d A_F \left(\frac{v_1 + v_2}{2}\right)^2 + E_{DT} \end{aligned} \tag{2.20}$$

rewriting E_{aero} and E_w together to

$$E_{eng} = mgh + \frac{m(v_1^2 - v_2^2)}{2} + 1.075 \cdot d_{real} \cdot \frac{1}{2}\rho C_d A_F \left(\frac{v_1 + v_2}{2}\right)^2 + E_{DT}, \tag{2.21}$$

gathering all the factors affecting torque in E_c

$$E_c = mgh + \frac{m(v_1^2 - v_2^2)}{2} + 1.075 \cdot d_{real} \cdot \frac{1}{2}\rho C_d A_F \left(\frac{v_1 + v_2}{2}\right)^2 \tag{2.22}$$

and turning Equation 2.21 to

$$E_{eng} = E_c + E_{DT}. \tag{2.23}$$

Since the energy loss from the drivetrain does not affect how much torque the engine has produce E_c can therefore be used directly as shown in Equation 2.24 as

$$\tau = \frac{E_c}{d_{real}} \cdot \frac{r_{wheel}}{GR}, \tag{2.24}$$

where τ together with the speed of the vehicle is put into the lookup table and used to calculate E_{DT} which is an energy strictly bigger than zero.
The energy loss of the drivetrain in the lookup table is described in Watt and therefore has to be transformed to energy in order to fit the rest of the code. This is done dividing it by the time aspect, or in terms known, expressed as

$$\frac{d_{real}}{V_x}. \tag{2.25}$$

2.4 Problem formulation - Search algorithm

Given an initial speed of 85 km/h on a 90 km/h road, iterating 4 more points ahead with 6 different speed options ranging from 85-95 km/h gives 1296 different speed options, where iterating through 5 points increases that number to 7776 different speed options. The formula for the number of speed options is given by

$$(SpeedResolution - 1)^{TopoPoints} = SpeedOptions, \tag{2.26}$$

if the starting speed is given. To visualise the problem, all the speed options are given in Figure 2.3:

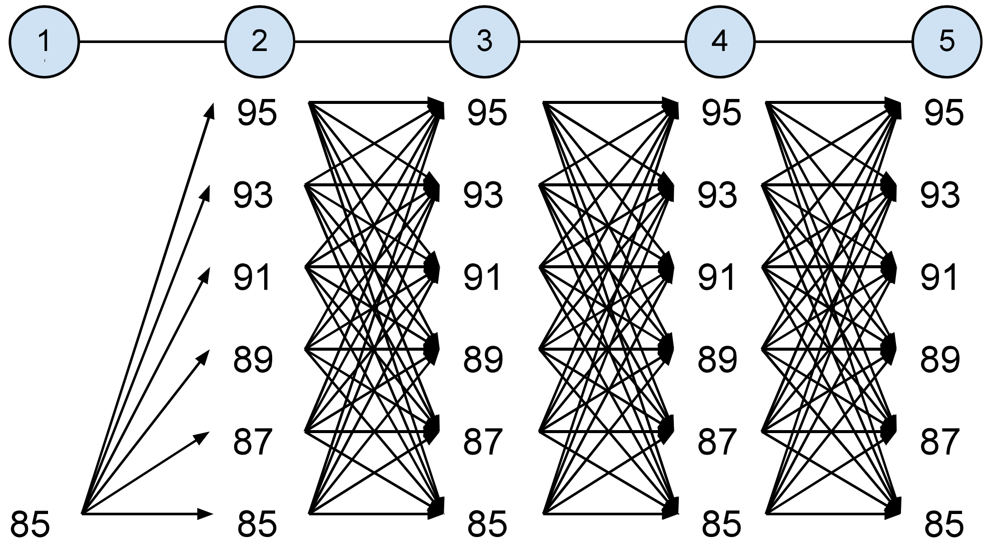

Figure 2.3: All the different speed profiles available for a simple flat topography, making up 7776 different options.

It is clear that some kind of algorithm narrowing the search down to a more realistic number might be helpful. This should be done while still finding the optimal speed profile but using much less computing power.

2.4.1 The algorithm

Dijkstra's algorithm [6] works with the concept of finding the next node to investigate based on the least amount of energy usage so far (or distance in path-finding applications). When it has investigated all its options from that node, it closes it for further investigation since its information has been extracted. The total energy use to get to that node can be updated but that doesn't cause the node to be re-opened and investigated yet again. Each node is allowed to only keep the best path to that node in memory and thus throw away all the other investigated alternatives.

Dijkstra is a so called "greedy algorithm" which means that it only searches what is locally optimal. To not explore all options sets demands on the mathematical properties to guarantee the optimal solution. The main property is that the algorithm cannot be said to find the optimal solution when negative node are involved. On the other hand it is optimal for problem with the right mathematical properties and a lot faster than looking through all the possible routes to find the optimal solution.

The order of which the nodes are investigated in this thesis is the same used in Breadth-first search algorithm [11]. Combined with Dijkstra algorithm to open and close nodes, a better adapted solution can be formulated which make the same choices but has fewer mathematical requirements. Instead of selecting the next choice by energy consumption, all the possible options can be investigated from all the speed alternatives. As in the case of Dijkstra, each node can be closed after an investigation and each node are allowed to keep only the option with the least amount of energy. For example if coming from the speeds 85, 90 and 95 km/h is investigated and it is going to the speeds 85, 90 and 95 km/h, 9 different iterations will be made with 3 investigations from each node. The following nodes will have 3 node but only 1 per node will be kept, a clarifying example is shown in Section A.1. The burden of proof does not have to do with in the order the search itself is conducted. The interesting question is if a search to a node has been conducted and there are several alternatives, can one be kept and the other ones thrown away?

Starting at the case shown in Figure 2.3 and looking at point 2 and 3, what has to been reduces is shown more clearly in Figure 2.4 as:

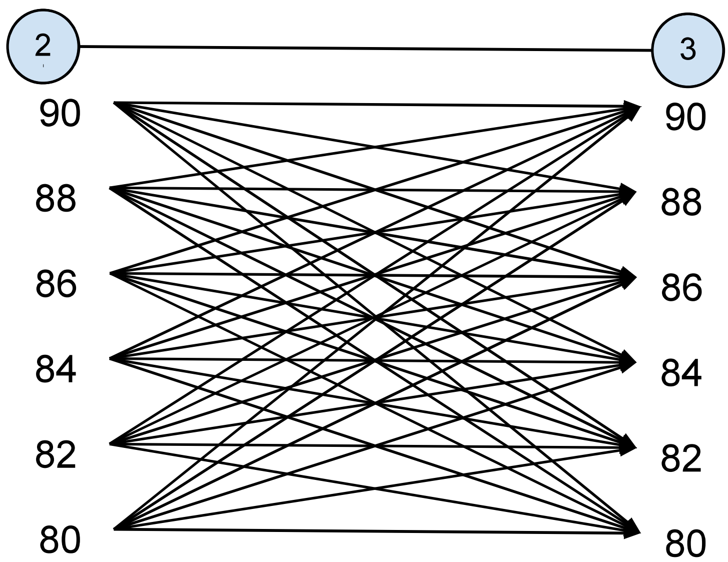

Figure 2.4: the points in the topography seen between point 2 and 3 in Figure 2.3, showing 6 options for every speed in point 3.

2.4.2 Proof

The answer to the question if all but one parent node can be thrown away is that if the vehicle itself is somehow different coming from 95 to 85 km/h compared to 85 to 85 km/h (other than energy consumption), then the answer is "no". If they are exactly the same the answer is "yes". Nothing the equations describing the vehicle model shown in Equation 2.19 and in greater detail in Equation 2.20 suggests that any kind of rate of change (acceleration for example) is important. Meaning that if something started to accelerate to get to a certain speed instead of having a constant speed will not affect the rest of the iterations other than having another total energy consumption.

Making those choices with the case shown in Figure 2.4, it can end up looking as shown in Figure 2.5 as:

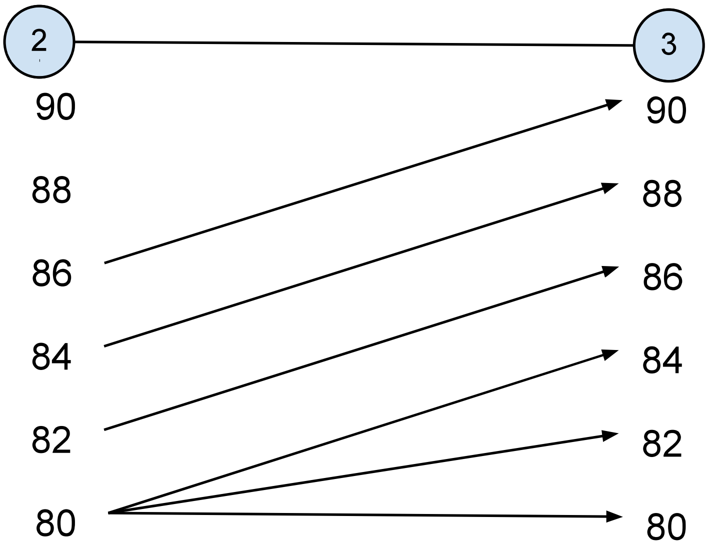

Figure 2.5: The speed options left from Figure 2.4 after only keeping the best alternatives for different options to point 3.

More about an in depth explanation about how the algorithm functions in different situations can be found in Section A.1.

2.4.3 Topography

The topography is represented in a discrete way where it has a fix number of points and each point has a position described by a certain height in the vertical-axis and a certain length in the horizontal-axis. The distances between the points in the given topography are assumed to be all different and to have noise in it that needs to be filtered. This can vary from case to case, but worst case is expected.

2.4.4 Designing the filter

The algorithms to the filter is divided in three different iterations. Starting with the *Matlab* function $sgolayfilt$ which is making extreme points less extreme by using a Savitzky-Golay filter. The filter takes several points into account when making its estimations, which is why it was chosen as the noise reducing filter. Next is *Matlabs* function *csapi* standing for "Cubic spline interpolation" which estimates a function between given points on a plane of which resolution is defined by the number of points in between the gaps of the points. The cubic spline interpolation sometimes gives very curvy results that simply should not be realistic and that is where $interp1$ is useful. $interp1$ simply draws a line between the points and add points onto that line. The advantage of using both algorithms is that they both go through all the initial points and can therefore be weighted until it best represents the reality.

The order in which the different functions are used is first $sgolayfilt$ to filter away the most extreme values and then a combination of the other two by putting different weights on them to match the reality as well as possible. Advantage of these functions is that an infinite amount of points can be added between each data points, this allows a lot of flexibility when chosing a fix distance between the points.

The two graphs put together is shown in Figure 2.6 where "$data3$" are the given topography points, "$data2$" the spline functions ($csapi$ and $interp1$), and "$data1$" represents the filtered given points. The data used has a fix x-axis distance of 20 meter and has a total of 1795 points, compared with the given 920 points.

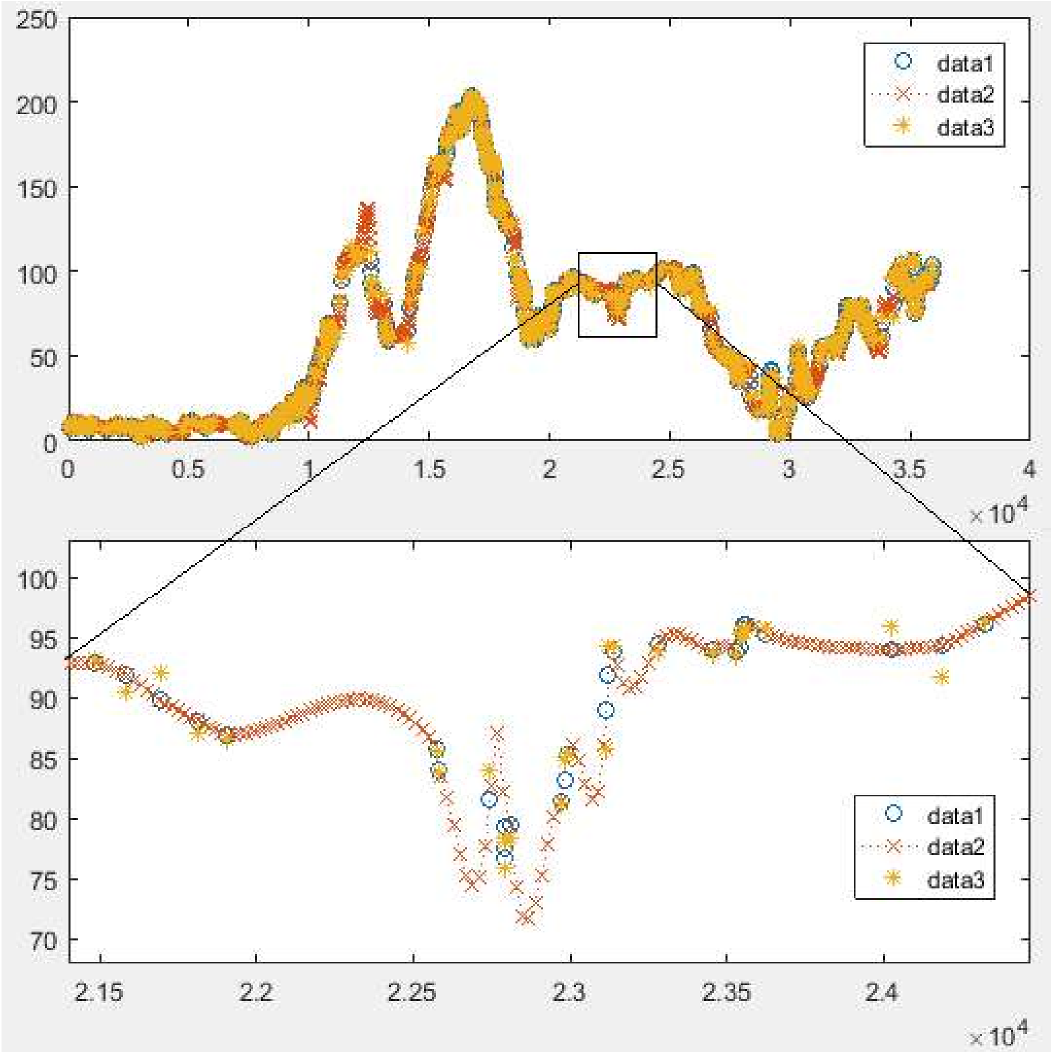

Figure 2.6: Showing the processing of the topography given from NEVS to make it more realistic and thus closer to the reality.

Chapter 3

Results

The simulations showing the different results are all optimal given the conditions. That is to say that for those options given for a topography, it is proven in Section 2.4.2 that there is no other speed profile that will save more energy.

This is not to confuse with a continuous system where a topography described by a function or several connected functions and the speed profile is algebraically derived, which would be the *true* optimal.

The advantage with a discrete system is that the proof is easily understood and it does not have to be converted to a continuous system and back to a discrete one since the original data can be used directly. It is not to say that the end result is better than a continuous system, which is outside the scope of this thesis. What is important to note is the pros and cons about different resolutions of the speed and the topography which Table 3.9 goes into detail in explaining.

To get a more understanding of what effect generative braking has, its result has been presented and explained for every topography it is used.

3.1 Computer used

The computer used to run these simulations presented in this chapter are an *acer aspire 5750G* which is a laptop that has gotten a slight upgrade in its RAM memory and changed from a hard drive to an SSD. The important specifications can be seen in the list below:

- Intel Core i7-2630QM 2GHz
- NVIDIA GeForce GT 540M
- 8 GB RAM
- 480 GB SSD

3.1.1 Flat surface

The most simple case of an algorithm is a flat surface shown in Figure 3.1 where no anomalies are expected. Looking at Equation 2.19, $E_p = 0$ and E_w is constant leaving the task to minimize

$$E_{eng} = E_{aero} + E_{DT} + E_k. \tag{3.1}$$

The aerodynamic drag E_{aero} is increased exponentially by increased and the drivetrain energy loss is increased by increased speed. An increase in kinetic energy E_k is a way to store energy if needed later on. As low speed as possible throughout the entire flat surface is expected because it gives the lowest values of E_{aero} and E_{DT}. The topography used has a distance between points of 10 m and a speed resolution of 0.025 km/h per point.

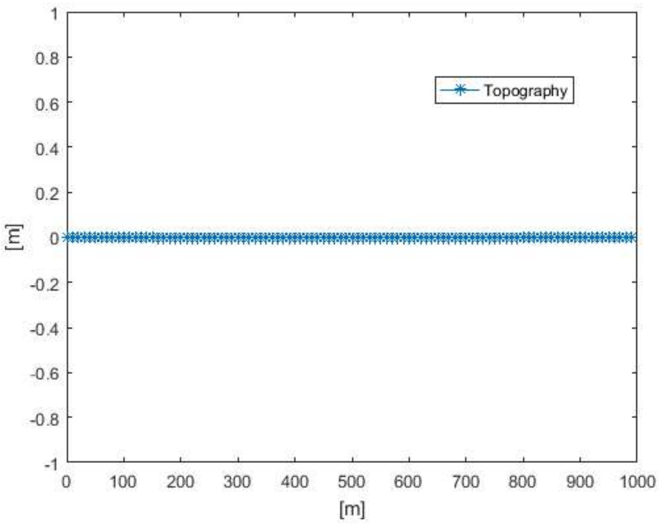

Figure 3.1: A simple flat topography, 1000 m long.

The average speed can be seen in Figure 3.2 as:

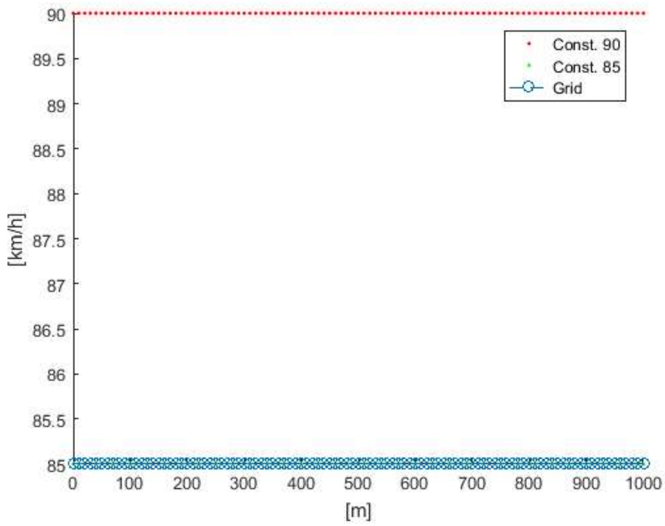

Figure 3.2: The resulting speed profile from the topography shown in Figure 3.1.

and the total energy use in Figure 3.3 as:

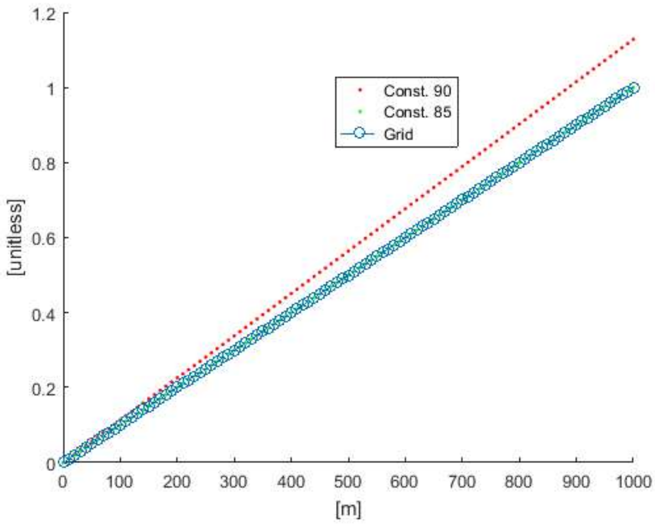

Figure 3.3: The normalized total energy use per meter from the topography shown in Figure 3.1.

3.2 Flat, uphill and flat

Is it possible to save energy by increasing the speed before the hill? Looking at the academic works mentioned in Section 1.6.2, all of them come to the conclusion "yes", but that is using a internal combustion engine. When investigating the contents of Equation 2.19 in Equation 2.20, the only factor that might punishing a sudden change in torque (τ) is E_{DT}. This is important because *if* there are no energy losses in the drivetrain by increasing the torque intensively at a hill by having a constant speed, a constant low speed is per definition always the most energy efficient speed. The topography used can be seen in Figure 3.4 as

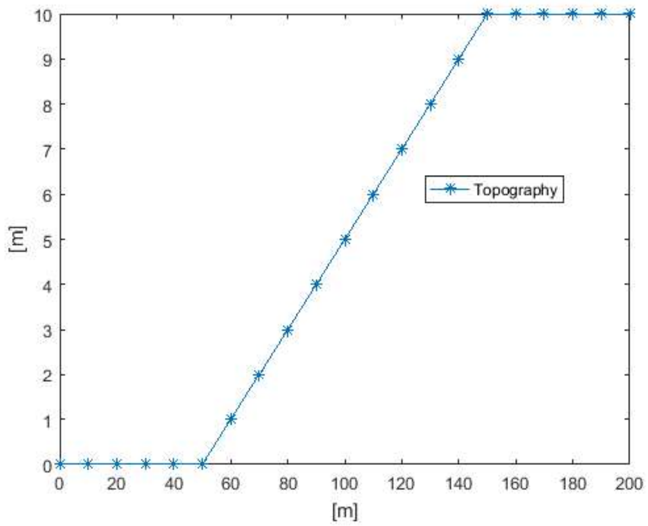

Figure 3.4: Flat, slope up and flat topography with a hill distance of 100 m placed in the center of a 200 m long road.

With a resolution as high as 0.01 km/h between 85 and 95 km/h using 21 points to describe the topography, the simulation took 70 s and resulted in a speed profile shown in Figure 3.5 as:

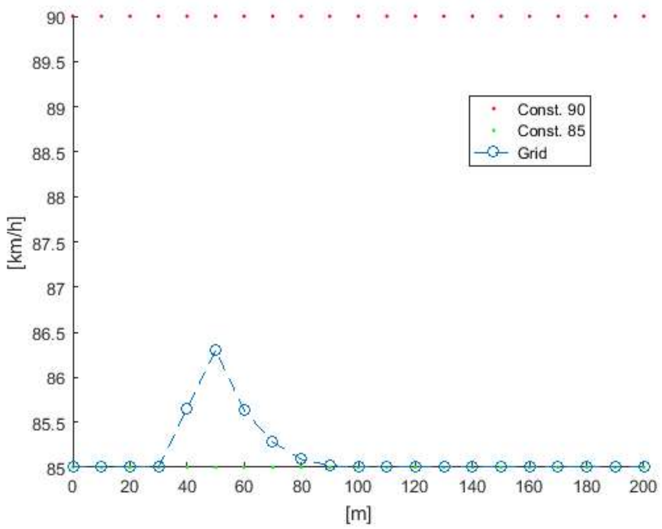

Figure 3.5: The resulting speed profile from the topography shown in Figure 3.4, with a resolution of 0.01 km/h.

The same topography but with a resolution of 0.25 km/h resulted in Figure

3.6

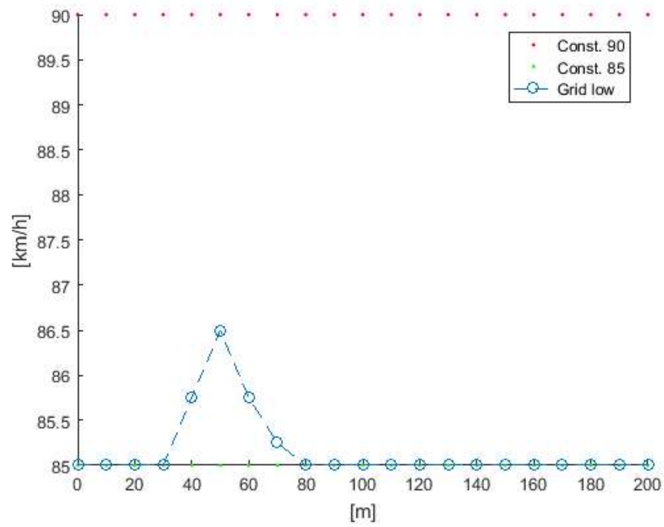

Figure 3.6: The resulting speed profile from the topography shown in Figure 3.4, with a resolution of 0.25 km/h.

The resulting total energy spending from 0.01 km/h resolution, is shown in Figure 3.7

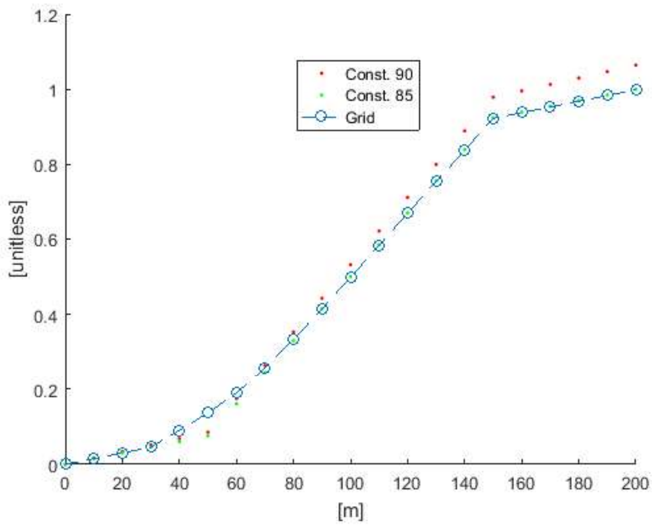

Figure 3.7: The normalized total energy use per meter from the topography shown in Figure 3.4, with a resolution of 0.01 km/h.

and the higher resolution saved 0.13% of energy compared to having a constant speed of 85 km/h while the lower resolution saved 0.11% in total. The result of having generative braking was not shown for this topography or these settings since it was not used.

3.3 Flat, downhill and flat

Preserving some of the kinetic energy built up when going downhill through increasing the speed seems intuitively like a good way of saving energy. Looking at the case when going downhill instead of uphill which was the case in Section 3.2 the topography used is:

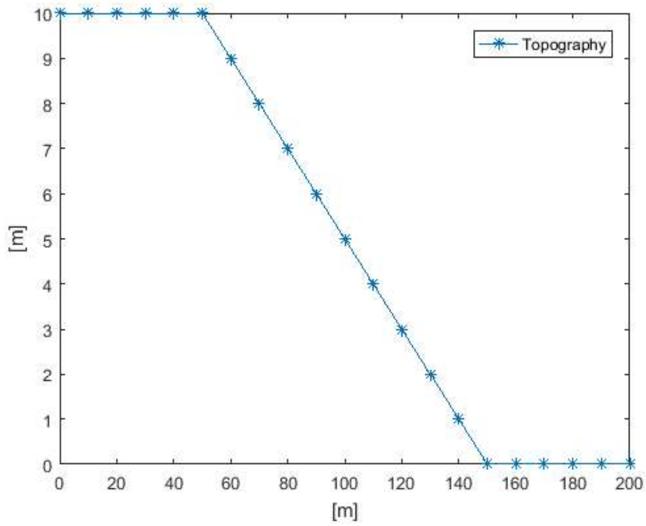

Figure 3.8: Flat, slope down and flat topography with a hill distance of 100 m placed in the center of a 200 m long road.

3.3.1 No regenerative braking

The corresponding speed profile using a resolution of 0.01 km/h is shown in Figure 3.9 as:

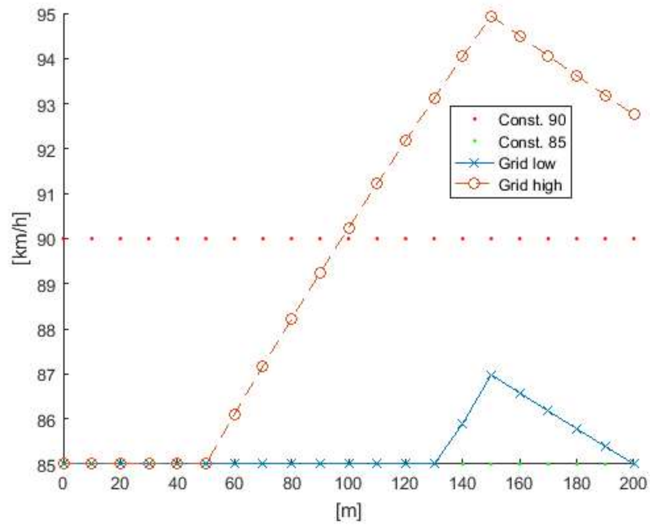

Figure 3.9: The two resulting speed profile from the topography shown in Figure 3.8, the fastest and the slowest using no regenerative braking.

Ehere the two solutions shown are the fastest and the slowest optimal solution in terms of average speed. This is common when there is a steep slope down and brakes needs to be used to maintain a speed within the set limits. The resulting total energy used is shown in Figure 3.10 as:

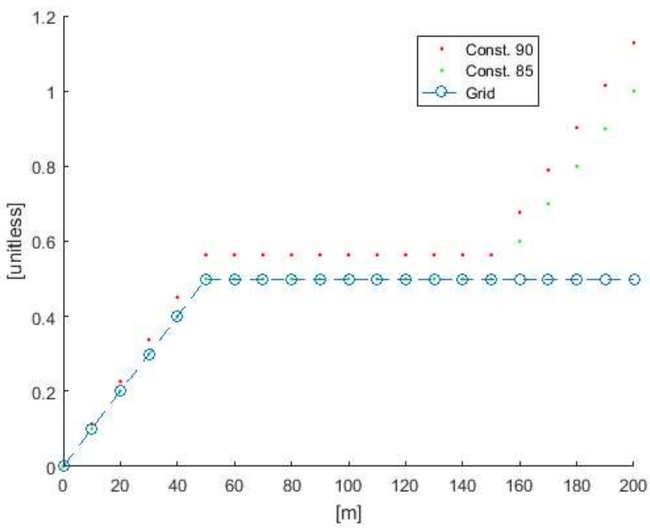

Figure 3.10: The normalized total energy use per meter from the topography shown in Figure 3.8. The fastest and the slowest has the same energy spending, using no regenerative braking.

where it is clear that the engine didn't have to do any work from the distanced $x = 50$[m] and as an experiment and thus spent 50% less than 85 km/h and 55.7% less than 90 km/h.

3.3.2 Allowing regenerative braking

Allowing generative braking made a big difference in this case where a constant speed of 85 km/h consumed 197% less energy compared with not having generative braking on. The end result went from spending energy to ending up getting energy. Using the same topography and the same resolution as in Section 3.3.1, the resulting speed profile is seen in Figure 3.11 as:

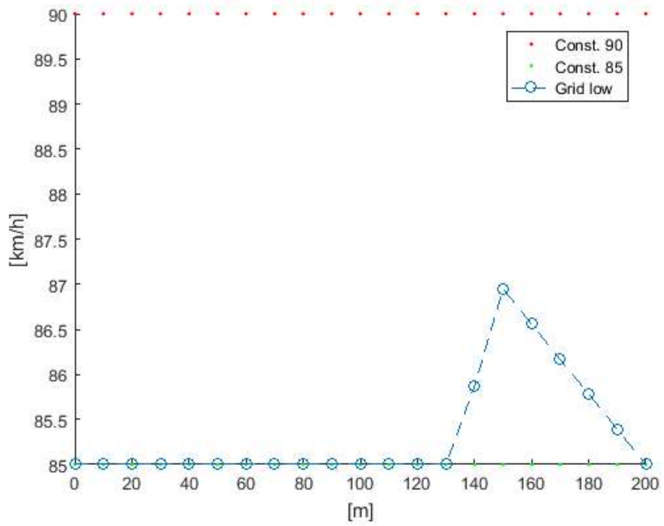

Figure 3.11: The resulting speed profile from the topography shown in Figure 3.8 with regenerative braking.

The resulting total energy is seen in Figure 3.12 as:

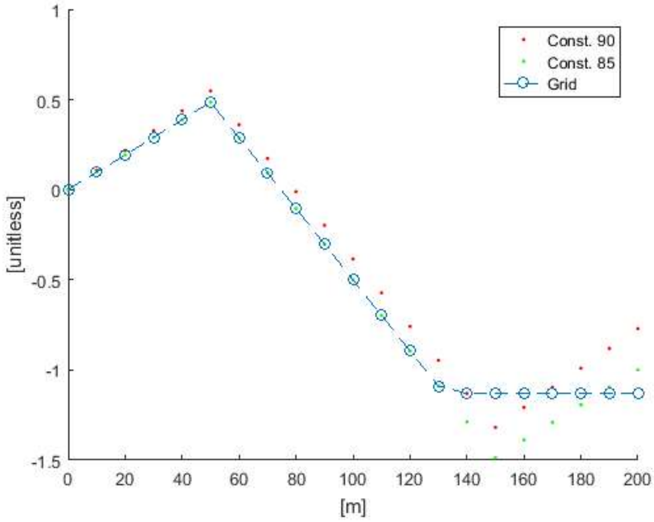

Figure 3.12: The normalized total energy use per meter from the topography shown in Figure 3.8 with regenerative braking.

3.4 Flat, downhill and long flat

The topography show in Figure 3.8 was modified to include a longer flat in the end to see how that changed the resulting speed profile. The used topography is seen in Figure 3.13:

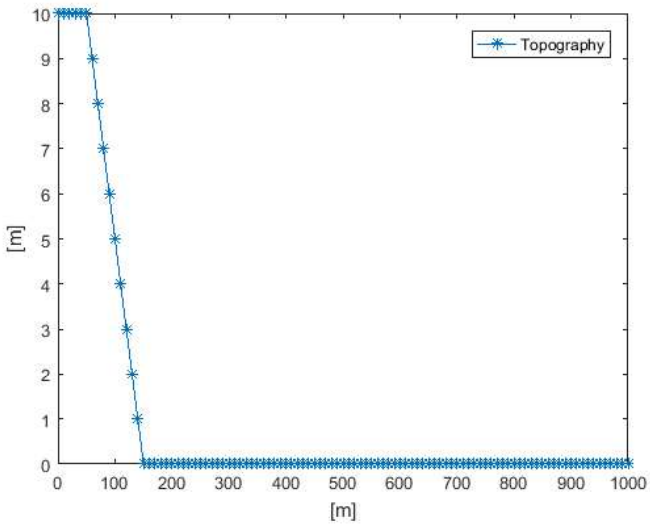

Figure 3.13: The same topography as in Figure 3.8, but with an extra 800 m added to the end of it.

3.4.1 No regenerative braking

With a 0.025 km/h resolution was used which resulted in a speed profile seen in Figure 3.14:

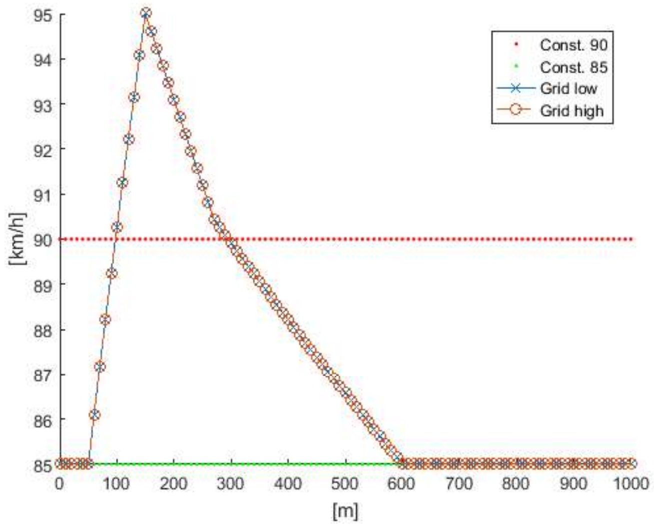

Figure 3.14: The resulting speed profile from the topography shown in Figure 3.13, with regenerative braking.

The and the resulting total energy use is seen in Figure 3.15 as

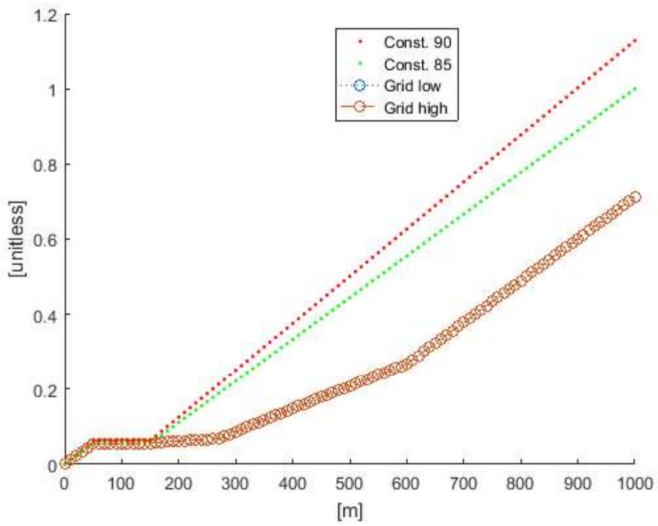

Figure 3.15: The normalized total energy use per meter from the topography shown in Figure 3.13, with regenerative braking.

An interesting perspective seen in Figure 3.16 is the energy use per node which shows the scenario in greater detail:

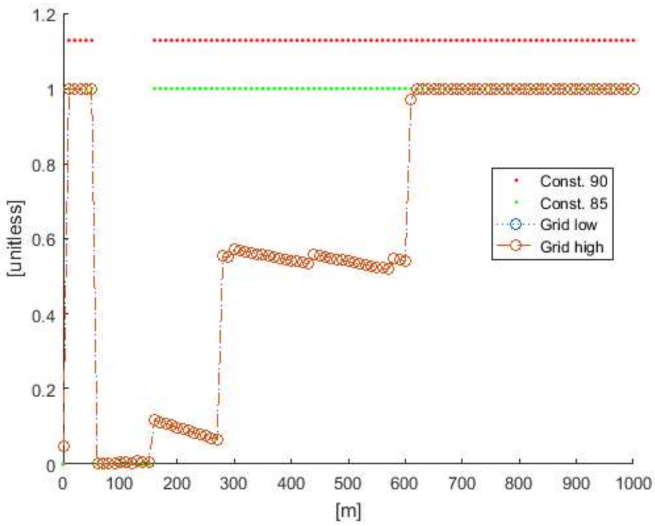

Figure 3.16: Normalized energy spending per node from the topography shown in Figure 3.13, with regenerative braking.

It is clear that the energy saving potential is greater when going downhill than going uphill.

3.4.2 Allowing regenerative braking

When allowing regenerative braking, the constant speed of 85 km/h used 29% less energy than when not allowing it. The resulting speed profile can be seen in Figure 3.17 as:

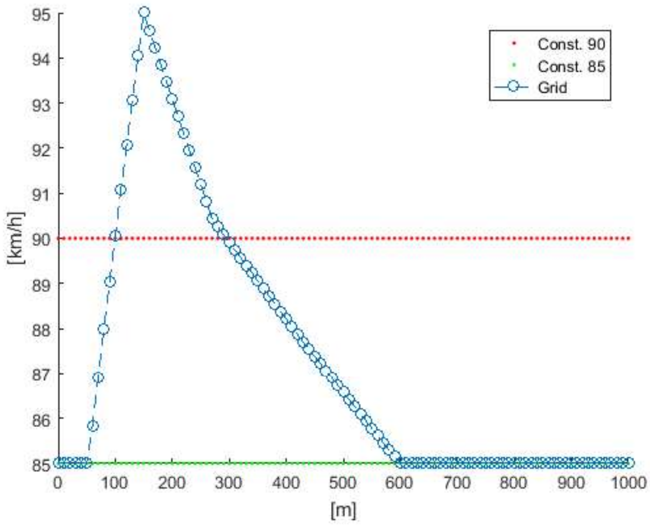

Figure 3.17: The resulting speed profile from the topography shown in Figure 3.13, with regenerative braking.

and the resulting speed profile in Figure 3.18 as:

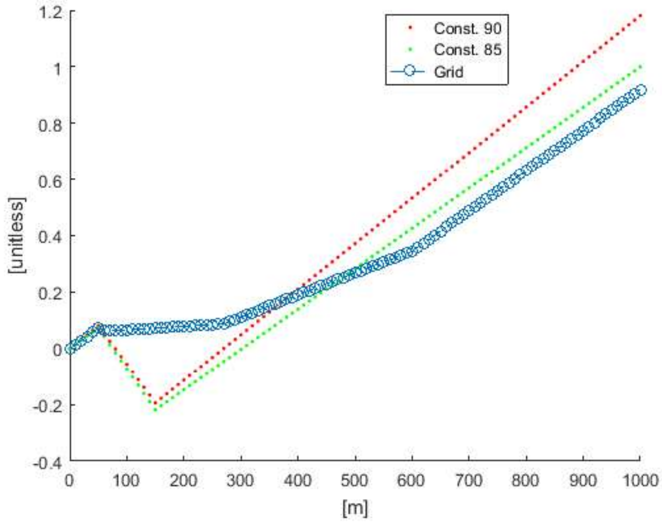

Figure 3.18: The normalized total energy use per meter from the topography shown in Figure 3.13, with regenerative braking.

and its corresponding engine energy spending per node is seen in Figure 3.19 as:

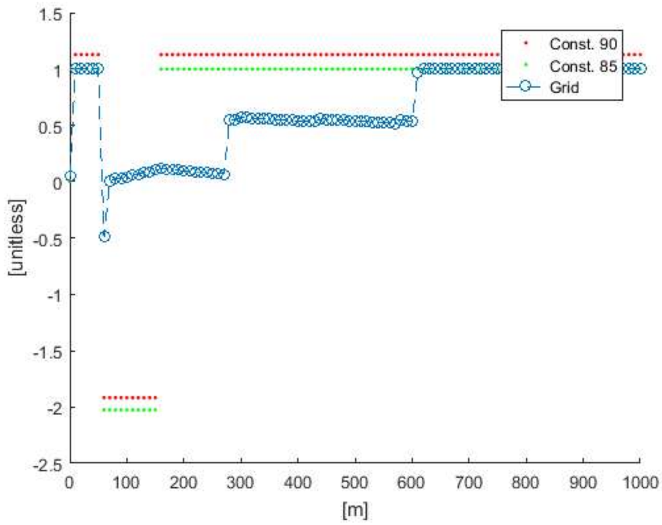

Figure 3.19: The resulting speed profile from the topography shown in Figure 3.13, with regenerative braking.

3.5 Randomized topography

Theoretically the algorithm proven in Section 2.4.2 and explained in Section A.1 should be sound, but to be even more certain a comparison with all the alternatives can be made. In Section 2.4 the problem of doing this is explained in detail but a short topography with a low resolution can still be simulated.

As expected, searching through all options yielded the same result as the algorithm used. The randomized topography used is seen in Figure 3.20 as:

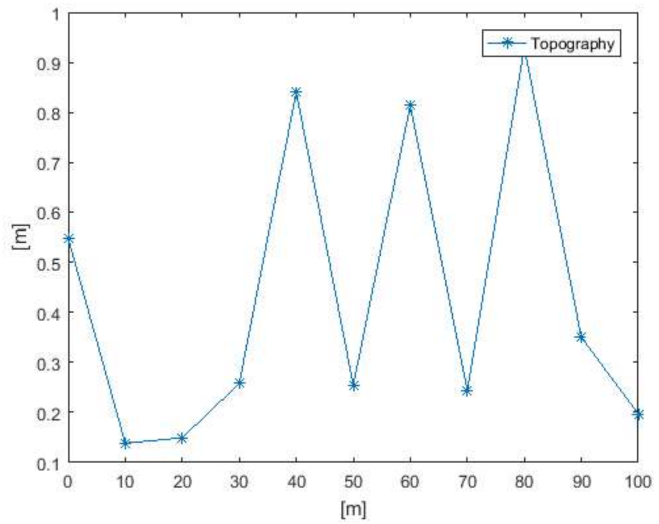

Figure 3.20: A randomized topography with an height of about 1 m and a length of 100 m.

The speed boundaries was set from 85 to 86 km/h with a resolution of 0.5 km/h. The algorithm looking at all alternatives did it in 65.7s and the Dijkstra inspired algorithm in 0.163s.

3.5.1 No regenerative braking

The speed profile found is shown in Figure 3.21 as:

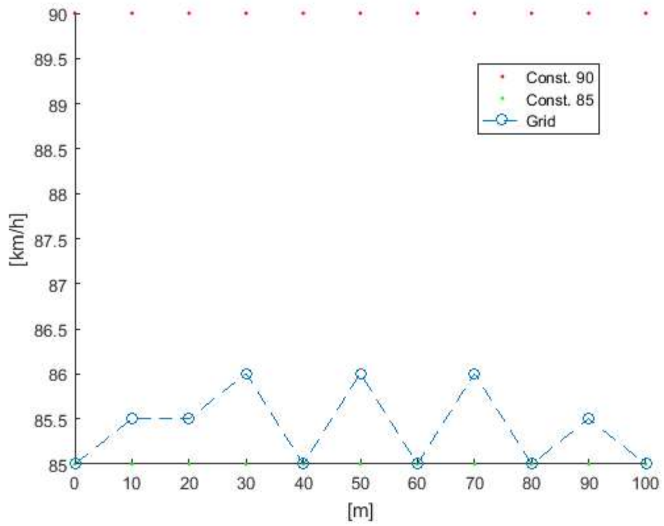

Figure 3.21: Resulting speed profile from the randomized topography shown in Figure 3.20 without regenerative braking.

The total energy usage is seen in Figure 3.22 as:

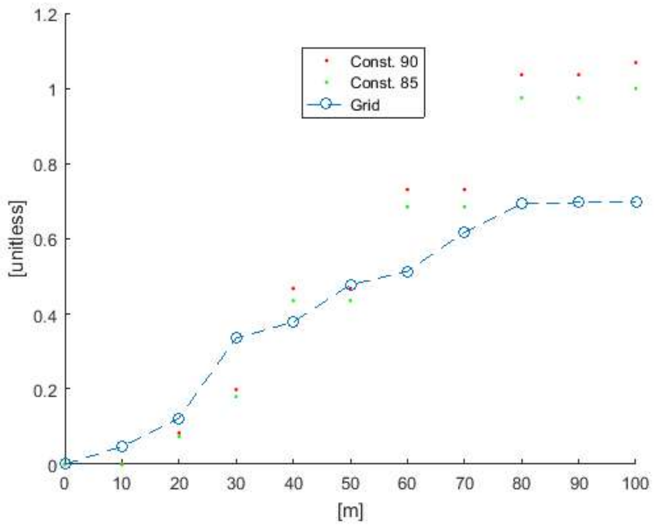

Figure 3.22: The normalized total energy use per meter from the topography shown in Figure 3.20 without regenerative braking.

3.5.2 Allowing regenerative braking

When allowing regenerative braking for the same topography, it consumed 26.9% less when going at a constant speed of 85 km/h. The speed profile can be seen in Figure 3.23 as:

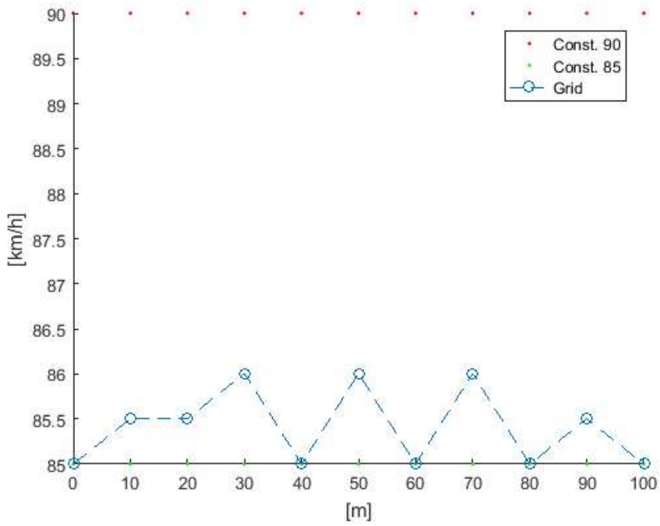

Figure 3.23: Resulting speed profile from the randomized topography seen in Figure 3.20 with regenerative braking.

Ending up saving 17.1% compared to a constant speed of 85 km/h, the total energy spending can be seen in Figure 3.24 as:

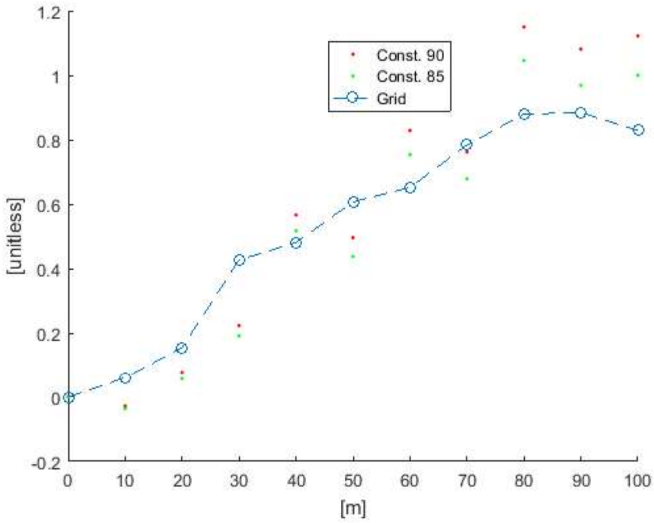

Figure 3.24: The normalized total energy use per meter from the topography shown in Figure 3.20 with regenerative braking.

3.6 Given topography

A test topography was given using free online services and can be seen in Figure 3.25:

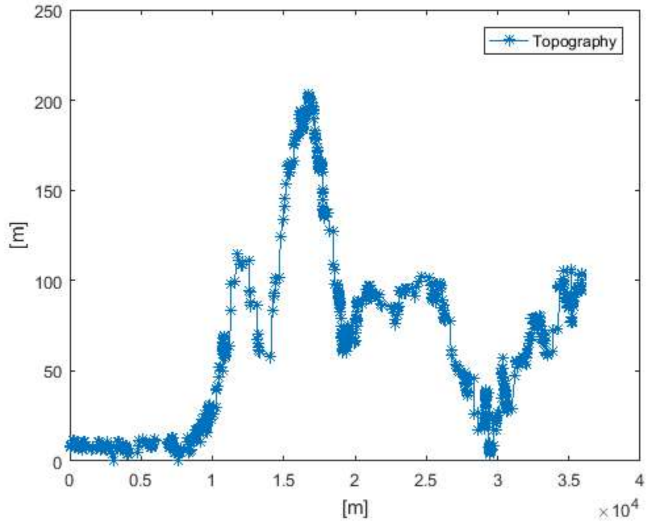

Figure 3.25: The topography given from NEVS.

and using the filter with a point distance of 50m in the x-axis gets the result seen in Figure 3.26:

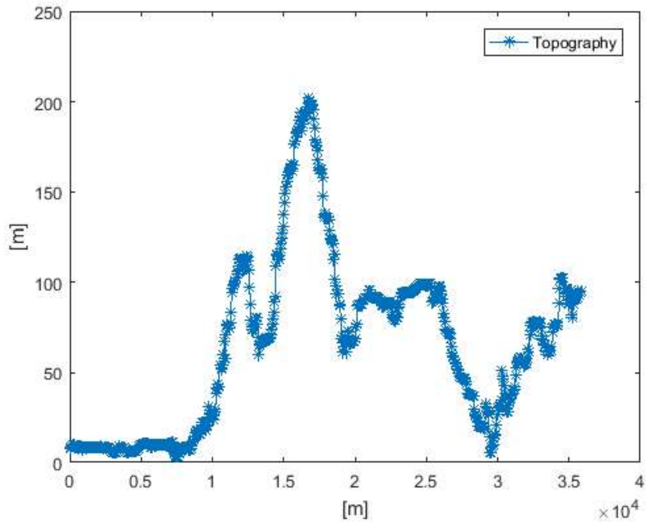

Figure 3.26: The given topography filtered with 50 m between each point, seen from the x-axis.

and a point distance of 10m in Figure 3.27:

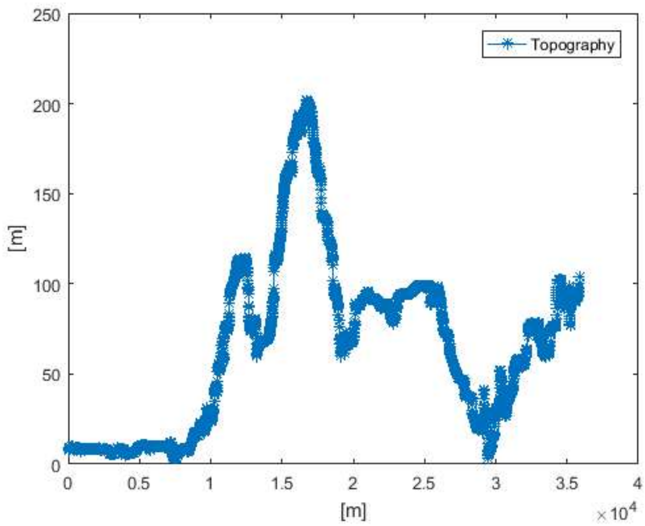

Figure 3.27: The given topography filtered with 10 m between each point, seen from the x-axis.

where a zoomed in case of the filtering can be seen in Figure 2.6 and the explanation in the same section as the figure.

3.6.1 Simulations

There is a cases of *terminated* found in Table 3.1 meaning that the jumps from one speed to another requires an unrealistic acceleration which is impossible. This is the single most important point when deciding the resolutions of the speed and the topography, being scarce enough to allow jumps from a low speed to a higher speed to be an option. To name a very extreme case to prove the point; if the distances in the topography is 1 m and the only allowed speeds are 85 and 95 km/h, the energy required to go from 85 and 95 km/h in only 1 meter is too great for it ever to be an option. The only options available is in that case therefore only the constant speed of 85 km/h (since it requires less energy 95 km/h). If then a very short but extreme upwards slope is in the topography, the result is that it cannot be done since the torque in that particular section is too high.

3.6.2 No regenerative braking

When not allowing the drivetrain to use the engine as a generator when possible saves energy when going downhill as a way of getting back the potential energy spent when going up. Other than making the comparison between this section and Section 3.6.3 the focus is to see how different resolutions in both topography and in speed affect the end result.

To explain the terms used in tables 3.1 through 3.8, ”Res” is the resolution used for the speed, ”% saved 85/90” is comparisons between a constant 85 and a constant 90 km/h compared to the result, ”Ex time” means execution time and ”Av speed” mean average speed. Under ”Av speed” there are two different speeds when using no regenerative braking. This is due to the dynamic system having several solutions with exactly the same energy consumption. Presented are the slowest and the fastest solution.

Table 3.1: 2 m between the nodes seen from the x-axis, without regenerative braking.

Res	% saved 85/90	Ex time	Av speed
2 km/h	*terminated*		
1 km/h	*terminated*		
0.5 km/h	*terminated*		
0.25 km/h	12.3%/2.18%	7450s	86.5/86.5 km/h

Table 3.2: 10 m between the nodes seen from the x-axis, without regenerative braking.

Res	% saved 85/90	Ex time	Av speed
2 km/h	8.9%/15.6%	12s	86.7/86.7 km/h
1 km/h	12.3%/18.7%	27s	86.9/87.0 km/h
0.5 km/h	14.6%/20.9%	118s	86.9/87.2 km/h
0.25 km/h	15.7%/22.0%	255s	86.7/87.0 km/h

Table 3.3: 20 m between the nodes seen from the x-axis, without regenerative braking.

Res	% saved 85/90	Ex time	Av speed
2 km/h	13.9%/18.3%	5s	87.2/87.4 km/h
1 km/h	16.6%/20.9%	10s	87.0/87.4 km/h
0.5 km/h	18.0%/22.3%	28s	86.7/87.2 km/h
0.25 km/h	18.4%/22.6%	77s	86.5/87.2 km/h

Table 3.4: 40 m between the nodes seen from the x-axis, without regenerative braking.

Res	% saved 85/90	Ex time	Av speed
2 km/h	15.7%/19.3%	3s	87.1/87.5 km/h
1 km/h	17.2%/20.8%	4s	86.7/87.3 km/h
0.5 km/h	17.6%/21.2%	11s	86.5/87.2 km/h
0.25 km/h	17.7%/21.3%	30s	86.3/87.2 km/h

3.6.3 Allowing regenerative braking

Table 3.5: 2 m between the nodes seen from the x-axis, with regenerative braking.

Res	% saved 85/90	Saved vs no regen	Ex time	Av speed
2 km/h	*terminated*			
1 km/h	*terminated*			
0.5 km/h	*terminated*			
0.25 km/h	12.3%/21.9%	%	5421s	86.5 km/h

Table 3.6: 10 m between the nodes seen from the x-axis, with regenerative braking.

Res	% saved 85/90	Saved vs no regen	Ex time	Av speed
2 km/h	4.8%/14.2%	19.2%	8s	86.2 km/h
1 km/h	6.7%/15.9%	19.2%	17s	86.2 km/h
0.5 km/h	7.8%/16.9%	19.2%	60s	86.2 km/h
0.25 km/h	8.6%/17.7%	19.2%	256s	86.3 km/h

Table 3.7: 20 m between the nodes seen from the x-axis, with regenerative braking.

Res	% saved 85/90	Saved vs no regen	Ex time	Av speed
2 km/h	2.1%/11.1%	51.9%	4s	85.6 km/h
1 km/h	2.5%/11.5%	51.9%	7s	85.7 km/h
0.5 km/h	2.8%/11.7%	51.9%	19s	85.7 km/h
0.25 km/h	2.9%/11.9%	51.9%	74s	85.7 km/h

Table 3.8: 40 m between the nodes seen from the x-axis, with regenerative braking.

Res	% saved 85/90	Saved vs no regen	Ex time	Av speed
2 km/h	0.2%/9.0%	65.1%	2s	85.2 km/h
1 km/h	0.2%/9.0%	65.1%	3s	85.2 km/h
0.5 km/h	0.2%/9.0%	65.1%	8s	85.2 km/h
0.25 km/h	0.2%/9.0%	65.1%	27s	85.2 km/h

Table 3.9 explains in great detail the pros and the cons of having different settings in the simulation. It is clear that there are advantages and disadvantages with all combinations.

Table 3.9: The pros and the cons of different resolutions.

		High res. speed	Low res. speed
High res. topo.	Pro	Closest to a continuous solution Most likely to work in real life	Easy to identifiy extreme hills through looking at the resulting speed profile
	Con	Takes long time to compute	Will only increase speed in extreme hills Cannot save energy in small topographical changes
Low res. topo.	Pro	Allows for small changes of the speed	Computes the fastest Quick check to learn about the characteristics of the topography
	Con	Misses small changes in the topography	Does not represent the reality well

Chapter 4

Analysis and discussion

4.1 Evaluating the method and the findings

Since this master thesis is not about comparing one method to another and then deciding which one is the best, comparing it to other available methods is difficult. One aspect that is especially difficult is to evaluate how it compares to execution time in order to get good results depending on resolution. What can be said is that there are either few or no other method that both can calculate the optimal solution without being affected by local minimums, for this particular problem. This opens up the possibility to solve advanced problems without having knowledge of control engineering methods in a very robust manner.

There are two main limitations of the method presented and best illustrated in Appendix A.1.1. The first being that there may be an unknown factor making it impossible to weigh options against each other, a solution for this is presented in Appendix A.2. Another limitation is the chosen resolution not being accurate enough when evaluating a road. Table 3.9 summarizes the pros and the cons from different resolutions.

The only way to know how it measure up to other solutions is to test two of them using the same conditions. Up until that time, much of this methods flaws will be an unknown.

4.2 Time consumption - punishing slow speeds

The resulting speed profiles from the simulations presented in tables 3.1 through 3.8 in Section 3.6.2 and in Section 3.6.3 shows that the average speed always were under the speed limit of 90 km/h. The difference between 86 km/h and 90 km/h is in 100 km 3 minutes and 6 seconds. This will ultimately lead to the traveling time being lengthened do a degree which might be unacceptable for the driver. To punish going too slow might at first thought be an option worth considering. After experimenting with it, it is obvious that it is very complicated, especially if the goal is to at the same time save energy.

An option making the driver feel more in control over this situation can be letting him or her adjusting the allowed difference for his or her vehicle. To set it to ±5 km/h for example and seeing the expected arrival time and how much energy potentially can be saved is handy to weigh the different alternatives available. Calculating this in real time with high accuracy can take a long time and a lookup table is recommended for fast results.

4.3 Implementation

Having extremely high resolution requiring a lot of computation power to calculate the recommended speed profile is in no way obstacle to implementing this solution. The EV can simply use an internally stored multidimensional lookup table accounting for all the different conditions that might happen when driving. The only thing that in practice is needed from the car is the location of the EV and information about the conditions. From that information, the best option can be selected.

4.4 Calculate in real time

If circumstances required calculations in real time, this could be done even if the end result might differ from one done in a more powerful computer given more time. The question is only how much internal computing power can the EV provided and what resolution is the best one that can be calculated using that power given. There is a constant time limit when traveling. The boundary condition is therefore that the calculations should be further ahead than the car itself moves.

The big limitation that should always be considered when making in real time calculations is that the calculated speed profile is only optimal to its final point and not further ahead. If the car were to deviate from the recommended speed, efforts should be made to return to it. If it deviates too much, the calculations have to be made yet again to guarantee an optimal recommendation.

4.5 Other applications

The only factor deciding if this method can be used for other applications (where the disturbance is known but random) is the possibility to sort out data. This is the method easiest understood from looking at the difference between Figure A.2 and A.3 in Appendix A.1. For example, in a cheap and simple way investigating the height options for an air plane or keeping a house within certain temperatures depending on expected temperature and electricity prices.

As long as many different options can be tested and only the best kept for further testing, the path finding algorithm presented in this thesis is a viable option.

Chapter 5

Conclusions and future work

5.1 The results

As seen in Table 3.9 there are a lot of Conclusions that can be drawn from having different resolutions of the topography and the speed options for every step. Important to notice is that the closer together the points in the resolution, a higher resolution in the speed options is needed. Conclusions can be drawn from Table 3.3 and Table 3.6 where 18.4% and 8.6% of energy was saved. One should be careful to make any conclusions from 3.5 since the topography could be somewhat faulty because of the filtering of the topography. A good rule of thumb is about 0.01 km/h per meter to be safe and get a decent estimation.

5.2 Applications

Given that the algorithm is tested and optimized to represent the reality as well as possible, energy saving speeds have three main applications. The first and simplest one is to act as a guide for the drive by suggesting speed that benefits the energy consumption. The second one is acting as a more energy efficient cruise controller. The third application is as a part of a self-driving car, giving the option to try saving as much energy as possible. All of the options need to be at least as safe as not using them, since safety never can be a compromise. The driver should not feel distracted by the reminder of what speed should be kept and the second and third option are therefore the main recommendation of application.

5.3 Charge left

When an estimation has been done how much energy it will consume driving it from point **A** to point **B**, a calculation of how much energy the battery has left can be done. If the destination is a place where there are no charging station available, several questions needs to be answered. One of them is "will the EV be able to get to a nearby charging station when going from the destination?" in other word, will the car be stranded there? If time isn't an issue, a stop can be made on the way and the problem can therefore be solved. Otherwise this has to be investigated further.

5.4 Further ideas about the filter

This thesis suggests in Section 2.4.4 3 tools that can be used to make the filtered topography to best represent the reality. Weighing these filters needs to be combined with more accurate measurement data until it matches reality as good as possible.

One way of collecting more accurate data about road topography and conditions is by using cars own internal sensors, that way many readings can be gathered useful for creating accurate data.

5.5 Admissibility of A* algorithm

As is explained in Section A.2, a bigger window to ensure that the chosen parents to the nodes were correct ones, might be needed. In that window an all-out search of all the different options is a way to solve that problem. Another way is to use the A* algorithm (or a similar one) to narrow the search down. For this still to work the A* algorithm has to be admissible in order to be optimal. Deriving an admissible function requires a lot of work but might be worth it if the window is too big.

5.6 Future work

What need to be taken into consideration to represent the reality are speed limitations due to weather, construction works and sharp curves to name a few. Adding those limitations into the algorithm only takes a few changes and will not affect finding the optimal speed given the circumstances but needs nevertheless to be done.

As this is thought of a tool in a bigger algorithm deciding the best route between two points on a map, it needs to be integrated in such an algorithm. If the energy evaluation between road options represents the reality well an A* search algorithm is suggested as one way to act as the main intelligence in the whole system. An easier approach requiring less research and work is using Dijkstra's search algorithm which can be done despite it not working with negative values.

Appendix A

Appendix

A.1 Resulting search algorithm

Dijkstra works roughly by checking all alternatives and then picking the best option (the process of being allowed to do so is the key to this whole thesis). When an alternative route is found to a previously explored node, the most efficient way to get there and the cost of getting there is updated. This particular scenario will never happen in this application because of two reasons. Firstly all the nodes only point at one direction and secondly the order being the one of a Breadth-first search algorithm. What happens is when an iteration of one point to the next is completed; the next node has several alternatives to choose from. Only one is kept which is allowed according to Section 2.4.2.

Following in Sections A.1.1 and A.1.2 is meant purely for understanding the algorithm better. Similar behaviour as in the examples can be seen in Section 3 but nothing that is exactly the same.

A.1.1 Flat and up

The slowest initial speed is chosen as the initial speed to see how increased speed can save energy. Figure A.1 shows how the first iteration starts from going at 85 km/h and from there reaching all the different alternatives.

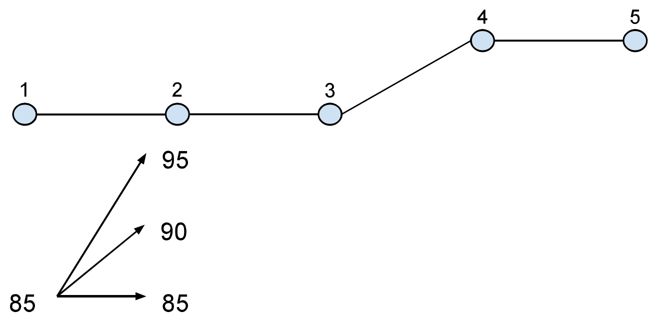

Figure A.1: Small slope up, first iteration.

Second iteration shown in Figure A.2 starts from 85, 90 and 95 km/h and from those speeds checks the energy consumption to 85, 90 and 95 resulting in 9 different checks.

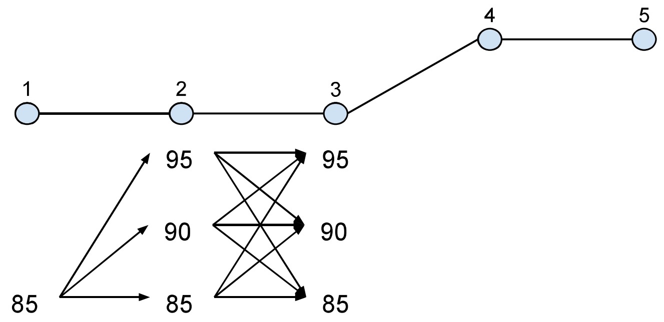

Figure A.2: Small slope up, second iteration.

Figure A.3 shows what was chosen from Figure A.2 based on what alternative consumed the least amount of total energy. As mentioned in the beginning of this section, for the engine model used for this example it makes sense to accelerate more gradually despite the increased average speed. Therefore 85, 90 and 95 km/h on point 1, 2 and 3 is chosen to reach 95 km/h in node 3 rather than 85, 85 and 95 because it results in a lower total energy consumption.

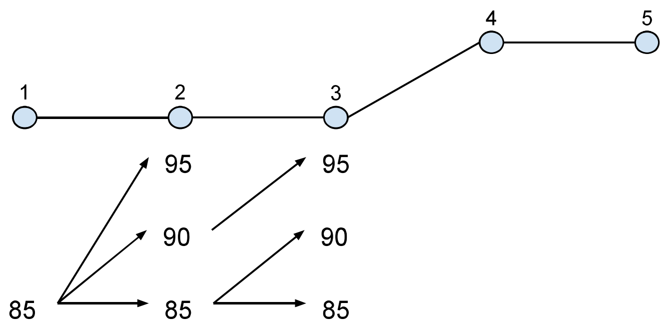

Figure A.3: Small slope up, third iteration.

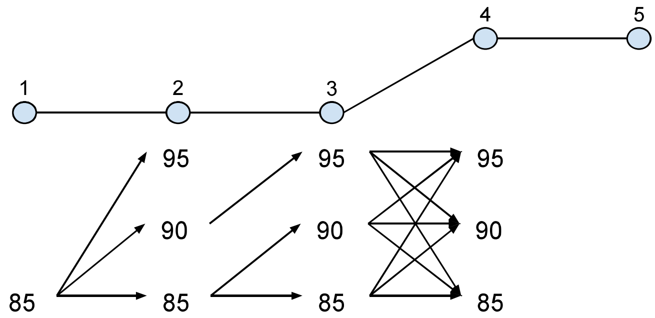

Figure A.4: Small slope up, forth iteration.

When going up a hill the engine has to increase its output significantly just to keep a constant speed. The alternative is to increase the speed bit by bit right before the hill. In this case that is seen as more energy efficient going from 85, 90, 95 and then 85 rather than going 85, 85, 85 and 85. That behaviour is illustrated in Figure A.4 and then Figure A.5 where the best speeds are chosen.

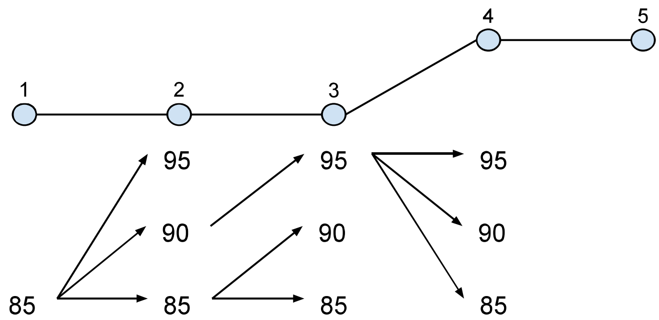

Figure A.5: Small slope up, fifth iteration.

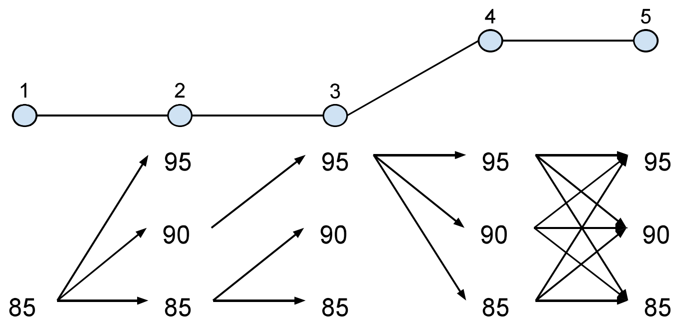

Figure A.6: Small slope up, sixth iteration.

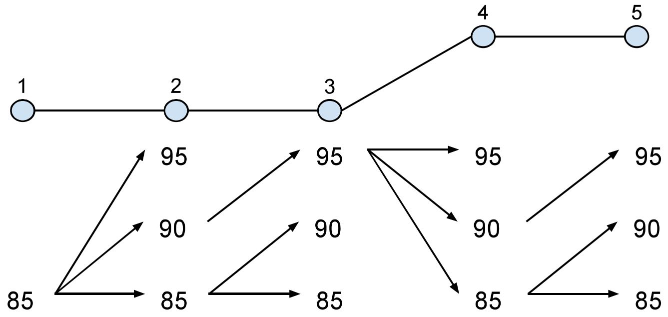

Figure A.7: Small slope up, seventh iteration.

Last iteration illustrated in Figure A.6 and A.7 is a repetition of happened before between point 2 and 3. When looking through the whole topography and the algorithm is done with this part it looks at the alternatives available in node 5 as speed 85, 90 and 95 to see which one consumes the least amount of energy in total. 85 is the obvious choice and after it is picked the algorithm backtracks and looks where it came from to choose that speed profile as the recommended one.

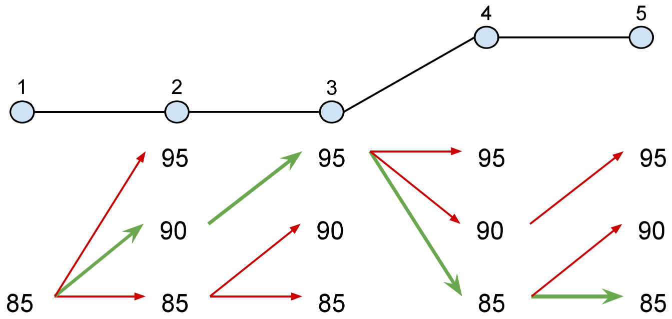

Figure A.8: Small slope up, eighth iteration.

A.1.2 Flat and down

For a different topography where it is going flat, down and then flat, a different speed profile than the one presented in Section A.1.1 is needed. As before and seen in Figure A.9 it starts at 85 km/h. The first interesting part is shown in Figure A.10 and A.11 where the EV goes downhill

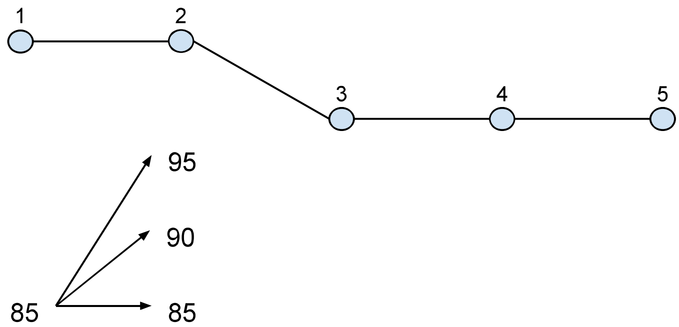

Figure A.9: Small slope down, first iteration.

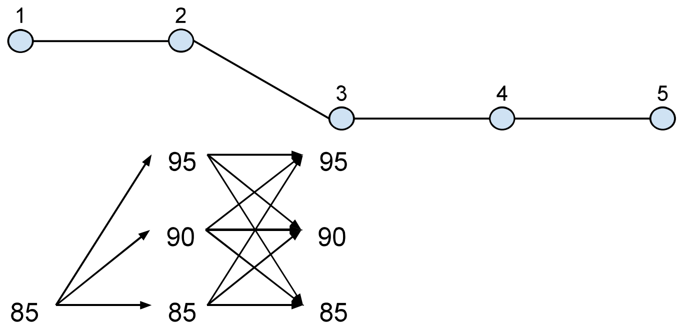

Figure A.10: Small slope down, second iteration.

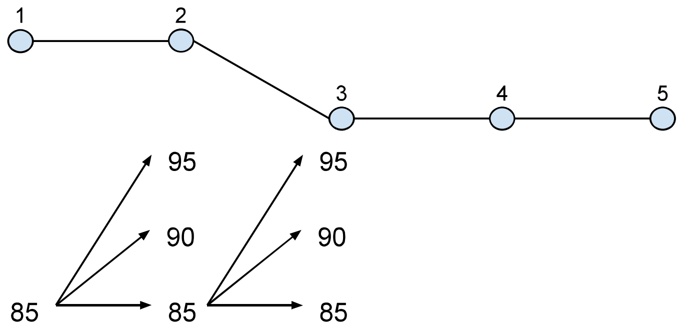

Figure A.11: Small slope down, third iteration.

As seen in Figure A.11, converting as much potential energy as possible to kinetic energy is the most energy efficient. This requires the least amount of energy to get from node 1 to 2 and then uses the most of the potential energy available when going downhill.

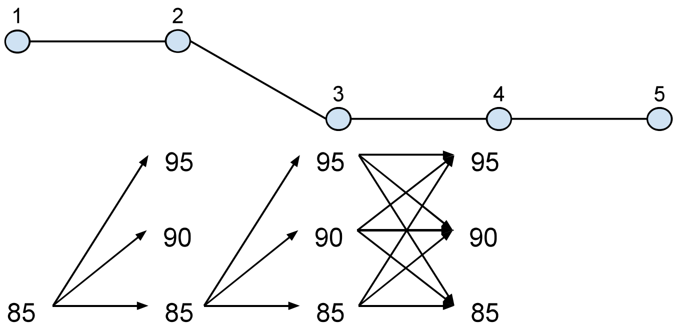

Figure A.12: Small slope down, forth iteration.

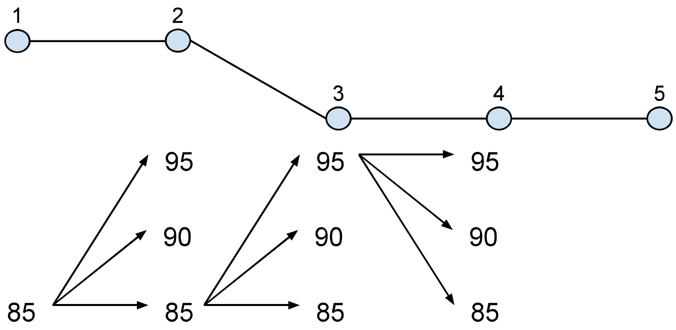

Figure A.13: Small slope down, fifth iteration.

Looking at Figure A.12 and A.13, the idea is that the hill is steep enough for the EV having to brake in order not to go faster than 90 km/h but just making it to 95 km/h without braking. This means that Going 85 and 90 km/h in node 4 requires little or no energy when starting at 95 km/h in node 3.

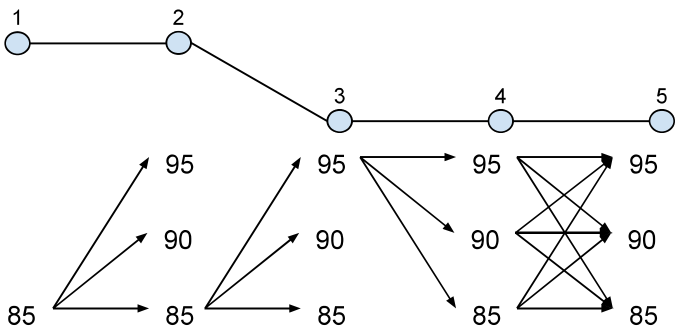

Figure A.14: Small slope down, sixth iteration.

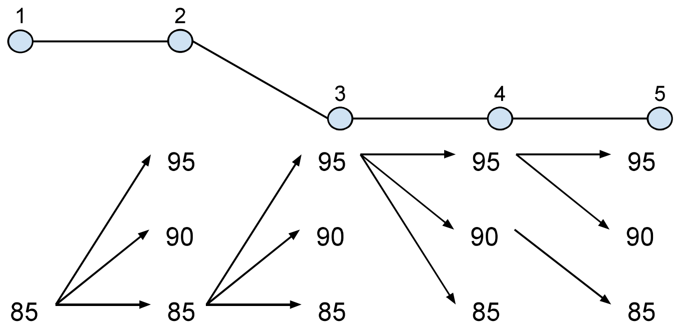

Figure A.15: Small slope down, seventh iteration.

The only difference of going at 90 km/h compared to 85 km/h in point 4 is that 90 km/h is faster, there is no difference in total energy consumed to that point because getting to 85 km/h just meant braking more. This leaves 90 km/h at an advantage since it has more kinetic energy which it can use when reaching 85 km/h in node 5. This is illustrated in Figure A.14 and A.15.

As for the case presented in Section A.1.1 in Figure A.8, Figure A.16 shows what speed option available in node 5 consumed the least amount of energy. Speed 85 km/h is picked and then backtracked to with what speeds got it to that point. The speed picked is marked with a green arrow and the ones not picked in red.

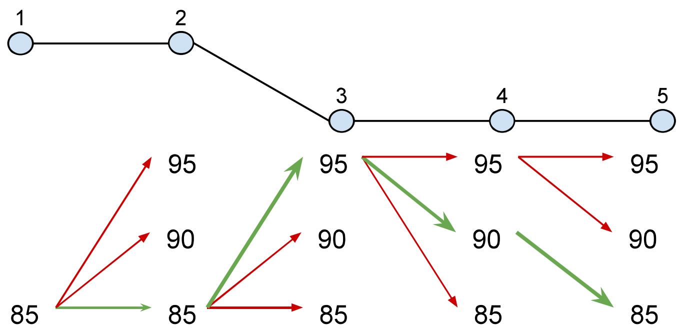

Figure A.16: Small slope down, eighth iteration.

A.1.3 Higher resolution

To make calculations more accurate and closer to continuous time, more speed options as illustrated in Figure A.17 is one of the two ways to achieve that. With a linear increase of the speed resolution comes an exponential cost in iterations. The other option is to do as illustrated in Figure A.18 and is to increase the topographical points which increases the cost linearly. The effects of balancing the two can be seen in Table 3.9 and simulation times in tables 3.1 through 3.4.

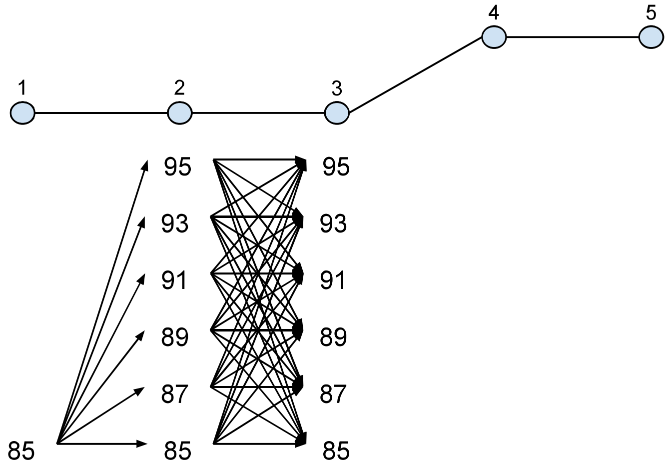

Figure A.17: Small slope up, high speed resolution.

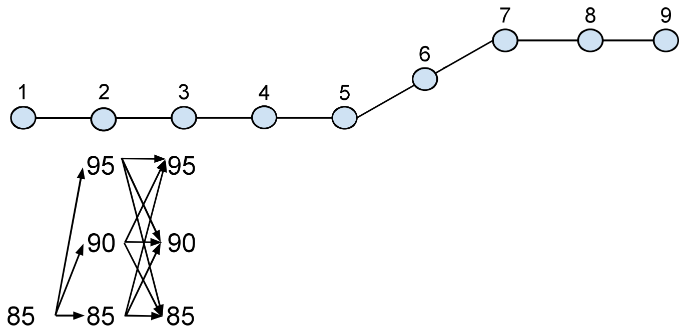

Figure A.18: Small slope up, high topography resolution.

A.2 Expanding the window

In case the proof made in Section 2.4.2 is wrong because of some missed factor (which is highly unlikely), there is an easy but expensive fix to that problem. Solution is simply to look further ahead at all the alternatives before making a decision on the particular spot. How that is done is illustrated through figures A.19 to A.23. For this point if the window is to be very large, algorithms such as A* search algorithm can be used to narrow the search down.

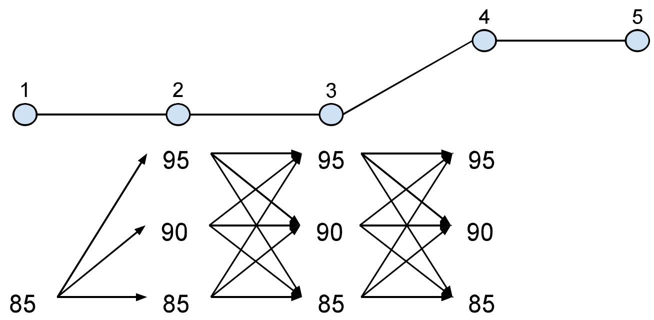

Figure A.19: Small slope up, expanding the window, first, second and third iteration.

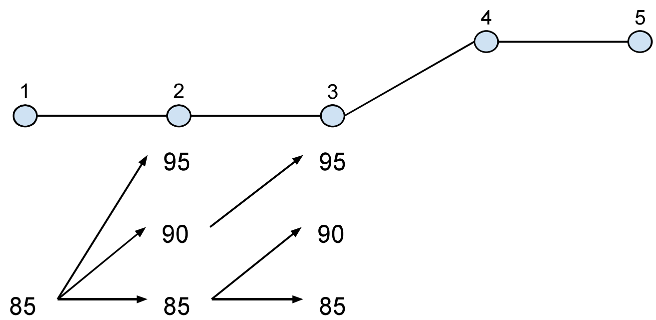

Figure A.20: Small slope up, expanding the window, forth iteration.

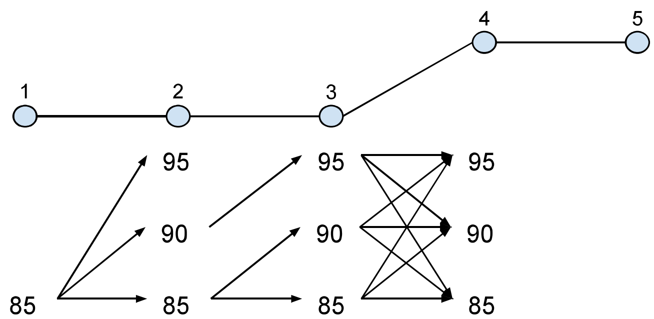

Figure A.21: Small slope up, expanding the window, fifth iteration.

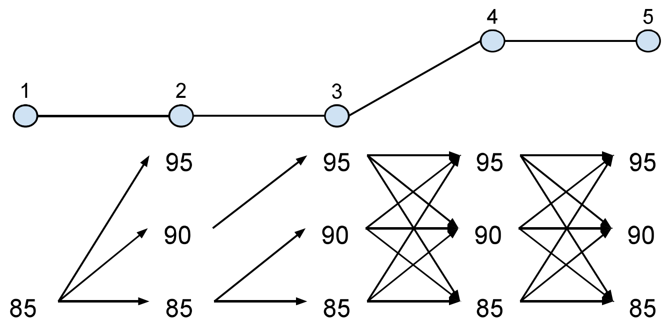

Figure A.22: Small slope up, expanding the window, sixth iteration.

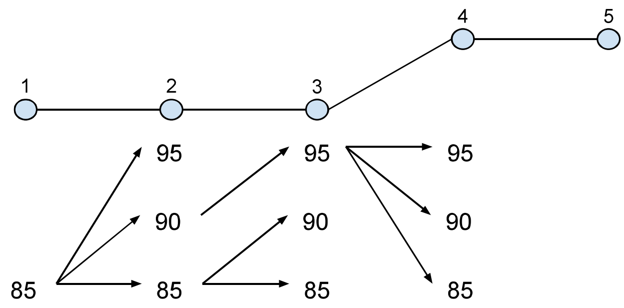

Figure A.23: Small slope up, expanding the window, seventh iteration.

Bibliography

[1] Hart, P., Nilsson, N., & Raphael, B. *A formal basis for the heuristic determination of minimum cost paths.* IEEE Transactions on Systems Science and Cybernetics (SSC), 4 (2), (1968), pages 100–107.

[2] Eric A. Hansen, Rong Zhou, *Anytime Heuristic Search*, Journal of Artificial Intelligence Research 28 , Mississippi State University, Mississippi State, USA (2007), page 268

[3] Rajesh Rajamani, Vehicle Dynamics and Control, First edition, (2006)

[4] Erik Hellström,Explicit use of road topography for model predictive cruise control in heavy trucks, Master's thesis, Linköping University, Linköping, Sweden, 21st February 2005

[5] Jaime Junell and Kagan Tumer, *Robust Predictive Criuse Control for Commercial Vehicles*, Academic Paper, Oregon State University, Corvallis, Oregon, USA, 2010

[6] Jan Lundgren, Mikael Rönnqvist and Peter Värbrand. Optimeringslära, upplaga 3:1. Studentlitteratur 2003

[7] John Wahnström, Energy Optimization for Platooning through Utilizing the Road Topography, Master thesis, Lund University, Lund, Sweden, 2015

[8] Mathias Mattsson och Rasmus Mehler, Optimal Vehicle Speed Control Using a Model Predictive Controller for an Overactuated Vehicle, Master thesis, Linköping University, Linköping, Sweden, 2015

[9] Wang, J. Alexander, L. and Rajamani, R., "Friction Estimation on Highway Vehicles Using Longitudinal Measurements", *ASME Journal of Dynamic systems, Measurement and Control*, Special Issue on Sensors, Vol. 126, No. 2, pp. 265-275, June 2004

[10] White, R.A. and Korst, H.H., "The Determination of Vehicle Drag Contributions from Coastdown Tests, "SAE Transactions, Vol. 81, paper 720099, 1972

[11] Chin Yang Lee, "An Algorithm for Path Connections and Its Applications". IRE Transactions on Electronic Computers, 1961