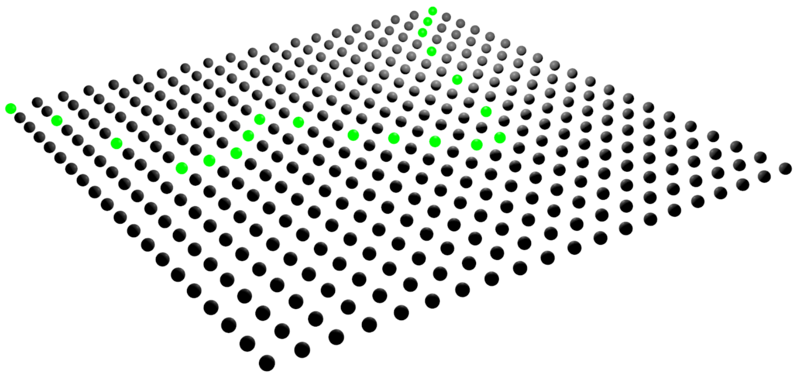
CHALMERS
UNIVERSITY OF TECHNOLOGY



Speed Optimization of an Electric Autonomous Vehicle

Master's thesis in mathematical engineering and computational science

WILLIAM ORTON SÖRENSEN

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2019

MASTER'S THESIS 2019:NN

Speed Optimization of an Electric Autonomous Vehicle

WILLIAM ORTON SÖRENSEN

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electric Engineering
division of Systems- and Control
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2019

Speed Optimization of an Electric Autonomous Vehicle
WILLIAM ORTON SÖRENSEN

Supervisor: Jonas Hellgren, Volvo Group Trucks Technology
Examiner: Jonas Fredriksson, Chalmers University of Technology

Master's Thesis 2019:NN
Department of Electrical Engineering
division of Systems- and Control
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: A visualization of how the optimization algorithm picks the states that constitute the optimal velocity profile.

Typeset in LaTeX
Printed by [Name of printing company]
Gothenburg, Sweden 2019

Speed Optimization of an Electric Autonomous Vehicl
WILLIAM ORTON SÖRENSEN
Department of Electrical Engineering
Chalmers University of Technology

Abstract

The purpose of this thesis is to construct and implement an optimization algorithm for choosing the optimal velocity profile for a fully electric vehicle from point A to point B. This is done by translating the speed optimization problem to a shortest path problem in graph theory. The shortest path problem has known solution methods.
To find an optimal velocity one also needs a vehicle model describing the behavior of the vehicle in relation to speed, acceleration, mass etc. The algorithm in this thesis is flexible. One can simply change the vehicle model and still use the same algorithm. So the speed optimization algorithm is not limited to electric vehicle. The concept would work for any type of vehicle.
A vehicle model for the fully electric VERA truck is described. The speed optimization algorithm is then tested with the VERA model on a chosen set of test routes followed by a real world route that is intended for the VERA truck. The results show velocity profile behaviors that are expected, for example that a heavy vehicle accelerates before a steep hill, indicating that the algorithm is giving appropriate results.

Acknowledgements

I am forever grateful for my supervisor, Jonas Hellgren. He has enthusiastically guided me through this thesis. Originally the intention was that Jonas would give a colleague the role as supervisor while Jonas was on parental leave. However, Jonas saw great potential in this thesis work and chose to continue as the supervisor. He has continuously expressed his belief in me and my work. This has really been motivating for me and I feel that I have made a significant contribution to the department and the development of self driving cars. I could not imagine a better supervisor. Therefor, from the bottom of my heart, I want to thank Jonas and the people I met at Volvo Group Trucks Technology for all the support and for an amazing experience.

Orton Sörensen, Gothenburg, June 2019

Contents

Nomenclature

Symbol	**Quantity**	**Unit**
m	Mass	kg
a	Acceleration	m/s^2
v	Velocity	m/s
f	Roll resistance	-
CdA	Air resistance coefficient	m^2
ρ_{Air}	Air density	kg/m^3
μ	Friction coefficient	-
R_{Wheel}	Wheel radius	m
g	Gravitational acceleration	m/s^2
α	Slope Angle	$^\circ$
ω	cost divisor	-

1 Introduction

In the transport industry there is a common goal to put autonomous vehicles on the road. A lot of research has been and is being done on the topic but there is still a long way to go before the first autonomous vehicles are rolling safely on the main roads. There are several things to think of such as safety, fuel efficiency, ethical and legal aspects, optimal speed and much more. This thesis will focus on developing an algorithm for a fully electric autonomous vehicle to choose an optimal velocity on a given route.

Besides articles, master's theses and books, products such as *I-see* by Volvo, *Scania Active Prediction* by Scania and *RunSmart Predictive Cruise* by Daimler exist for fuel optimization of conventional vehicles with drivers. Conventional vehicles are vehicles powered fully or partially by a combustion engine. These products are neatly summarized and explained in a previous master thesis by Thomas Köhler [2]. The products are all designed to reduce fuel consumption for a horizon of a few kilometers ahead. A common denominator for these products is that they reduce fuel consumption by for example increasing the speed before a steep slope and slowing down before a down hill occurs and avoids gear shifting. These are behaviors that truck drivers are taught during training. However, it is easy to imagine that over long distances this behavior may be hard to consistently maintain for a driver. This is why these products have such a great fuel saving potential.

1.1 Objectives

In the pursuit of producing autonomous trucks the department Vehicle Automation(VA) at the company Volvo Group Truck Technology(VGTT) want to conduct research on the topic of vehicle speed optimization on fully electric vehicles. More specifically VGTT has a concept truck called VERA. The VERA truck is supposed to be a fully autonomous truck for transportation of goods at low velocity(below 50 km/h). VA have created this thesis in order initiate the research on speed optimization of the VERA truck.

The optimal velocity profile on a given path between point A and point B is the one letting the vehicle arrive safely, minimizing operating costs and within a reasonable amount of time. Operating costs is a combination of costs such as energy costs, driver costs, wear and tear etc.

The main objectives are:

- What is the research front in the field of speed optimization?
- How can an optimal vehicle speed profile be designed for fully electric autonomous vehicles?

1.2 Problem description

This section aims to give a general mathematical problem description of the speed optimization problem and a few limitations of the project.

Consider a route from point A to point B with total distance D, where x is the position along the route, $x \in [0, D]$. The following is assumed to be given about the route:

- the altitude is given as a function of the position, $\mathrm{h}(x)$, and
- a lower and upper speed limit $v_{\mathrm{min}}(x)$ and $v_{\mathrm{max}}(x)$ respectively.

Figure 1.2 is an example of what the altitude profile of a route may look like.

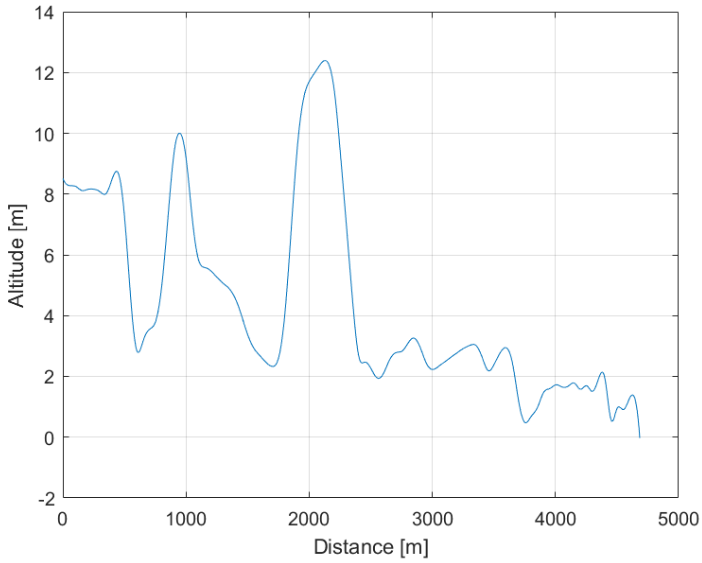


Figure 1.1: Example of the altitude profile for a given route.

The problem of finding an optimal velocity profile, $v(x)$, can be described mathematically as a non-linear optimization problem. For an optimization problem there is an objective function that is to be minimized or maximized. The objective function defines the goal of the optimization problem. In this case the goal is to minimize time and operating costs. These terms often contradict each other. Meaning that low time cost implies high operating cost and vice versa. The objective function is thus a trade off between total travel time and operating costs. In order to prioritize between the two a factor $\omega \in [0, 1]$ is introduced. Moreover, time and operating cost have different units. Time has the unit seconds but operating costs may be a combination of costs such as energy consumption and battery wear. Typically a big truck will require several thousand joules of energy to move 10 meters while it may only require a few seconds of time. Hence, a normalization factor is introduced for each type of cost that is considered.

Define $T(v)$ as the total time required to travel from point A to point B with velocity profile v. The cost of time is then

$$\text{timeCost}(v) = \frac{1}{T_{\text{norm}}} T(v). \tag{1.1}$$

Moreover, define the cost of operation as the following

$$\begin{aligned} \text{operatingCost}(v) = &\frac{1}{\text{C1}_{\text{norm}}} \text{C1}(v) \\ &+ \frac{1}{\text{C2}_{\text{norm}}} \text{C2}(v). \end{aligned} \tag{1.2}$$

where $\text{C1}(v)$ and $\text{C2}(v)$ are examples of two different cost types that depend on the velocity profile v. The normalization factors $T_{\text{norm}}, \text{C1}_{\text{norm}}, \text{C2}_{\text{norm}}$ are yet to be defined. They are simply stated here to give a broader perspective of the problem. They will be defined later on.

The general optimization problem can then be formulated as

$$\underset{v}{\text{minimize}} \quad \omega \cdot \text{timeCost}(v) + (1-\omega) \cdot \text{operatingCosts}(v) \tag{1.3a}$$

$$\text{subject to} \quad v \in \text{Feasible}(v_{\text{Start}}), \tag{1.3b}$$

$$S = \int_0^D v(x)\mathrm{d}x, \tag{1.3c}$$

$$v(0) = v_{\text{Start}}, \tag{1.3d}$$

$$v(D) = v_{\text{End}}, \tag{1.3e}$$

$$v_{\text{Min}} \leq v \leq v_{\text{Max}} \tag{1.3f}$$

Note that we want to find the optimal function v describing the velocity for the route between point A and point B with initial velocity v_{Start} and end velocity v_{End}. Moreover, Feasible is the set of all physically feasible velocity profiles. A few examples of the optimal velocities can be seen in figure 1.2

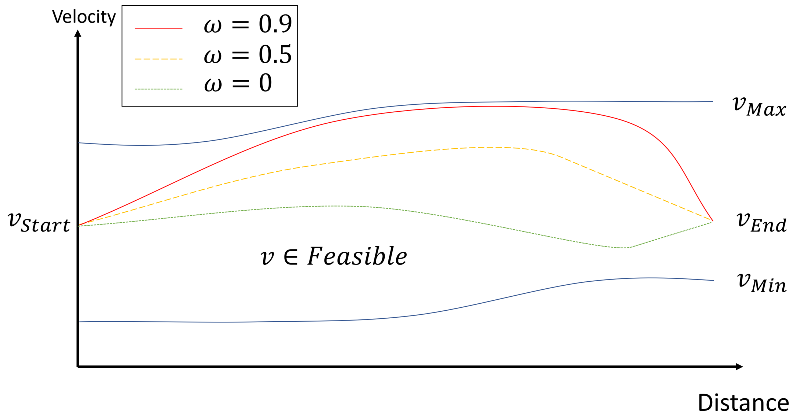


Figure 1.2: How optimal velocities may look like with a few different ω

1.3 Limitations

The problem is limited by the following limitations:

- No surrounding traffic is considered. Traffic is very stochastic and at this stage of research into speed optimization traffic it is too complex to model.
- The vehicle is considered fully electric.
- The speed optimization problem described above will not be solved analytically. The goal is to give a numerical approximation to the function v.
- The state of charge of a battery is assumed to be constant for a given route.

1.4 Related work

In this section previous articles and existing products relating to the topic of speed optimization will be presented and briefly described.

Several internal VGTT documents related to speed optimization have been studied. However, they are not published. Other published articles, that are at the front of speed optimization research, are listed below.

The optimal speed control problem for fully electric vehicles is not a well studied topic in the literature, but there exists a number of attempts on how to deal with the nonlinear, non-convex optimization problem. In [14], a method of reducing fuel consumption and the number of gear changes on an intercity highway by setting up a mixed integer linear optimization problem is presented. As a linear optimization problem becomes much easier to solve compared to a non-linear optimization problem. The objective is limited to only minimize operation costs by minimizing the fuel consumption. Other articles that focus on minimizing energy consumption and costs are [8] and [9].

An alternative approach is to use dynamic programming as presented in [4]. The objective of this article is to minimize time and fuel consumption for heavy diesel trucks. The algorithm presented in the article is implemented on a real truck driving on a highway. The results show that the fuel consumption is reduced. Another dynamic programming approach that considers the tradeoff between fuel, time and driver comfort is discussed in [11].

Another popular method for a lot of research is to apply machine learning techniques. [5] is a master thesis that applies machine learning to minimize the ownership cost for long haul hybrid trucks. Costs that are included in the thesis is fuel consumption, battery degradation and disc brake degradation. A reinforcement learning algorithm is used to find that the ownership cost is reduced by roughly 4% compared to conventional trucks. This thesis was not about speed optimization but important to read due to the mathematical modelling that was done in it.

A collected set of articles can be found in [6]. They describe approaches to speed optimization, modelling and simulation techniques for heavy trucks. Specifically, the article includes research on optimal control of vehicle propulsion, optimal gear engagement and disengagement to minimize fuel consumption and A set of test routes are constructed to test the algorithm.

The author's of [7] introduce influence diagrams to generate an optimal speed profile for a formula one car on a specific racing track. The generated speed profiles are claimed to correspond well to test driving of test pilots where the goal was to minimize the total lap time.

The articles above try to solve single vehicle speed optimization. However, in reality there are multiple vehicles on the road. To reduce the total amount of fuel consumption produced by vehicles one should also consider a multiple vehicle speed optimization method. [10] suggests a game theoretic cooperative speed optimization to minimize vehicles idling times and the number of stops at signalized intersections.

To summarize this section there seems to be a fair amount of work on speed optimization for conventional vehicle but not so much on fully electrical vehicles. The related work given in this section has influenced the ideas and equations presented in this thesis. For example they have given inspiration on how to present results and what is reasonable to include in the mathematical model of the vehicle.

2
Theory

The topics presented in this chapter are necessary for the understanding of the algorithm that generates the optimal velocity profile.

2.1 Basic graph theory

A graph consists of nodes and edges. A node can be an abstract object while an edge represents relationships between the nodes. For example, assume we are at a party consisting of 20 guests. Each guest can then be seen as a node. If two guests shake hands a connection (an edge) between the two guests (the nodes) will appear, and we can view this as a graph, see figure 2.1. In this example the edges are undirected, as it does not matter if guest 1 shake hands with guest 2 or vice versa.

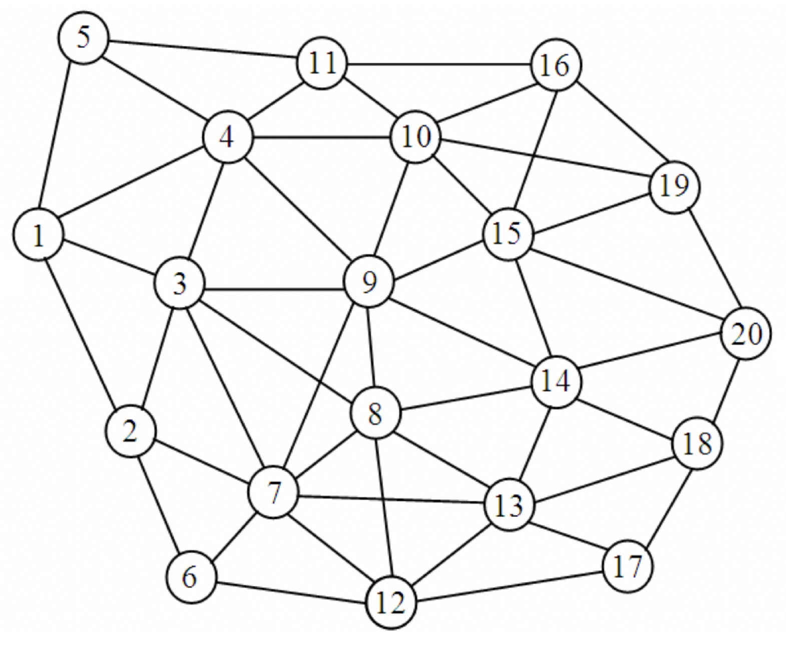


Figure 2.1: An example of an undirected graph with 20 nodes.

Another example where directed and undirected edges can be present is in the shortest path problem where each node represents a location. The possibility of moving from one location to another location is represented by an edge. This edge is directed or undirected depending on if its possible to go the same way back. The cost of moving from one node to another node is called an edge cost or weight. The edge cost can for example represent required energy or a fee. The shortest path between node s and node t is the sequence of nodes minimizing the total edge costs necessary to move from node s to t.

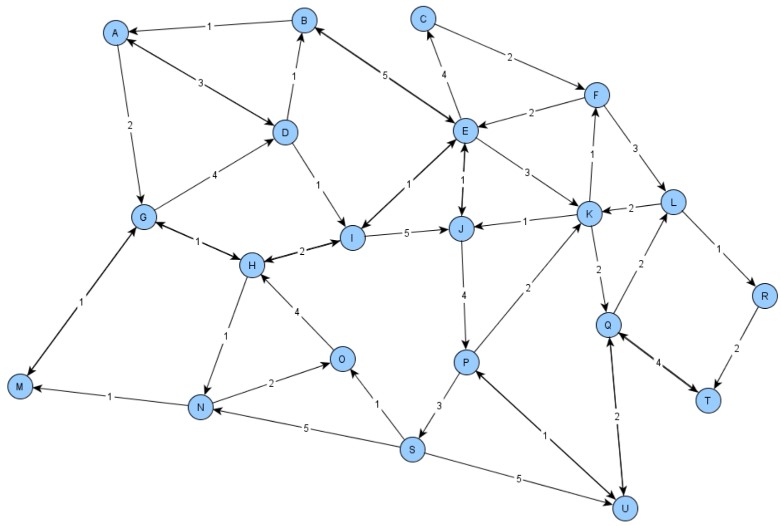


Figure 2.2: An example of a graph with directed and undirected edges with edge costs/weights

2.2 Bellman's principle of optimality

Assume a given initial state x_0 and a set of all possible state as X. Let $a_t \in \Gamma(x_k)$ define an element in the set of actions that can be taken from state x_t. When action a is taken from state x a new state is given by $T(x, a)$. The cost of this action is defined as $F(x, a)$. A finite horizon decision problem can then be defined as

$$V(x_0) = \min_{\{a_t\}_{t=0}^{n}} \sum_{t=0}^{n} F(x_t, a_t) \tag{2.1}$$

subject to

$$a_t \in \Gamma(x_t) \tag{2.2}$$
$$x_{t+1} = T(x_t, a_t) \tag{2.3}$$
$$\forall t = 0, 1, 2, ..., n \tag{2.4}$$

Solving this gives the optimal policy $a_0, a_1, a_2, ..., a_n$
Bellman's principle of optimality states:

> *An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision*, [1].

This principle means that the above problem can be simplified to

$$V(x) = \min_{a \in \Gamma(x)} \{F(x, a) + V(T(x, a))\} \quad \forall x \in X. \tag{2.5}$$

This equation is called the Bellman equation and is a vital part of the algorithm generating the optimal velocity profile. There is no point in trying to describe something that has already been well described. The following description of the Bellman equation comes from wikipedia: "The Bellman equation is classified as a functional equation, because solving it means finding the unknown function V, which is the value function. Recall that the value function describes the best possible value of the objective, as a function of the state x. By calculating the value function, we will also find the function a(x) that describes the optimal action as a function of the state; this is called the policy function".

2.3 Dynamic Programming

"Those who do not remember the past are bound to repeat it"

- Dynamic Programming

Dynamic programming consists of recursively breaking down a big problem into sub problems and saving the solutions to the sub-problems in order to reduce the number of calculations needed to solve the big problem.

The following example illustrates this principle of dynamic programming. Consider the problem of finding the n:th Fibbonacci number f_n. The Fibbonacci sequence has the following recursive formula

$$f(n) = f(n-1) + f(n-2) \tag{2.6}$$
$$f(1) = 1 \tag{2.7}$$
$$f(0) = 0. \tag{2.8}$$

The following figure illustrates how $f(6)$ is broken down into sub problems.

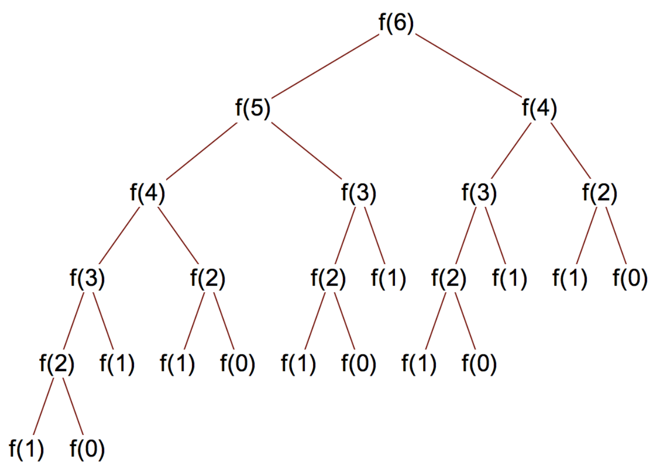


Figure 2.3: The 6:th fibonacci number broken down into sub-problems

One way to solve this problem is by simply calculating all values. However, one may notice two occurrences of for example $f(4)$. One on the left and one on the right.

By allocating some memory for calculations we only need to calculate $f(4)$ once. The calculated value of $f(4)$ is saved and reused when seeing $f(4)$ again. This is dynamic programming.

2.4 Shortest path algorithm

Algorithm 1 finds the shortest path from node s to node t for a specific type of graph. The graph must have the following properties.

1. Edges are directed.
2. May not contain cycles. A graph has a cycle if one can find a path where the same edge can be chosen an arbitrary number of times.
3. From each node the number of edges that must be traversed to reach the target node is known

The algorithm makes use of Bellman's principle of optimality on line 5. By storing $V(x)$ from line 5 one can save a lot of computations. This is where dynamic programming comes in to play.

Let s and t be the source and target node respectively. Let K be the number of edges between node s and node t.

Algorithm 1 Shortest path algorithm for directed graphs with no cycles and K edges between node s and node t

1: $V(x) = \infty$ for all $x \in$Nodes
2: $V(t) = 0$
3: **for** $k = 1, 2, ..., K - 1$ **do**
4:     **for** $x \in$ Nodes s.t. x is k edges from node t **do**
5:         $V(x) = \min_{y \in \Gamma(x)} \{F(x, y) + V(T(x, y))\}$
6:     **end for**
7: **end for**
8: $V(s) = \min_{y \in \Gamma(s)} \{F(s, y) + V(T(s, y))\}$

At every node "what is the cheapest way to reach node t?" is asked. Asking this question in a specific order and saving the answer is key idea of this algorithm.

2.5 An illuminating example

In this example we will apply algorithm 1 to the shortest path problem presented in figure 2.4.

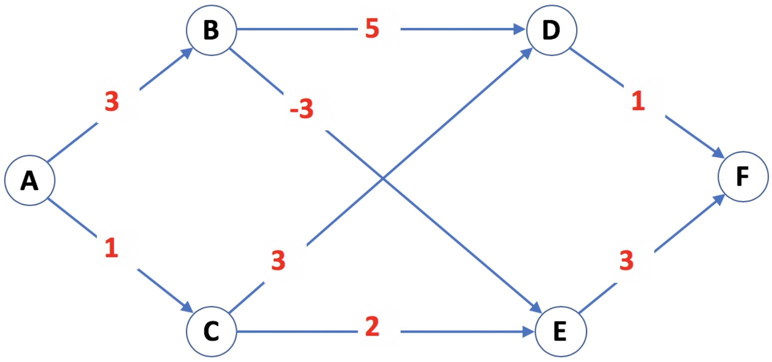


Figure 2.4: An example of the shortest path problem. Red numbers are edge costs. The objective is to find the shortest path from node A to node F.

The objective is to find an optimal path from node A to node F. Red numbers are edge costs. To understand the algorithm lets translate the notation and sets used in algorithm 1 to the problem at hand.

A state x is a node, $x \in \{A, B, C, D, E, F\}$. s is node A and t is node F. The set $\Gamma(x)$ is the set of nodes we can move to from node x, so $y \in \Gamma(x)$ is a node. $T(x, y)$ will be the next node if we take action y from node x. In this case: taking action y from nodes x means that we move to node y. Thus $T(x, y) = y$. $V(x)$ is the cost to node t from node x. $F(x, y)$ is the edge cost between node x and node y.

1. $V(x) = \infty$ for $x \in \{A, B, C, D, E, F\}$
2. $V(F) = 0$
3. $k = 1$ implies that we calculate $V(D)$ and $V(E)$. The possible set of actions from node D and E is $\Gamma(D) = \Gamma(E) = F$. Hence

$$V(D) = F(D, F) + V(T(D, F)) = 1 + V(F) = 1 \tag{2.9}$$

and

$$V(E) = F(E, F) + V(T(E, F)) = 3 + V(F) = 3 \tag{2.10}$$

4. $k = 2$ implies that we calculate $V(B)$ and $V(C)$. The possible set of actions from node B and C is $\Gamma(B) = \Gamma(C) = \{D, E\}$. Hence

$$V(B) = \min\{F(B, D) + V(T(B, D)), F(B, E) + V(T(B, E))\} \tag{2.11}$$
$$= \min\{F(B, D) + V(D), F(B, E) + V(E)\} \tag{2.12}$$
$$= \min\{6, 0\} = 0 \tag{2.13}$$

and

$$V(C) = \min\{F(C, D) + V(T(C, D)), F(C, E) + V(T(C, E))\} \tag{2.14}$$
$$= \min\{F(C, D) + V(D), F(C, E) + V(E)\} \tag{2.15}$$
$$= \min\{4, 5\} = 4 \tag{2.16}$$

5. Now look at the source node A. The possible set of actions from node A is $\Gamma(A) = \{B, C\}$. Hence

$$V(A) = \min\{F(A,B) + V(T(A,B)), F(A,C) + V(T(A,C))\} \tag{2.17}$$
$$= \min\{F(A,B) + V(B), F(A,C) + V(C)\} \tag{2.18}$$
$$= \min\{3, 5\} = 3 \tag{2.19}$$

Figure 2.5 is an illustration of the solution. The green parenthesis above the nodes contains the information $(V(x), y)$ where $V(x)$ is the cost to go to node F from node x and y is the next optimal node in the sequence of optimal nodes between x and F.

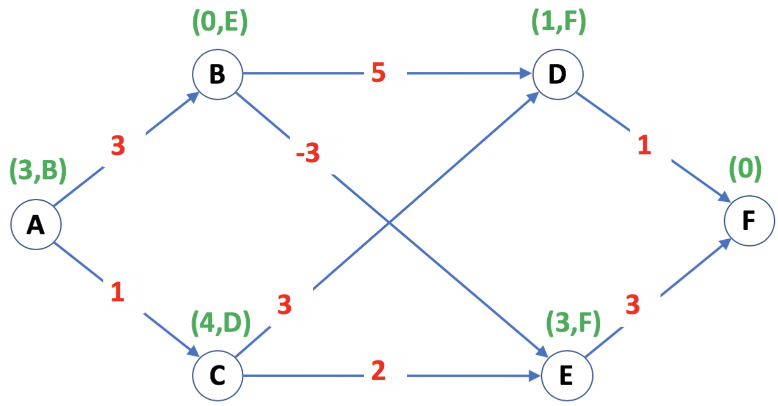


Figure 2.5: An example of the shortest path problem. The green markings above the nodes is the cost to node F and the next optimal node respectively.

It is now easy to find the optimal path from node A to node F by looking at the green markings above node A. The cheapest cost and optimal path is 3 and consists of the sequence (A, B, E, F).

The speed optimization problem is converted into a shortest path problem. However, in order to make this conversion a mathematical vehicle model is necessary.

3

Vehicle motion and Powertrain modeling

The purpose of this chapter is to give an overview of the physical relations describing vehicle motion and modeling of the systems included in the powertrain. These relations are necessary for determining the edge costs, see section 4.2. Additional and more detailed equations expressing the relations can be found in Appendix A.

3.1 Longitudinal motion model

Propulsion force, torque and power, $F_{\text{Propulsion}}, T_{\text{Propulsion}}, P_{\text{Propulsion}}$ are related to acceleration a, speed v and slope angle α.

$$F_{\text{Propulsion}} = F_{\text{Propulsion}}(a, v, \alpha) \tag{3.1}$$
$$T_{\text{Propulsion}} = T_{\text{Propulsion}}(a, v, \alpha) \tag{3.2}$$
$$P_{\text{Propulsion}} = P_{\text{Propulsion}}(a, v, \alpha) \tag{3.3}$$

From $P_{\text{Propulsion}}$ and auxiliary power need P_{Aux}, system power need P_{System} is derived by

$$P_{\text{System}} = P_{\text{Propulsion}} + P_{\text{Aux}} \tag{3.4}$$

More details regarding the longitudinal motion model can be viewed in appendix A.

3.2 Powertrain overview and constraints

As figure 3.1 illustrates, the powertrain is fully electric.

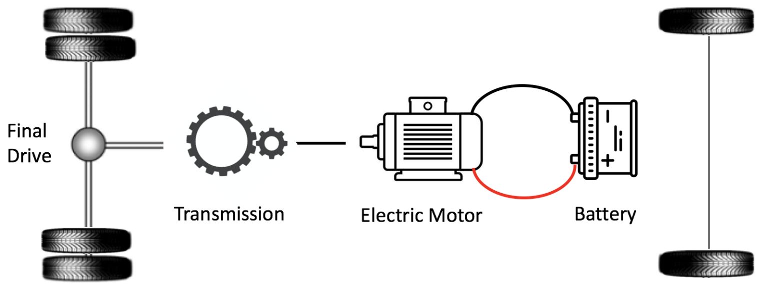


Figure 3.1: Powertrain layout

Mathematical modeling of the powertrain is needed to transform the power/torque request for a specific driving situation into terms that affects the operating cost of the vehicle. The bullets below expresses aspects assumed and/or considered for the modelling. More information about most statements is presented later in the chapter.

- The transmission ratio is constant, i.e. there is only one gear.
- Electric motor efficiency is constant but power flow direction dependent. A constant electric motor efficiency also makes the modelling of gear changes unnecessary.
- Mechanical brakes are actuated as little as possible but when absolutely necessary.
- A battery cell is modeled with a simple equivalent circuit model, it contains a constant voltage source in series with a resistor.
- The time derivative of battery state of health SoH is modeled.
- Battery power capability is dependent on battery state of charge SoC.
- Limitations of the system is handled by constraints, these are listed in table 3.1. These limitations mean that some driving situations become non feasible. For example will high acceleration not be possible due to restricted battery and/or electric machine power.

Ci	**Expression**	**Comment**
C1	$v_{\text{Min}} \leq v \leq v_{\text{Max}}$	Speed limits
C2	$a_{\text{Min}} \leq \dot{v} \leq a_{\text{Max}}$	Acceleration limits
C3	$\lvert F_{\text{Propulsion}} \rvert \leq F_{\text{FrictionMax}}$	Propulsion force is restricted by road friction
C4	$P_{\text{ElectricMotor}} \leq P_{\text{ElectricMotorMax}}$	Electric motor power is restricted
C5	$T_{\text{ElectricMotor}} \leq T_{\text{ElectricMotorMax}}$	Electric motor torque is restricted
C6	$\omega_{\text{ElectricMotor}} \leq \omega_{\text{ElectricMotorMax}}$	Electric motor revs is limited
C7	$T_{\text{BrakeMin}} \leq T_{\text{Brake}}$	Torque that the brakes can produce is limited
C8	$U_{\text{CellMin}} \leq U_{\text{Cell}} \leq U_{\text{CellMax}}$	Battery cell voltage is restricted
C9	$I_{\text{CellMin}} \leq I_{\text{Cell}} \leq I_{\text{CellMax}}$	Battery cell current is restricted

Table 3.1: Constraints expressing vehicle and Powertrain limitations

Note that the constraints in table 3.1 can be chosen for each position on a route.

For example v_{Max} can be different for different positions on the route.

3.3 Electric motor and brake model

The electric motor is a component that converts electrical energy into mechanical energy. The power through the electric motor is modeled by a constant but propulsion sign dependent efficiency $\phi_{\text{ElectricMotor}}$, or mathematically

$$\dot{E}_{\text{ElectricMotor}} = P_{\text{ElectricMotor}}\phi_{\text{ElectricMotor}}(T_{\text{Propulsion}}). \tag{3.5}$$

where $\dot{E}_{\text{ElectricMotor}}$ is the power required from the electric motor including energy loss while $P_{\text{ElectricMotor}}$ is the power required from the electric motor excluding the energy loss.

Brake torque actuation can be seen as an additional control freedom in the system. Therefore, logic describing the brake torque actuation is needed. The reasoning for such a logic is that brake usage shall be minimized because braking means loosing power that can be used for recharging the battery. By the optimization problem described in equation 3.6 and Table 3.2 the mechanical brakes will be used as little as possible.

$$\max \quad T_{\text{Brake}} \tag{3.6}$$

T_{Brake} should be maximized because T_{Brake} is negative when the brakes are applied and zero else. Maximizing T_{Brake} will then result in minimal brake usage.

Ci	Expression	Comment
C1	$T_{\text{Brake}} \leq 0$	Braking torque is always negative
C2	$T_{\text{ElectricMotor}} \geq T_{\text{ElectricMotorMin}}$	Relevant when the electric motor is acting as generator.
C3	$P_{\text{ElectricMotor}} \geq P_{\text{ElectricMotorMin}}$	Relevant when the electric motor is acting as generator.
C4	$P_{\text{Battery}} \geq P_{\text{BatteryMin}}$	There is a lower limit on battery power regeneration

Table 3.2: Constraints in optimization problem for brake actuation logic.

The results of this optimization problem is then used to calculate 3.5. More details about electric motor and brake modelling can be found in Appendix A.

3.4 Battery model

The battery is assumed to be built up from a discrete number of battery packs, typically 1-4 packs is used in vehicles. Each pack includes a number of cells in series, typically in the order of a few hundreds, see figure 3.3

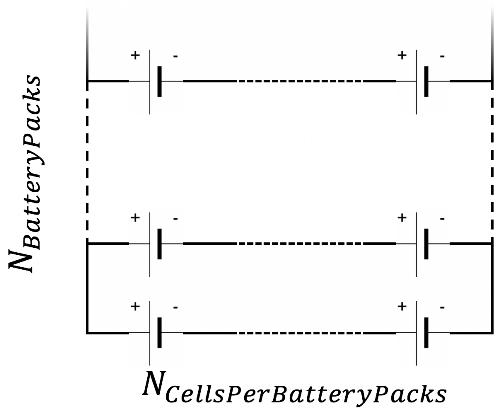


Figure 3.3: Battery model

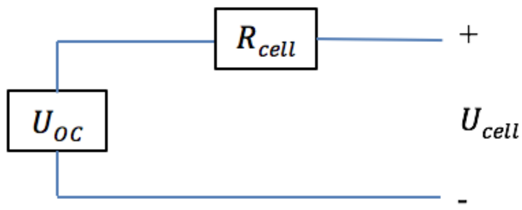


Figure 3.4: Battery cell model, where U_{OC} is nominal voltage

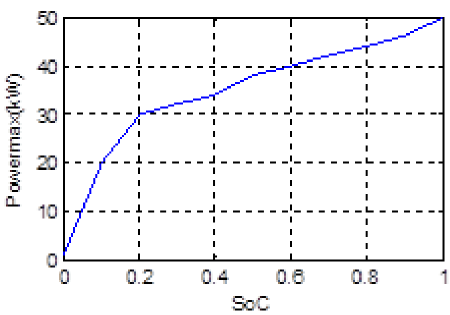
The power through the battery is modeled by

$$\dot{E}_{\text{Battery}} = P_{\text{Battery}} \phi_{\text{Battery}} \tag{3.7}$$

$$P_{\text{Battery}} = \dot{E}_{\text{ElectricMotor}} + P_{\text{Aux}} \tag{3.8}$$

where $\dot{E}_{\text{Battery}}$ is the power required from the electric motor including energy loss while P_{Battery} is the power required from the electric motor excluding the energy loss. ϕ_{Battery} is the battery efficiency. It is described and calculated in A. P_{Aux} is the power required from running electrical components in the vehicle such as lights.

It is possible to take and give energy to the battery. Charging a battery is a bit like filling up a balloon where the air in the balloon represents energy. If the battery is fully charged the state of charge(SoC) is 1. In this state we can get the highest amount of power from the battery. However, it also means that the battery can not be charged any more. On the opposite end, if the battery has no charge($SoC = 0$), no power can be generated from it. However, the battery can be charged with a higher power. Figure 3.2 shows the relationsship between $P_{\text{BatteryMin}}$ and $P_{\text{BatteryMax}}$ vs SoC.



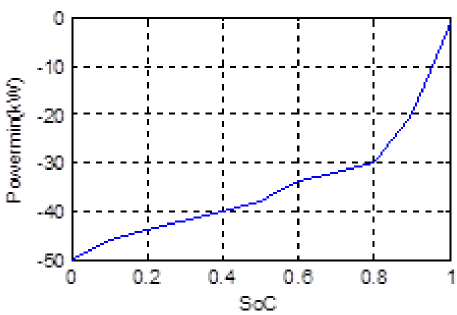


Figure 3.2: Relationsship between $P_{\text{BatteryMin}}$ and $P_{\text{BatteryMax}}$ vs SoC.

In this work the battery cell model variant, often called R_{Internal} and illustrated in Figure 3.4, has been used. The model implies, for example, that when power is taken

from the battery, the cell terminal voltage U_{Cell} decreases due to the cell resistance R_{cell}. The ohmic power losses $I_{\text{Cell}}^2 \cdot R_{\text{Cell}}$ can be regarded as wasted energy. This model is used to calculate the battery efficiency in appendix A.
The battery degradation model can be summarized by

$$\Delta SoH = \dot{SoH}(P_{\text{Cell}}, N_{\text{Max}}(P_{\text{Cell}}))\Delta t \tag{3.9}$$

The equation prescribes that for a specific cell power P_{Cell} and change in time Δt the change in battery state of health is ΔSoH. Figure 3.5 illustrates how the maximum number of full cycles depends on cell power. A typical value of the maximum number of full cycles N_{Max} is 5 k. A reflection from the figure is that loading a battery with a high power will degrade it quicker.

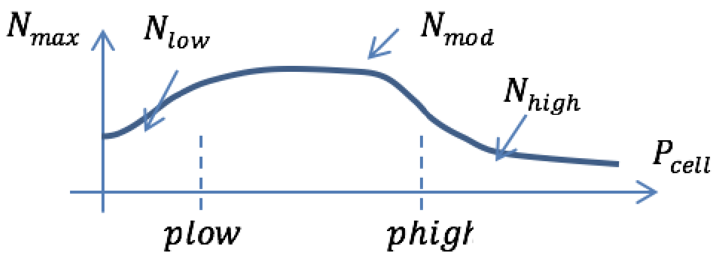


Figure 3.5: Schematic plotting of N_{Max} function

More details about battery modelling can be found in Appendix A

4

Graph Construction and Optimization

The purpose of this chapter is to give a detailed explanation of the graph followed by the algorithm that finds the optimal velocity profile. Understanding the graph, i.e. nodes and edges, described below is a vital part of understanding the algorithm that gives the optimal velocity.

4.1 Nodes

Consider a matrix of size $m \times n$ where each element is a node. Each node is defined by a velocity and a position. The row index indicates which velocity the node has and the column index indicates which position it has. This implies that there are n different distance points. For each distance point there are m possible velocities. An increasing row index corresponds to an increasing velocity. An increasing column index corresponds to an increasing distance from the initial position. The set of distance points is defined as

$$D = \{d_1, d_2, ..., d_n\}.$$

A position d_i is associated with a specific known height h_i for $i = 1, ..., n$. The set of velocities is defined as

$$V = \{v_1, v_2, ..., v_m\}.$$

A node x is defined by its velocity v_x and its distance point d_x. The set of nodes is defined as

$$\text{Nodes} = V \times D \tag{4.1}$$

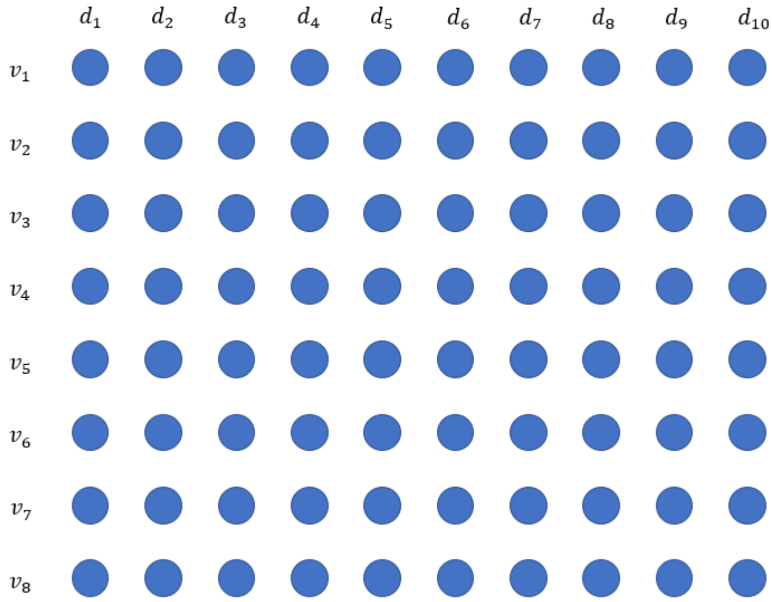


Figure 4.1: The graphical illustration of a set of nodes where there are eight different velocities and 10 different distance points.

4.2 Edges

It is only possible to move from the distance point d_j to distance point d_{j+1} for all $j = 1, ..., n-1$. d_{j+1} is defined as a successor to d_j and d_j is defined as a predecessor to d_{j+1} for all $j = 1, ..., n-1$. Moreover, node y is said to be a successor to node x if the distance point of node y is a successor to the distance point of node x. The distance, height and velocity point of a node x is denoted d_x, h_x and v_x respectively. The distance between node x and node y is

$$d_y - d_x. \tag{4.2}$$

The height difference between node x and node y is

$$h_y - h_x. \tag{4.3}$$

and the slope angle between node x and node y is

$$\alpha = \arctan(\frac{h_y - h_x}{d_y - d_x}). \tag{4.4}$$

With these definitions it is possible to calculate the necessary functions to determine if the constraints in table 3.1 are satisfied. If they are satisfied then there exists an edge between node x and node y. The set of edges can then be defined as

$$\text{Edges} = \{(x, y) \in \text{Nodes}^2, \tag{4.5}$$
$$d_y \text{ is a successor to } d_x, \tag{4.6}$$
$$\text{and all constraints from table 3.1 are satisfied}\} \tag{4.7}$$

To move a vehicle with a velocity v_x at position d_x to a new velocity and position v_y and d_y is associated with a certain cost defined as the edge cost $c_{\text{edge}}(x, y)$. The edge cost consists of a time cost $c_{\text{time}}(x, y)$ and an operation cost $c_{\text{operation}}(x, y)$.

$$c_{\text{edge}}(x, y) = \omega \cdot c_{\text{time}}(x, y) + (1 - \omega) \cdot c_{\text{operation}}(x, y) \tag{4.8}$$

where $\omega \in [0, 1]$ is a coefficient to set the priority between time and operation costs. The time cost is how long time it takes to move from node x to node y. The operation cost can consist of any cost that can be mathematically defined. However, to limit the scope of this thesis, the operation cost $c_{\text{operation}}(x, y)$ consists of an energy cost $c_{\text{energy}}(x, y)$ and a state of health cost $c_{\text{stateOfHealth}}(x, y)$. The state of health cost is the wear and tear cost of using the battery.

$$c_{\text{operation}}(x, y) = c_{\text{energy}}(x, y) + c_{\text{stateOfHealth}}(x, y) \tag{4.9}$$

Since there is a total of three different types of costs, there are three different types of units to consider. The unit of energy (Joule) cannot be compared with the unit of time (seconds). Hence, each type of cost z must be normalized with the factor N_z.

$$c_{\text{time}}(x, y) = \frac{1}{N_{\text{time}}} \text{time}(x, y) \tag{4.10}$$

$$c_{\text{energy}}(x, y) = \frac{1}{N_{\text{energy}}} \text{energy}(x, y) \tag{4.11}$$

$$c_{\text{stateOfHealth}}(x, y) = \frac{1}{N_{\text{stateOfHealth}}} \text{StateOfHealth}(x, y). \tag{4.12}$$

where $\text{time}(x, y)$ is time in seconds, $\text{energy}(x, y)$ is the energy cost in *Euro* and $\text{StateOfHealth}(x, y)$ is the state of health cost in *Euro*. The normalization factor N_z for cost z can be chosen in several ways. Define the following sets

$$C_{\text{time}} = \{\text{time}(x, y) : (x, y) \in \text{Edges}\} \tag{4.13}$$

$$C_{\text{energy}} = \{\text{energy}(x, y) : (x, y) \in \text{Edges}\} \tag{4.14}$$

$$C_{\text{stateOfHealth}} = \{\text{stateOfHealth}(x, y) : (x, y) \in \text{Edges}\}. \tag{4.15}$$

Then, the normalization factors are chosen as following

$$N_{\text{time}} = \text{median}\{x \in C_{\text{time}} : x \neq 0\} \tag{4.16}$$

$$N_{\text{energy}} = \text{median}\{x \in C_{\text{energy}} : x \neq 0\} \tag{4.17}$$

$$N_{\text{stateOfHealth}} = \text{median}\{x \in C_{\text{stateOfHealth}} : x \neq 0\}. \tag{4.18}$$

Now the time, energy and state of health can be defined. Constant acceleration is assumed between node x and y if node y is a successor to node x. The velocity function $v(t)$ and total time $\text{time}(x, y)$ required between node x and node y is then found by solving the following equation and differential equation.

$$d_y - d_x = \int_0^{\text{time}(x,y)} v(t)\text{d}t \tag{4.19}$$

$$v'(t) = a = \frac{v_y - v_x}{\text{time}(x, y)}, v(0) = v_x. \tag{4.20}$$

Solving for v and $\mathrm{time}(x, y)$ gives

$$v(t) = at + v_x \tag{4.21}$$

$$\mathrm{time}(x, y) = 2\frac{d_y - d_x}{v_y - v_x}. \tag{4.22}$$

The total price in euro for taking energy from the battery $\mathrm{energy}(x, y)$ to go from node x to node y is defined as

$$\mathrm{energy}(x, y) = p_{\mathrm{Energy}} \int_0^{\mathrm{time}(x,y)} \dot{E}_{\mathrm{Battery}} \mathrm{d}t \tag{4.23}$$

where p_{Energy} is the price for energy and $\dot{E}_{\mathrm{Battery}}$ is defined in chapter 3.

The total price in euro for wear and tear on the battery $\mathrm{stateOfHealth}(x, y)$ to go from node x to node y is defined as

$$\mathrm{stateOfHealth}(x, y) = p_{\mathrm{Battery}} \int_0^{\mathrm{time}(x,y)} \dot{\mathrm{SoH}} \mathrm{d}t \tag{4.24}$$

where p_{Battery} is the total battery price and $\dot{\mathrm{SoH}}$ is defined in chapter 3.

Figure 4.2 shows an example of the feasible edges(green) and non feasible edges(red) from one node. An edge is feasible if all the constraints from section 3.1 hold. Else the edge is not feasible. There is an upper limit to how many edges that are feasible. Figure 4.2 has $n = 10$ distance points and $m = 8$ velocity points. The maximum number of feasible edges is then $(n - 1) \cdot m^2 = 576$. This means that 576 edge costs must be calculated and checked if they are physically feasible. However, that is a toy example. For a real route one may have distance points every ten meters for a route of one kilometer and setting velocity points in the range from 0 to 20 with step size 0.1. This gives $m = 200$, $n = 100$ and a total number of $m \cdot n \cdot m = 4 \cdot 10^6$ edges. Calculating this many edge costs takes some time. However, this can be done in a pre processing stage.

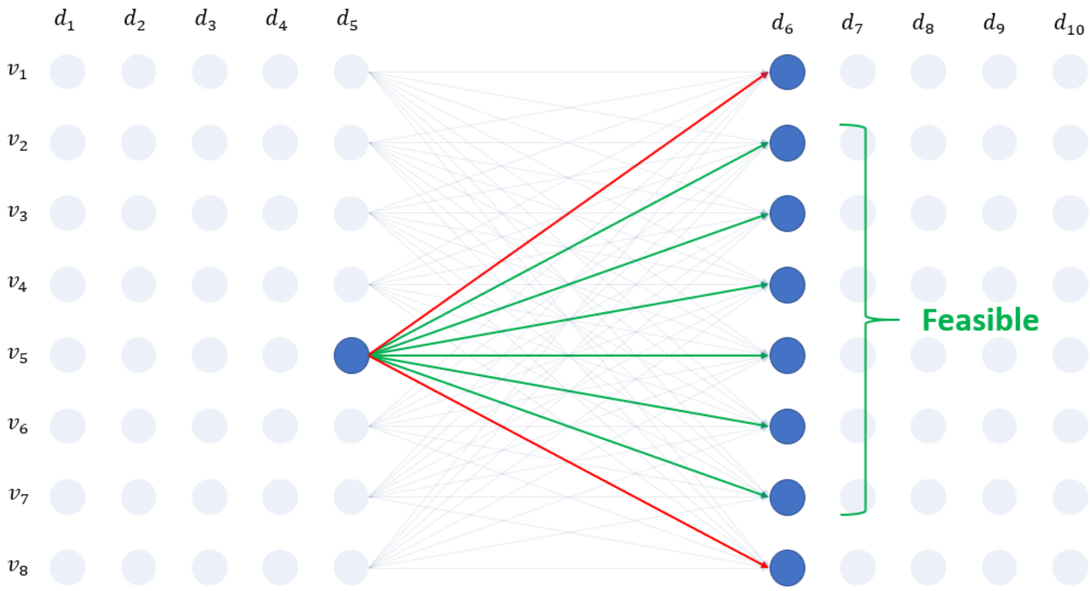


Figure 4.2: This is an example of the set of feasible(green) and non feasible(red) edges from a node. An edge is feasible if all the constraints from section 3.1 hold. Else the edge is not feasible.

To summarize this section: For every edge the time [s], energy [euro] and battery wear and tear [euro] costs are calculated as well as if the edge is feasible or not. The edge costs are then calculated by normalizing time, energy and battery wear and tear costs and then combining these into a single scalar value with no unit.

4.3 Trajectory Optimization

Let a state be a node. Recall that a node x is defined by a velocity v_x and a distance point d_x. Let $s = (v_s, d_s)$ and $t = (v_t, d_t)$ be the source and target node respectively. Algorithm 1 will then find the optimal and feasible velocity at each distance point from the source node to the target node. Denote the distance points between node s and node t as $d_s = d_1, ..., d_n = d_t$. $V(x)$ will denote the cost to go from node x to the target node t. Moreover, the set of feasible nodes to go to from node x is defined by the set

$$\Gamma(x) = \{y \in \text{Nodes} : (x, y) \in \text{Edges}\}. \tag{4.25}$$

This set represents the end nodes of the green edges in figure 4.2. If action y is chosen from node x then the next node is y. Hence,

$$T(x, y) = y. \tag{4.26}$$

The cost of taking action y at node x is

$$F(x, y) = c_{\text{edge}}(x, y). \tag{4.27}$$

By applying algorithm 1 to the graph defined in the previous sections for a given route, an optimal velocity profile is efficiently generated. Figure 4.3 shows an example of this.

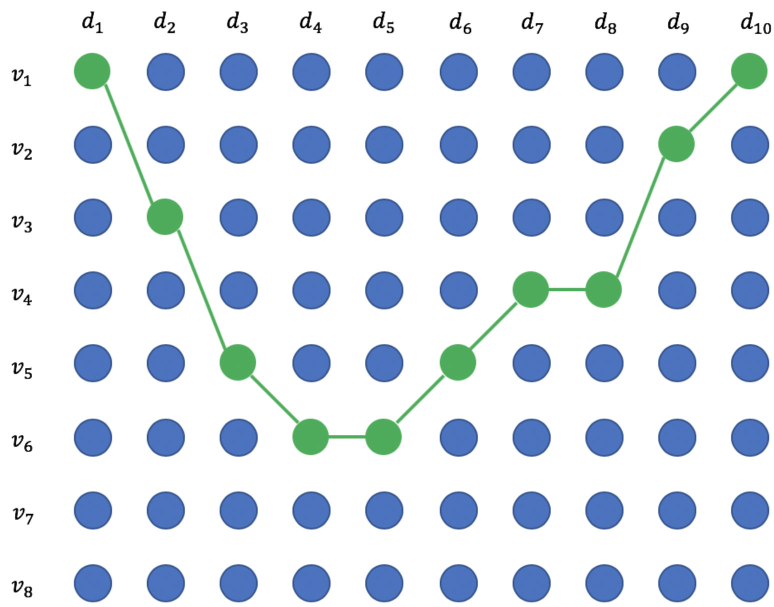


Figure 4.3: A graphical example of the optimal(green) nodes chosen by the algorithm. The source node is $s = (v_1, d_1)$ and the target node is $t = (v_1, d_{10})$. The optimal velocity profile from this figure would be $(v_1, v_3, v_5, v_6, v_6, v_5, v_4, v_4, v_2, v_1)$.

5
Result

The purpose of this chapter is to test the algorithm and the vehicle model to make sure that appropriate velocity profiles are generated. Unless anything else is stated, a default set of vehicle parameters are used and it is defined in Appendix B. For example the default mass of the vehicle is $m = 30$ $[kkg]$. The default step size for velocity and distance step size is $dv = 0.1$ $[m/s]$ and $dx = 10$ $[m]$ respectively. Any route has sections of flat, uphill and downhill sections. To make sure that the model and optimization algorithm is behaving well, three test routes have been created: route 1, 2 and 3. If the algorithm can give reasonable results on each test route it is also reasonable to assume that it can handle more complex routes which consists of several variations of the test routes. Moreover, a time performance analysis of the algorithm can be found in Appendix A. One thing all routes have in common is that no optimal speed exceeds 50 $[km/h] = 13.9$ $[m/s]$. This is because there is a limit for the angular velocity of the electric motor. The default maximum angular velocity of the electric motor $\omega_{\text{ElectricMotorMax}} = 340$ $[rad/s]$ is chosen because this is what the VERA vehicle is capable of.

5.1 Route 1 - Flat

This route is designed to analyze the vehicle behaviour on a flat route. The altitude and slope angle α can be seen in figure 5.1

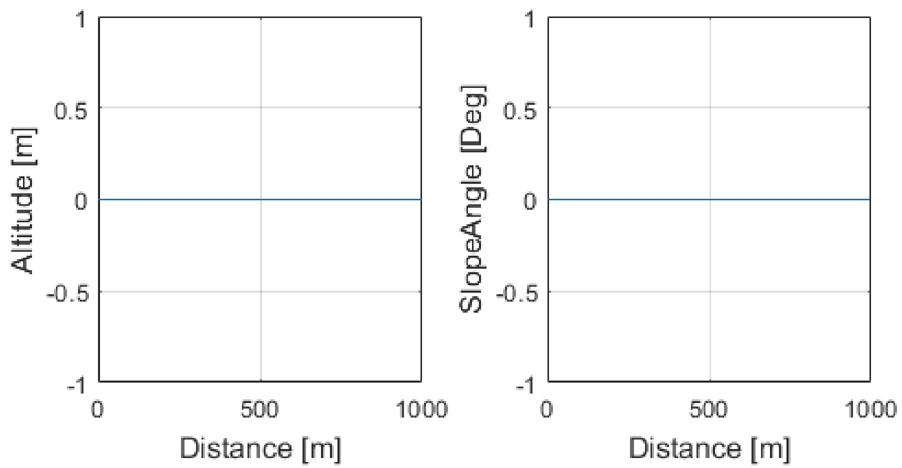


Figure 5.1: The flat route. Altitude profile to the left and slope angle α on the right

The optimal velocity generated for this route can be seen in figure 5.2. If one looks at the plot for $\omega = 1$, which means that time is solely prioritized, we see that the vehicle initially accelerates as much as physically possible until it reaches the chosen maximum velocity of 13.9 $[m/s]$. The maximum velocity is then maintained and at the end applies the maximum braking power at the end to save time. For $\omega = 0.5$ the vehicle initially accelerates maximally but instead of reaching maximum velocity it finds an optimal velocity to maintain. in order to save on operation costs.

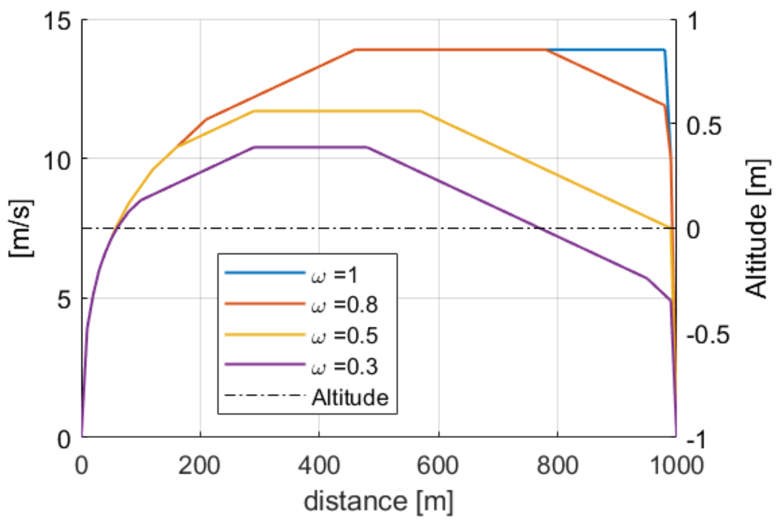


Figure 5.2: Optimal velocity on a flat route for a few different values of ω

In table 5.1 the total time in seconds and operation costs(energy cost and state of health) in euro can be seen. For high omega, time is low while operation costs are high. For low omega time is high and operation costs are low. Which is the desired behavior.

ω	Time [s]	Energy cost [€]	Batt. degr. cost [€]
1	89	0.15	0.25
0.8	90	0.13	0.25
0.5	107	0.09	0.19
0.3	129	0.07	0.15

Table 5.1: Total time and total operation(energy and state of health) costs on a flat route for a few different values of ω

In figure 5.3 one can see the Pareto front. The plot gives a graphical illustration of the trade off between operation cost and time for different ω. One can for example see that using $\omega = 0.3$ instead of $\omega = 1$ will increase time by 45% ut reduce operation costs by 45%.

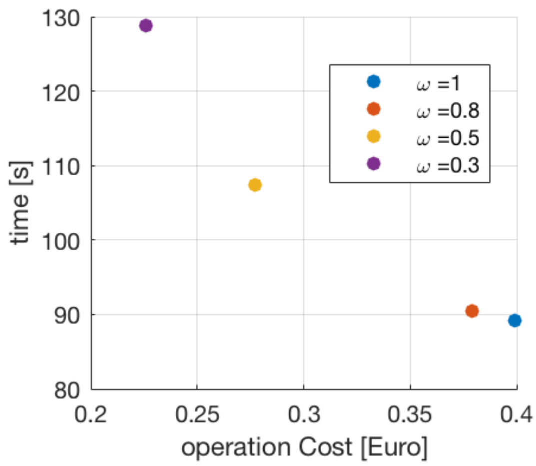


Figure 5.3: Pareto front. Total Operation Cost vs total Time.

Figure 5.4 shows the torque and power of the electric motor, torque for the brakes and the power required from the battery for $\omega = 1$. The plot is used to make sure the vehicle properties are behaving appropriately. One reflection is that battery power is limiting when accelerating.

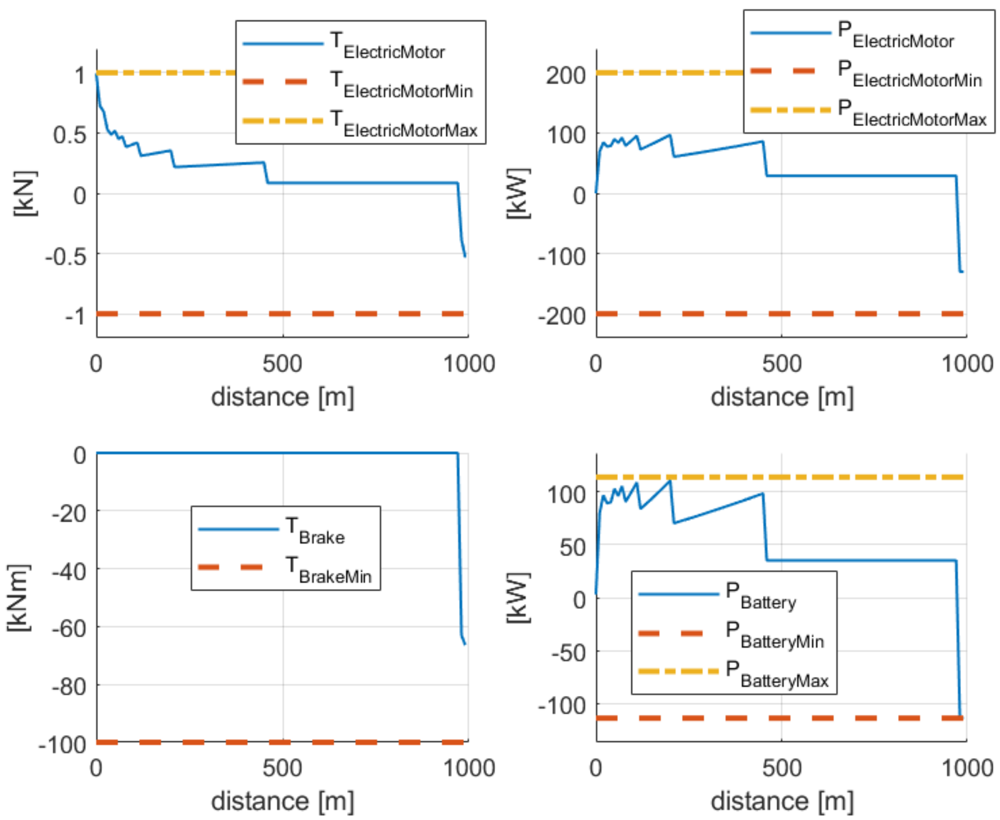


Figure 5.4: Torque and power for the electric motor, torque for the brakes and power required from the battery for $\omega = 1$

5.2 Route 2 - Up hill

This route is designed to analyze the vehicle behaviour when approaching a positive slope. The route altitude and slope angle α can be seen in figure 5.5.

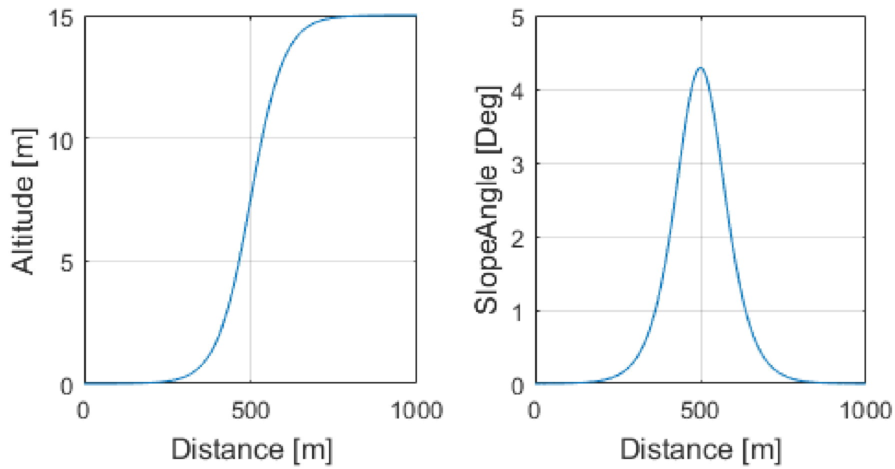


Figure 5.5: The uphill route. Altitude profile to the left and slope angle α on the right

One would expect a heavy vehicle such as a truck to not be able to maintain velocity on a steep slope because of some limitations such as battery power. This is exactly what can be seen in figure 5.6. Looking at the optimal velocity profile for $\omega = 1$ the vehicle accelerates as much as physically possible before approaching the steep part of the slope. In the steeper parts of the slope the vehicle slows down because the vehicle cannot produce enough energy to maintain its velocity. When the vehicle has overcome the steep part it starts accelerating again and at the end the brakes are applied to save time.

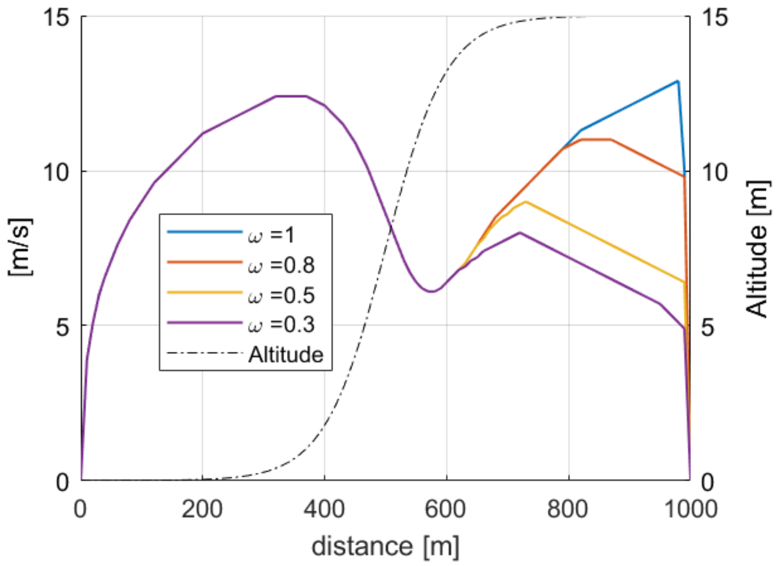


Figure 5.6: Optimal velocity on a up hill route for a few different values of ω

Similar behaviour can be seen for the other ω. However, for lower ω the algorithm chooses to not accelerate as much in order to save on operation costs. But heavy braking is applied in the end in order to save on the time cost.

Table 5.2 shows that ω is doing an appropriate priority of time vs operation costs. For high omega time is low while operation costs are high. For low omega time is high and operation costs are low. Which is the desired behavior.

ω	Time [s]	Energy cost [€]	Batt. degr. cost [€]
1	111	0.28	0.42
0.8	113	0.25	0.38
0.5	123	0.22	0.33
0.3	131	0.21	0.33

Table 5.2: Total time and total operation(energy and state of health) costs on an up hill route for a few different values of ω

In figure 5.7 one can see the Pareto front. The plot gives a graphical illustration of the trade off between operation cost and time for the different ω. One can for example see that using $\omega = 0.3$ instead of $\omega = 1$ will increase time by 18% but reduce operation costs by 23%.

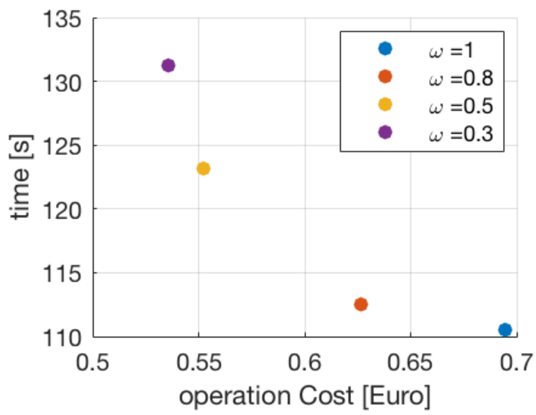


Figure 5.7: Pareto front. Total Operation Cost vs total Time.

In figure 5.8 The torque and power of the electric motor is plotted as well as the power required from the battery for $\omega = 1$. One would hope to see a smoother plot for these properties. Why this occurs remains to be researched properly. Never the less, these graphs show what stops the vehicle from doing something that is not physically possible. For this case, the battery maximum battery power is setting the limits.

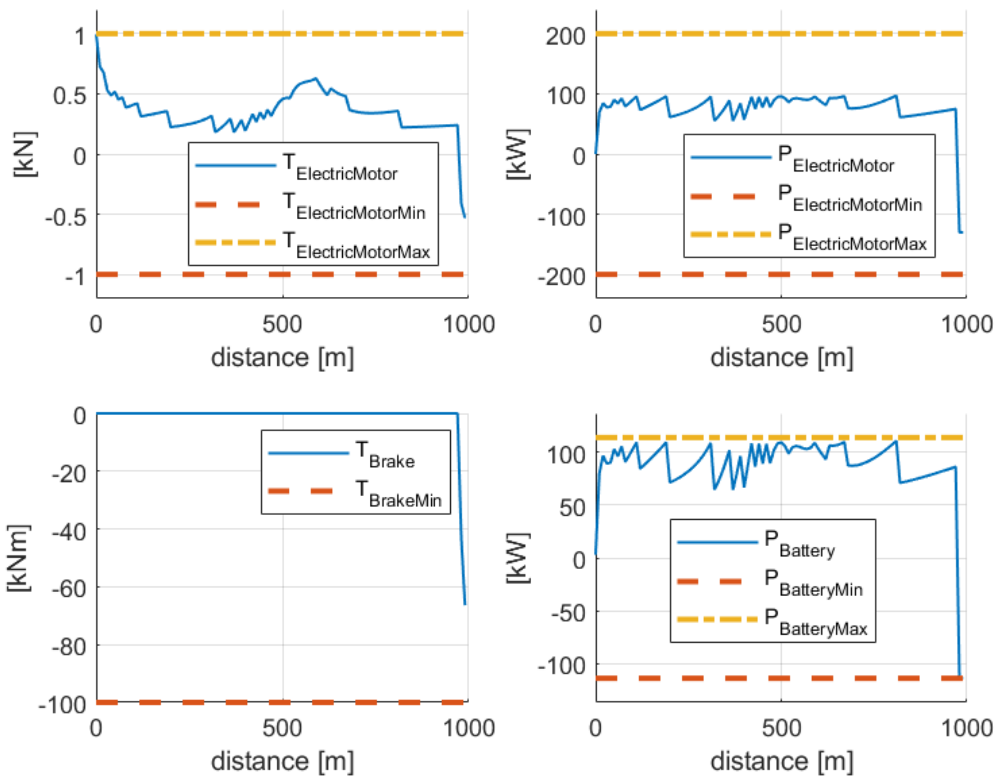


Figure 5.8: Torque and power for the electric motor and power required from the battery for $\omega = 1$

5.3 Route 3 - Down hill

This route is designed to analyze the vehicle behaviour when approaching a negative slope. The route altitude and slope angle α can be seen in figure 5.9

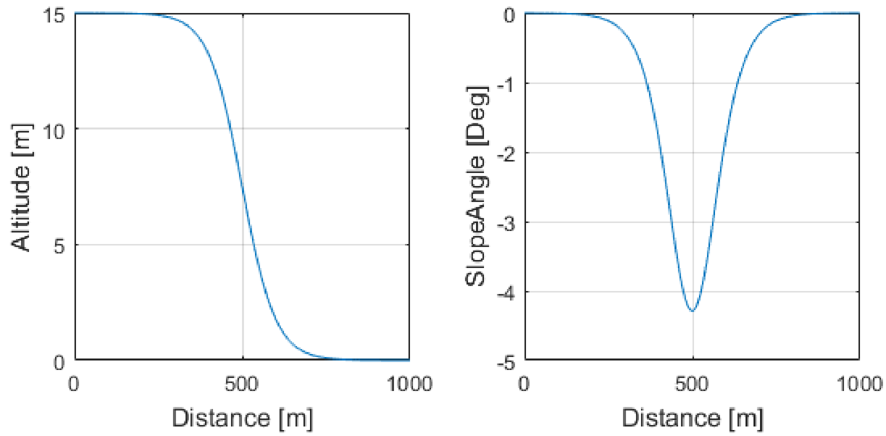


Figure 5.9: The downhill route. Altitude profile to the left and slope angle α on the right

In this route one would expect that the vehicle accelerates by electric power to begin with. Followed by gravity taking over to accelerate the vehicle down the slope where it will free roll to the end of the route and brake at the end. Looking at figure 5.10 confirms this behavior.

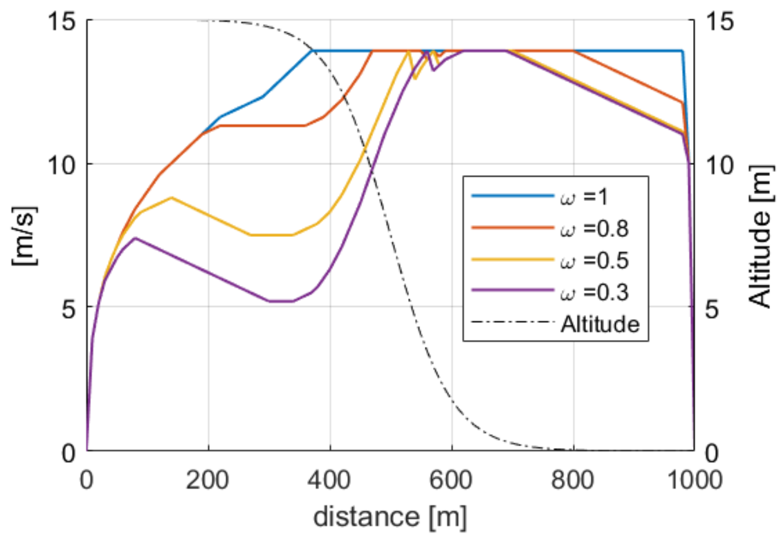


Figure 5.10: Optimal velocity on a down hill route for a few different values of ω

Table 5.3 shows that ω is doing an appropriate priority of time vs operation costs.

ω	Time [s]	Energy cost [€]	Batt. degr. cost [€]
1	89	0.064	0.35
0.8	92	0.035	0.26
0.5	106	0.009	0.18
0.3	123	-0.004	0.15

Table 5.3: Total time and total operation(energy and state of health) costs on x route for a few different values of ω

An interesting thing to note is that for $\omega = 0.3$ the vehicle is actually gaining energy which is indicated by the negative energy cost. One can for example see that using $\omega = 0.3$ instead of $\omega = 1$ will increase time by 38% but reduce operation costs by 65%.

In figure 5.11 one can see the Pareto front. The plot gives a graphical illustration of the trade off between operation cost and time for the different ω.

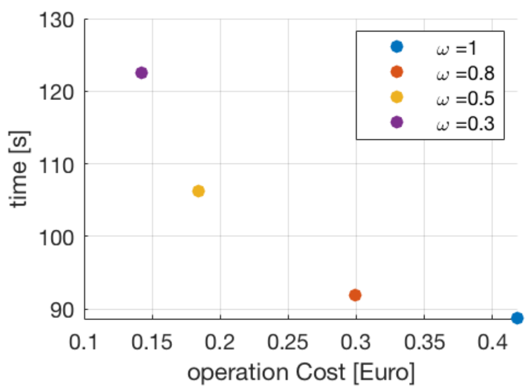


Figure 5.11: Pareto front. Total Operation Cost vs total Time.

To prove that before the slope the power comes from the battery and in the down hill the acceleration comes from gravity while the battery is regaining energy take a look at 5.12. Here we see that P_{Battery} is positive before and after the slope and negative in the slope for $\omega = 1$. This shows that the battery is gaining energy when going down hill. Moreover, the velocity is growing as its going down hill. The reason for this can only be that gravity is working as expected. At the steepest part of the slope the gravity is trying to accelerate the vehicle beyond its limitations, so the brakes have to be applied. This is why the little bump occurs for the brake torque at $500[m]$.

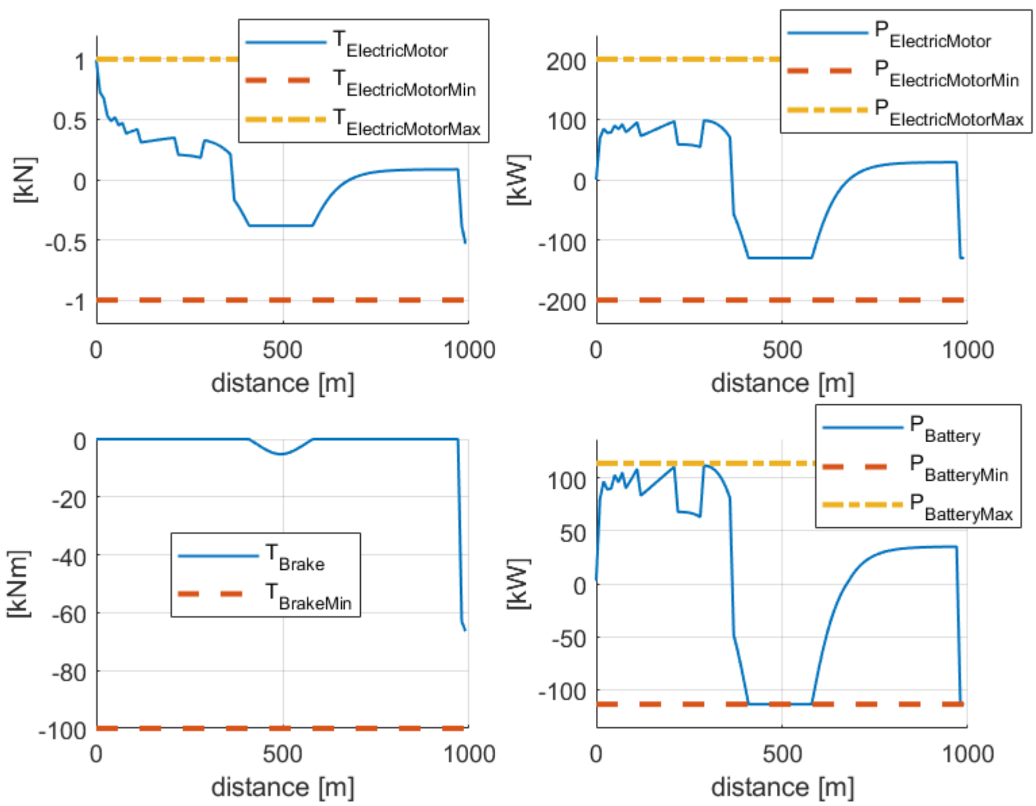


Figure 5.12: Torque and power for the electric motor and power required from the battery for $\omega = 1$.

5.4 Route 4 - Constant down hill

This route is designed to analyze the vehicle behaviour when going down hill with a constant slope angle. The route altitude and slope angle α can be seen in figure 5.13

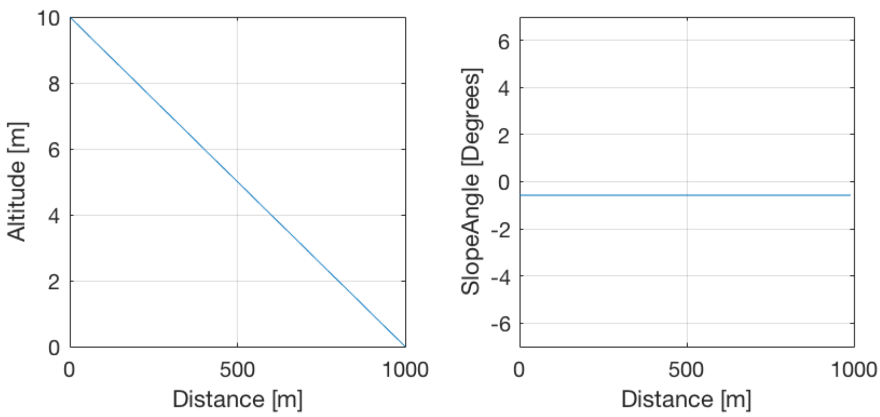


Figure 5.13: The constant downhill route. Altitude profile to the left and slope angle α on the right

The expected behavior is that the vehicle accelerates to a fixed speed and takes on a constant energy regeneration rate for some ω. The optimal velocity for a few different ω can be seen in figure 5.14 and table 5.4. The vehicle starts bye accelerating and for $\omega = 0.5$ and $\omega = 0.3$ the algorithm find an optimal constant velocity to maintain.

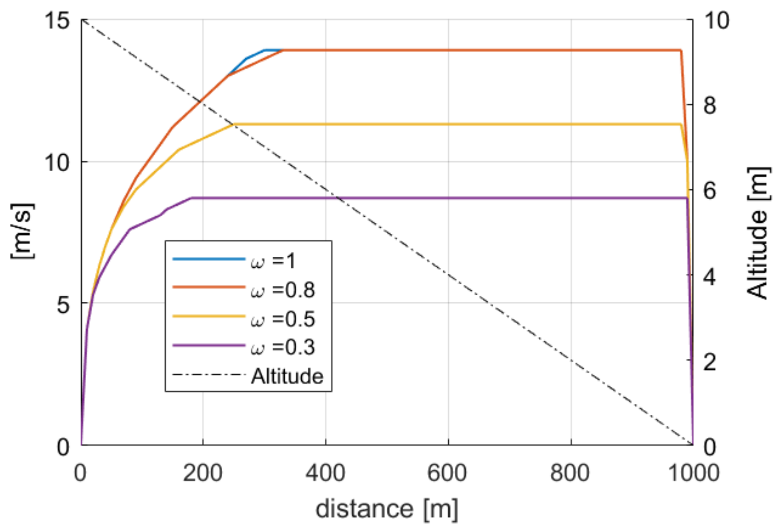


Figure 5.14: Optimal velocity on a constant down hill route for a few different values of ω

Table 5.3 shows how ω is prioritizing between time and operation costs. Even though there is a constant down hill, the battery is not gaining energy because initially it will need to spend energy accelerating the vehicle to the optimal velocity. However, if the vehicle were to start at the optimal velocity a negative energy cost would occur.

ω	Time [s]	Energy cost [€]	Batt. degr. cost [€]
1	86	0.066	0.168
0.8	86	0.066	0.165
0.5	99	0.032	0.138
0.3	124	0.008	0.111

Table 5.4: Total time and total operation(energy and state of health) costs on constant down hill route for a few different values of ω

In figure 5.15 one can see the Pareto front. The plot gives a graphical illustration of the trade off between operation cost and time for the different ω. One can for example see that using $\omega = 0.3$ instead of $\omega = 1$ will increase time by 44% but reduce operation costs by 49%.

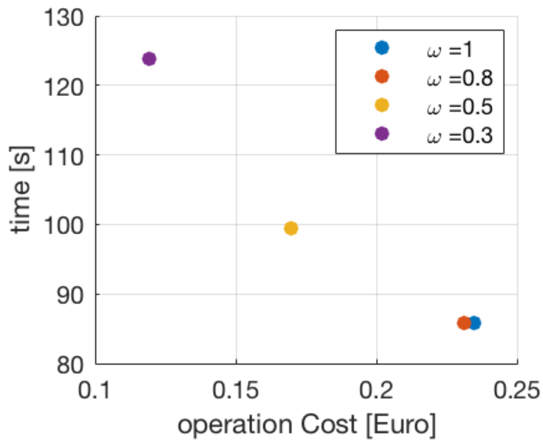


Figure 5.15: Pareto front. Total Operation Cost vs total Time.

Figure 5.16 shows the torque and power for the electric motor and power required from the battery for $\omega = 1$. It shows that the battery is regaining energy when the constant optimal velocity is found.

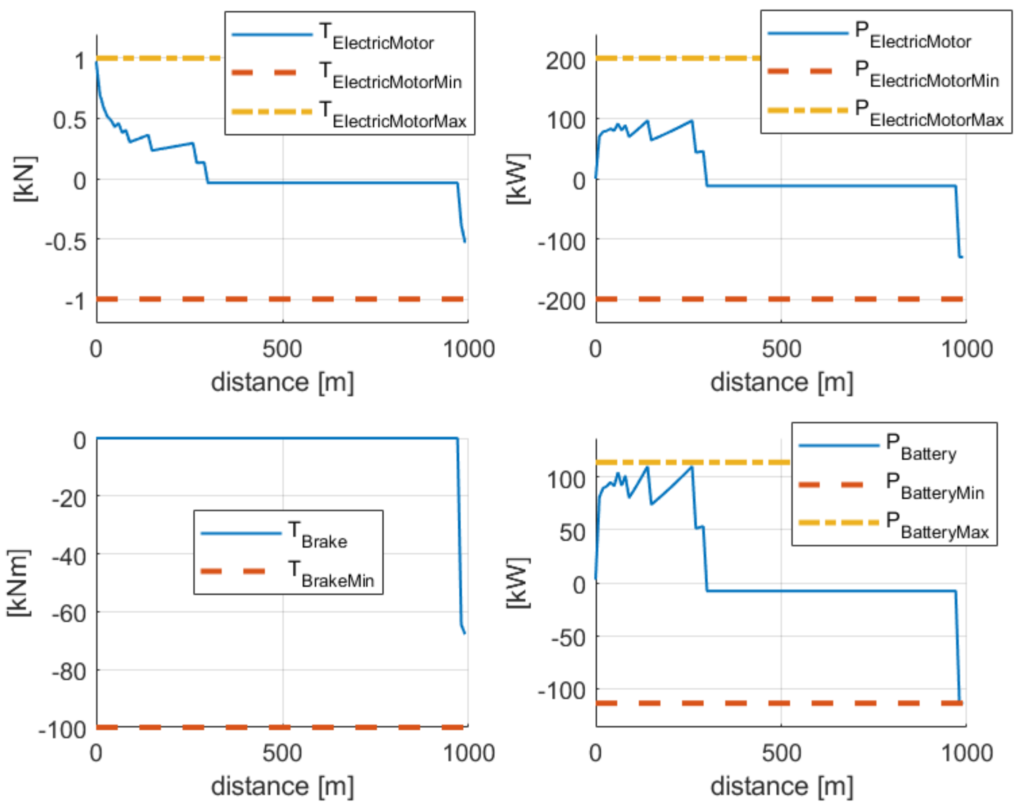


Figure 5.16: Torque and power for the electric motor and power required from the battery for $\omega = 1$.

5.5 Route 5 - The Vera Route

This is a route that exists in the real world. The route altitude and slope angle α can be seen in figure 5.17

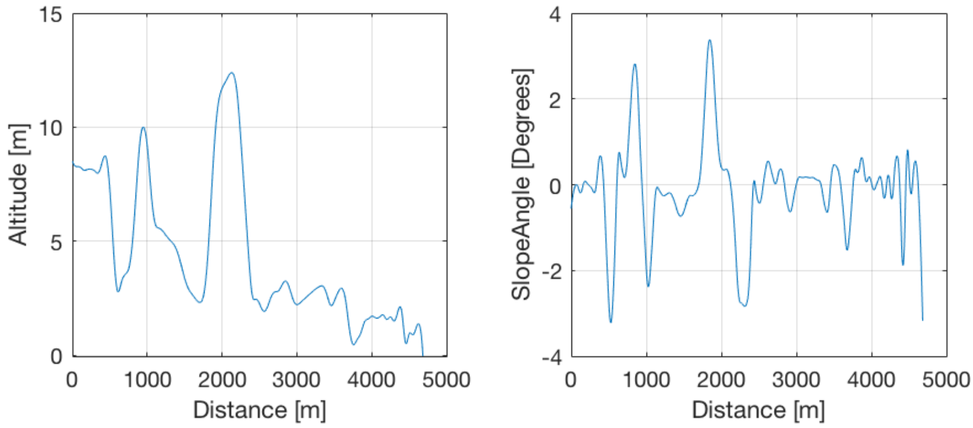


Figure 5.17: A route that exists in the real world, a.k.a. "The Vera route". Altitude profile to the left and slope angle α on the right.

The optimal speed profile can be seen in figure 5.18. It is hard to analyze each individual part of this route to make sure the a correct behavior is being generated.

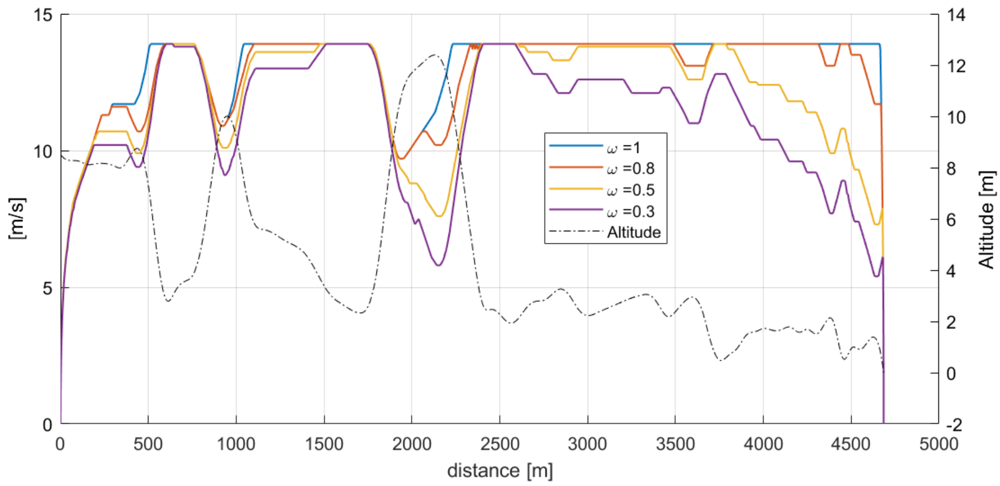


Figure 5.18: The optimal velocity profile for a few different ω

ω	Time [s]	Energy cost [€]	Batt. degr. cost [€]
1	367	0.39	0.98
0.8	374	0.33	0.76
0.5	405	0.28	0.62
0.3	447	0.26	0.53

Table 5.5: Total time and total operation(energy and state of health) costs on x route for a few different values of ω

However, this result is definitely non trivial and based on the previous test routes it is reasonable to say that this result is not unrealistic. Moreover, from table 5.5 one can conclude that the total energy consumed for $\omega = 1, 0.8, 0.5, 0.3$ is $3.9, 3.3, 2.8, 2.6$ [kWh] respectively. This gives an energy consumption per kilometer of $0.83, 0.70, 0.60, 0.55$ respectively. The energy consumption for a heavy vehicle is roughly 1 [kWh/km] according to [15], so this confirms that the results can be considered reasonable and theoretically a great improvement.

In figure 5.19 one can see the Pareto front. The plot gives a graphical illustration of the trade off between operation cost and time for the different ω. One can for example see that using $\omega = 0.3$ instead of $\omega = 1$ will increase time by 22% but reduce operation costs by 42%.

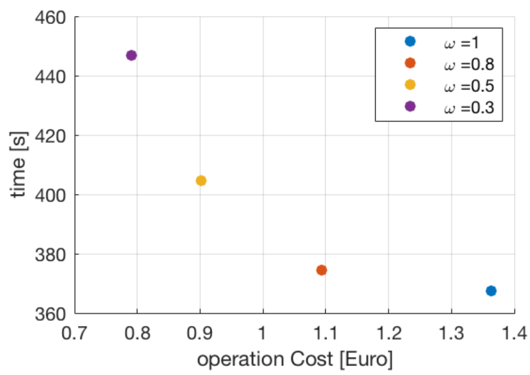


Figure 5.19: Pareto front. Total Operation Cost vs total Time.

In figure 5.20 one can see the torque and power for the electric motor and power required from the battery for $\omega = 1$. One reflection is that on numerous occasions the battery is close to or reaches its power limits.

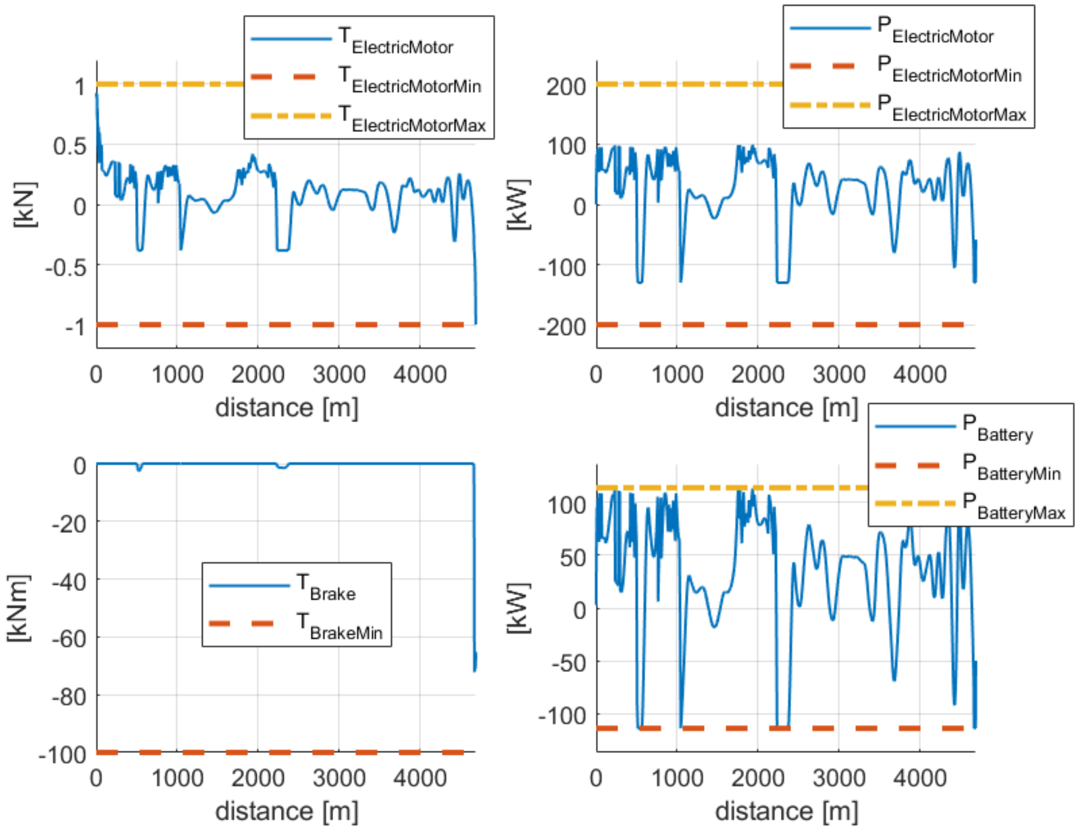


Figure 5.20: Torque and power for the electric motor and power required from the battery for $\omega = 1$.

6
Conclusion

It is reasonable to say that the method for developing an optimal speed profile that has been presented has its pros and cons. This chapter will include a set of paragraphs mentioning both pros and cons but also other topics worth talking about regarding this thesis work.

The idea of defining the vehicle's state as nodes and choosing edge costs to define the cost of changing states state generalizes the problem. One can for example choose to include gear changes. This could then for example be implemented as another dimension of the state space. Edge costs can also be chosen to anything that can be mathematically defined. Moreover, the speed optimization problem is translated to a shortest path problem which is a known problem with several solution methods. The solution method is specialized for the specific type of graphs that arise in this thesis. The costs that define the edge costs in this thesis (time, energy and battery degradation) are first attempts at speed optimization, with promising results. Furthermore, there is no doubt that if further research into the mathematical model would be done the results would be even better and more realistic.

One large flaw in the algorithm is that edge costs between node x and node y can only include information from the two nodes x and node y. No previous information of how the vehicle got to node x can be assumed. The consequence of this is that a constant state of battery charge must be assumed along the route that is being optimized. This is not how reality works. In reality the vehicle would loose its charge gradually along the route depending on the velocity profile. One way around this problem would be to assume a state of charge at each node. However, this would be a heuristic attempt at addressing the issue. Another option would be to speed optimize smaller sections at a time. Lets say the route is 10 [km] long, then perhaps one option is to speed optimize every km individually. Then each sub-route could have a unique state of charge.

The results presented in chapter 5 are promising. Fuel saving behaviors that are previously known are recognized in the figures. Another reflection is that the battery ware and tare costs always seem to be higher than the energy costs. These costs are of course directly effected by the assumed energy and battery price. Moreover, the model behaves in line with what is physically feasible. There are quite a few constraints and parameters to the mathematical model. It is hard to visualize and test all details but thorough testing has been done. Testing the mathematical model involves for example checking that certain constraints are broken in the right situations etc. There are many tests that can be done and properties that can be plotted. However, at some point one has to accept the model and move on.

7

Future work

This chapter presents some ideas of what to research in order to further develop this algorithm.

7.1 Look into the saw like behavior

One interesting thing to find out is why the saw like behavior appears in for example 5.2 for $\omega = 0.3$. It could be the case that this is a consequence of the algorithm and that there is nothing wrong mathematically or algorithmically. Moreover, I doubt this behavior would occur unless we could gain energy by slowing down or going down hill. Never the less, this could be researched further.

7.2 Parallel computing

As previously mention, the pre-processing of the edge costs takes a significant amount of time. It is a fact that each edge cost can be calculated independently. In order to reduce the time for pre-processing this could be done in parallel. However, this requires some work. But there is no doubt that this would improve the computation time.

7.3 Improve the model

The mathematical model described in the thesis can be improved or extended. One could for example include more costs or find better ways to represent battery ware and tare. There are many things that can be thought off. However, the idea of the algorithm to solve the speed optimization problem still holds. One can add or remove any type of cost and still use the same algorithm. However, the costs that are added between two nodes must be independent of how we got to the start node. An example of a cost that can not be added is the cost of the battery's state of charge. The state of charge depends on how the vehicle reaches a node. The state of charge must thus be assumed constant over the route that is to be optimized. This is one of the larger drawbacks of the mode. Finding a way to overcome this would be a big improvement.

7.4 Improve the normalization factor

Choosing a better normalization factor will have an influence on the result because the normalization factor has a direct impact on the cost function that is being minimized. The normalization factors in this thesis were chosen without much motivation. However, they seem to work. But this is an area that definitely can be improved.

7.5 Find better step size

The step size between velocity and distance points could be chosen more optimally. In this thesis they have been chosen to be the same for the test cases. However, the algorithm is designed to be able to handle different step sizes for both velocity and distance points. One could for example imagine that at inclination changes there could be smaller step sizes while on long straight slope they can be more distant.

7.6 Compare with a reference velocity

It is natural to have a real vehicle drive a certain route and measure its velocity and energy consumption and compare that result with what the algorithm generates. One then gets the answer to how well the mathematical model represents reality. This would also give some guidelines to where the mathematical model would need improvement.

7.7 Logic for setting end velocity

For this thesis the end velocity has been assumed to be zero. In the real world this may not always reasonable to do. An example would be a speed optimization over a really long route, perhaps several hundred km. It would take too long time to speed optimize the whole route. So it would be reasonable to divide the route into sections. Each section must have an end velocity. Another suggestion for future work is there for to look into how to choose the end velocity of each section.

Bibliography

[1] Bellman, R.E. 1957. Dynamic Programming. Princeton University Press, Princeton, NJ. Republished 2003: Dover, ISBN 0-486-42809-5.

[2] Köhler, T. (2017). *Mission Optimized Speed Control* (Unpublished master's thesis). Clausthal University of Technology, Clausthal-Zellerfeld, Germany.

[3] Sudhanva B., Jincan H. (2017) *Mission Optimized Speed Control* (Unpublished master's thesis). Kungliga Tekniska Höskolan, Stockholm, Sweden

[4] Erik Hellström, Maria Ivarsson, Jan Åslund, Lars Nielsen. (2009), *Look-ahead control for heavy trucks to minimize trip time and fuel consumption*, Control Engineering Practice,(17),2,245-254.

[5] Ragnar E.(2017), *Setting the Equivalence Factor in Equivalent Consumption Minimization Strategy using Reinforcement Learning Based Controller* (Unpublished master's thesis). Chalmers University of Technology, Gothenburg, Sweden

[6] Anders Fröberg (2008), *Efficient Simulation and Optimal Control for Vehicle Propulsion*, ISBN 978-91-7393-904-1

[7] Vaclav Kratochvil and Jiri Vomlel (2017), *Influence diagrams for the optimization of a vehicle speed profile*, International Journal of Approximate Reasoning 88 (2017) 567-586

[8] V.V.MonastyrskyI.M.Golownykh *Rapid computation of optimal control for vehicles*, Transportation Research Part B: Methodological Volume 27, Issue 3, June 1993, Pages 219-227.

[9] Y.Saboohi, H. Farzaneh, *Model for developing an eco-driving strategy of a passenger vehicle based on the least fuel consumption*, Applied Energy Volume 86, Issue 10, October 2009, Pages 1925-1932.

[10] Mahmoud Faraj, Baris Fidan, and Vincent Gaudet, *Cooperative Autonomous Vehicle Speed Optimization near Signalized Intersections*, Faculty of Engineering, University of Waterloo, ON, Canada.

[11] *Dynamic Programming for fuel consumption optimization on light vehicle*, 6th IFAC Symposium Advances in Automotive Control, Munich, Germany, July 12-14, 2010.

[12] Jonas Hellgren, *Mobile phone application for vehicle energy consumption evaluation - Algorithm description v2* (internal Volvo document)

[13] Jonas Hellgren, *Vehicle model* (internal Volvo document)

[14] Alexey V., Olof L. (2018) *Optimization vehicle speed for saving fuel.* (Internal document, Volvo).

[15] Jonas Hellgren. Senior principal engineer. Volvo Group Trucks Technology.

A

Appendix 2

A.1 Longitudinal motion model

Figure A.1 shows the different forces acting on the vehicle in an arbitrary state.

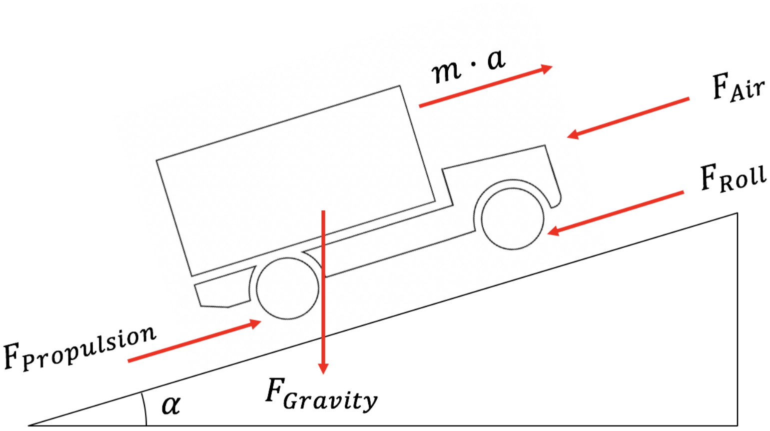


Figure A.1: The forces on an autonomous vehicle in an arbitrary state.

The equations that follow come from an internal Volvo document [12]. However, they can be derived by basic knowledge of mechanics.

$$F_{\text{Gravity}} = mg\sin(\alpha) \tag{A.1}$$

$$F_{\text{Roll}} = fmg\cos(\alpha) \tag{A.2}$$

$$F_{\text{Air}} = \frac{1}{2}\rho_{\text{Air}}CdAv^2 \tag{A.3}$$

$$F_{\text{Propulsion}} = ma + F_{\text{Gravity}} + F_{\text{Roll}} + F_{\text{Air}} \tag{A.4}$$

$$P_{\text{Propulsion}} = F_{\text{Propulsion}}v \tag{A.5}$$

$$T_{\text{Propulsion}} = F_{\text{Propulsion}}R_{\text{Wheel}} \tag{A.6}$$

$$F_{\text{FrictionMax}} = \mu F_{\text{Normal}} \tag{A.7}$$

$$F_{\text{Normal}} = mg\cos(\alpha). \tag{A.8}$$

$P_{\text{Propulsion}}$ and $T_{\text{Propulsion}}$ describes how much power and torque respectively is required for the vehicle to maintain its velocity. But they can also be negative. $P_{\text{Propulsion}}$ or $T_{\text{Propulsion}}$ can in some situations be too large for the vehicle to handle.

Meaning that the vehicle can not produced enough power or torque to maintain its velocity. If so, the situation is not physically feasible. However, if $P_{\text{Propulsion}}$ and $T_{\text{Propulsion}}$ is negative there is a lower limit of how much power and torque the vehicle can use to restore energy in the battery. If this lower limit is reached the brakes can be applied to make the situation physically feasible. However, there is also a limit to how much power and torque can go into the brakes. So a model that correctly distributes the power and torque between the electric motor and brakes must be constructed.

A.2 Electric motor and brake model

The brakes are applied when the electric motor and/or the battery can not take up any more energy. We want to minimize the brake usage because we are wasting energy when applying the brakes. Note that T_{Brake} is never positive because we can never use the brakes to propel the vehicle forward. Hence, to minimize the brake usage we want to maximize T_{Brake}. With the following optimization problem the electric motor will be used to control the vehicle as much as possible.

$$\underset{T_{\text{Brake}}}{\text{maximize}} \quad T_{\text{Brake}} \tag{A.9a}$$

$$\text{subject to} \quad T_{\text{Propulsion}} = T_{\text{ElectricMotor}} r_{\text{Total}} + T_{\text{Brake}}, \tag{A.9b}$$

$$T_{\text{Brake}} \leq 0, \tag{A.9c}$$

$$T_{\text{ElectricMotorMin}} \leq T_{\text{ElectricMotor}}, \tag{A.9d}$$

$$P_{\text{ElectricMotorMin}} \leq P_{\text{ElectricMotor}}, \tag{A.9e}$$

$$P_{\text{BatteryMin}} \leq P_{\text{Battery}} \tag{A.9f}$$

where

$$P_{\text{Brake}} = T_{\text{Brake}} \omega_{\text{Wheel}} \tag{A.10}$$

$$P_{\text{ElectricMotor}} = T_{\text{ElectricMotor}} \omega_{\text{ElectricMotor}} \tag{A.11}$$

$$P_{\text{Battery}} = T_{\text{ElectricMotor}} \phi_{\text{ElectricMotor}} + P_{\text{Aux}}. \tag{A.12}$$

This optimization problem is mathematically equivalent to

$$\underset{T_{\text{ElectricMotor}}}{\text{minimize}} \quad T_{\text{ElectricMotor}} \tag{A.13a}$$

$$\text{subject to} \quad T_{\text{Propulsion}} \leq T_{\text{ElectricMotor}} r_{\text{Total}}, \tag{A.13b}$$

$$T_{\text{ElectricMotorMin}} \leq T_{\text{ElectricMotor}}, \tag{A.13c}$$

$$P_{\text{ElectricMotorMin}} \leq T_{\text{ElectricMotor}} \omega_{\text{ElectricMotor}}, \tag{A.13d}$$

$$P_{\text{BatteryMin}} \leq T_{\text{ElectricMotor}} \phi_{\text{ElectricMotor}} + P_{\text{Aux}} \tag{A.13e}$$

The analytical solution to this optimization problem is thus

$$T_{\text{ElectricMotor}} = \max\{\frac{T_{\text{Propulsion}}}{r_{\text{Total}}}, T_{\text{ElectricMotorMin}}, \tag{A.14}$$

$$\frac{P_{\text{ElectricMotorMin}}}{\omega_{\text{ElectricMotor}}}, \frac{P_{\text{BatteryMin}} - P_{\text{Aux}}}{\omega_{\text{ElectricMotor}}}\} \tag{A.15}$$

With this analytical solution the remaining properties can easily be calculated.

$$T_{\text{Brake}} = T_{\text{Propulsion}} - T_{\text{ElectricMotor}} r_{\text{Total}} \tag{A.16}$$
$$P_{\text{Brake}} = T_{\text{Brake}} \omega_{\text{Wheel}} \tag{A.17}$$

The result of the above optimization problem is then used to calculate the energy drawn from the electric motor $\dot{E}_{\text{ElectricMotor}}$.

$$\dot{E}_{\text{ElectricMotor}} = P_{\text{ElectricMotor}} \phi_{\text{ElectricMotor}} \tag{A.18}$$
$$\phi_{\text{ElectricMotor}} = \begin{cases} \frac{1}{\eta_{\text{ElectricMotor}}}, & \text{if } F_{\text{Propulsion}} \geq 0 \\ \eta_{\text{ElectricMotor}}, & \text{if } F_{\text{Propulsion}} < 0 \end{cases} \tag{A.19}$$
$$P_{\text{ElectricMotor}} = T_{\text{ElectricMotor}} \omega_{\text{ElectricMotor}} \tag{A.20}$$
$$\omega_{\text{ElectricMotor}} = \omega_{\text{Wheel}} r_{\text{Total}} \tag{A.21}$$
$$\omega_{\text{Wheel}} = v / R_{\text{Wheel}} \tag{A.22}$$
$$r_{\text{Total}} = r_{\text{Transmission}} r_{\text{FinalDrive}} \tag{A.23}$$

A.3 Battery model

The following set of equations describes how the battery efficiency ϕ_{Battery} is calculated. The model was created with help from Jonas Hellgren, supervisor of this thesis.

$$I_{\text{Cell}} = I_{\text{Pack}} = \frac{P_{\text{Pack}}}{U_{\text{Pack}}} = \frac{P_{\text{Battery}} / N_{\text{Packs}}}{U_{\text{Cell}} N_{\text{CellsPerPack}}} \tag{A.24}$$
$$U_{\text{Cell}} = U_{\text{CellNominal}} - I_{\text{Cell}} R_{\text{Cell}} \tag{A.25}$$

Solving for I_{Cell} gives

$$b = U_{\text{CellNominal}} / R_{\text{Cell}} \tag{A.26}$$
$$c = P_{\text{Battery}} / (R_{\text{Cell}} N_{\text{CellsPerPack}} N_{\text{Packs}}) \tag{A.27}$$
$$I_{\text{Cell}} = -\frac{b}{2} + \sqrt{\frac{b}{2}^2 - c} \tag{A.28}$$

The other solution to the second degree polynomial is trashed because it gives unreasonable results. For example that the I_{Cell} will be negative at all times. So then we can calculate

$$P_{\text{Celli}} = U_{\text{Cell}} I_{\text{Cell}} \tag{A.29}$$
$$P_{\text{Celle}} = U_{\text{CellNominal}} I_{\text{Cell}} \tag{A.30}$$
$$\eta_{\text{Charge}} = \frac{P_{\text{Celle}}}{P_{\text{Celli}}} \tag{A.31}$$
$$\phi_{\text{Battery}} = \eta_{\text{Charge}} \tag{A.32}$$

The battery's state of health SoH is a measurement of the ware and tare on the battery. Thus, $\dot{\text{SoH}}$ represents the ware and tare per second. The following model comes from an internal volvo document [13]. It is used without deeper understanding of the equations. However, this part of the model was intensely tested to confirm that reasonable results were generated.

$$\dot{\text{SoH}} = \frac{|P_{\text{Cell}}|}{2N_{\text{CyclesMax}} \cdot \text{Enb}} \tag{A.33}$$

$$P_{\text{Cell}} = \frac{P_{\text{Battery}}}{\text{N}_{\text{Cells}}} \tag{A.34}$$

$$\text{N}_{\text{Cells}} = \text{N}_{\text{Packs}}\text{N}_{\text{CellsPerPack}} \tag{A.35}$$

$$\begin{aligned} \text{N}_{\text{CyclesMax}} = {} & \text{N}_{\text{Low}} \\ & + (\text{N}_{\text{Mod}} - \text{N}_{\text{Low}})\frac{1}{1 + \text{Exp}(-\text{k}_{\text{Low}}(P_{\text{Cell}} - P_{\text{Low}}))} \\ & + (\text{N}_{\text{High}} - \text{N}_{\text{Mod}})\frac{1}{1 + \text{Exp}(-\text{k}_{\text{Mod}}(P_{\text{Cell}} - P_{\text{Mod}}))} \end{aligned} \tag{A.36}$$

$$\text{Enb} = \text{C}_{\text{Cell}}\text{U}_{\text{CellNominal}} \tag{A.37}$$

The parameters for this model are set to the following

$$N_{\text{Low}} = 2000 \tag{A.38}$$

$$N_{\text{Mod}} = 4200 \tag{A.39}$$

$$N_{\text{High}} = 1000 \tag{A.40}$$

$$k_{\text{Low}} = 0.1 \tag{A.41}$$

$$k_{\text{Mod}} = 0.02 \tag{A.42}$$

$$P_{\text{Low}} = 50 \tag{A.43}$$

$$P_{\text{Mod}} = 300 \tag{A.44}$$

A.4 Model analysis

The purpose of this chapter is to show what is physically possible before applying the model to speed optimization algorithm for the test routes. A neat way to do this is to make a scatter plot on a 2d plane where the x-axis represents the vehicle velocity v, the y-axis represents the vehicle acceleration a and do this for different slope angles α. Each colored point in the graph represents a scenario that is physically feasible. If a point is white, it is not physically feasible. The red line simply indicates where the zero line lies.

In figure A.2 one can see what is physically feasible for a certain set of and the magnitude of the property $P_{\text{Propulsion}}$. $P_{\text{Propulsion}}$ is the required energy for the scenario. For example we can read that it is not possible to have the combination $v = 15, a = 0, \alpha = -2$ but it is not possible to have $v = 15, a = 0, \alpha = 2$. This tells us that for high speeds and steep enough slopes the vehicle cannot accelerate.

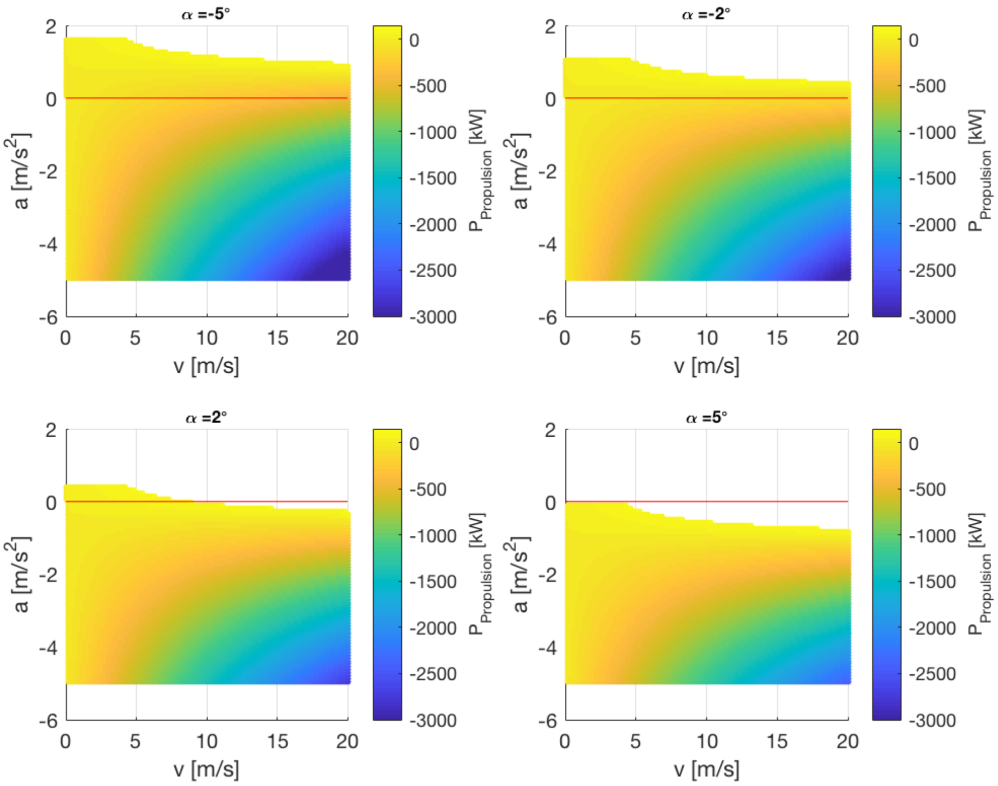


Figure A.2: Figures showing $P_{\text{Propulsion}}$ and what is physically possible combinations of velocity, acceleration for a few different slope angles. Red line indicates the zero line. Whites region is not physically possible.

A.5 Result: Preprocessing and optimization time performance

In this section results regarding the time performance of the preprocessing and optimization algorithm will be analyzed. All the calculations in this section were done on the same computer. Assume we have a graph with m velocity points at each of the n distance points. This means that there are a total of mn nodes in the graph and m^2n edges to calculate. Calculating an edge means to calculate the time cost, operation costs and if it is feasible. Preprocessing time is the time it takes to calculate all edges. Optimization time is the time it takes to run algorithm 1 on the described graph. The time will vary on different machines but this section gives an indication of the time that is required. But the purpose of this section is to show how fast the algorithm generates a result. Looking at table A.1 one can see how the preprocessing time is drastically increasing with the number of distance points n but the optimization time is still very efficient.

m	n	Preprocessing time [s]	Optimization time[s]
201	101	225	0.13
201	201	407	0.25
201	401	760	0.51

Table A.1: preprocessing and optimization time as a result of the number of nodes

In table A.2 the number of velocity points m has been increased and as an effect of this there is an even higher increase in preprocessing time. However, the optimization time remains efficient.

m	n	Preprocessing time[s]	Optimization time[s]
201	101	225	0.13
401	101	877	0.28
801	101	3437	0.61

Table A.2: preprocessing and optimization time as a result of the number of nodes

From these two tables it is reasonable to conclude that the preprocessing time is probably going to be a problem that needs to be addressed if this algorithm were to be used on real trucks.

B
Appendix 3

B.1 Vehicle parameters

Symbol	Value	Unit	Comment
Other			
airDensity	1.225	kg/m^3	
g	9.82	m/s^2	
frictionCoefficient	0.9		
Vehicle			
accelerationMin	-5	m/s^2	
accelerationMax	5	m/s^2	
ratioTransmission	2		
ratioFinalDrive	6		
mass	30	kkg	
rollResistance	0.005		
airResistance	6.2		
powerAux	3	kW	Power required for GPS, lights etc
radiusWheel	0.491	m	
torqueBrakeMin	-100	kNm	
ElectricMotor			
torqueElectricMotorMin	1	kNm	
torqueElectricMotorMax	1	kNm	
powerElectricMotorMin	-200	kW	
powerElectricMotorMax	200	kW	
etaElectricMotor	0.9		
omegaElectricMotorMax	340	1/s	Angular velocity of the electric motor
Battery			
numberOfCellsPerPack	180		
numberOfPacks	4		
voltageCellNominal	3.7	Volt	
resistanceCell	0.002	Ohm	
ampsCell	133200		
priceBattery	83782.8	euro	
priceEnergy	0.1	euro/kWh	
voltageCellMin	2.6	Volt	
voltageCellMax	4.15	Volt	
currentCellMax	400	A	
cellCapacityNominal	13320	As	
stateOfCharge	0.5	-	

Figure B.1: Default vehicle parameters