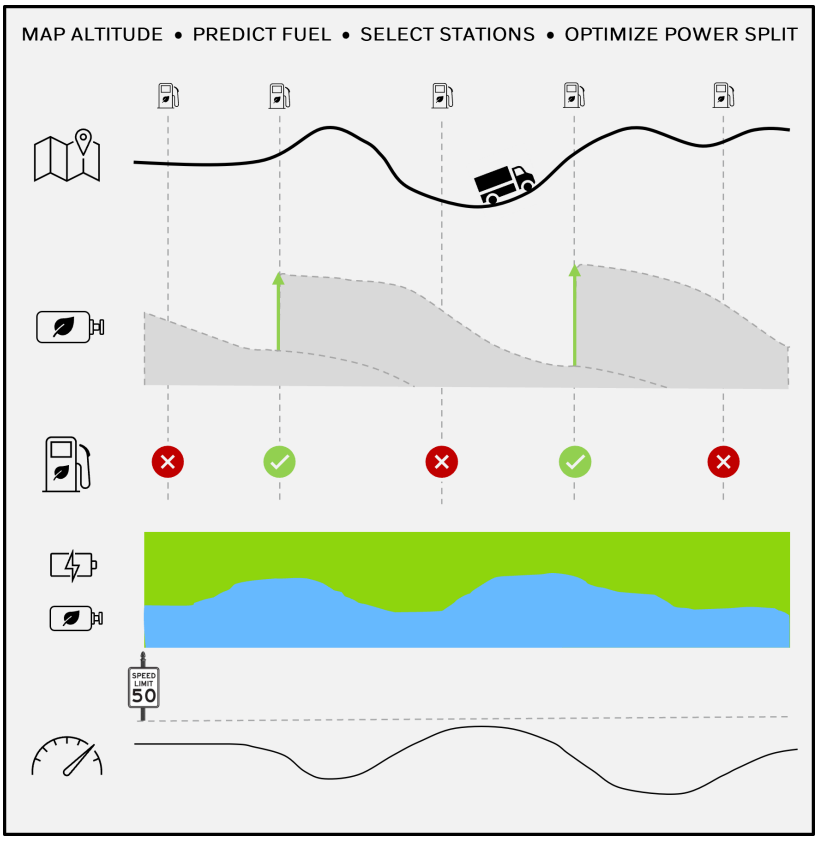


Mission Management for Fuel Cell Electric Trucks – a Bilevel Approach

Optimise power split, speed and refuelling in the presence of uncertain external factors

Master's thesis in Systems, Control and Mechatronics

ILYA KUANGALIYEV

OSCAR MARK

DEPARTMENT OF ELECTRICAL ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY

Gothenburg, Sweden 2024

www.chalmers.se

Master's thesis 2024

Mission Management for Fuel Cell Electric Trucks - a Bilevel Approach

Optimise power split, speed and refuelling in the presence of uncertain external factors

ILYA KUANGALIYEV
OSCAR MARK

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Division of Systems and control
Chalmers University of Technology
Gothenburg, Sweden 2024

Mission Management for Fuel Cell Electric Trucks - a Bilevel Approach
Optimise power split, speed and refuelling in the presence of uncertain external factors
ILYA KUANGALIYEV, OSCAR MARK

Supervisors: Saurabh Suman, Olof Lindgärde, Volvo Group Technology
Examiner: Nikolce Murgovski, Department of Electrical Engineering

Master's Thesis 2024
Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Illustration of mission management for fuel cell electric trucks

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2024

Mission Management for Fuel Cell Electric Trucks - a Bilevel Approach
Optimise power split, speed and refuelling in the presence of uncertainty
ILYA KUANGALIYEV, OSCAR MARK
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Fuel cell electric vehicles (FCEVs) have a crucial role in the transition to fossil-free travel and transport. In order to fully leverage their potential, proper mission planning of such vehicles should be conducted. This thesis aims to solve the mixed-integer problem of mission management with a hierarchical bilevel approach. The first layer solves a simplified mixed-integer problem of choosing hydrogen stations to refuel at and the amount to refuel with the help of stochastic dynamic programming, allowing us to account for uncertain external factors to obtain a more robust solution. The second layer becomes a smooth problem that can be solved with more complex and comprehensive dynamics. In addition, the benefits of using non-equidistant stage sampling were evaluated. To evaluate the robustness and optimality of the proposed approach 3 tests were conducted. The simulations were conducted with a 45-ton truck using 3 routes: Zagreb–Paris (1559 km), Gothenburg–Kiruna (1793 km), and Gothenburg–Rødby (461 km). Overall, according to the results, the bilevel approach can yield solutions comparable in optimality to the ones that were obtained with different strategies. In some cases, the proposed approach produced better results when it comes to robustness in the presence of disturbances, but the results were not conclusive enough to answer either of the research questions with a high level of confidence. Even though the non-equidistant stage sampling did not show any substantial increase in simulation accuracy, it was shown that stage and DP stage resolutions can be reduced without significant losses in optimality, making the computation time almost 3 times smaller.

Keywords: Fuel cell electric trucks, Mission Management, Stochastic optimal control, Mixed-integer nonlinear program.

Acknowledgements

We would like to extend our heartfelt gratitude to our supervisors, Saurabh Suman and Olof Lindgärde. Their support and advice throughout the project has been invaluable. By giving us a good understanding of the background and inspiring us with a lot of great ideas for implementation they helped the project to advance without a single hitch. We also want to thank our examiner, Nikolce Murgovski, for being so dedicated to helping us with every issue we had and sharing all his experience in optimal control. Finally, we want to thank our peer, Isac Borghed, who was doing his thesis on a similar topic together with us at Volvo Group Technology, for great and effective collaboration and partnership.

Ilya Kuangaliyev, Gothenburg, June 2024
Oscar Mark, Gothenburg, June 2024

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

DC	Direct current
GPS	Global positioning system
ICE	Internal combustion engine
IVP	Initial value problem
BEV	Battery electric vehicle
EM	Electric machine
ESS	Energy storage system
HRS	Hydrogen refuelling station
MSE	Mean squared error
MPC	Model predictive control
MINLP	Mixed-integer nonlinear program
NLP	Nonlinear program
NMPC	Nonlinear model predictive control
FC	Fuel cell
FCEV	Fuel cell electric vehicle
HEV	Hybrid electric vehicle
PEM	Predictive energy management
PEMFC	Proton-exchange membrane fuel cell
RPM	Revolutions per minute
DP	Dynamic programming
SDP	Stochastic dynamic programming
SoC	State of charge

Nomenclature

Below is the nomenclature of indices, sets, parameters, and variables that have been used throughout this thesis.

Indices

i	State index
n	Stage number in the NLP
k	Stage number in the MINLP

Sets

$\mathbb{X}$	State constraint set
$\mathbb{X}_f$	Terminal state constraint set
$\mathbb{U}$	Control input constraint set
$\mathbb{K}_{station}$	Set of stages in MINLP that have a hydrogen station

Parameters

$\mathbf{x}_0$	Initial condition
$\mathbf{x}_{target}$	Target terminal state value
m	Vehicle mass
m_b	Battery mass
A_f	Vehicle effective frontal surface area
C_d	Vehicle aerodynamic drag coefficient
C_{rr}	Vehicle rolling resistance coefficient
k_{tm}	Gear ratio of the transmission
k_{amb}	Coefficient of heat exchange between the battery's case and air

g	Free fall acceleration
ρ_{air}	Air density
η_{em}	Electric machine efficiency
η_{tm}	Transmission efficiency
η_{th}	Battery thermal system efficiency
$P_{fc,max}$	Fuel cell maximum power output
$P_{fc,min}$	Fuel cell minimum power output
$\dot{P}_{fc,max}$	Fuel cell maximum power output rate
$\dot{P}_{fc,min}$	Fuel cell minimum power output rate
$F_{T,max}$	Maximum traction force
$F_{br,max}$	Maximum friction brake force
$\bar{\mathrm{H}}_{2,min}$	Minimum normalised hydrogen level
$\bar{\mathrm{H}}_{2,max}$	Maximum normalised hydrogen level
SoC_{min}	Minimum SoC level
SoC_{max}	Maximum SoC level
C_b	Battery electric capacity
C_{H_2}	Hydrogen tank capacity
c_b	Battery specific heat capacity
N	Number of stages in the NLP
K	Number of stages used in the MINLP
M	Number of quantised state values used in SDP
R	Tyre effective radius
t_{rf}	Full tank refuelling time
t_q	Station queue time
v_{max}	Maximum vehicle speed
$T_{b,min}$	Minimum battery temperature
$T_{b,max}$	Maximum battery temperature
λ	Discount factor
Δs	Spatial step
ΔH	Hydrogen's lower heating value
κ	Risk tolerance (Maximum risk of failure)

Variables

$\mathbf{x}$	State vector
$\mathbf{u}$	Control input vector
$\mathbf{y}$	Outputs vector
t	Time
t_K	Total travel time in NLP
t_N	Total travel time in MINLP
v	Vehicle longitudinal velocity
v_{rel}	Vehicle speed relative to the air
s	Vehicle position along the path
q	Battery charge
SoC	Battery state of charge
I_b	Battery current
H_2	Hydrogen level in the vehicle tank
$\mathrm{H}_{2,rf}$	Hydrogen refuelled at a hydrogen station
$\bar{\mathrm{H}}_2$	Normalised hydrogen level in the vehicle tank
$\bar{\mathrm{H}}_{2,rf}$	Normalised hydrogen refuelled at a hydrogen station
T_b	Battery temperature
P_{em}	Electric machine power demand
P_{fc}	Fuel cell power output
P_b	Battery power output
P_{br}	Friction brake power
P_{th}	Battery thermal system power
E_{br}	Energy lost from friction breaking
Q_{act}	Battery active heat generation rate
Q_{pass}	Battery passive heat generation rate
Q_{amb}	Battery dissipated heat rate
F_T	Traction force
F_{fc}	Traction force from the fuel cell
F_b	Traction force from the battery
F_{br}	Friction brake force
F_{rr}	Rolling resistance force
F_{air}	Aerodynamic drag force
F_g	Gravitational force due to road slope
F_z	Normal reaction force of the road

s_{H_2}	Slack variable for $\bar{\mathrm{H}}_2$ bound relaxation
s_{SoC}	Slack variable for SoC bound relaxation
s_t	Slack variable for t bound relaxation
τ_{em}	Electric machine torque
ω_{em}	Electric machine angular velocity
$\delta_{\boldsymbol{\pi}}$	Binary variable used for feasibility check in the MINLP
δ_{st}	Binary variable identifying a hydrogen station's presence
ω_t	Cost of time in the NLP
ω_{H_2}	Cost of hydrogen in the NLP
ω_s	Cost of slack variables in the NLP
ω_{br}	Cost of friction breaking in the NLP
ε_e	Cost of deviating from desired state targets
ε_{br}	Cost of friction braking in the MINLP
ε_{fuel}	Cost of fuel (hydrogen) in the MINLP
ε_{fail}	Cost of mission failure in the MINLP
ε_{stop}	Cost of making a stop to refuel in the MINLP
ε_{tr}	Cost of breaching the fuel cell's transient's maximum allowed level in the MINLP
ε_{llim,x_i}	Cost of breaching a state's lower bound in the MINLP
ε_{ulim,x_i}	Cost of breaching a state's upper bound in the MINLP
ε_C	Cost of mission completion in the MINLP
$\mu_{P_{em}}$	Power demand's mean
$\sigma_{P_{em}}$	Power demand's standard deviation
γ	Vehicle's gear

Functions

$\mathbf{p}(s)$	Road parameters
$\alpha(s)$	Road grade
$\varphi(s)$	Vehicle heading angle
$\theta(s)$	Wind heading angle
$T_{amb}(s)$	Ambient temperature
$w(s)$	Wind speed
$\beta(\cdot)$	Wind effective angle of attack

$v_{lim}(s)$	Road speed limit
$\Delta v_{tr}(s)$	Traffic speed reduction
$P_{aux}(T_{amb})$	Auxiliary power demand
$R_b(SoC, T_b)$	Battery internal resistance
$P_{em,loss}(P_{em})$	Electric machine loss
$P_{b,max}(SoC, T_b)$	Battery maximum power output
$P_{b,min}(SoC, T_b)$	Battery minimum power output
$U_{oc}(SoC, T_b)$	Battery open circuit voltage
$\mathrm{H}_{2,con}(P_{fc})$	Hydrogen consumption rate
$J(\cdot)$	Objective function to minimise in the nonlinear problem
$P_{b,loss}(\cdot)$	Battery heat loss
$P_{fc,heat}(P_{fc})$	Fuel cell heat loss
$\Pr(\cdot)$	Probability function
$f(\cdot)$	Probability density function of the uncertainty
$\mathbf{f}(\cdot)$	Continuous state dynamics
$\mathbf{g}(\cdot)$	Nonlinear inequality constraints
$k_{em}(\gamma)$	Gear ratio of the motor
$L(\cdot)$	Cost function used in the MINLP
$L(\cdot)$	Cost function used in the MINLP
$L(\cdot)_{dr}$	Cost function of driving used in the MINLP
$L(\cdot)_{lim}$	Cost function of leaving the state constraint set used in the MINLP
$L(\cdot)_{tr}$	Cost function of breaching the maximum allowed level of the fuel cell's transient used in the MINLP
$L(\cdot)_{rf}$	Cost function of refuelling used in the MINLP
$T(\cdot)$	Terminal cost function used in the MINLP
$Q(\cdot)$	Bellman action-value function
$V(\cdot)$	Bellman value function
$\boldsymbol{\pi}$	Optimal policy from dynamic programming
$\delta_{Dirac}(s)$	Dirac delta function
$\eta_{fc}(P_{fc})$	Fuel cell efficiency

Contents

List of Figures

List of Algorithms

List of Tables

1
Introduction

Humanity has set ambitious goals for reducing worldwide carbon emissions to fight climate change. As transport is responsible for about a quarter of the EU's total CO2 emissions, 27% of which come from heavy-duty trucks [1], the market actors can not ignore this challenge. From a lifecycle perspective, most of the emissions occur during the use phase of vehicles. Therefore, one of the ways that companies can reduce emissions is to manufacture and encourage the use of vehicles running on more sustainable sources of energy.

Today plenty of resources are spent on developing alternatives to the internal combustion engine. An example of such is fuel cell electric vehicles. One challenge in bringing these vehicles to the market is the lack of infrastructure required to charge or refuel. To ensure the adoption of these new technologies in the commercial sector additional services need to be provided that plan routes and suggest where to stop to refuel, as such decisions might have unintuitive optimal solutions.

A common way of approaching the mission management task is viewing it as an optimisation problem and solving it using nonlinear model predictive control (NMPC) [2]. Some of the previous work in this field make it evident that mission management for fuel cell electric vehicles using NMPC is not a trivial task for many reasons. Firstly, problems arise from the specific behaviour of the fuel cell, which makes it necessary to use an additional short-term energy storage system (ESS), in other words, a battery. The ESS lets the vehicle brake regeneratively, increasing energy efficiency, and provides an additional power source during acceleration and hill climbing, reducing power fluctuations for the fuel cell, which increases its lifetime [3]. This, however, in turn leads to some complications. To fully utilise the ESS the battery state of charge (SoC) must be used optimally. For instance, the battery SoC should be sufficiently high before a large hill climb, and nearly empty before a steep descent, in order to efficiently assist the fuel cell and recuperate otherwise lost energy. Having two energy sources also complicates the optimisation task as it increases the number of states in the vehicle, where one state tracks the level of hydrogen in the tank and another tracks the battery's charge. This in turn means the controller needs to know the road topography ahead of time and plan accordingly.

Another complication arises from the fact that the task contains some binary decision variables, such as whether to stop at a hydrogen station to fuel or not. This makes the optimisation problem mixed-integer which is computationally demanding to approach using only NMPC as then it has to be solved for many combinations of

the binary variables. In this project, a two-layer hierarchical approach is developed, where the discrete decision variables are determined at the top level using stochastic dynamic programming (SDP) leaving the residual problem to be solved at the lower level as a smooth nonlinear program (NLP).

1.1 Previous work

One of the main works that this project is based on is that by Sundberg and Bragde [4]. In their thesis, the same problem is approached only from the NMPC perspective, where some smoothening techniques are used to omit the integer part of the objective function. Also, some complexity-reducing assumptions such as simplified thermal dynamics of the battery were made in that work which will be challenged in this work. Another example of using optimal control applied to the modelling of FCEVs is [5] where a supervisory control is conducted to ensure an optimal power split between the battery and the fuel cell. In [6] one can find how power split optimisation is conducted with model predictive control (MPC).

The area of vehicle routing and planning has been studied for decades. Optimal vehicle routing problems consider single or multiple vehicles travelling along a graph, potentially loading/unloading goods or passengers at certain points and minimising a cost often associated with fuel, time, or more recently, environmental impact [7]. For an overview, see the book by Toth and Vigo [8]. For the single vehicle mission, eliminating the routing problem and instead considering a fixed route decreases model complexity, allowing instead for more complex vehicle modelling, e.g., hybrid vehicles where a powertrain split can be optimised [9]. Early work considering hybrid electric vehicles (HEVs), by Rajagopalan and Washington [10] and Lin *et al.* [11], utilise GPS technology to optimise the power split between the internal combustion engine (ICE) and the electric machine (EM) for a short horizon based on topography and traffic data. In papers by Kolmanovsky *et al.* [12] and by Johannesson *et al.* [13], certain parameters are modelled as stochastic. In Sundberg and Bragde's thesis, and in the paper by Johannesson *et al.* the problem is discretised spatially, as space-dependent dynamics become easier to formulate than time-dependent dynamics and maintain problem convexity. In her master thesis, Katsagyri [14] samples the road at non-equidistant points to capture more detail in the road characteristics and in doing so improves accuracy.

When it comes to thermal modelling of the battery, Hamednia *et al.* [15] discusses optimal (battery) thermal management for battery electric vehicles (BEVs) from the mission planning perspective in their paper. In it, they also separate driving and charging dynamics into a hybrid dynamical system, where the charging dynamics are modelled in a time domain, whereas the driving dynamics are modelled in a spatial domain. Another example is a paper by Murashko *et al.* [16], in which they describe the modelling thermal management system for the lithium-ion battery pack in hybrid electric vehicles (HEVs).

Different hierarchical approaches have also been incorporated quite extensively in

past works. One example is a paper by Johannesson *et al.* [17], in which the solution is based on a 3-layer structure to approach the mixed-integer problem, where each layer operates on different timescales with different update frequencies, prediction horizons, and model abstractions. Another example of such an approach is described in a paper by Ganesan *et al.* [18], where the lower layer is intended to solve the mixed-integer problem of gear selection using receding horizon MPC with different numerical relaxations.

Methods accounting for stochastic factors have also been used in quite a few works, one example being a paper by Wang *et al.* [19], where optimal energy scheduling for a hybrid fuel cell/PV/battery system in the presence of uncertain factors is done based on stochastic dynamic programming. In their paper, Ritter *et al.* [20] incorporate stochastic model predictive control for the energy management of HEVs.

1.2 Limitations

In order to reduce the complexity of the problem, no other vehicle dynamics than the longitudinal ones are considered in this work, which means that only the forces that influence a vehicle's movement in the direction of travel are accounted for. For the same reason, no fluid dynamics of hydrogen during refuelling is considered in the work.

According to EU laws, truck drivers must take a 45-minute long break after $4\frac{1}{2}$ hours of driving at the latest, and must not drive for more than 9 hours a day, including breaks, except for twice a week where it may be extended to 10 hours [21]. This factor significantly complicates the solution of the mission management problem, therefore an optimisation problem considering these constraints will be left for future work.

1.3 Main research questions

Through implementing a hierarchical approach for handling the mixed-integer problem, the authors hope to answer the following research questions:

- Can splitting the mixed-integer problem into two layers, solving the integer decisions in the higher layer, thus making the second layer a smooth non-linear problem, yield a similar optimal solution as the MINLP problem?
- Further, can modelling certain dynamics as stochastic processes improve the robustness of the refuelling station selection?
- Can non-equidistant stage sampling help reduce the computational heaviness of the problem without significant losses in accuracy and optimality?

2
Theory

This chapter details how the vehicle and its powertrain are modelled, detailing the background to help understand the equations used in modelling.

2.1 Vehicle modelling

As previously stated only the longitudinal dynamics are considered as the truck is driving, the problem can thus be considered 2-dimensional, one being the spatial dimension and the other being the temporal dimension.

2.1.1 Longitudinal dynamics

The equation of motion of the truck in the longitudinal direction can be written as

$$m\dot{v} = F_T - F_{rr} - F_{air} - F_g - F_{br}, \tag{2.1}$$

where, on the left side, m is the mass of the truck and $\dot{v}$ is the time derivative of the truck's speed. On the right-hand side, F_T is the traction effort generated by the EM, F_{rr} is the sum of all rolling resistance forces acting on each tyre, F_{air} is the aerodynamic drag force, and F_g is the gradient force due to the road slope. The equation of motion is missing longitudinal slip losses, which appear when the wheel rotation speed differs from the corresponding vehicle speed. Modelling slip losses require tracking wheel rotation speed, which significantly increases the complexity of the model, and as these losses are mostly relevant during acceleration and deceleration modelling them provides little increase in accuracy, especially in the mission management context. A more detailed description of traction effort F_T is done in section 2.1.2 whereas the other forces are discussed here.

Rolling resistance

Rolling resistance is the resistive force that arises as a wheel is rolling on the surface of the road. The deformation of the tyre during rolling is both elastic and viscous, so-called viscoelastic, due to the properties of rubber material. In the elastic component of the deformation, no permanent (plastic) deformations occur, which is, however, not the case for the viscous component as this deformation can not be fully recovered after unloading. During viscous deformation, friction between the molecules causes a conversion of the mechanical energy transferred to the tyre into heat, hence leading

to energy losses [22]. Mathematically these losses are expressed as a rolling resistance force that can be written as

$$F_{rr} = C_{rr} F_z, \tag{2.2}$$

where C_{rr} is the rolling resistance coefficient of the tyres and F_z is the sum of all the normal reaction forces of the road acting on each tyre. The rolling resistance coefficient C_{rr} is a measure of the force needed for a vehicle to maintain a constant speed with zero road grade and zero air resistance per unit weight. The rolling resistance coefficient is usually found empirically and is approximately equal to 0.0035 for a 40-tonne Volvo FH truck operating in the summer [23]. External factors such as rain, snow and the road surface can have significant effects on the rolling resistance coefficient, increasing it by upwards of 50% [24][25]. For a truck driving on a road, the normal force is $F_z = mg\cos\alpha$, where α is the road grade. The rolling resistance force can then be rewritten as

$$F_{rr} = C_{rr} mg\cos\alpha. \tag{2.3}$$

Aerodynamic drag

Aerodynamic drag forces, F_{air} arise as an object moves through a fluid. The equation is given by

$$F_{air} = \frac{1}{2}\rho_{air} C_d A_f v^2, \tag{2.4}$$

where ρ_{air} is the density of air, v the vehicle's longitudinal speed, and $C_d A_f$, often combined as one parameter denoted *drag area*, is the product of the drag coefficient and the effective frontal area of the vehicle respectively. Considering the dynamics of wind speed and heading the equation can be expanded into

$$F_{air} = \frac{1}{2}\rho_{air} C_d A_f(\beta) v_{rel}^2, \tag{2.5}$$

where the parameter $C_d A_f$ is now a function of the effective angle of attack β and the vehicle's speed relative to the air around it, v_{rel}, both of which are given by the equations

$$\beta = \arctan\frac{v_{rel,y}}{v_{rel,x}} - \varphi - \pi \tag{2.6}$$

$$v_{rel} = \sqrt{v_{rel,x}^2 + v_{rel,y}^2}, \tag{2.7a}$$

$$v_{rel,x} = w\cos\theta - v\cos\varphi, \tag{2.7b}$$

$$v_{rel,y} = w\sin\theta - v\sin\varphi. \tag{2.7c}$$

The parameter w is the wind speed, φ is the vehicle's heading angle and θ is the wind heading angle, both relative to a set coordinate frame. An illustration of all those speeds can be seen in Figure 2.1. The value of $C_d A_f(\beta)$ has been determined empirically for a range of common angles β.

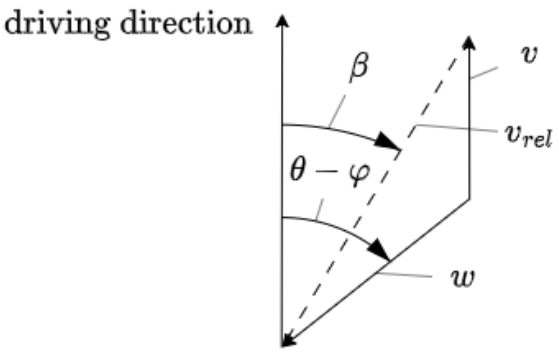


Figure 2.1: Illustration of wind's relative speed.

Gradient force

The part of the gravitational force acting in the direction of the vehicle's heading is given by

$$F_g = mg \sin \alpha, \tag{2.8}$$

where a positive slope angle α, signifying an increase in altitude ahead, results in a resistive gradient force and a negative slope angle results in an additive gradient force.

Braking force

Braking of the vehicle is done with the help of either friction brakes or regenerative brakes, however, in the context of this work, all the variables that have *br* as an index, will refer to the friction brakes only. More about the regenerative brake force will be discussed further in this chapter.

2.1.2 Powertrain

The vehicle considered in this thesis is a heavy-duty fuel-cell electric truck. A fuel cell converts hydrogen gas to electric energy which, assisted by a lithium-ion battery, drives an electric machine. The battery can be charged by the fuel cell or by regenerative braking but cannot be charged by plug-in.

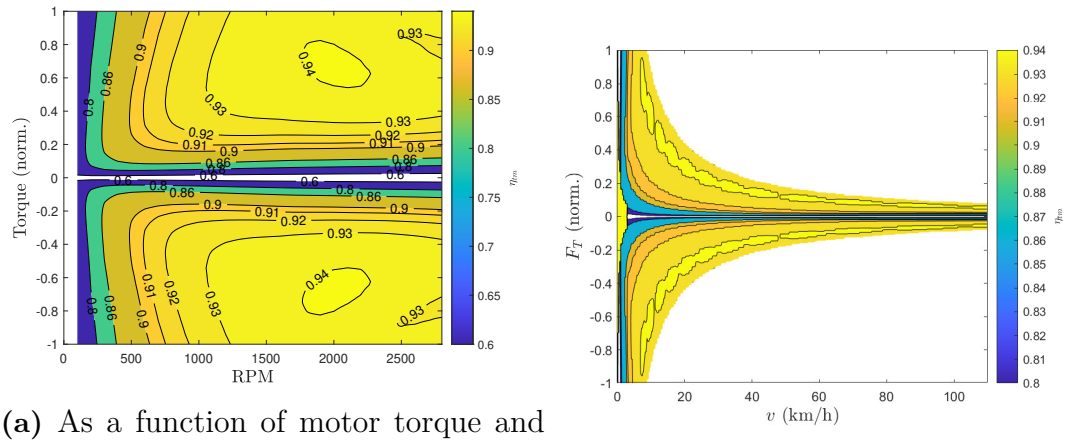
Electric machine

The EM converts electric energy into tractive effort, driving the vehicle in the desired direction. The conversion from motor torque τ_{em} to traction force F_T depends on the gear ratio k_{em} of the selected gear γ, the fixed gear ratio of the transmission k_{tm}, the transmission efficiency η_{tm} and the effective tyre radius R as such

$$F_T = \frac{\eta_{tm} k_{em}(\gamma) k_{tm} \tau_{em}}{R}. \tag{2.9}$$

The efficiency η_{tm} depends on τ_{em} and the motor RPM. An efficiency map can be seen in Figure 2.2a. In turn, the RPM depends on the vehicle speed and the gear selection. The field of optimising gear selection has been well studied [18].

Inherently it is an integer problem which brings complexity to the NLP layer. It is subsequently assumed that a low-level controller performs gear selection to achieve close to optimal efficiency. By doing so, the efficiency mapping can be made directly from vehicle speed to traction effort, as in Figure 2.2b.



(a) As a function of motor torque and RPM.

(b) At optimal gear.

Figure 2.2: Transmission efficiency maps as a function of motor torque and RPM and at optimal gear.

The direct relationship between τ_{em} and motor power P_{em} is given by the angular speed of the wheels as

$$\tau_{em} = \frac{\eta_{em} P_{em}}{\omega_{em}}, \tag{2.10}$$

which, under the assumption that there is zero slip between the wheels and the road, is a function of the vehicle speed as such

$$\tau_{em} = \frac{\eta_{em} P_{em} R}{k_{em} k_{tm}(\gamma) v}. \tag{2.11}$$

Putting this into equation (2.9) shows the relation between motor input power P_{em} from the DC-bus, vehicle speed v, and traction force F_T to be

$$F_T = \eta_{tm} \eta_{em} \frac{P_{em}}{v}. \tag{2.12}$$

For more convenience when solving the problem using numerical methods, the efficiency can be represented by an EM loss term

$$F_T = \frac{P_{em} - P_{em,loss}}{v}, \tag{2.13}$$

where $P_{em,loss} = (1 - \eta_{tm}\eta_{em}) P_{em}$ and $P_{em,loss} > 0$ for both positive and negative motor powers.

Fuel cell

A fuel cell is an electrochemical system that is intended to convert the chemical energy of a fuel (hydrogen in this case) and an oxidising agent (oxygen in this case) into electricity through a pair of redox reactions. Polymer electrolyte membrane fuel cells (PEMFC) utilise a membrane that is permeable for protons, but not for the reactant gases and electrons [26]. Thanks to that property, this kind of fuel cell is also known as a proton-exchange membrane fuel cell. As shown in Figure 2.3, hydrogen enters the fuel cell at the anode and splits into protons and electrons. The electrical chemical reaction that occurs at the anode is

$$\mathrm{H_2} \longrightarrow 2\,\mathrm{H^+} + 2\,\mathrm{e^-}. \tag{2.14}$$

The protons pass through the membrane while the electrons flow through an external circuit, producing an electric current. After that, the oxygen entering the cathode is combined with the resulting protons and electrons to produce water (either gas or liquid) and heat. The reaction on the cathode's side is thus

$$\mathrm{O_2} + 4\,\mathrm{H^+} + 4\,\mathrm{e^-} \longrightarrow 2\,\mathrm{H_2O}. \tag{2.15}$$

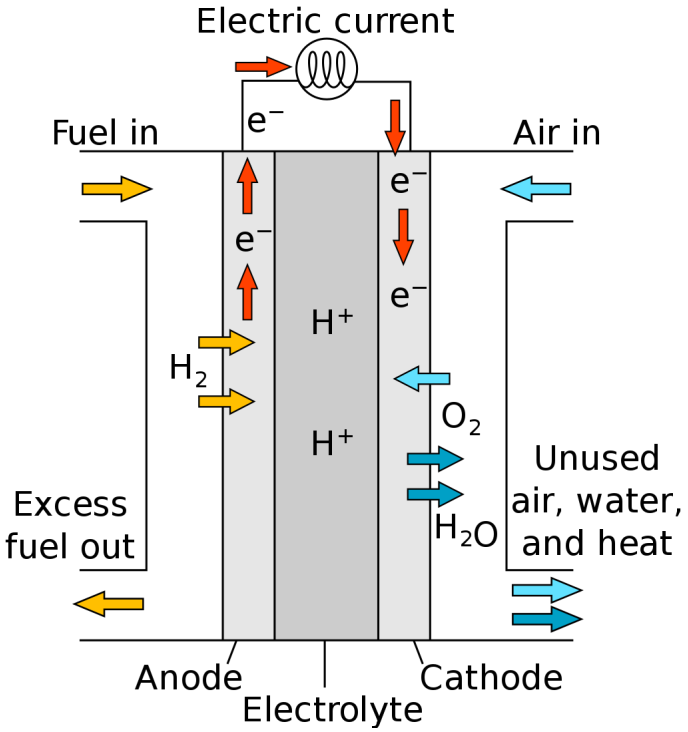


Figure 2.3: Illustration of PEMFC's principle of operation. [27], CC0

The entire reaction that occurs in the PEMFC is

$$2\,\mathrm{H_2} + \mathrm{O_2} \longrightarrow 2\,\mathrm{H_2O}. \tag{2.16}$$

In the context of the problem, knowing how a fuel cell's hydrogen consumption and efficiency vary depending on its power is important. According to [28], hydrogen

consumption can be approximated as a quadratic function of the fuel cell's electrical power P_{fc} as follows

$$\mathrm{H}_{2,con}(P_{fc}) = -(a_2 P_{fc}^2 + a_1 P_{fc} + a_0), \tag{2.17}$$

where $a_0, a_1, a_2 \geq 0$. Hydrogen power can then be written as

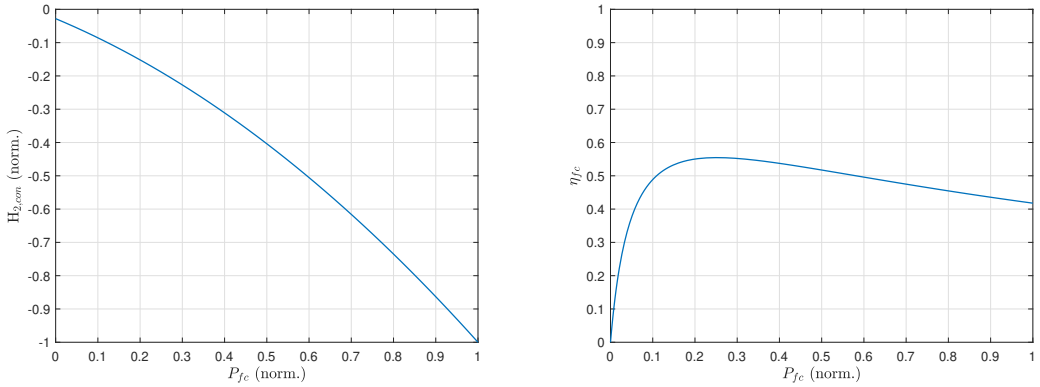
$$P_{\mathrm{H}_2}(P_{fc}) = \mathrm{H}_{2,con}(P_{fc})\Delta H = -(a_2 P_{fc}^2 + a_1 P_{fc} + a_0)\Delta H, \tag{2.18}$$

where ΔH is hydrogen's lower heating value.

The efficiency of a fuel cell represents the share of chemical (hydrogen) energy that is converted into electrical energy. The heat that is emitted during the chemical reaction is considered to be a loss and most of it is removed by the cooling system. Thus, fuel cell efficiency can be written as

$$\eta_{fc}(P_{fc}) = \frac{P_{fc}}{|P_{\mathrm{H}_2}(P_{fc})|} = \frac{P_{fc}}{(a_2 P_{fc}^2 + a_1 P_{fc} + a_0)\Delta H}. \tag{2.19}$$

The fuel cell hydrogen power and efficiency curves can be seen in Figures 2.4a and 2.4b, respectively.



(a) Hydrogen power. **(b)** Fuel cell efficiency.

Figure 2.4: Hydrogen power and fuel cell efficiency, as functions of fuel cell power.

Accounting for the hydrogen consumption by the fuel cell and the amount refuelled at hydrogen stations, the differential equation of the hydrogen level in the tank can be written as follows

$$\dot{\mathrm{H}}_2 = \begin{cases} \mathrm{H}_{2,con}, & \mathrm{H}_{2,rf} > 0 \\ \mathrm{H}_{2,con} + \delta_{st}\dfrac{C_{\mathrm{H}_2}}{t_{rf}}, & \mathrm{H}_{2,rf} = 0 \end{cases}, \tag{2.20}$$

which after substitution becomes

$$\dot{\mathrm{H}}_2 = \begin{cases} -(a_2 P_{fc}^2 + a_1 P_{fc} + a_0), & \mathrm{H}_{2,rf} > 0 \\ -(a_2 P_{fc}^2 + a_1 P_{fc} + a_0) + \delta_{st}\dfrac{C_{\mathrm{H}_2}}{t_{rf}}, & \mathrm{H}_{2,rf} = 0 \end{cases}, \tag{2.21}$$

where $H_{2,rf}$ is the amount of hydrogen refuelled at a hydrogen station, C_{H_2} is the capacity of the vehicle's hydrogen tank, t_{rf} is the time needed to refuel the hydrogen tank to the fullest, δ_{st} is a binary variable identifying the presence of a hydrogen station nearby at the current time. The refuelling time of FCEVs is usually below 15 minutes [29], which makes it reasonable to neglect the time dynamics of refuelling and consider the refuelling rate constant.

It is important to note that operation with low fuel cell power P_{fc} should be avoided as it leads to degradation of the catalyst in the fuel cell [30]. At the same time, operating at high power levels is also detrimental to the fuel cell as it may incur fuel starvation, which degrades both the catalyst and gas diffusion layers in the fuel cell through carbon oxidation [31]. Constraining the fuel cell to work within a desired operating region can thus lengthen its lifespan.

Since the fuel cell must operate in a very narrow temperature range around 70 °C, the heat generated during its operation must be systematically removed [32]. The heat generated by the fuel cell system can be approximated as

$$P_{fc,heat}(P_{fc}) = c_2 P_{fc}^2 + c_1 P_{fc} + c_0 \tag{2.22}$$

where $c_0, c_1, c_2 \geq 0$.

Battery

The battery model is comprised of lithium-titanium-oxide (LTO) cells. These offer a faster charge/discharge cycle and a longer life cycle than other lithium-based batteries at the cost of lower terminal voltage [33]. Connecting several of these cells in parallel yields a battery pack with a capacity C_b. The normalised battery charge, the so-called SoC is defined as the relation between the current battery charge q and its capacity

$$SoC = \frac{q}{C_b}. \tag{2.23}$$

The SoC's dynamics can then be described as

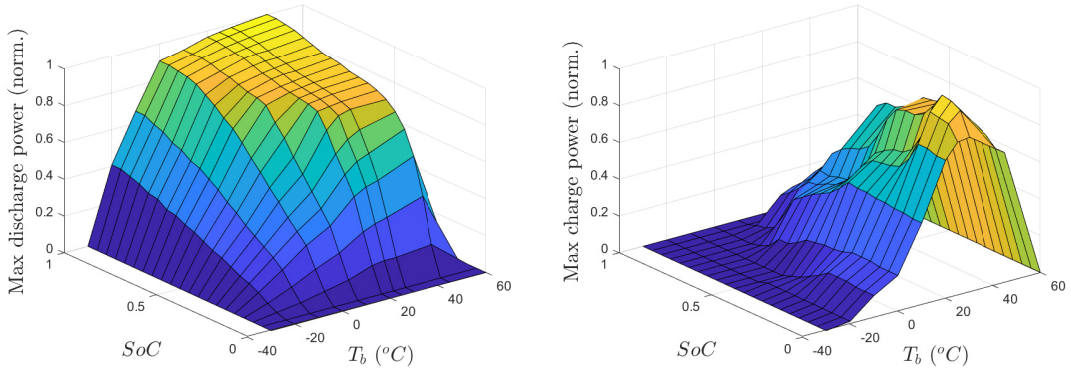
$$\dot{SoC} = \frac{I_b}{C_b}, \tag{2.24}$$

where I_b is the battery current. As the power of any circuit equals voltage multiplied by the current in it, the relation between the normalised battery discharge rate $\dot{SoC}$ and the output power of the battery P_b is given by

$$\dot{SoC} = \frac{-P_b}{C_b U_{oc}(SoC, T_b)}, \tag{2.25}$$

where the battery open circuit voltage U_{oc} is a function of SoC and the battery's internal temperature T_b that is shown in Figure 2.6a, and P_b is the battery power that lies within an interval that is determined by SoC and T_b: $P_b \in [P_{b,min}(SoC, T_b), P_{b,max}(SoC, T_b)]$. P_b is positive when the battery is discharging, and negative when charging. The upper bounds on the battery's discharge and

charge powers as functions of battery temperature and SoC are shown in Figures 2.5a and 2.5b, respectively.



(a) Discharge power bound. **(b)** Charge power bound.

Figure 2.5: Normalised charge and discharge power bounds, as functions of battery temperature and state of charge.

Battery temperature can be modelled with the help of the heat balance equation

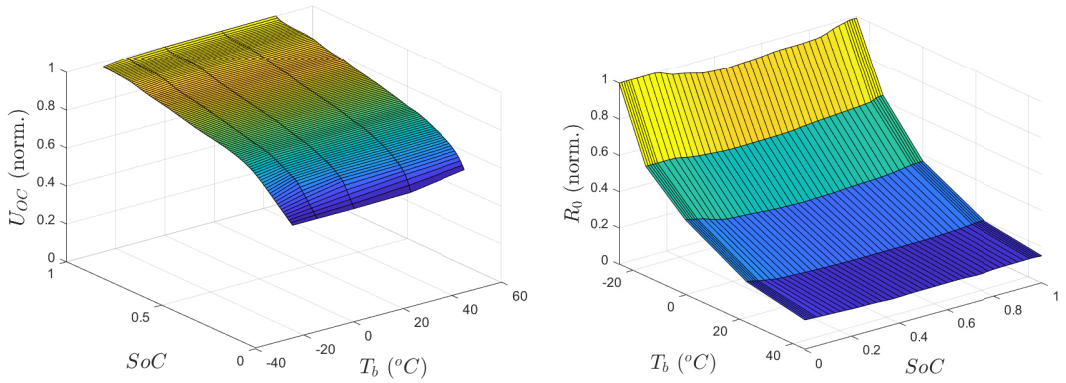
$$\dot{T}_b = \frac{Q_{pass} + Q_{act} + Q_{amb}}{c_b m_b} \tag{2.26}$$

where c_b and m_b are the battery's specific heat capacity and total mass, respectively, Q_{pass} is the rate of the heat generated by passive heat sources affecting the battery's temperature, Q_{act} is the active heat rate supplied to or removed from the battery by external auxiliary components that are intended to control battery's temperature, and Q_{amb} is the rate of the heat dissipated into the ambient air. It is worth noting that the incorporated model assumes a uniform distribution of temperature along the battery, which substantially reduces the complexity of the problem.

Passive heat ultimately consists of the heat losses generated due to internal resistance in the battery. These losses can be modelled according to Joule's law as

$$Q_{pass} = R_b(SoC, T_b) I_b^2 = R_b(SoC, T_b) \frac{P_b^2}{U_{oc}^2(SoC, T_b)}, \tag{2.27}$$

where R_b is the battery's internal resistance which significantly decreases at higher battery temperatures as one can see in Figure 2.6b.



(a) Open circuit voltage. **(b)** Internal resistance.

Figure 2.6: (a): Normalised open circuit voltage and internal resistance, as functions of battery temperature and SoC.

Active heat generation rate can be defined as

$$Q_{act} = \eta_{th} P_{th}, \tag{2.28}$$

where P_{th} is the power of the battery thermal system and η_{th} is its efficiency. P_{th} is positive when heat is supplied to and negative when heat is removed from the battery.

Heat dissipated into the ambient air can be modelled using Fourier's law of heat conduction

$$Q_{amb} = k_{amb}(T_{amb} - T_b), \tag{2.29}$$

where T_{amb} is the temperature of the ambient air and $k_{amb} > 0$ is the coefficient of heat exchange between the battery's case and air, which normally is dependent on the vehicle's speed but in this work will be chosen as constant for simplicity.

The battery modelling is based on [15] and [34], so for more detailed derivations check out those.

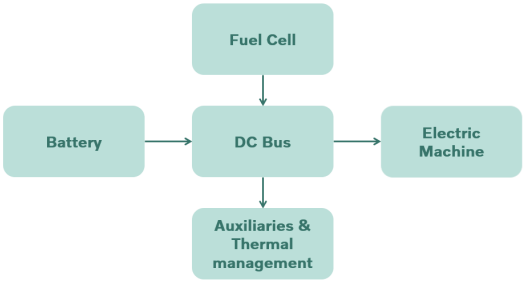
Power balance equations

The fuel cell and battery supply power to the DC Bus, which then passes it to the EM, battery and fuel cell thermal systems and other auxiliary loads. The battery can be charged either through regenerative braking or by the fuel cell. The battery cannot be charged and discharged at the same time. There are 4 different cases of power flow in the vehicle:

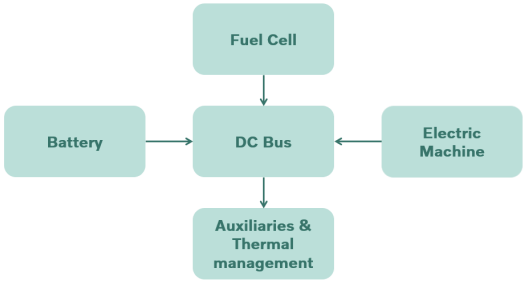
(a) Both the battery and the fuel cell provide power to the EM and for the auxiliaries and thermal management needs as in Figure 2.7a.

(b) The vehicle is braking regeneratively but all the power goes to the auxiliaries and thermal management needs as in Figure 2.7b. This case is not realistic as the fuel cell's minimum operating power is always larger than the auxiliaries and thermal management needs.

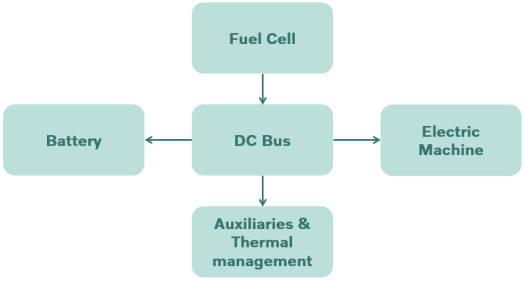
(c) The battery is charging from the fuel cell as in Figure 2.7c.
(d) The battery is charging from the regenerative brakes as in Figure 2.7d.



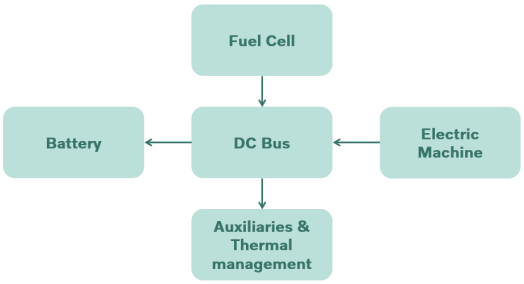
(a) Both the battery and the fuel cell act as power sources.



(b) All the power goes to the auxiliaries and thermal management.



(c) Battery charging from the fuel cell.



(d) Battery charging from the regenerative brakes.

Figure 2.7: Schemes representing the cases of the vehicle's power flow.

The power balance equation can be written as

$$P_{em} = P_{fc} + P_b - P_{fc,heat} - |P_{th}| - Q_{pass} - P_{aux}, \tag{2.30}$$

where P_{aux} is the auxiliary power mainly used for air conditioning in the cabin. Combining this with equations (2.13) and (2.30) gives an equation for traction effort

$$F_T = \frac{1}{v}(P_{fc} + P_b - P_{fc,heat} - |P_{th}| - Q_{pass} - P_{aux} - P_{em,loss}). \tag{2.31}$$

2.2 Optimal control

A common way of describing a system with nonlinear dynamics in continuous time is by using a system of differential equations of the form

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t)), \tag{2.32}$$

where t is time, $\mathbf{x}$ is the state vector and $\mathbf{u}$ is the (control) input vector. The state vector contains, at minimum, all the information necessary to unambiguously describe the current state of the system. The input vector represents an external impact coming from an actuator that is intended to affect the state of the system.

Optimal control aims to find an optimal control input trajectory $\mathbf{u}^*(t)$ and a corresponding optimal state trajectory $\mathbf{x}^*(t)$ that minimises objective, or also called cost, function J. Usually, the actuator is limited by some region of operation, control constraint set $\mathbb{U} \in \mathbb{R}^m$, which the control trajectory must fulfil. The same applies to the state vector which must lie within a constraint set $\mathbb{X} \in \mathbb{R}^n$. In addition, if the system is analysed in a set time interval $[t_0, t_N]$, there will also appear an initial state $\mathbf{x}(t_0) = \mathbf{x}_0$ and a terminal set $\mathbb{X}_f$ which $\mathbf{x}(t_N)$ must belong to. Those control trajectories, which meet the control and state constraints and lead the state trajectory to the target set $\mathbb{X}_f$ are called *feasible*. It is worth mentioning that the control constraint set $\mathbb{U}$ largely depends on the initial state $\mathbf{x}_0$, but also that it is generally not fixed as it may vary with time t and the state value $\mathbf{x}$. The same goes for the state constraints $\mathbb{X}$ as they can also be time-dependent. All the aforementioned constraints can be expressed as general inequality constraints of the form $\mathbf{g}(t, \mathbf{x}(t), \mathbf{u}(t)) \leq 0$. So finally we can mathematically formulate the continuous-time optimal control problem as

$$\begin{aligned}
\underset{\mathbf{x},\,\mathbf{u}}{\text{minimize}} \quad & J(t, \mathbf{x}(t), \mathbf{u}(t)) && \text{(2.33a)}\\
\text{subject to} \quad & \dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t)), && \text{(2.33b)}\\
& \mathbf{x}(t_0) = \mathbf{x}_0, && \text{(2.33c)}\\
& \mathbf{g}(t, \mathbf{x}(t), \mathbf{u}(t)) \leq 0, && \text{(2.33d)}
\end{aligned}$$

Optimal control problems are typically solved numerically with the help of computational software, therefore the dynamics described in (2.32) must be discretised, which can be done by solving the initial value problem (IVP) described in (2.34) using integration methods such as Euler or other higher order Runge-Kutta methods. After discretisation, the continuous objective function, dynamics and constraints need to be evaluated at $N + 1$ time instances $t = t_0, t_1, ..., t_N$ only.

$$\mathbf{x}(t_n) = \mathbf{x}_n, \quad \dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t)), \quad \text{for} \quad t \in [t_n, t_{n+1}] \tag{2.34}$$

It is assumed in this work that $\mathbf{u}$ is a piece-wise constant vector, that is

$$\mathbf{u}(t) = \mathbf{u}(t_n), \quad \forall\, t \in [t_n, t_{n+1}). \tag{2.35}$$

2.2.1 Stochastic optimal control

Commonly, system dynamics are modelled to contain parameters representing some sort of uncertainty. These can stem from external factors or internal mismatches in the modelled and actual dynamics. In the presence of disturbance with uncertain behaviour, here denoted $\mathbf{w}$, the aim is to find the control input trajectory $\mathbf{u}^*(t)$ that minimises the expected value of the cost function.

$$\underset{\mathbf{x},\,\mathbf{u}}{\text{minimize}} \quad \mathbb{E}\left[J(t,\mathbf{x}(t),\mathbf{u}(t))\right] \tag{2.36a}$$

$$\text{subject to} \quad \dot{\mathbf{x}}(t) = \mathbf{f}(t,\mathbf{x}(t),\mathbf{u}(t),\mathbf{w}(t)), \tag{2.36b}$$

$$\mathbf{x}(0) = \mathbf{x}_0, \tag{2.36c}$$

$$\mathbf{g}(t,\mathbf{x}(t),\mathbf{u}(t)) \leq 0, \tag{2.36d}$$

$$\mathbf{w}(t) \sim f_{\mathbf{w}}(\mathbf{w}(t)), \tag{2.36e}$$

where $f_{\mathbf{w}}$ is the probability density function (pdf) of the uncertainty.

2.2.2 Mixed-integer optimisation

If the problem described in equation (2.36) has one or several variables that are restricted to be an integer, for instance, $u_i \in \mathbb{U}_i \subset \mathbb{Z}$, the cost function becomes non-smooth and non-differentiable, forcing one to solve a separate optimal control problem for many combinations of the discrete variables, which is, of course, very computationally demanding. However, dynamic programming is one of the algorithms that can handle such problems without any increase in the computational burden and more about it is written in the next subsection.

2.3 Dynamic programming

Invented by Richard Bellman, dynamic programming (DP) has been used as an optimisation method since the 1950s and, naturally, has also found application in optimal control [35, 36]. As with optimal control, its purpose is to find a feasible optimal control trajectory $\mathbf{u}^*(t)$ for any initial state $\mathbf{x}_0$ over the time interval $[t_0, t_N]$. As there are infinite time instances in the considered time interval, discretising the system is necessary, as mentioned in section 2.2.

By discretising dynamical system for $K+1$ time instances t in $[t_0, t_K]$ the problem can be considered to be split into K subproblems, where for problem $k = 0, \ldots, K-1$ the initial state $\mathbf{x}_0 = \mathbf{x}_k$ is constrained by $\mathbb{X}(t_k)$, likewise, for the terminal state $\mathbf{x}_N = \mathbf{x}_{k+1} \in \mathbb{X}(t_{k+1})$, and the optimal input $\mathbf{u}_k^*$ minimises the cost function $J(t,\mathbf{x},\mathbf{u})$ over the interval $[t_k, t_{k+1}]$. This requires that the cost function J can also be split, such that over a time interval $[t_k, t_{k+1}]$, J only depends on the input $\mathbf{u}$ and state $\mathbf{x}$ trajectories within that interval. In more abstract terms, the cost at time t_k can be expressed as the cost of getting from $\mathbf{x}_k$ to $\mathbf{x}_{k+1}$ and the cost associated with getting from $\mathbf{x}_{k+1}$ to $\mathbf{x}_K$.

If it is possible splitting the problem in such a way the necessary conditions for DP are fulfilled, and an action-value equation can be introduced, defined by the Bellman equation

$$Q(t,\mathbf{x},\mathbf{u}) = L(t,\mathbf{x},\mathbf{u}) + V(t^+,\mathbf{x}^+), \tag{2.37}$$

where the first term L is commonly referred to as the stage cost, that is the cost associated with taking the action $\mathbf{u}$ at time t in state $\mathbf{x}$ and V is the optimal cost of

the remaining subproblems, meaning V is the optimal solution to the action-value function Q at the next time instance t^+ and the resulting state values $\mathbf{x}^+$ from taking control action $\mathbf{u}$, expressed as

$$V(t, \mathbf{x}) = \min_{\mathbf{u}} Q(t, \mathbf{x}, \mathbf{u}). \tag{2.38}$$

The control input $\mathbf{u}$ that minimises the action-value function at time t in state $\mathbf{x}$ is referred to as the optimal policy, given by

$$\boldsymbol{\pi}(t, \mathbf{x}) = \operatorname*{argmin}_{\mathbf{u}} Q(t, \mathbf{x}, \mathbf{u}), \tag{2.39}$$

which, when combined with equation (2.38) gives the Bellman equation for the value function as

$$V(t, \mathbf{x}) = L(t, \mathbf{x}, \boldsymbol{\pi}(t, \mathbf{x})) + V(t^+, \mathbf{x}^+), \tag{2.40}$$

expressing the value in state $\mathbf{x}$ at time t.

The purpose of DP is to consecutively determine the value function and the policy for all states and for each time step to then save this policy as a look-up table allowing one to determine the optimal input trajectory $\mathbf{u}^*(t)$ for any given initial value $\mathbf{x}(t_0)$ by iteratively looking up the policy $\boldsymbol{\pi}(\mathbf{x})$ and integrating the dynamics to attain $\mathbf{x}^+$. Noting how the action-value function depends on the value for future actions but is independent from actions take at previous time instances one can conclude that the policy must be determined recursively, starting at t_K and stepping backwards in time, commonly known as backward induction. The backward induction is initialized by defining the value at the final time instance t_K as

$$V(t_K, \mathbf{x}) = T(\mathbf{x}), \tag{2.41}$$

where T is common notation for the terminal cost function. Thus, the cost function J has been split according to

$$J(t, \mathbf{x}(t), \mathbf{u}(t)) = \sum_{i=1}^{K-1} L(t_i, \mathbf{x}(t_i), \mathbf{u}(t_i)) + T(\mathbf{x}(t_K)). \tag{2.42}$$

2.3.1 Stochastic dynamic programming

In the presence of external stochastic factors in the system, the model dynamics become uncertain such that the continuous system dynamics become

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t)), \tag{2.43}$$

thus making $\mathbf{x}^+$ and therefore $V(t^+, \mathbf{x}^+)$ stochastic variables in the value function in equation (2.40), which require additional measures to be taken in the approach described in the previous section. In order to still fulfill the criteria for DP the uncertainty $\mathbf{w}$ of the system must have the Markov property, meaning the uncertainty of the system at a certain time instance can only depend on the state $\mathbf{x}$ and input $\mathbf{u}$ at that time instance. This makes the optimal control problem a Markov decision process, and using DP to solve it is referred to as stochastic dynamic programming

(SDP)[37].

Since the value $V(\mathbf{x})$ is now stochastic, its values in the Bellman equations (2.37) and (2.40) are represented by the expected values

$$Q(t, \mathbf{x}, \mathbf{u}) = L(t, \mathbf{x}, \mathbf{u}) + \mathbb{E}\left[V(t^+, \mathbf{x}^+) \mid t, \mathbf{x}, \mathbf{u}\right]. \tag{2.44a}$$

$$V(t, \mathbf{x}) = L(t, \mathbf{x}, \boldsymbol{\pi}(\mathbf{x})) + \mathbb{E}\left[V(t^+, \mathbf{x}^+) \mid t, \mathbf{x}, \boldsymbol{\pi}\right], \tag{2.44b}$$

To calculate the expected value it is convenient to introduce a probability density function that would describe the conditional probability that the system's next state value will be $\mathbf{x}^+$ given that the system currently is at state $\mathbf{x}$, the action $\mathbf{u}$ is taken and the stochastic uncertainty $\mathbf{w}$ affects the system, at time instance t. Written mathematically, that would be $f_{\mathbf{x}^+|\,\mathbf{t},\mathbf{x},\mathbf{u}}(\mathbf{x}^+)$, which together with the definition of mathematical expected value can be used to write the Bellman equations for the stochastic case as

$$Q(t, \mathbf{x}, \mathbf{u}) = L(t, \mathbf{x}, \mathbf{u}) + \int_{-\infty}^{\infty} V(t^+, \mathbf{x}^+) f_{\mathbf{x}^+|\,\mathbf{t},\mathbf{x},\mathbf{u}}(\mathbf{x}^+) d\mathbf{x}^+, \tag{2.45a}$$

$$V(t, \mathbf{x}) = L(t, \mathbf{x}, \boldsymbol{\pi}(\mathbf{x})) + \int_{-\infty}^{\infty} V(t^+, \mathbf{x}^+) f_{\mathbf{x}^+|\,\mathbf{t},\mathbf{x},\mathbf{u}}(\mathbf{x}^+) d\mathbf{x}^+. \tag{2.45b}$$

3
Methods

This chapter details how the hierarchical strategy was set up, how the route was sampled, how cost functions were modelled and how the strategy evaluation was set up.

3.1 Solution strategy

Before going to concrete mathematical formulations of the problems to solve, it is worth describing the general strategy that was used to solve the stated problem of mission management.

As mentioned before, the problem contains binary variables corresponding to decisions to stop at hydrogen stations along the route. These decision variables make the objective function discontinuous and, therefore non-differentiable and unapproachable as an NLP. Thus it becomes a mixed-integer nonlinear program (MINLP) that has a combinatorial nature and is very computationally expensive to solve as the solution lies in solving a separate NLP problem for many combinations of the binary variables as stated in 2.2.2.

In this context, the work in this thesis seeks to investigate the benefits and drawbacks of an approach with the problem decomposed into two parts. The first part, the so-called top layer, was to solve the MINLP problem with a method that can handle binary variables, in this case SDP. Since dynamic programming explores every possible combination of states and inputs its computational complexity increases exponentially with the number of states and inputs, as well as stochastic variables. Due to this the mathematical model used in this layer had to undergo some simplifications to account for that. One of the main simplifications was the exclusion of the thermal management of the battery from the scope of the optimisation, thus making battery temperature a fixed parameter for the entire route. As illustrated in Figure 3.1, the layer takes mission and vehicle parameters as input and outputs the optimal station selection, eliminating this decision from the second part, the so-called bottom layer.

Since the binary decisions are taken the problem is no longer mixed-integer and a global NLP optimisation can be conducted with more accurate and comprehensive dynamics, more inputs and more states. This layer outputs the final speed, battery temperature and power split trajectories. Unlike the top layer, the bottom

layer does not account for any uncertain parameters.

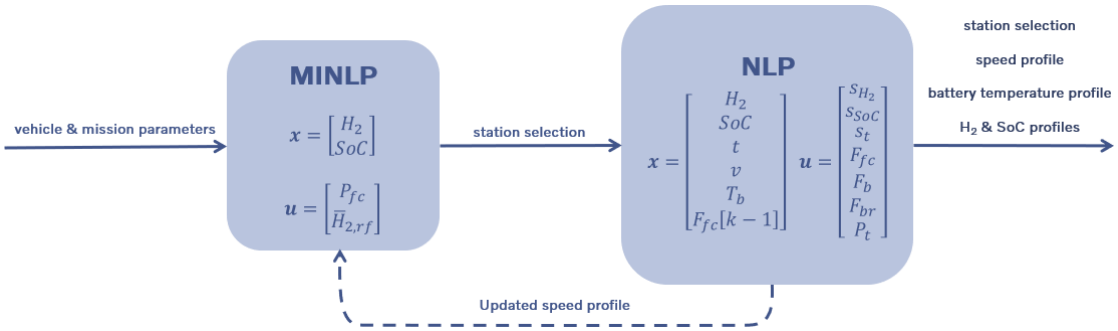


Figure 3.1: Block diagram illustrating the hierarchical approach.

It is essential to note that in order to reduce the number of states, speed was not optimised in the top layer and a predefined base speed profile was calculated instead. Therefore, a feedback loop with an optimised speed profile from the bottom layer to the top layer was implemented as an option. This feedback is illustrated with a dashed line in Figure 3.1.

3.2 Discretisation

As mentioned in section 2.2, the problem needs to be discretised to be solved numerically by a computer. The problem is split into a number of stages, K for the top layer, and N for the bottom layer, for which an optimisation problem is solved separately. The route thus has to be divided into K and N road segments respectively, over which the dynamics are integrated in the solving process. This section describes how the problem is discretised and how the data is sampled to reach the desired number of data points.

3.2.1 Space sampling

Since the positions of hydrogen stations are fixed along the route, it poses a more convenient solution to solve the problem in the spatial domain, meaning that all the time derivatives written in section 2.1 have to be rewritten as spatial ones. For the rest of this paper, the notation x' will represent the space derivative of an arbitrary variable x, i.e. $x' = dx/ds$, where s is the vehicle's position along the route. The transition from the time derivative $\dot{x}$ to the space domain is done using the equation

$$x' = \frac{dx}{ds} = \frac{dx}{dt}\frac{dt}{ds} = \frac{\dot{x}}{v}. \tag{3.1}$$

The differential equations (2.1), (2.21), (2.25) and (2.26) thus become

$$mv' = \frac{1}{v}(F_T - F_{rr} - F_{air} - F_g - F_{br}), \tag{3.2a}$$

$$\mathrm{H}_2' = -\frac{1}{v}(a_2 P_{fc}^2 + a_1 P_{fc} + a_0) + \delta_{st}\delta_{Dirac}(s)\mathrm{H}_{2,rf}, \tag{3.2b}$$

$$SoC' = \frac{-P_b}{vC_bU_{oc}(SoC, T_b)}, \tag{3.2c}$$

$$T_b' = \frac{Q_{pass} + Q_{act} + Q_{amb}}{vc_bm_b}. \tag{3.2d}$$

$\delta_{Dirac}(s)$ in (3.2b) is the Dirac delta function needed to demonstrate that refuelling happens instantly in the space domain. Spatial sampling will be used for both layers in the control hierarchy. The Bellman equations for stochastic dynamic programming, given in (2.44), will also have to be rewritten in the space domain

$$Q(s, \mathbf{x}, \mathbf{u}) = L(s, \mathbf{x}, \mathbf{u}) + \mathbb{E}\left[V(s^+, \mathbf{x}^+) \mid s, \mathbf{x}, \mathbf{u}\right]. \tag{3.3a}$$

$$V(s, \mathbf{x}) = L(s, \mathbf{x}, \boldsymbol{\pi}(\mathbf{x})) + \mathbb{E}\left[V(s^+, \mathbf{x}^+) \mid s, \mathbf{x}, \boldsymbol{\pi}\right], \tag{3.3b}$$

3.2.2 Data downsampling

Originally, road data parameters such as position, altitude, speed limits and weather are provided at a much higher resolution than the desired number of stages in the solver. Hence, the data must be downsampled to K and N data points, a process that can have a significant effect on the simulation's accuracy. Unless the original number of data points is evenly divisible by the desired data points the data can not be downsampled into segments of equal distance, so-called equidistantly, without some sort of mathematical interpolation. The process of interpolating and then downsampling data introduces data points that were originally not part of the data set which risks decreasing the accuracy of the data.

Additionally, different road segments affect the dynamics differently. A long flat road segment requires only two data points to be accurately described, while a hilly segment of the same length requires more data points to be described with equal accuracy. The parameter with the largest effect on dynamics is altitude. By downsampling the road data using an algorithm that preserves the highest detail in the altitude profile, while not introducing interpolated data points, the simulation should in theory become more accurate with the same number of data points N.

Such an algorithm would be computationally intensive, however, an algorithm aiming to achieve close to optimal results can have a much lower computation time at the trade-off of optimality. One created by the authors of [4] was used to downsample data non-equidistantly to preserve the detail in altitude with the intention of decreasing computation time. In essence, the algorithm splits the data into windows and eliminates redundant data points. A way to evaluate the detail retention is to upsample to the original resolution using first-order hold and calculate the mean squared error (MSE). Another proposed metric is the cumulative error in energy consumption when simulating a truck driving the downsampled routes compared to the original route.

A comparison of equidistant and non-equidistant downsampling was done for two routes. For the first route, of length 1793 km, the data was downsampled to

$N = 100, \ldots, 1000$ stages, while for the second route, of length 461 km, the data was downsampled to $N = 100, \ldots, 500$ stages. To calculate the energy consumption a speed profile was generated by generating a noisy speed profile around 70 km/h and integrating equation (3.2a) using explicit Euler.

Additional parameters such as speed limits and station positions were considered, but ultimately no changes were made to the algorithm. This means that currently speed limit changes and hydrogen stations are positioned at the longitudinally closest data point post-downsampling.

3.3 Problem formulations

In this section the models and cost functions of the two layers are detailed.

3.3.1 Top layer – stochastic dynamic programming

The purpose of the top layer is to select stations from the available ones at which to stop and fuel during the mission and decide how much hydrogen should be refuelled.

Mathematical model

States and inputs of the system will thus be

$$\mathbf{x}(s) = \begin{bmatrix} SoC \\ \bar{\mathrm{H}}_2 \end{bmatrix}, \qquad \mathbf{u}(s) = \begin{bmatrix} P_{fc} \\ \bar{\mathrm{H}}_{2,rf} \end{bmatrix}, \tag{3.4}$$

where $\bar{\mathrm{H}}_2 = \mathrm{H}_2/C_{\mathrm{H}_2}$ is the normalised hydrogen level in the tank and $\bar{\mathrm{H}}_{2,rf} = \mathrm{H}_{2,rf}/C_{\mathrm{H}_2}$ is the normalised amount of hydrogen refuelled at a hydrogen station. Note that if $\bar{\mathrm{H}}_{2,rf} = 0$, the vehicle should pass the station without refuelling, which implies that this input doubles as a binary variable.

As vehicle speed is neither a state nor an input it must be deterministic, or modelled as a stochastic variable. Preprocessing of vehicle and road data, such as speed limits, power output limits, road grade, desired arrival time and maximum risk of failure is done to produce a predefined base speed profile. The algorithm for this can be found in Appendix A.1.

The dynamics used to find $\mathbf{x}^+$ in the SPD algorithm in equation (3.3) start with the power input to the EM, given by combining equations (3.2a) and (2.12)

$$P_{em} = \eta_{tm}\eta_{em} v \left(v' m v + F_{rr} + F_g + F_{air}\right). \tag{3.5}$$

As road parameters are known, as well as a predefined speed trajectory, the input power to the EM can be determined and is henceforth referred to as the power demand, as it governs how much power needs to be output by the DC bus to meet the demand from the machine to fullfil the differential equations. One can notice that

P_{em} depends on several external factors, such as road grade, traffic speed, wind direction and speed and rolling resistance which all display a random behaviour in real life, or are impossible to predict flawlessly hours in advance. This makes modelling P_{em} as a single stochastic variable a suitable choice, to represent the uncertainty while reducing the computational complexity. In short, a number of vehicle and mission-specific parameters determine the shape of the probability distribution of P_{em}, which differs for each spatial step. A more detailed description of P_{em} modelling is done later in this section.

Assuming constant battery temperature, the vehicle's power balance (**??**) can be simplified to

$$P_{fc} + P_b = P_{fc,heat} + P_{aux} + P_{b,loss} + P_{em} + P_{em,loss}, \tag{3.6}$$

where, as battery temperature is outside of the optimisation scope in this layer, battery losses are modelled as a quadratic fit independent of the battery temperature

$$P_{b,loss} = (b_0 - b_1 SoC)P_b^2, \qquad b_0, b_1 \geq 0. \tag{3.7}$$

This makes equation (3.6) quadratic in P_b and can for a given P_{em} and a selected P_{fc} be solved to find P_b. If, at a certain spatial step, no solution exists, or the solution exceeds the state of charge-dependent power output limit $P_{b,max}(SoC)$ the power demand can not be met for the given P_{fc} and it is not considered for the policy $\boldsymbol{\pi}$. If the solution exceeds the regenerative charging power limit $P_{b,min}(SoC)$, or the power needed to recharge the battery fully, given from inverting (3.9), another term is added to the power balance, namely the braking power P_{br}, to allow for a solution to be found, given the braking power is minimised. Equation (3.6) then becomes

$$P_{fc} + P_b - P_{br} = P_{fc,heat} + P_{aux} + P_{b,loss} + P_{em} + P_{em,loss}. \tag{3.8}$$

It should be noted that friction brakes P_{br} are applied independently of the EM and do not induce additional losses in the EM, thus this representation is not perfectly accurate. Finally, after P_b has been calculated, the next state values $\mathbf{x}^+$ can be found by integrating differential equations

$$\mathbf{x}' = \begin{bmatrix} SoC' \\ \bar{\mathrm{H}}_2' \end{bmatrix} = \begin{bmatrix} -\dfrac{P_b}{vC_b U_{oc}(SoC)} \\ -\dfrac{1}{vC_{\mathrm{H}_2}}\left(a_2 P_{fc}^2 + a_1 P_{fc} + a_0\right) + \delta_{st}\delta_{Dirac}\bar{\mathrm{H}}_{2,rf} \end{bmatrix}. \tag{3.9}$$

Objective function

The dynamics in section 3.3.1 return three outputs: the battery power consumption P_b, the braking power required P_{br}, and a binary variable $\delta_{\boldsymbol{\pi}}$ indicating whether the input should be considered for the policy or not, determined by its feasibility, as mentioned above.

$$\mathbf{y}(s) = \begin{bmatrix} P_b \\ P_{br} \\ \delta_{\boldsymbol{\pi}} \end{bmatrix} \tag{3.10}$$

Additionally, a separate policy is determined for each of the control inputs, as

$$\boldsymbol{\pi}(s, \mathbf{x}(s)) = \begin{bmatrix} P^*_{fc} \\ \bar{\mathrm{H}}^*_{2,rf} \end{bmatrix}. \tag{3.11}$$

The terminal cost function is defined as

$$T(\mathbf{x}(s_N)) = \mathbf{x}_e^\top \boldsymbol{\varepsilon}_e \mathbf{x}_e + \varepsilon_C, \qquad \mathbf{x}_e = \mathbf{x}(s_N) - \mathbf{x}_{target}, \tag{3.12}$$

where

$$\boldsymbol{\varepsilon}_e = \begin{bmatrix} \varepsilon_{e,SoC} & 0 \\ 0 & \varepsilon_{e,\mathrm{H}_2} \end{bmatrix} \tag{3.13}$$

are the costs of deviating from the desired SoC and H_2 targets and ε_C is the cost of mission completion. The cost at any other spatial point is the sum of the cost of driving and refuelling, as long as the risk of failure is below or equal our risk tolerance κ, as such

$$L(s) = \begin{cases} L_{dr}(s, \mathbf{x}(s), \mathbf{u}(s)) + L_{rf}(s, \mathbf{x}(s), \mathbf{u}(s)), & \Pr\left(\delta_{\boldsymbol{\pi}}(s) \mid \mathbf{u}(s)\right) \leq \kappa \\ \infty, & \Pr\left(\delta_{\boldsymbol{\pi}}(s) \mid \mathbf{u}(s)\right) > \kappa \end{cases}, \tag{3.14}$$

The cost function of driving is in turn the sum of three costs

$$L_{dr}(s, \mathbf{x}(s), \mathbf{u}(s)) = L_{lim}(\mathbf{x}(s)) + L_{tr}(\mathbf{x}(s), \mathbf{u}(s)) + L_{br}(\mathbf{y}(s)), \tag{3.15}$$

where the cost of leaving the state constraint set is

$$L_{lim}(\mathbf{x}(s)) = L_{lim,SoC}(\mathbf{x}(s)) + L_{lim,\mathrm{H}_2}(\mathbf{x}(s)), \tag{3.16}$$

$$L_{lim,x_i}(\mathbf{x}(s)) = \begin{cases} \varepsilon_{llim,x_i} \dfrac{x_{i,min} - x_i(s)}{x_{i,min}}, & x_i(s) < x_{i,min} \\ \varepsilon_{ulim,x_i} \dfrac{x_i(s) - x_{i,max}}{x_{i,max}}, & x_i(s) > x_{i,max} \\ \varepsilon_{fail}, & x_i(s) \notin \mathbb{X}_i = \\ 0, & \text{otherwise} \end{cases}, \tag{3.17}$$

where ε_{ulim,x_i} and ε_{llim,x_i} are the costs of breaching a state's upper and lower bounds $\{x_{i,min}, x_{i,max}\} \subseteq \mathbb{X}_i$, respectively, and ε_{fail} is the cost of failing the mission by exiting the feasible state space $\mathbb{X}_i$, the cost of breaching the maximum allowed level of the fuel cell's transient is

$$L_{tr}(\mathbf{x}(s), \mathbf{u}(s)) = \begin{cases} \varepsilon_{tr}, & |P^*_{fc}(s^+, \mathbf{x}^+(s)) - P_{fc}(s)| > P'_{fc,max} \Delta s(s) \\ 0, & \text{otherwise} \end{cases}, \tag{3.18}$$

where ε_{tr} is the cost of exceeding the fuel cell's transient's maximum allowed level, and the cost of using friction brakes

$$L_{br}(\mathbf{y}(s)) = \varepsilon_{br} \frac{P_{br}(s) \Delta s(s)}{F_{br,max} v(s)}, \tag{3.19}$$

where ε_{br} is the cost of friction braking. The cost function of refuelling is calculated as follows

$$L_r(s, \mathbf{x}(s), \mathbf{u}(s)) = \begin{cases} \delta_{st}(s) \left(\varepsilon_{fuel}(s) \bar{\mathrm{H}}_{2,rf}(s) C_{\mathrm{H}_2} + \varepsilon_{stop}(s) \right) & \bar{\mathrm{H}}_{2,rf}(s) > 0 \\ 0, & \bar{\mathrm{H}}_{2,rf}(s) = 0 \end{cases}, \tag{3.20}$$

where ε_{fuel} and ε_{stop} are the costs of fuel (hydrogen) and stopping, respectively.

Since the output $\mathbf{y}(s)$ is stochastic, making $L(s)$ stochastic, both terms in the action-value function in equation (2.44a) are stochastic and the action-value function is calculated as

$$Q(s, \mathbf{x}(s), \mathbf{u}(s)) = \mathbb{E}\left[L(s, \mathbf{x}(s), \mathbf{u}(s)) + V(s^+, \mathbf{x}^+(s)) \mid s, \mathbf{x}(s), \mathbf{u}(s) \right]. \tag{3.21}$$

The value function and optimal policy for each input is saved at every stage as

$$V[k, \mathbf{x}, \mathbf{u}] = \min_{\mathbf{u}} Q(s[k], \mathbf{x}, \mathbf{u}) \tag{3.22a}$$

$$P^*_{fc}[k, \mathbf{x}] = \operatorname*{argmin}_{P_{fc}} Q(s[k], \mathbf{x}, \mathbf{u}), \qquad k = 1, \ldots, K \tag{3.22b}$$

$$\bar{\mathrm{H}}^*_{2,rf}[k, \mathbf{x}] = \operatorname*{argmin}_{\bar{\mathrm{H}}_{2,rf}} Q(s[k], \mathbf{x}, \mathbf{u}), \qquad \delta_{st}(s[k]) = 1 \tag{3.22c}$$

As these are only saved for a discrete set of stages, states and inputs linear interpolation is done to calculate the value and policy at any point in space and for any state and input values with the exception of refuelling decisions $\bar{\mathrm{H}}_{2,rf}$. As this input has an inherent binary interpretation its value is often discontinuous with regard to state values. Hence, it is important to identify discontinuities and interpolate using a different method around these. An example of this is shown in Figure 3.2 where an estimation of the discontinuity is illustrated and the difference between linear and nearest interpolation is shown. An intuitive explanation of this is that the truck will either pass the station or stop and refuel enough to finish the mission without stopping again. Refuelling less than the policy causes the truck to need another stop later on, which is not cost-effective.

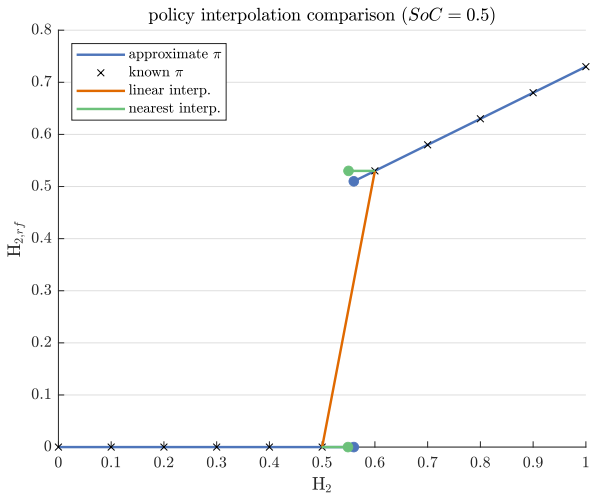


Figure 3.2: Comparison of interpolation methods in discontinuous region.

Stochastic variables

Implementing dynamic programming means quantising states, inputs and disturbances in a range of interest as $x_i = x_i^1, \ldots, x_i^{M_{x_i}} \in \mathbb{X}_i$ where M_{x_i} represents the number of discrete values at which to determine a policy for state x_i. Likewise, M_{u_i} denotes the number of inputs to consider for the policy of input u_i, and M_{w_i} the number of samples from the random distribution of w_i to perturb the system with. The number of calculations to be made for each spatial sample can thus be calculated by $\prod_j M_j$. Because of this, the number of states and inputs is already kept to a minimum; likewise, this is desired for the number of disturbances. Hence, it is of interest to approximate the disturbances with a single variable and associated probability density function.

The variables that were modelled as stochastic were wind speed w, wind angle θ, rolling resistance coefficient C_{rr} due to road conditions, mass m, and speed reduction due to traffic Δv_{tr}. As all of these eventually affect the power demand P_{em} it was chosen as the variable which the single disturbance should affect.

The vehicle dynamics are not linear in all stochastic variables, nor are all of the variables modelled as Gaussian distributions, hence it is not possible to implicitly derive a density function for a desired combined variable. However, by sampling each stochastic variable as shown in Figure 3.3 and calculating the power demand for the sampled values, akin to a Monte Carlo method, it is possible to determine a combined probability distribution from the resulting data points. The distribution could then be approximated using a normal distribution as shown in Figure 3.4, allowing it to be represented mathematically using only two parameters, a mean and a variance.

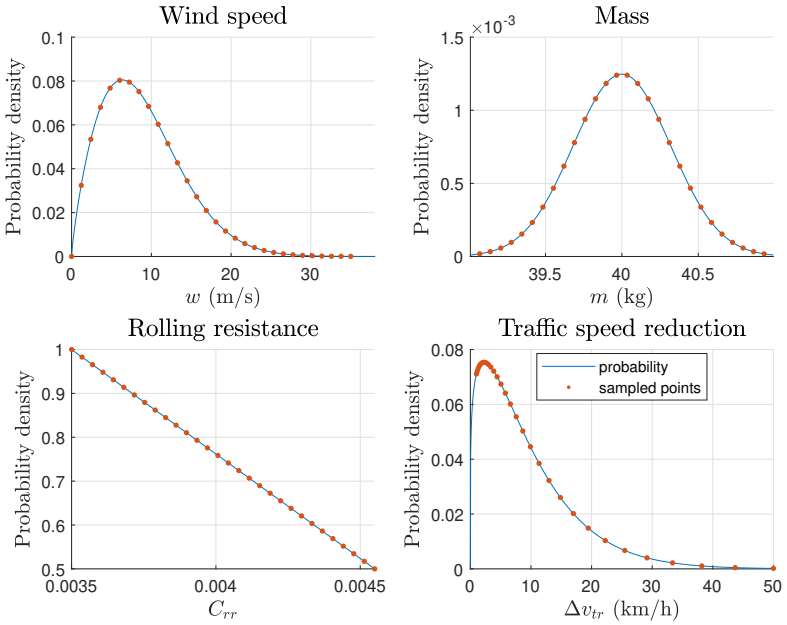


Figure 3.3: Illustration of stochastic variables' sampling.

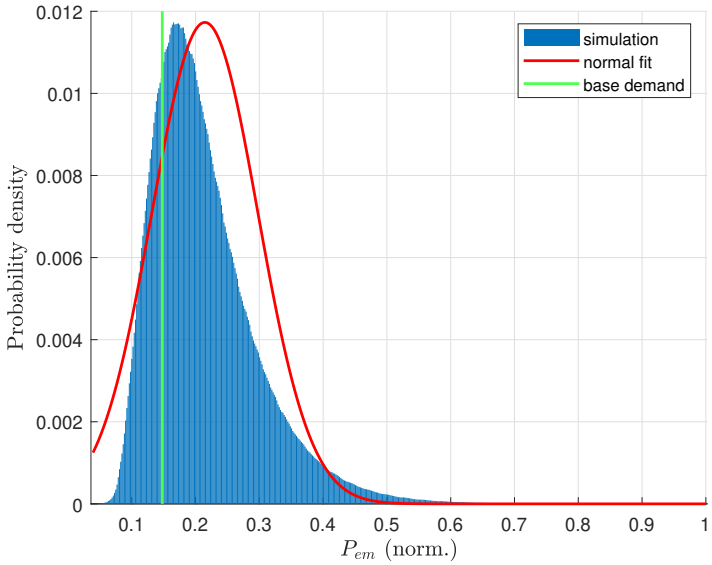


Figure 3.4: Illustration of normal approximation of the stochastic power demand. Probability density function of total power demand ($v = 70$ km/h, $\alpha = 0$, $v' = 0$).

As some stochastic variables were modelled as a distribution around a given base value, i.e. mass was normally distributed around a known approximate vehicle mass m and resulting speed was the product of the base speed v and the speed reduction from traffic, the resulting power demand's mean $\mu_{P_{em}}$ and standard deviation $\sigma_{P_{em}}$ become a function of some base values, as well as some spatially varying parameters such as road grade and acceleration, which have a large effect on the power demand.

$$\mu_{P_{EM}}(s) = \mu_{P_{EM}}(C_{rr0}, m, v(s), v'(s), \alpha(s)) \tag{3.23a}$$

$$\sigma_{P_{EM}}(s) = \sigma_{P_{EM}}(C_{rr0}, m, v(s), v'(s), \alpha(s)) \tag{3.23b}$$

A joint probability distribution for vehicle speed given power demand was also determined, with covariance 1, as the speed is used in the integration step to calculate the total power consumption, see equation (3.9). Consequently, the distribution of energy consumption varies less than that of power demand, as outliers in power demand correlate with a decreased speed, and a change in speed entails an implicit change in the step size in the Euler integration.

3.3.2 Bottom layer – non-linear programming

Mathematical model

After refuelling decisions have been made at the upper level and the problem is no longer mixed-integer, an NLP with more accurate and comprehensive dynamics can be formulated. The states and inputs of the system in discrete distance space thus

become

$$\mathbf{x}[n] = \begin{bmatrix} SoC \\ \bar{\mathrm{H}}_2 \\ t \\ v^2 \\ T_b \\ F_{fc}[n-1] \end{bmatrix}, \qquad \mathbf{u}[n] = \begin{bmatrix} s_{\mathrm{H}_2} \\ s_{SoC} \\ s_t \\ F_{fc} \\ F_b \\ F_{br} \\ P_{th} \end{bmatrix}, \tag{3.24}$$

where $s_{\mathrm{H}_2}, s_{\mathrm{SoC}}, s_{\mathrm{t}}$ are slack variables that are used for bound relaxation for H_2, SoC, t, respectively.

Using distance sampling substantially increases the complexity of the state dynamics, especially for the vehicle's speed. This complexity can be partly eliminated by using v^2 instead of v as a state representing the vehicle's speed and that is thanks to the following property

$$\frac{dv^2}{ds} = 2v\frac{dv}{ds} = 2v\frac{dv}{dt}\frac{dt}{ds} = 2\dot{v}. \tag{3.25}$$

In order to make the equation of speed dynamics linear in control inputs, some power control inputs have been replaced by their force counterparts. The relation between the two can be described as

$$F_{fc} = \frac{P_{fc}}{v}, \tag{3.26a}$$

$$F_b = \frac{P_b}{v}, \tag{3.26b}$$

$$F_{br} = \frac{P_{br}}{v}. \tag{3.26c}$$

Using the mathematical relations written before and accounting for some transformation made specifically for this layer, the state dynamics become

$$SoC' = \frac{-F_b}{C_b U_{oc}(SoC, T_b)}, \tag{3.27a}$$

$$\bar{\mathrm{H}}_2' = -\frac{1}{\sqrt{v^2}C_{\mathrm{H}_2}}\left(a_2 F_{fc}^2 v^2 + a_1 F_{fc}\sqrt{v^2} + a_0\right) + \delta_{st}\delta_{Dirac}\bar{\mathrm{H}}_{2,rf}, \tag{3.27b}$$

$$t' = \frac{1}{\sqrt{v^2}} + \delta_{st}\delta_{Dirac}\cdot(t_{rf} + t_q), \tag{3.27c}$$

$$(v^2)' = \frac{2}{m}\Big(F_{fc} + F_b - F_{br} - \\ - \frac{Q_{pass} + P_{EM,loss} + |P_{th}| + P_{fc,heat} + P_{aux}}{\sqrt{v^2}} - F_{rr} - F_{air} - F_g\Big), \tag{3.27d}$$

$$T_b' = \frac{Q_{pass} + Q_{act} + Q_{amb}}{\sqrt{v^2} c_b m_b}. \tag{3.27e}$$

t_q in equation (3.27c) is queing time of a hydrogen station. The complete discretised mathematical formulation of the problem can be found in Appendix B. It is solved numerically with the help of Embotech's FORCESPRO solver [38] which utilises a Newton method-based approach by the name of nonlinear interior point (IP). More information about the solver can be found in [4] and [39]. The problem is solved to minimise the objective function described below.

Objective function

The objective function for the bottom layer is defined as follows

$$J(\mathbf{x}, \mathbf{u}) = \omega_t t[N] + \\ + \sum_{n=1}^{N} \left(\omega_s(s_{\mathrm{H}_2}[n] + s_{SoC}[n] + s_t[n]) + \omega_{\mathrm{H}_2}\delta_{st}[n]H_{2,rf}[n] + \omega_{br}F_{br}[n]\Delta s[n]\right), \quad (3.28)$$

where time cost ω_t is used to minimise the time spent en route, hydrogen cost ω_{H_2} is used to minimise the refuelled amount of hydrogen, slack variables costs ω_s is used to minimise their use and enforce constraint fulfilment and cost of friction braking ω_{br} is used to minimise the use of friction brakes.

3.4 Simulation setups

Reiterating the research questions asked:

- Can splitting the mixed-integer problem into two layers, solving the integer decisions in the higher layer, thus making the second layer a smooth non-linear problem, yield a similar optimal solution as the smoothened MINLP problem?
- Further, can modelling certain dynamics as stochastic processes improve the robustness of the refuelling station selection?

Aiming to answer these questions a number of simulation tests were set up, each quantifying the claims above or additional metrics that were deemed interesting. The test cases aim to answer the following questions:

Test 1 Does the bilevel approach produce results of equal or lower total cost, and if so, how consistently? In the case of disturbances, does the stochastic strategy find a feasible solution more consistently than the deterministic strategy?

Test 2 Are there any possible benefits to be had when iterating the two layers until convergence?

Test 3 What is the tradeoff between computational time and optimality or robustness? Can sampling the route non-equidistantly assist the strategy in any of the above questions?

3.4.1 Test 1 – Robustness analysis

The first set of simulations was done to evaluate the optimality and robustness of station selection using the two-layer approach. The following strategies were compared:

SDP$_\kappa$ Use a bilevel approach with stochastic dynamic programming selecting stations, with a maximum failure risk of κ.

DP Use a bilevel approach with deterministic dynamic programming selecting stations.

Sigm Use a sigmoid activation function, similar to the one in [4], to smoothen the MINLP problem allowing stations to be selected directly in one layer.

Heur Use a bilevel approach with a heuristic algorithm based on estimated power consumption selecting stations. The algorithm can be found in Appendix A.2.

The simulations were conducted on two routes: one from Zagreb, Croatia to Paris, France (1559 km) and another from Gothenburg, Sweden to Kiruna, Sweden (1793 km). In Tables 3.1 and 3.2, the parameters of the hydrogen stations that lie along the chosen routes are listed. Using the different strategies listed above as the top layer, selecting stations and refuelling amount, the evaluation was then based on the total cost J, given from running the top layer results in the bottom layer (NLP). The same simulation was then done but with a constant disturbance applied to the rolling resistance to simulate wet road conditions due to precipitation. The purpose of this was to see whether the station selection and refuelling amount were sufficient in an uncertain scenario. The parameters used for the top and bottom layer were $K = 600$, $N = 800$, $M_{x_1} = M_{x_2} = 31$, $M_u = 21$, $M_w = 13$ and with initial and terminal conditions $x_1[0] = x_2[0] = 0.90$ and $x_1[N] = 0.30$, $x_2[N] = 0.15$.

Table 3.1: Station positions and hydrogen prices for the Zagreb–Paris route.

Station	Position [km]	Price [SEK/kg]	Station	Position [km]	Price [SEK/kg]
1	93.64	60.3	17	587.18	67.0
2	96.56	83.0	18	602.79	62.9
3	109.24	59.7	19	748.12	74.5
4	141.43	84.7	20	821.28	72.8
5	156.06	69.6	21	824.20	73.5
6	223.36	73.9	22	877.85	61.8
7	249.70	79.3	23	922.72	62.5
8	273.11	70.6	24	977.34	61.4
9	310.17	62.9	25	1056.34	64.0
10	323.83	59.1	26	1084.62	57.5
11	342.36	55.3	27	1122.66	58.3
12	376.50	78.3	28	1152.91	86.2
13	474.04	67.9	29	1164.61	86.1
14	503.30	58.0	30	1362.61	59.1
15	549.14	83.3	31	1392.85	53.4
16	565.72	62.1	32	1426.01	59.7

Table 3.2: Station positions and hydrogen prices for the Gothenburg–Kiruna route.

Station	Position [km]	Price [SEK/kg]	Station	Position [km]	Price [SEK/kg]
1	6.73	67.1	19	667.27	59.4
2	13.46	77.7	20	670.63	80.5
3	48.22	52.5	21	712.12	86.4
4	50.47	63.1	22	745.77	63.5
5	70.65	57.6	23	753.62	76.7
6	172.70	55.7	24	802.96	83.2
7	176.07	59.0	25	855.67	83.8
8	235.51	64.6	26	905.01	55.5
9	259.06	66.4	27	907.26	53.9
10	288.21	71.4	28	908.38	58.4
11	327.46	67.2	29	1025.01	83.2
12	344.29	76.5	30	1160.71	55.9
13	373.44	59.7	31	1166.31	67.2
14	382.42	83.2	32	1167.43	86.0
15	513.63	53.5	33	1304.25	71.2
16	527.08	67.1	34	1503.87	76.7
17	529.33	72.1	35	1790.96	63.5
18	617.92	57.4	36	1792.09	76.5

3.4.2 Test 2 – Iterative method

The premise of this test is that on the one hand, the optimal station selection is dependent on the speed travelled, as that determines the efficiency of the energy consumption and the total trip time, while on the other hand, the optimal speed profile is dependent on which stations were selected, i.e. when the truck needs to lower the speed to have sufficient energy until the next station is reached. This two-way dependency is one reason why the hierarchical strategy can not be considered globally optimal, as it optimises one variable at a time. One strategy to mitigate this drawback of the bilevel approach is to iterate the two layers until the solution potentially converges to a solution upon which no further improvements can be made. This is no guarantee that the solution will be globally optimal, but rather improving the solution marginally.

The idea is thus to run the mission optimiser, save the speed profile and rerun the mission optimiser using this optimal speed profile to potentially get a new optimal station selection, for which a new optimal speed profile is found, as illustrated in Figure 3.1. When doing so, a question arises: Should the SDP policy be regenerated using the new speed profile as well, or just the forward simulation? Generating the SDP policy makes up a majority of the computation time of the mission optimiser, incentivising reusing the policy generated initially. On the contrary, the policy is determined for a certain speed profile, meaning if this changes significantly the policy is no longer optimal and might not give desirable results.

The simulations run in this test were done for three different routes, with two sets of parameter values, specifically the stage resolution for the two layers, K and N, and trying both approaches of regenerating the SDP policy for each iteration and utilising the initial SDP policy. This comes out to 12 simulations in total, for which the parameters are detailed in Table 3.3. All simulations use non-equidistant sampling, and a risk tolerance of $\kappa = 0.05$. As conditions for convergence, the stations selected should be the same for three consecutive iterations and the total cost J should not increase or decrease by more than 1% from the previous iteration, with a maximum of 10 iterations allowed.

Table 3.3: Initial and terminal state constraints, stage quantisation and SDP regeneration decision for simulations 1-12.

simulation	$x_1[0]$	$x_2[0]$	$x_1[N]$	$x_2[N]$	K	N	reg. SDP
Zagreb – Paris							
1	0.90	0.90	0.30	0.15	250	500	no
2	0.90	0.90	0.30	0.15	250	500	yes
3	0.90	0.90	0.30	0.15	500	1000	no
4	0.90	0.90	0.30	0.15	500	1000	yes
Gothenburg – Kiruna							
5	0.90	0.90	0.30	0.15	250	500	no
6	0.90	0.90	0.30	0.15	250	500	yes
7	0.90	0.90	0.30	0.15	500	1000	no
8	0.90	0.90	0.30	0.15	500	1000	yes
Gothenburg – Rødby							
9	0.90	0.90	0.75	0.75	250	500	no
10	0.90	0.90	0.75	0.75	250	500	yes
11	0.90	0.90	0.75	0.75	500	1000	no
12	0.90	0.90	0.75	0.75	500	1000	yes

3.4.3 Test 3 – Resolution analysis

Another set of simulations was be aimed at investigating how different resolutions of the state vector quantisation in the MINLP M and stage quantisation in both problems K, N can affect the optimality of the overall solution. The purpose of this test is to determine whether the two-layer solution can perform competitively at a similar computation time as the other solutions. The simulations were conducted based on a route from Gothenburg, Sweden to Kiruna, Sweden (1793 km). All combinations of the following parameters were tested:

- $M_{x_1} = M_{x_2} = \{11, 21, 31\}$,
- $N = K = \{400, 800, 1200, 1600\}$,

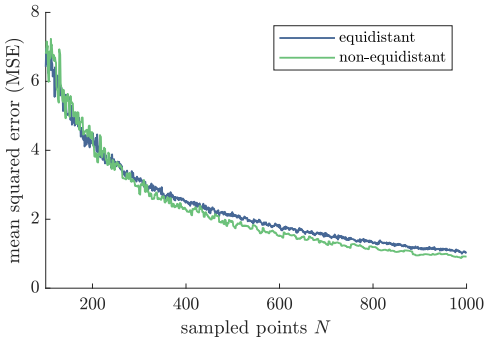
and the parameters total cost J, the total amount of refuelled hydrogen, the selected stations, the travel time t_N, and the computation time will be evaluated. Initial and terminal conditions were $x_1[0] = x_2[0] = 0.90$, $x_1[N] = x_2[N] = 0.2$. One set of simulations were run using equidistant sampling, and one using non-equidistant sampling.

4

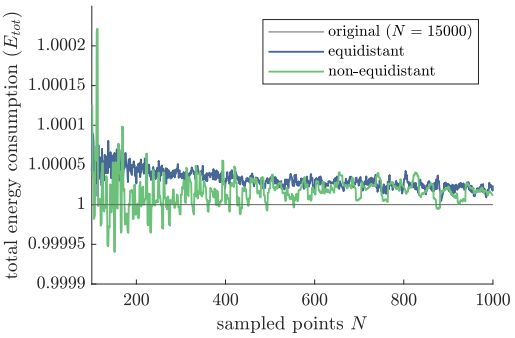
Results

4.1 Downsampling

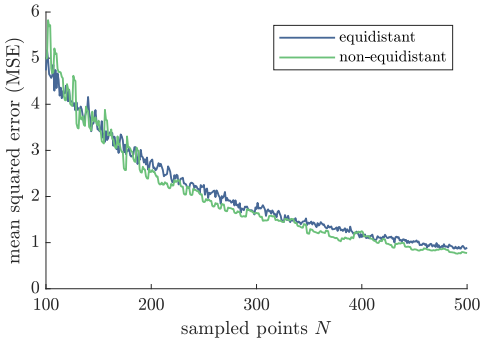
The results for the comparison between the equidistant and non-equidistant downsampling of the road data is shown in Figure 4.1. An exponential fit was made to the MSE data for route one, illustrated in Figure 4.2. This shows that for any N over 175, a 10% increase in N is required to achieve the same MSE for equidistant as when using non-equidistant. Looking at total energy consumption error the non-equidistant sampling performs better on average than equidistant for route one and worse on average for route two. The total error in energy consumption is notably less than 0.3% for as low as 100 stages using both methods.



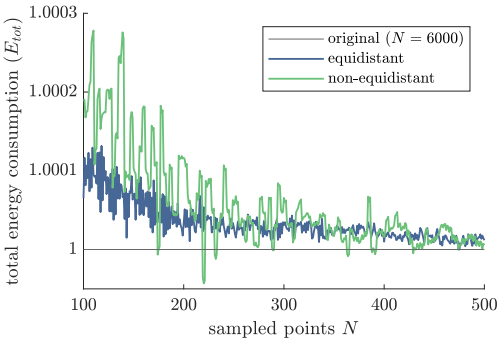
(a) MSE for route one.



(b) Energy consumption for route one.



(c) MSE for route two.



(d) Energy consumption for route two.

Figure 4.1: Comparison of equidistant and non-equidistant sampling for two different routes, with original resolutions of $N = 15000$ and $N = 6000$, respectively.

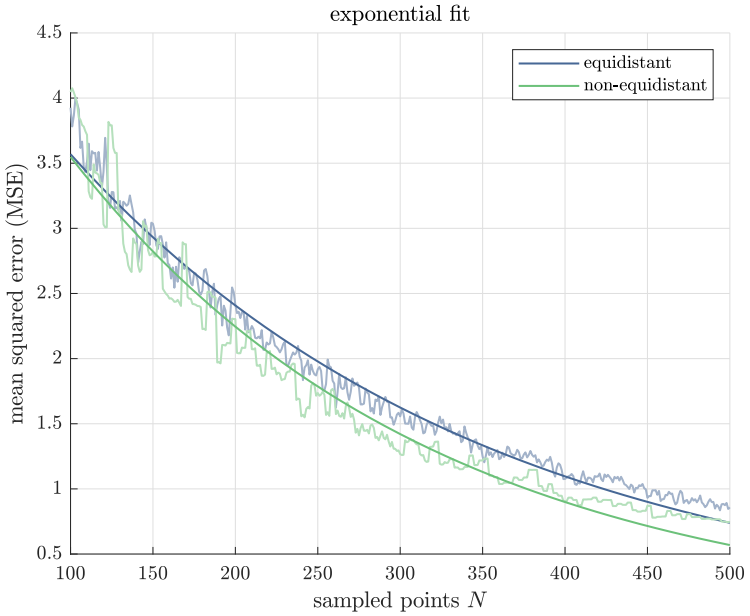


Figure 4.2: exponential fit of MSE data for route one.

4.2 Test 1 – Robustness analysis

The results of the simulations for the two chosen routes with and without uncertainty added can be seen in Tables 4.1 - 4.4 below. A dagger (†) signifies a failure of convergence of the solver in the bottom layer, determined either from the solver's output message or by observing the results. Failure to converge can be a consequence of the station selection being infeasible, but such deductions should be made with caution. Due to the non-linearity of the problem tiny changes in the parameters can cause the solver to not converge to a solution.

Table 4.1: Simulation results with different methods for the route Zagreb–Paris with no uncertainty applied.

method	J	t_N [min]	Selected stations	$\sum \bar{\mathrm{H}}_{2,rf}$	E_{br} $[J]$	t_{comp} [sec]
variable H_2 cost, equidistant sampling						
SDP$_{\kappa=0.05}$	394664	1301	18 31	0.7463	1598	68.61
SDP$_{\kappa=0.50}$	394592	1301	18 31	0.7471	1598	55.65
DP	†	†	22 28	0.4473	†	56.67
sigm	559505	1864	16	0.3857	1599	80.76
heur	394739	1302	20 31	0.7437	1598	20.09
constant H_2 cost, equidistant sampling						
SDP$_{\kappa=0.05}$	393973	1301	18 31	0.7379	1598	54.01
SDP$_{\kappa=0.50}$	394612	1302	18 31	0.7518	1598	55.75
DP	385738	1283	17	0.4479	4488	56.74
sigm	536466	1859	16	0.3931	1599	40.01
heur	395381	1302	20 31	0.7437	1598	19.65
constant H_2 cost, non-equidistant sampling						
SDP$_{\kappa=0.05}$	394281	1302	17 24	0.8058	1598	67.09
SDP$_{\kappa=0.50}$	400229	1322	17 28 30	0.7861	1598	68.73
DP	†	†	18	0.4445	†	56.53
sigm	504915	2307	16	0.3687	1599	39.92
heur	394106	1302	20 31	0.7437	1598	19.67
variable H_2 cost, non-equidistant sampling						
SDP$_{\kappa=0.05}$	397766	1307	19 32	0.7341	1598	67.41
SDP$_{\kappa=0.50}$	402050	1308	19 32	0.7372	1598	67.97
DP	†	†	23	0.4209	†	56.93
sigm	486618	2133	17 26	0.3973	1599	40.93
heur	398233	1309	21 32	0.7279	1598	38.13

Table 4.2: Simulation results with different methods for the route Zagreb–Paris, with uncertainty applied after the station selection.

method	J	t_N [min]	Selected stations	$\sum \bar{\mathrm{H}}_{2,rf}$	E_{br} $[J]$	t_{comp} [sec]
variable H_2 cost, equidistant sampling						
SDP$_{\kappa=0.05}$	701179	2015	17 28	0.9448	1602	44.59
SDP$_{\kappa=0.50}$	707140	2018	17 28	0.9439	1602	45.88
DP	516429	1710	20 21 26	0.5479	1598	34.03
sigm	509478	1689	11 16	0.4200	1598	123.04
heur	682114	1976	18 28	0.9357	1602	15.44
constant H_2 cost, equidistant sampling						
SDP$_{\kappa=0.05}$	724601	2063	14 28	0.9538	1603	68.86
SDP$_{\kappa=0.50}$	708528	2019	13 25	0.9554	1602	70.32
DP	504097	1671	20	0.5819	1598	33.65
sigm	513491	1689	11 16	0.4198	1598	82.87
heur	681484	1976	18 28	0.9357	1602	16.21
constant H_2 cost, non-equidistant sampling						
SDP$_{\kappa=0.05}$	†	†	15 26	0.9568	†	72.05
SDP$_{\kappa=0.50}$	†	†	15 25	0.9486	†	42.91
DP	510064	1691	18 21	0.5914	1598	32.60
sigm	509803	1690	11 16	0.4208	1598	81.09
heur	681523	1976	18 28	0.9357	1602	15.43
variable H_2 cost, non-equidistant sampling						
SDP$_{\kappa=0.05}$	†	†	18 29	0.9359	†	70.91
SDP$_{\kappa=0.50}$	†	†	18 29	0.9417	†	67.98
DP	581831	1718	21 27 29	0.5762	1601	55.19
sigm	563614	1708	17 27	0.5559	1600	76.90
heur	†	†	19 29	0.9186	†	38.60

In Tables 4.1 and 4.2 the deterministic DP strategy performs the best, cost-wise, while failing to converge three out of eight times. The heuristics algorithm performs on par with the SDP strategy for the deterministic simulations and outperform it for the uncertain simulations. The sigmoid strategy is the only one to converge every time, performing the worst in the deterministic set, but the best out of all the strategies in the uncertain set. Perhaps this is a consequence of the uncertainty being part of the bottom layer optimisation and not simulated in an additional plant-controller model, indicating that the method of evaluation is flawed.

Table 4.3: Simulation results with different methods for the route Gothenburg–Kiruna with no uncertainty applied.

method	J	t_N [min]	Selected stations	$\sum \bar{\mathrm{H}}_{2,rf}$	E_{br} $[J]$	t_{comp} [sec]
constant H_2 cost, equidistant sampling						
SDP$_{\kappa=0.05}$	397956	1294	18	0.7042	16002	67.24
SDP$_{\kappa=0.50}$	397826	1293	18	0.7016	16002	69.00
DP	396182	1292	23	0.4725	16001	58.51
sigm	481240	1383	17	0.4297	160020	18.80
heur	409092	1330	20, 31	0.7424	16005	44.11
variable H_2 cost, equidistant sampling						
SDP$_{\kappa=0.05}$	415933	1352	18, 28	0.8137	16009	52.03
SDP$_{\kappa=0.50}$	403478	1314	19, 20	0.7285	16003	73.81
DP	396013	1292	19	0.4948	16001	64.27
sigm	481137	1383	20, 31	0.4305	160020	16.60
heur	408950	1330	20, 31	0.7424	16005	45.21
variable H_2 cost, non-equidistant sampling						
SDP$_{\kappa=0.05}$	397340	1294	19	0.7286	16003	65.88
SDP$_{\kappa=0.50}$	397701	1294	18	0.6979	16002	65.56
DP	395816	1292	17	0.4704	16001	60.35
sigm	866507	1394	17	0.4534	3982674	16.57
heur	409507	1330	20, 21	0.7424	16005	50.17
constant H_2 cost, non-equidistant sampling						
SDP$_{\kappa=0.05}$	934964	1300	19	0.7057	5365848	70.13
SDP$_{\kappa=0.50}$	648533	1302	19	0.7071	2496247	68.39
DP	647081	1301	24	0.4848	2496246	58.76
sigm	866605	1393	17	0.4551	3980611	18.01
heur	655704	1326	21, 32	0.7337	2496249	44.22

Table 4.4: Simulation results with different methods for the route Gothenburg–Kiruna, with uncertainty applied after the station selection.

method	J	t_N [min]	Selected stations	$\sum \bar{\mathrm{H}}_{2,rf}$	$E_{br}\,[J]$	t_{comp} [sec]
variable H_2 cost, equidistant sampling						
SDP$_{\kappa=0.05}$	505343	1648	4 23 30 31	2.0834	1598	52.52
SDP$_{\kappa=0.50}$	505400	1649	4 23 30 31	2.0830	1598	15.00
DP	†	†	4 26 31	1.7397	†	28.22
sigm	†	†	9 14 15 22 29 31	1.8598	†	51.05
heur	498882	1628	5 23 30 31	2.0250	1598	2.90
constant H_2 cost, equidistant sampling						
SDP$_{\kappa=0.05}$	503405	1648	4 16 29 31	2.0948	1598	7.40
SDP$_{\kappa=0.50}$	503238	1649	4 19 30 31	2.1028	1598	13.09
DP	556520	2468	3 24 30	1.7538	693787385	26.10
sigm	†	†	14 15 22 29 31	1.7359	†	56.88
heur	499019	1628	5 23 30 31	2.0250	1598	53.96
constant H_2 cost, non-equidistant sampling						
SDP$_{\kappa=0.05}$	505391	1649	4 22 30 31	2.0857	1598	66.82
SDP$_{\kappa=0.50}$	504057	1649	1 18 29 31	2.0996	1598	70.11
DP	569126	1808	4 26 31	1.7445	1595	89.24
sigm	†	†	14 15 22 29 31	1.7520	†	61.44
heur	498183	1628	5 23 30 31	2.0250	1598	55.24
variable H_2 cost, non-equidistant sampling						
SDP$_{\kappa=0.05}$	510085	1664	5 25 34 36	2.0994	1598	100.08
SDP$_{\kappa=0.50}$	516100	1684	5 25 32 33 36	2.0997	1598	110.06
DP	†	†	5 29 36	1.7496	†	76.08
sigm	†	†	1 3 4 15 18 25 27 28 33 36	2.0541	†	48.36
heur	503535	1643	7 26 34 36	2.0250	1598	46.46

For Tables 4.3 and 4.4 the SDP strategy is outperformed by the DP strategy in the deterministic case, while performing better in the uncertain case, albeit being outperformed by the heuristics algorithm there. The sigmoid strategy has the worst performance, failing to find a solution in the uncertain case.

4.3 Test 2 – Iterative method

Presented below are the results from running the simulations in test 2. The first two simulations for each route used $K = 250$ and $N = 500$ while the latter used $K = 500$ and $N = 1000$. Even-numbered simulations, on the right-hand side, regenerated a new SDP policy for each iteration while odd-numbered simulations, on the left-hand side, reused the initial policy. The first set of 12 simulations, seen in Tables 4.5-4.7, were run using the seed `rng(5)` for the random number generator, which determines how the hydrogen prices vary between stations. The second set, seen in Tables 4.8-4.10, used a new seed for each simulation resulting in prices varying between simulations run with and without regenerating the SDP policy. Note that for route two, Gothenburg-Kiruna, the number of stations available decreases with the lower stage resolution, as nearby stations are combined, resulting in the indices not being comparable between stage resolutions and the randomly generated costs differing.

Table 4.5: Results from simulations for route one, Zagreb–Paris, using the same seed for every simulation.

simulation 1				
iter.	J	t_N [min]	stations	$\sum \bar{\mathrm{H}}_{2,rf}$
1	403718	1299	18	0.7084
2	423368	1295	18	0.7138
3	406036	1301	18	0.7139
4	398754	1295	18	0.7141
5	406004	1301	18	0.7138
6	467642	1300	18	0.7141
7	484385	1304	18	0.7139
8	403719	1298	18	0.7182
9	468267	1295	18	0.7138
10	476583	1300	18	0.7137

simulation 2				
iter.	J	t_N [min]	stations	$\sum \bar{\mathrm{H}}_{2,rf}$
1	403718	1299	18	0.7084
2	†	†	†	†

simulation 3				
iter.	J	t_N [min]	stations	$\sum \bar{\mathrm{H}}_{2,rf}$
1	413230	1326	19 32	0.7168
2	417976	1331	18 27	0.8146
3	409419	1341	18 27 29	0.8122
4	405712	1323	18 27	0.8089
5	403655	1322	18 27	0.8089
6	403320	1321	18 27	0.8101

simulation 4				
iter.	J	t_N [min]	stations	$\sum \bar{\mathrm{H}}_{2,rf}$
1	413230	1326	19 32	0.7168
2	†	†	†	†

Table 4.6: Results from simulations for route two, Gothenburg–Kiruna, using the same seed for every simulation.

simulation 5				
iter.	J	t_N [min]	stations	$\sum \bar{\mathrm{H}}_{2,rf}$
1	$1.1 \cdot 10^7$	2630	16 28	0.9520
2	745434	2461	15 26	1.2761
3	†	†	†	†

simulation 6				
iter.	J	t_N [min]	stations	$\sum \bar{\mathrm{H}}_{2,rf}$
1	$1.1 \cdot 10^7$	2630	16 28	0.9520
2	711221	2318	15 26	1.2674
3	†	†	†	†

simulation 7				
iter.	J	t_N [min]	stations	$\sum \bar{\mathrm{H}}_{2,rf}$
1	490930	1619	18 32	0.9892
2	492082	1619	18 32	1.0158
3	492070	1618	18 32	1.0160

simulation 8				
iter.	J	t_N [min]	stations	$\sum \bar{\mathrm{H}}_{2,rf}$
1	490930	1619	18 32	0.9892
2	492052	1617	18 32	1.0166
3	492312	1618	18 32	1.0163

Table 4.7: Results from simulations for route three, Gothenburg–Rødby, using the same seed for every simulation.

simulation 9				
iter.	J	t_N [min]	stations	$\sum \bar{\mathrm{H}}_{2,rf}$
1	116464	383	9	0.3193
2	120941	397	9	0.3319
3	118311	389	9	0.3270
4	119430	392	9	0.3272

simulation 10				
iter.	J	t_N [min]	stations	$\sum \bar{\mathrm{H}}_{2,rf}$
1	116464	383	9	0.3193
2	699926	1597	9	0.5409
3	†	†	†	†

simulation 11				
iter.	J	t_N [min]	stations	$\sum \bar{\mathrm{H}}_{2,rf}$
1	122220	400	9	0.3289
2	136394	448	9	0.3384
3	142515	468	9	0.3440
4	150348	490	9	0.3497
5	165948	537	9	0.3630
6	201265	612	9	0.3922
7	†	†	†	†

simulation 12				
iter.	J	t_N [min]	stations	$\sum \bar{\mathrm{H}}_{2,rf}$
1	122220	400	9	0.3289
2	†	1758	12	0.6063
3	†	†	†	†

Table 4.8: Results from simulations for route one, Zagreb–Paris, using a different seed for every simulation.

simulation 1				
iter.	J	t_N [min]	stations	$\sum \bar{\mathrm{H}}_{2,rf}$
1	437415	1299	18	0.6918
2	421984	1358	17 20 24 30	0.7145
3	410299	1319	17 20	0.7524
4	410465	1319	17 20	0.7522

simulation 2				
iter.	J	t_N [min]	stations	$\sum \bar{\mathrm{H}}_{2,rf}$
1	405938	1315	16 30	0.7131
2	†	†	†	†

simulation 3				
iter.	J	t_N [min]	stations	$\sum \bar{\mathrm{H}}_{2,rf}$
1	409438	1322	19 32	0.7163
2	†	2298	16 26	0.8237
3	406646	1326	16 26	0.8721
4	404271	1324	16 26	0.8218

simulation 4				
iter.	J	t_N [min]	stations	$\sum \bar{\mathrm{H}}_{2,rf}$
1	432181	1336	19 32	0.7284
2	†	†	†	†

Table 4.9: Results from simulations for route two, Gothenburg–Kiruna, using a different seed for every simulation.

simulation 5				
iter.	J	t_N [min]	stations	$\sum \bar{\mathrm{H}}_{2,rf}$
1	487589	1611	17 26	0.9702
2	487641	1612	17 26	0.9771
3	406646	1326	16 26	0.8721
4	404271	1324	16 26	0.8218

simulation 6				
iter.	J	t_N [min]	stations	$\sum \bar{\mathrm{H}}_{2,rf}$
1	487653	1611	17 27	0.9616
2	487499	1611	17 27	0.9688
3	406646	1326	16 26	0.8721
4	404271	1324	16 26	0.8218

simulation 7				
iter.	J	t_N [min]	stations	$\sum \bar{\mathrm{H}}_{2,rf}$
1	496783	1639	21 25 33	0.9736
2	497745	1638	21 25 33	0.9691
3	406646	1326	16 26	0.8721
4	404271	1324	16 26	0.8218

simulation 8				
iter.	J	t_N [min]	stations	$\sum \bar{\mathrm{H}}_{2,rf}$
1	496804	1639	23 32 33	0.9698
2	491992	1619	23 32	0.9886
3	406646	1326	16 26	0.8721
4	404271	1324	16 26	0.8218

Table 4.10: Results from simulations for route three, Gothenburg–Rødby, using a different seed for every simulation.

simulation 9				
iter.	J	t_N [min]	stations	$\sum \bar{\mathrm{H}}_{2,rf}$
1	114569	377	10	0.2819
2	123673	407	10 11	0.3288
3	122332	402	10 11	0.3252
4	122295	402	10 11	0.3233

simulation 10				
iter.	J	t_N [min]	stations	$\sum \bar{\mathrm{H}}_{2,rf}$
1	115177	378	12	0.3176
2	4438936	1651	12	0.5764
3	†	†	†	†

simulation 11				
iter.	J	t_N [min]	stations	$\sum \bar{\mathrm{H}}_{2,rf}$
1	121791	398	10	0.3284
2	126756	415	10	0.3332
3	124980	409	10	0.3322
4	122295	402	10 11	0.3233

simulation 12				
iter.	J	t_N [min]	stations	$\sum \bar{\mathrm{H}}_{2,rf}$
1	118808	389	12	0.3262
2	433304	1255	12	0.5202
3	†	†	†	†

From the results, there is a general pattern of simulations where the SDP policy is regenerated ending up infeasible, as illustrated by the † symbols. The cause of this is presumed to be the speed profile determined by the deterministic bottom layer optimization not being robust enough for a SDP policy to be found at the risk tolerance of $\kappa = 0.05$ used. A possible way to mitigate this is to use the algorithm to preprocess the speed profile given by the bottom layer, as done initially. Simulations 5 and 6 in Table 4.9 seem to suggest that the stage quantization $K = 250$ and $N = 500$ is insufficient to consistently find an optimal solution.

Notably, numerous simulations converges to a lower trip cost, as seen in simulation 3 in Table 4.5, simulations 1 and 3 in Table 4.8 and all simulations in Table 4.9. In total, 8 out of 24 simulations resulted in a lower cost after iterating. In simulation 3 in Table 4.5 and simulation 5-8 in Table 4.9 the station selection changes which brings a significant improvement, as seen in the decrease in cost. For other simulations, the trip cost instead diverges or stays stagnant. Simulation 1 in Table

4.5 illustrates the inconsistent behaviour of the NLP solver, as the only parameter that changes between attempts at finding the optimal solution is the initial guess and the refuel amount $\sum \bar{\mathrm{H}}_{2,rf}$ which differs by at most 1.4%, with total trip cost J differing by as much as 19.5% between solutions.

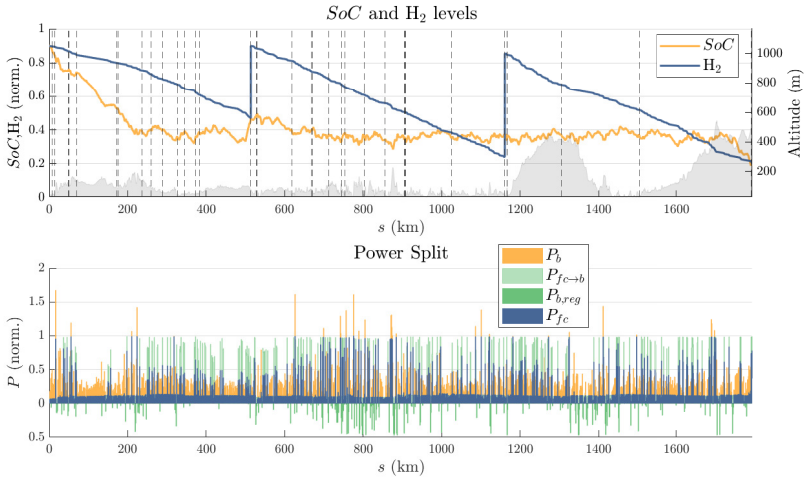
4.4 Test 3 – Resolution analysis

This section contains the results of the test that was conducted based on the description in subsection 3.4.3. In Figure 4.3 and Table 4.11, one can find test results obtained with equidistant stage sampling, and in Figure 4.4 and Table 4.11 the ones obtained with non-equidistant stage sampling.

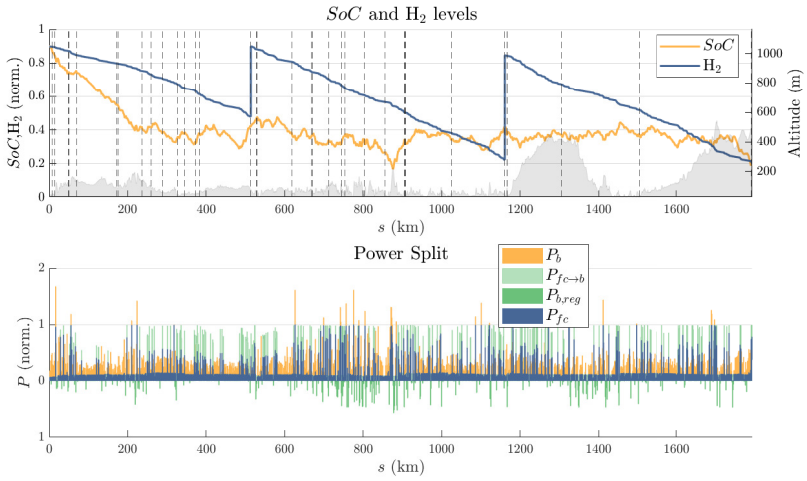
Results presented in Tables 4.11 and 4.12 demonstrate that solving with $M_{x_1} = M_{x_2} = 11$ yielded a solution with two more refuelling stops than with $M_{x_1} = M_{x_2} = 21$ and $M_{x_1} = M_{x_2} = 31$, which in turn resulted in a higher travel time and total cost. Similar results were seen when using non-equidistant sampling. However, there was no difference in the stations selected and no significant difference in the amount of refuelled hydrogen with 21 and 31 state values in SDP, whereas the computation time differed by between 30 and 50 seconds depending on the stage resolution. Stations 15 and 30 were selected for refuelling which, according to Table 3.2, have the lowest prices in their respective parts of the route. This further suggests that the SDP strategy reduces the total cost of operation as desired.

When it comes to stage sampling N and K, as shown in Table 4.11, using a lower stage resolution consistently yielded lower energy losses which turned out to be almost linear to the number of stages N, meaning that it most probably was by caused scaling in cost function and specific behaviour of the solver rather than resolution of the stage discretisation. Thus, the difference in cost and brake energy, and implicitly accuracy, is small relative to the difference in computation time, which is approximately halved when going from $N = K = 1600$ to $N = K = 800$. Worth noting is that in one of the simulations, a resolution of $N = K = 400$ was insufficient for the solver to converge. This suggests a trade-off between reliability and accuracy is necessary to achieve the desired results, as ultimately the mission optimiser needs to be reliable in providing a solution.

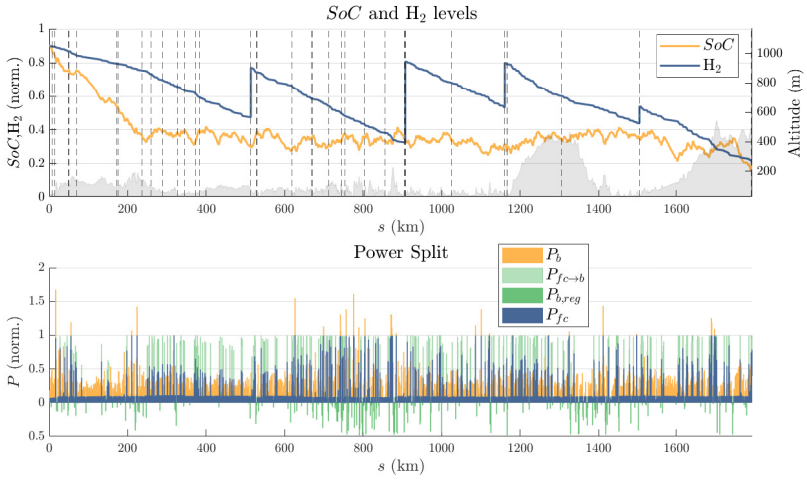
The results when running the same simulations using non-equidistant stage sampling, which are demonstrated in Table 4.12, suggest that the difference in performance is minimal. Losses from friction braking are identical to the losses when using equidistant sampling. Noticeably, the simulation which was infeasible in Table 4.11 was solvable when using non-equidistant sampling. However, no conclusive assumptions can be made based on this as the sample size is too low and only one route and vehicle is considered. Although the stations selected were the same, the trip time was slightly longer for the non-equidistant simulations, differing by an average of 1.2% which is reflected in the trip cost as well. This can be explained by an in-



(a) Simulation results for $M_{x_1} = M_{x_2} = 31$



(b) Simulation results for $M_{x_1} = M_{x_2} = 21$



(c) Simulation results for $M_{x_1} = M_{x_2} = 11$

Figure 4.3: $\bar{\mathrm{H}}_2$ and *SoC* trajectories together with the power split for the Gothenburg–Kiruna route with equidistant sampling with $N = K = 1600$. Subfigures **(a)**–**(c)** correspond to different numbers of quantised state values in SDP.

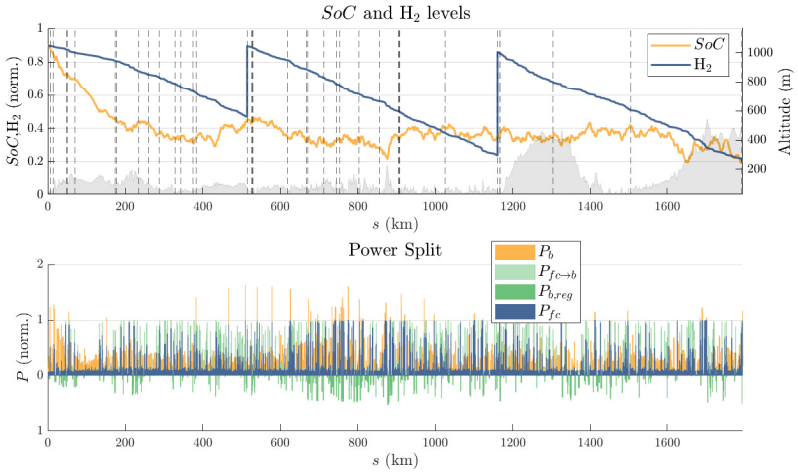
Table 4.12: Simulation results with different SDP state quantisation and stage resolution with non-equidistant stage sampling for the route Gothenburg–Kiruna.

$M_{x_1} = M_{x_2}$	$N = K$	J	t_N [min]	Selected stations	$\sum \bar{\mathrm{H}}_{2,rf}$	$E_{br}\,[J]$	t_{comp} [sec]
31	1600	493493	1620	15, 30	1.056	3198	231.01
	1200	497093	1622		1.042	2398	183.8
	800	492097	1618		1.030	1598	121.76
	400	495038	1634		0.991	798	66.6
21	1600	493511	1619	15, 30	1.058	3198	186.5
	1200	497282	1623		1.045	2398	138.3
	800	491902	1617		1.038	1598	90.6
	400	495222	1634		1.018	798	61.3
11	1600	506863	1662	15, 27, 30, 34	1.159	3198	147.8
	1200	510501	1665		1.139	2398	125.9
	800	504525	1658		1.123	1598	80.1
	400	507578	1674		1.060	798	42.33

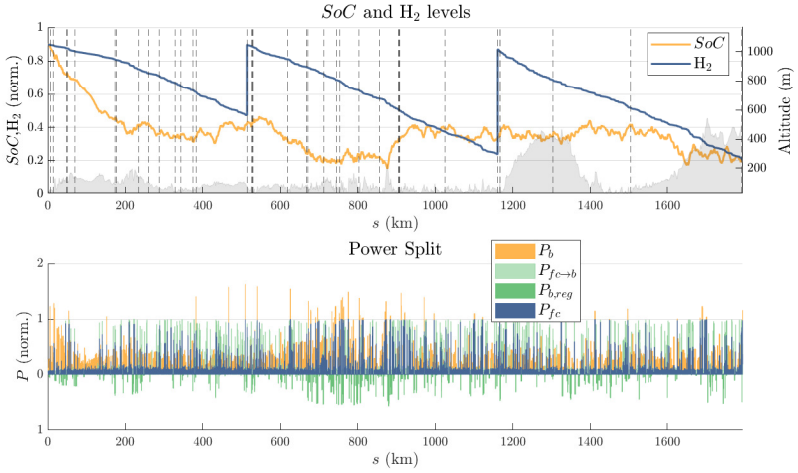
creased accuracy of the solution, although such claims would have to be investigated further.

Table 4.11: Simulation results with different SDP state quantisation and stage resolution with equidistant stage sampling for the route Gothenburg–Kiruna.

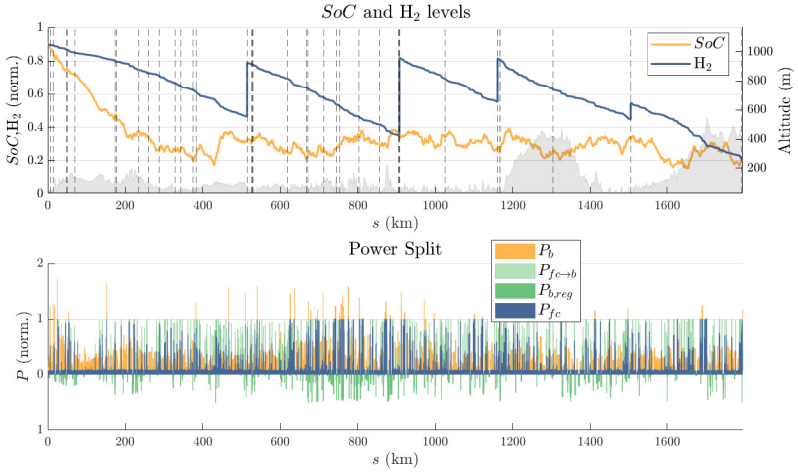
$M_{x_1} = M_{x_2}$	$N = K$	J	t_N [min]	Selected stations	$\sum \bar{\mathrm{H}}_{2,rf}$	$E_{br}\,[J]$	t_{comp} [sec]
31	1600	488376	1605	15, 30	1.048	3198	236.9
	1200	487542	1605		1.038	2398	159.8
	800	485240	1598		1.022	1598	95.2
	400	485467	1604		1.011	798	56.3
21	1600	488236	1605	15, 30	1.047	3198	172.6
	1200	487559	1605		1.043	2398	134.3
	800	485305	1598		1.033	1598	77.56
	400	†	†		1.029	†	†
11	1600	501133	1646	15, 27, 30, 34	1.148	3198	120.5
	1200	500238	1645		1.131	2398	91.0
	800	497800	1639		1.103	1598	63.0
	400	498178	1645		1.096	798	39.6



(a) Simulation results for $M_{x_1} = M_{x_2} = 31$.



(b) Simulation results for $M_{x_1} = M_{x_2} = 21$.



(c) Simulation results for $M_{x_1} = M_{x_2} = 11$.

Figure 4.4: $\bar{\mathrm{H}}_2$ and SoC trajectories together with the power split for the Gothenburg–Kiruna route with non-equidistant sampling with $N = K = 1600$. Subfigures **(a)**–**(c)** correspond to different numbers of quantised state values in SDP.

5
Discussion

In this chapter, the results from the simulations are evaluated, along with the methods used. Potential conclusions are drawn, flaws in the methodology are pointed out and future work to be done on the subject is proposed.

5.1 Downsampling

From the plots in Figure 4.1 it is questionable whether there is a performance improvement from using non-equidistant sampling, as hypothesised. The results vary depending on the route and other parameters, which suggests a more quantitative analysis is needed. The non-equidistant sampling method appears to be consistently outperforming the equidistant method after a certain number of stages when using the MSE metric. As the computation time appears to scale linearly with N, further discussed in the results for test 3, the results suggest a 9% decrease in computation time can be achieved by using non-equidistant sampling with the same accuracy.

However, when looking at the energy consumption metric the non-equidistant method appears far less consistent. It is possible that generating a more accurate speed profile could yield different results. Considering the overall small magnitude of the differences between the two downsampling methods, the potentially deciding metric could be the difference in solver stability and runtime.

Another consideration is that the algorithm used has no minimum or maximum stage length which means the distance between samples can vary significantly. Limiting this variation could increase stability in the mission optimiser.

5.2 Robustness Analysis

There will always be a trade-off in optimality when aiming for robustness. A strategy designed to mitigate the worst-case scenario will consequently perform worse in an ideal scenario than a strategy built only for the ideal scenario.

The task of evaluating the robustness of a strategy is no trivial question. The desire for robustness stems from the existence of disturbances and model errors in real-life systems. A fair evaluation should then be to use the strategies to be compared to acquire a policy, that is a set of stations to refuel at, the amount of hydrogen and the power split along the mission, and then test these policies in an environment

where realistic disturbances are applied. These can be external factors such as a traffic jam or a change in weather or internal factors such as deviations in battery voltage or fuel cell health. As the strategy's output is not a controller, but simply reference signals and binary choices handed over to a lower-level controller this is not as trivial as simulating a plant model with disturbances applied. Even with access to such a controller and plant, there are still several questions that arise: can the mission manager be rerun during the trip, in the case of strong deviation from plan? Can station selection be changed in that case? Is the refuelling amount set as well, or can it change depending on consumption? Answering such questions is outside of the scope of the thesis, but without it evaluating the strategy in a holistic scope is inconclusive.

Clearly defining robustness and how it should be quantified is also essential. As stated, in this context a robust strategy performs satisfactorily even when model errors or external disturbances are present. To quantify this a number of test cases were set up that simulate the policies from different strategies under a range of different disturbances. The more robust strategy is the one that completes the mission most frequently, without considerable deviation in cost or time. In this regard, a few suggestions for future work can be written. First, as the only output of the top layer is station selection, disturbances in station parameters should have been added instead of the ones that were used in this work (road and weather parameters etc.). That could be increased queueing times, the chance of a particular station being closed etc. and with those disturbances many cycles should be run to see how optimal and robust the approach is on average. Reducing the computation time of the solution by, for instance, increasing the efficacy of the DP algorithm or simplifications in the mathematical models will be necessary for conducting such an extensive evaluation. Second, when talking about the two layers working in tandem, having stochastic variables in the bottom layer as well would significantly help to leverage from uncertainty accounting of the proposed approach, which would, however, also increase the complexity of the problem in the bottom layer that is complex already now, but at least the possibility of doing that should be considered.

The decision to eliminate the refuelling amount as input from the bottom layer and the option of rerunning the mission optimiser is meant to simulate a hypothetical future hydrogen infrastructure where booking of hydrogen stations is done at the start of the workday, including the amount to fuel. This has the consequence of locking the amount of available energy during the trip, which with its fixed length and routing leaves speed as the only optimisation variable from a holistic perspective. However, without any such constraints, the need for a robust mission optimiser ceases to exist as the mission optimiser can be run consecutively and adapt to both model errors and disturbances. Thus, the assumption that the mission optimiser is run once at the start of the day was motivated.

In the results from test 1, the performance of the heuristics algorithm and the inconsistent performance of the SDP strategy against the DP strategy indicate that the evaluation method has room for improvements.

5.3 Iterative method

In the hypothetical scenario where the bottom layer has a low computation time, there is little drawback to iterating the solution as done in test 2 in an attempt to find an improved station selection or refuelling policy. Bringing the potential upside of improving the mission plan a third of the time, there is no risk associated with doing this either as ultimately the policy that gives the best cost can be saved, whether it appears in the first or last iteration. On the contrary, it can be argued that the results in test 2 demonstrate the inconsistency of both the top and bottom-layer strategies. The mission manager must ultimately be reliable, both with providing a solution and with ensuring it is optimal.

However, due to the nonlinear nature, as seen in the unpredictable behaviour of the solver, the end result may require several attempts to find an optimal solution, where the strategy used in test 2 is one such strategy showing promising results.

5.4 Resolution analysis

Regarding the resolution of the SDP state value quantisation, naturally, there will always be a trade-off between computation time and optimality, however, the results of test 3 demonstrated that computation time can be reduced from around 230 seconds to less 100 seconds without any loss in optimality and changes in station selection.

Test 3 also provided a practical comparison between equidistant and non-equidistant sampling, providing some basis to answer the question of whether non-equidistant stage sampling better captures the same road features even when using a lower resolution. As was shown by the results, no radical change in the performance was seen, meaning that the use of non-equidistant stage sampling has to be further studied including simulating using more cycles with different altitude profiles.

First-order explicit Runge-Kutta integrator was used in this work and as it was also suggested in [4], a higher-order one could be used and integrated with non-equidistance stage sampling for even higher accuracy in future work.

5.5 Dynamic programming

The purpose of this work was to implement and evaluate a bilevel approach to the mission management problem, utilising stochastic dynamic programming as a top layer to select stations and refuelling amounts, reducing the MINLP problem to an NLP problem. The dynamic programming algorithm was developed from scratch in MATLAB, utilising its powerful numerical computation capabilities. Although DP splits the problem into K subproblems these can not be solved in parallel due to the Bellman equation. Inherently, DP is considered a brute force method, ensuring global optimality, although only for the discretised problem. Thus, the quantisation

of the route into so-called stages, of the states, the inputs and the stochastic variables, as well as the methods used to quantise these are all important to consider as they all have a large effect on the quality of the results, but also computation time. Just as the route can be sampled non-equidistantly, attempting to capture more detail, so can the states or the input, e.g., providing more detail in the common operating region.

As the dynamics and the objective of the top layer differ from the bottom layer, so does the cost function. This poses a problem, where it is difficult to ensure the two solvers combine to minimise a shared objective. Using monetary values, where possible, ensures a common unit between terms in the cost function, but where such practice is not feasible, e.g., for slack variables, tuning is required to ensure that the solvers provide desirable results.

6
Conclusion

This thesis proposes a bilevel approach to solve the problem of mission management for fuel cell electric trucks. The first layer solves a mixed-integer problem with simplified dynamics to choose hydrogen stations to refuel at and the amount to refuel with the help of stochastic dynamic programming, allowing one to account for uncertain external factors to obtain a more robust solution. The second layer becomes a smooth problem that can be solved with more complex and comprehensive dynamics using Newton's method-based solver. The outputs, namely speed profile, power split, battery temperature profile, and chosen hydrogen stations, can be used by an onboard online controller as reference values. The idea is that the solution can be run before the trip and rerun during the trip if necessary, in case of a large deviation from the original reference, for example.

To answer the research question, 3 tests were conducted. The subject of the evaluation was the optimality and robustness that the proposed approach can yield, and how iterative solution and non-equidistant stage sampling can facilitate the improvement of the metrics mentioned above. The simulations were conducted with a 45-ton truck using 3 routes: Zagreb–Paris (1559 km), Gothenburg–Kiruna (1793 km), and Gothenburg–Rødby (461 km). Overall, according to the results, the bilevel approach can yield solutions comparable in optimality to the ones that were obtained with different strategies. The proposed approach can also produce better results when it comes to robustness in the presence of disturbances, but still, the results were not conclusive enough to answer either of the research questions with a high level of confidence. Even though the non-equidistant stage sampling did not show any substantial increase in simulation accuracy, it was shown that stage and DP stage resolutions can be reduced without significant losses in optimality, making the computation time almost 3 times smaller.

The authors of this work believe that the proposed approach has a huge potential that was not fully demonstrated by the results above and the suggestions that were written in Chapter 5 should be taken into account in any future work based on this paper.

Bibliography

[1] European Environment Agency. *CO2 emissions from cars: facts and figures (infographics)*. Mar. 2019. URL: https://www.europarl.europa.eu/topics/en/article/20190313STO31218/co2-emissions-from-cars-facts-and-figures-infographics.

[2] G. Goodwin, M.M. Seron, and J.A. De Doná. *Constrained Control and Estimation*. 2004, p. 206.

[3] U.S. Department of Energy - Energy Efficiency and Renewable Energy. *How Do Fuel Cell Electric Vehicles Work Using Hydrogen?* URL: https://afdc.energy.gov/vehicles/how-do-fuel-cell-electric-cars-work.

[4] Daniel Sundberg and Andreas Bragde. "Mission management for fuel cell heavy-duty trucks". In: (2023).

[5] G. Paganelli et al. "OPTIMAL CONTROL THEORY APPLIED TO HYBRID FUEL CELL POWERED VEHICLE". In: *IFAC Proceedings Volumes* 35.1 (2002). 15th IFAC World Congress, pp. 253–258. ISSN: 1474-6670. DOI: https://doi.org/10.3182/20020721-6-ES-1901.01510. URL: https://www.sciencedirect.com/science/article/pii/S1474667015399316.

[6] Vincenzo Alfieri et al. "Optimal Control of a Fuel Cell Electric Vehicle (FCEV) Powertrain". In: (Aug. 2022). DOI: 10.4271/2022-01-1131.

[7] Sevgi Erdoğan and Elise Miller-Hooks. "A Green Vehicle Routing Problem". In: *Transportation Research Part E: Logistics and Transportation Review* 48.1 (2012). Select Papers from the 19th International Symposium on Transportation and Traffic Theory, pp. 100–114. ISSN: 1366-5545. DOI: https://doi.org/10.1016/j.tre.2011.08.001. URL: https://www.sciencedirect.com/science/article/pii/S1366554511001062.

[8] Paolo Toth and Daniele Vigo. *Vehicle Routing*. Ed. by Daniele Vigo and Paolo Toth. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2014. DOI: 10.1137/1.9781611973594. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9781611973594. URL: https://epubs.siam.org/doi/abs/10.1137/1.9781611973594.

[9] Ilya V. Kolmanovsky and Dimitar P. Filev. "Terrain and Traffic Optimized Vehicle Speed Control". In: *IFAC Proceedings Volumes* 43.7 (2010). 6th IFAC Symposium on Advances in Automotive Control, pp. 378–383. ISSN: 1474-6670. DOI: https://doi.org/10.3182/20100712-3-DE-2013.00079. URL: https://www.sciencedirect.com/science/article/pii/S1474667015368580.

[10] Arun Rajagopalan and Gregory Washington. "Intelligent Control of Hybrid Electric Vehicles Using GPS Information". In: June 2002. DOI: 10.4271/2002-01-1936.

[11] Chan-Chiao Lin, Huei Peng, and J.W. Grizzle. "A stochastic control strategy for hybrid electric vehicles". In: *Proceedings of the 2004 American Control Conference*. Vol. 5. 2004, 4710–4715 vol.5. DOI: 10.23919/ACC.2004.1384056.

[12] I. Kolmanovsky, I. Siverguina, and B. Lygoe. "Optimization of powertrain operating policy for feasibility assessment and calibration: stochastic dynamic programming approach". In: *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)*. Vol. 2. 2002, 1425–1430 vol.2. DOI: 10.1109/ACC.2002.1023221.

[13] Lars Johannesson, Mattias Asbogard, and Bo Egardt. "Assessing the Potential of Predictive Control for Hybrid Vehicle Powertrains Using Stochastic Dynamic Programming". In: *IEEE Transactions on Intelligent Transportation Systems* 8.1 (2007), pp. 71–83. DOI: 10.1109/TITS.2006.884887.

[14] G.-E Katsargyri et al. "Optimally Controlling Hybrid Electric Vehicles using Path Forecasting". In: July 2009, pp. 4613–4617. DOI: 10.1109/ACC.2009.5160504.

[15] Ahad Hamednia et al. "Optimal Thermal Management, Charging, and Eco-Driving of Battery Electric Vehicles". In: *IEEE Transactions on Vehicular Technology* 72.6 (2023), pp. 7265–7278. DOI: 10.1109/TVT.2023.3240279.

[16] Kirill Murashko et al. "Modelling of the battery pack thermal management system for Hybrid Electric Vehicles". In: *2014 16th European Conference on Power Electronics and Applications, EPE-ECCE Europe 2014* (Aug. 2014). DOI: 10.1109/EPE.2014.6910774.

[17] Lars Johannesson et al. "Predictive energy management of hybrid long-haul trucks". In: *Control Engineering Practice* 41 (2015), pp. 83–97. ISSN: 0967-0661. DOI: https://doi.org/10.1016/j.conengprac.2015.04.014. URL: https://www.sciencedirect.com/science/article/pii/S096706611500088X.

[18] Anand Ganesan, Sebastien Gros, and Nikolce Murgovski. "Numerical Strategies for Mixed-Integer Optimization of Power-Split and Gear Selection in Hybrid Electric Vehicles". In: *IEEE Transactions on Intelligent Transportation Systems* 24.3 (2023), pp. 3194–3210. DOI: 10.1109/TITS.2022.3229254.

[19] Xianlian Wang et al. "Stochastic dynamic programming based optimal energy scheduling for a hybrid fuel cell/PV/battery system under uncertainty". In: *Process Safety and Environmental Protection* 165 (2022), pp. 380–386. ISSN: 0957-5820. DOI: https://doi.org/10.1016/j.psep.2022.07.025. URL: https://www.sciencedirect.com/science/article/pii/S0957582022006267.

[20] Andreas Ritter et al. "Long-term stochastic model predictive control for the energy management of hybrid electric vehicles using Pontryagin's minimum principle and scenario-based optimization". In: *Applied Energy* 322 (2022), p. 119192. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2022.119192. URL: https://www.sciencedirect.com/science/article/pii/S0306261922005608.

[21] Directorate-General for Mobility and European Commission Transport. *Driving time and rest periods*. 2006. URL: https://transport.ec.europa.eu/transport-modes/road/social-provisions/driving-time-and-rest-periods_en.

[22] William D. Callister and David G. Rethwisch. *Materials Science and Engineering*. John Wiley & Sons, Inc., 2020. ISBN: 9781119453918.

[23] Henrik Stenvall. "Driving resistance analysis of long haulage trucks at Volvo". In: (2022).

[24] Jerzy Ejsmont et al. "Influence of Road Wetness on Tire-Pavement Rolling Resistance". In: *Journal of Civil Engineering and Architecture* 9 (Nov. 2015). DOI: 10.17265/1934-7359/2015.11.004.

[25] Ulf Sandberg et al. *Road surface influence on tyre/road rolling resistance*. 2011. URL: https://urn.kb.se/resolve?urn=urn:nbn:se:vti:diva-15699.

[26] Sahra Hamdollahi and Luo Jun. "A REVIEW ON MODELING OF PROTON EXCHANGE MEMBRANE FUEL CELL: Review paper". In: *Chemical Industry amp; Chemical Engineering Quarterly* 29.1 (Oct. 2022), pp. 61–74. DOI: 10.2298/CICEQ220126014H. URL: https://www.ache-pub.org.rs/index.php/CICEQ/article/view/1024.

[27] Mattuci. *Proton Exchange Fuel Cell Diagram*. URL: https://commons.wikimedia.org/wiki/File:Proton_Exchange_Fuel_Cell_Diagram.svg.

[28] Qian Xun, Nikolce Murgovski, and Yujing Liu. "Joint Component Sizing and Energy Management for Fuel Cell Hybrid Electric Trucks". In: *IEEE Transactions on Vehicular Technology* 71.5 (2022), pp. 4863–4878. DOI: 10.1109/TVT.2022.3154146.

[29] Volvo Trucks. *When and why fuel cell trucks*. URL: https://www.volvotrucks.com/en-en/news-stories/stories/2022/nov/when-and-why-fuel-cell-truck.html.

[30] V.B. Avakov et al. "Lifetime prediction for the hydrogen–air fuel cells". In: *Russian Journal of Electrochemistry* 51 (2015), pp. 570–586.

[31] Xiaosong Hu et al. "Cost-optimal energy management of hybrid electric vehicles using fuel cell/battery health-aware predictive control". In: *ieee transactions on power electronics* 35.1 (2019), pp. 382–392.

[32] Huy Quoc Nguyen and Bahman Shabani. "Proton exchange membrane fuel cells heat recovery opportunities for combined heating/cooling and power applications". In: *Energy Conversion and Management* 204 (2020), p. 112328. ISSN: 0196-8904. DOI: https://doi.org/10.1016/j.enconman.2019.112328. URL: https://www.sciencedirect.com/science/article/pii/S0196890419313354.

[33] Thomas Nemeth et al. "Lithium titanate oxide battery cells for high-power automotive applications – Electro-thermal properties, aging behavior and cost considerations". In: *Journal of Energy Storage* 31 (2020), p. 101656. ISSN: 2352-152X. DOI: https://doi.org/10.1016/j.est.2020.101656. URL: https://www.sciencedirect.com/science/article/pii/S2352152X20314936.

[34] Changfu Zou et al. "Power capability prediction for lithium-ion batteries using economic nonlinear model predictive control". In: *Journal of Power Sources* 396 (2018), pp. 580–589. ISSN: 0378-7753. DOI: 10.1016/j.jpowsour.2018.06.034. URL: https://www.sciencedirect.com/science/article/pii/S037877531830630X.

[35] Richard E. Bellman. *Dynamic Programming*. Rand Corporation research study. Princeton University Press, 1957. URL: https://books.google.se/books?id=rZW4ugAACAAJ.

[36] Olle Sundström, Daniel Ambühl, and L. Guzzella. "On Implementation of Dynamic Programming for Optimal Control Problems with Final State Constraints". In: *http://dx.doi.org/10.2516/ogst/2009020* 65 (Sept. 2009). DOI: 10.2516/ogst/2009020.

[37] Craig Boutilier, Richard Dearden, and Moisés Goldszmidt. "Stochastic dynamic programming with factored representations". In: *Artif. Intell.* 121 (2000), pp. 49–107. URL: https://api.semanticscholar.org/CorpusID:10757431.

[38] Embotech AG. *FORCESPRO*. 2023. URL: https://forces.embotech.com/.

[39] A. Zanelli et al. "FORCES NLP: An efficient implementation of interior-point methods for multistage nonlinear nonconvex programs". In: *International Journal of Control* 93 (2020), pp. 13–29. DOI: 10.1080/00207179.2017.1316017.

A

Algorithms used in solution and evaluation

A.1 Speed preprocessing algorithm

Algorithm 1 is the algorithm that creates a speed profile from the road speed limits for the SDP algorithm.

Algorithm 1 Speed preprocessing

Let $\mathbf{p}$ contain all necessary information about the route and vehicle.
Let κ be the user specified maximum risk tolerance.
Let the operator $\text{shift}^+(v)$ shift the positions in v to the next spatial step.
Let ρ be the factor with which to iteratively lower the speed.

$P_{max} \leftarrow P_{fc,max} + P_{b,max}(SoC_{min})$ ▷ Calculate the worst case max power output.
$P_{min} \leftarrow P_{fc,min} + P_{b,min}(SoC_{max})$ ▷ Do the same for the max power intake.
$\mathbf{v} \leftarrow \mathbf{v}_{lim}$ ▷ Start from the road speed limits.
$feasible \leftarrow FALSE$ ▷ Assume infeasibility.
while NOT $feasible$ **do** ▷ Iterate until trip feasible.
  $\mathbf{P}_{em} \sim \Pr(P_{em} \,|\, \mathbf{p}, \mathbf{v})$ ▷ Sample power demand from distribution.
  $s_{P^+_{em}} \leftarrow \{s \,|\, \Pr(P_{em} > P_{max}) > \kappa\}$ ▷ Identify where power demand is not met,
  $s_{P^-_{em}} \leftarrow \{s \,|\, \Pr(P_{em} < P_{min}) > \kappa\}$ ▷ and similarly, where brakes are needed.
  $s_{ch} \leftarrow s_{P^+_{em}} \bigcup s_{P^-_{em}}$ ▷ Combine the two sets of stages.
  $s_{ch,next} \leftarrow \{s \,|\, \in s_{ch} \wedge v'(s) > 0\}$ ▷ Identify if this is due to high acceleration,
  $s_{ch,this} \leftarrow s_{ch}/s_{ch,next}$ ▷ in which case ignore those stages,
  $s_{ch} \leftarrow s_{ch,this} \bigcup \text{shift}^+(s_{ch,next})$ ▷ but change the speed at the next stage.
  $v(s_{ch}) \leftarrow v(s_{ch})\rho$ ▷ Lower the speed incrementally at those stages.
  $v' \leftarrow \frac{dv}{ds}$ ▷ Recalculate the speed differential.
  if $s_{ch} = \emptyset$ **then**
    $feasible \leftarrow TRUE$ ▷ Break loop when no change is needed.
  end if
end while

A.2 Heuristic algorithm

Algorithm 2 is the heuristic algorithm that, for evaluation purposes, is supposed to serve as the top layer in the proposed approach and imitate refuelling decisions similar to the ones of a real human.

Algorithm 2 Heuristic station selection

let $f(x)$ convert hydrogen and SoC to an arbitrary energy unit.
let ϕ denote the estimated energy consumption rate per kilometre driven, based on average data.

$E_{init} \leftarrow f(x_0)$ ▷ Estimate the total initial energy of the truck.
$E_{min} \leftarrow f(x_{min})$ ▷ Similarly, estimate the minimum energy level allowed.
$E_{terminal} \leftarrow f(x_N)$ ▷ And do the same for the desired terminal energy.
$E^+_{max} \leftarrow f(\mathrm{H}_{2,max} - \mathrm{H}_{2,min})$ ▷ Estimate the maximum energy refuelling amount.

$S \leftarrow s_N$ ▷ Initialize the total trip length in kilometers.
$s \leftarrow 0$ ▷ Initialize the distance traveled.
$E \leftarrow E_{init}$ ▷ Initialize the current energy.
$E_{finish} \leftarrow (s_N - s)\phi$ ▷ Calculate the energy needed to reach the end.

while $E_{finish} > E - E_{terminal}$ **do**
  $s^+_{tank} \leftarrow (E - E_{min})/\phi$ ▷ Calculate the drivable distance on fuel.
  $s_{reach} \leftarrow s + s^+_{tank}$ ▷ Determine the furthest reachable point on fuel.
  $s_{st} \leftarrow \max\{s \mid \delta_{st}(s) \wedge s < s_{reach}\}$ ▷ Identify the furthest reachable station.
  save s_{st} to list $\mathbf{s}_{st}$.
  $E \leftarrow E - (s_{st} - s)\phi$ ▷ Calculate energy after driving to that station.
  $d \leftarrow s_{st}$ ▷ Update distance driven.
  $E_{finish} \leftarrow (s_N - s)\phi + E_{terminal} - E$ ▷ Update energy needed to finish trip.
  $E^+_s \leftarrow \min\{E_{finish}, E^+_{max}\}$ ▷ Refuel either fully or enough to finish trip.
  $\mathrm{H}_{2,rf} \leftarrow f^{-1}(E^+_s)$ ▷ Convert refuel amount back to hydrogen.
  save $\mathrm{H}_{2,rf}$ to list $\mathbf{H_2}_{,rf}$.
  $E \leftarrow E + E^+_s$ ▷ Update energy.
end while

return list of stations $\mathbf{s}_{st}$ and refuel amount $\mathbf{H_2}_{,rf}$.

B

Nonlinear problem formulation

Below is the complete mathematical formulation of the nonlinear problem discretised with the first-order Runge Kutta method to minimise the given objective function representing the cost of a mission.

$$\underset{\mathbf{x},\,\mathbf{u}}{\text{minimize}} \quad \sum_{n=1}^{N} \Big(\omega_s s_{\mathrm{H}_2}[n] + \omega_s s_{SoC}[n] + \omega_s s_t[n] + \omega_{\mathrm{H}_2}\delta_{st}[n]\bar{\mathrm{H}}_{2,rf}[n] + \omega_{br}F_{br}[n]\Delta s[n]\Big) + \\ + \omega_t t[N] \tag{B.1a}$$

subject to

$$\mathbf{x}[0] = \mathbf{x}_0, \tag{B.1b}$$

$$SoC[n+1] = SoC[n] - \Delta s[n]\frac{F_b[n]}{C_b U_{oc}(SoC[n], T_b[n])}, \tag{B.1c}$$

$$\bar{\mathrm{H}}_2[n+1] = \bar{\mathrm{H}}_2[n] + \\ + \Delta s[n]\left(-\frac{1}{\sqrt{v^2[n]}C_{\mathrm{H}_2}}\left(a_2 F_{fc}^2[n]v^2[n] + a_1 F_{fc}\sqrt{v^2[n]} + a_0\right) + \delta_{st}[n]\bar{\mathrm{H}}_{2,rf}[n]\right), \tag{B.1d}$$

$$t[n+1] = t[n] + \Delta s[n]\frac{1}{\sqrt{v^2[n]}} + \delta_{st}[n](t_{rf}[n]\bar{\mathrm{H}}_{2,rf}[n] + t_q[n]), \tag{B.1e}$$

$$v^2[n+1] = v^2[n] + \Delta s[n]\frac{2}{m}\Big(F_{fc}[n] + F_b[n] - F_{br}[n] - \\ - \frac{Q_{pass} + P_{EM,loss} + |P_{th}[n]| + P_{fc,heat} + P_{aux}}{\sqrt{v^2[n]}} - F_{rr} - F_{air} - F_g\Big), \tag{B.1f}$$

$$T_b[n+1] = T_b[n] + \Delta s[n]\frac{Q_{pass}[n] + Q_{act}[n] + Q_{amb}[n]}{\sqrt{v^2[n]}c_b m_b}, \tag{B.1g}$$

$$P_{fc,min} \le F_{fc}\sqrt{v^2[n]} \le P_{fc,max}, \tag{B.1h}$$

$$P_{b,min}(SoC[n], T_b[n]) \le F_b[n]\sqrt{v^2[n]} \le P_{b,max}(SoC[n], T_b[n]), \tag{B.1i}$$

$$F_{fc}[n] + F_b[n] \le F_{T,max}, \tag{B.1j}$$

$$0 \le F_{br}[n] \le F_{br,max}, \tag{B.1k}$$

$$-\dot{P}_{fc,max} \le (F_{fc}[n+1] - F_{fc}[n])\frac{v^2[n]}{\Delta s[n]} \le \dot{P}_{fc,max}, \tag{B.1l}$$

$$\bar{\mathrm{H}}_{2,min} - s_{\mathrm{H}_2}[n] \le \bar{\mathrm{H}}_2[n] \le \bar{\mathrm{H}}_{2,max} + s_{\mathrm{H}_2}[n], \tag{B.1m}$$

$$SoC_{min} - s_{SoC}[n] \leq SoC[n] \leq SoC_{max} + s_{SoC}[n], \tag{B.1n}$$

$$\sqrt{v^2[n]} \leq v_{max}[n], \tag{B.1o}$$

$$T_{b,min} \leq T_b[n] \leq T_{b,max} \tag{B.1p}$$

$$P_{th}[n] \leq P_{th,max} \tag{B.1q}$$

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY