

# Scientific Proceedings of the Study Group Mathematics with Industry SWI 2023

swi-wiskunde.nl







platform wiskunde nederland



university of groningen



Scientific Proceedings 170th European Study Group with Industry

SWI 2023

Groningen, January 30 - February 3, 2023

Editors: Julian Koellermeier, Pieter Tibboel, Stephan Trenn Cover design: Kitty Molenaar



# Contents

### Preface

1	Dat	Data-driven Parameters Tuning for Predictive Performance Improve-									
	mer	nt of Wire Bonder Multi-body Model 1									
	1.1	Introduction									
	1.2	Problem Description									
	1.3	Approach									
	1.4	Modelling									
		1.4.1 Toy Models									
		1.4.2 Alternative Model of the Chassis									
	1.5	Measured Data									
	1.6	Optimization Methods									
		1.6.1 Cost Function $\ldots \ldots \ldots$									
		1.6.2 Sensitivity Analysis									
		1.6.3 Experiment 1: Varving the Inertial Parameters									
		1.6.4 Experiment 2: Varying the Parasitic parameters									
		1.6.5 Experiment 3: Exploring the z-Stage 18									
	17	Some Ideas on Data Analysis									
	1.8	Discussion and Recommendations									
<b>2</b>	Frequency Domain Characteristics in Nonlinear Trimmer Dynamics 25										
	2.1	Introduction									
	2.2	Model									
		2.2.1 Nondimensionalisation									
	2.3	Asymptotic analysis									
	2.4	Discussion and recommendations 34									
	2.5	Acknowledgements									
3	Modelling the heat transition 37										
	3.1	Problem statement - The heat transition									
	3.2	Graph Theory and the Monte Carlo Method									
	-	3.2.1 Graph									

 $\mathbf{v}$ 

		3.2.2	Monte Carlo Simulation						41			
		3.2.3	Advantages of The Framework						43			
	3.3	Framework of Energy Hubs							44			
		3.3.1	Introduction to Energy Hubs						44			
		3.3.2	Mathematical Modeling						44			
		3.3.3	An optimal capacity design problem (OCDP)						51			
		3.3.4	Case study: municipality of Rotterdam						51			
	3.4	nodel - a MIP formulation						56				
		3.4.1	Model description						58			
		3.4.2	Proof of concept implementation						59			
		3.4.3	Adding time dependency to the model						68			
		3.4.4	Notes on the kilometer function						70			
		3.4.5	Recommendations and Future work						71			
	3.5	Conch	usion						71			
		3.5.1	Recommendations and future work						72			
	3.6	Apper	ndix: Matlab code for optimizing energy hub						72			
	3.7	Matla	b code for cost model $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$	•	•		•		83			
4	Ster	tereo 3D cloud motion from tandem satellites										
	4.1	Introd	luction						96			
	4.2	Frame	work Implementation						96			
	4.3	4.3 Methodology							96			
		4.3.1	Extract positions from cloud observations						97			
		4.3.2	Cloud reconstruction						100			
		4.3.3	Cloud motion - bulk motion						102			
		4.3.4	Statistical shape analysis						104			
		4.3.5	Optical Flow						106			
	4.4	Concle	usion						107			
		4.4.1	Limitation						108			
5	Тор	Dutcl	h Solar Racing									
	An	ovel s	trategy model approach for the Bridgestone W	/oi	rlo	1 9	30	la	r			
	Challenge								111			
	5.1	Proble	em Formulation						112			
		5.1.1	Details of the race and available data						112			
	5.2	Optim	nization strategies						113			
		5.2.1	Data-based long-term strategy						114			
		5.2.2	Optimisation-based long-term strategy						121			
		5.2.3	Optimisation-based short-term strategy						128			
		5.2.4	Distance maximization under stochastic uncertainty						130			
	5.3	Conch	usions and future work						136			

\_\_\_\_

# Preface

The 2023 edition of the Study Group Mathematics with Industry (SWI) took place in Groningen for the first time, and it was the 170th European Study Group with Industry (ESGI). More than forty mathematicians from nine different countries responded to our invitation to solve challenging problems from five companies and organizations for a week in the stimulating atmosphere of the Academy Building of the University of Groningen.

The five problems came from globally operating companies Philips and ASMTP, from the European Space Agency (ESA) in collaboration with the Dutch KNMI, from the Dutch consulting firm DHM, and from the local Top Dutch Solar Racing team. On the following pages, we provide the scientific proceedings of the exciting problems and the solutions proposed by the SWI participant groups. A more accessible public proceedings version in Dutch can be found on the SWI website.

As organisation committee, we would like to thank NWO and platform wiskunde nederland for their support, and we are also grateful to the national SWI coordinators, Vivi Rottschäfer and Alessandro Di Bucchianico, for their advice and support.

The SWI 2023 organisation committee Stephan Trenn (chair) Ineke Schelhaas (Management Office Assistant) Tsega Gebretekle Tamás Görbe Hildeberto Jardon Kojakhmetov Matthijs de Jong Julian Koellermeier Nikolay Martynchuk Tobias Müller Pieter Tibboel Advisors SWI: Juan Peypouquet Wim Krijnen \_\_\_\_

# Chapter 1

# Data-driven Parameters Tuning for Predictive Performance Improvement of Wire Bonder Multi-body Model

XIAODONG CHENG<sup>1</sup>, ALESSANDRO DI BUCCHIANICO<sup>2</sup>, NAJMEH JA-VANMARDI<sup>3</sup>, MATTHIJS DE JONG<sup>3</sup>, EMIL LYKKE DIGET<sup>4</sup>, COLIN PLEASE<sup>5</sup>, DOMENICO LAHAYE<sup>6</sup>, QIYAO PENG<sup>7</sup>, CORDULA REISCH<sup>8</sup> AND DAVIDE SCLOSA<sup>9</sup>

**Abstract** This report describes work performed during SWI 2023 at the University of Groningen in relation with Problem 1 posed by the company ASMPT.

ASMPT makes a very large number of different machines for manufacturing of electronic devices. They have detailed simulation software of one of these machines and they compare the results of this with physical experimental results. There is a significant difference between the simulated and measured data, and it is the goal of this work to study how to estimate the parameters in the simulation model using the experimentally measured frequency response.

 $^2\mathrm{Eindhoven}$  University of Technology, The Netherlands

<sup>&</sup>lt;sup>1</sup>Wageningen University and Research, The Netherlands

<sup>&</sup>lt;sup>3</sup>Groningen University, The Netherlands

<sup>&</sup>lt;sup>4</sup>University of Southern Denmark

<sup>&</sup>lt;sup>5</sup>Oxford University, United Kingdom

<sup>&</sup>lt;sup>6</sup>Delft University of Technology, The Netherlands

<sup>&</sup>lt;sup>7</sup>Leiden University, The Netherlands

<sup>&</sup>lt;sup>8</sup>Technische Universität Braunschweig, Germany

<sup>&</sup>lt;sup>9</sup>Vrije University Amsterdam, The Netherlands

First, two toy models are studied to understand the challenges of parameter estimation in the frequency domain. Later, optimization methods are applied. Several different approaches of reducing the dimensionality of the parameter space are explored, including determining the parameter sensitivity. A suggestion for increasing the detail of the model, specifically related to the machine base, is also outlined.

In the summary, we supply a discussion of the key insights we gained during the week.

KEYWORDS: Frequency domain, optimisation, sensitivity analysis.

# 1.1 Introduction

With the increase in demand of electrical components the manufacturing has to be increasingly efficient and reliable. ASMPT is a world spanning company that produces as very large number of different machines for rapid manufacturing of electronic devices. Among these are wire bonding machines, shown in Figure 1.1, used in chip manufacturing. Wire bonding is the process of installing thin wires that connect internal chip areas to external breakout pads, see Figure 1.2.

Crucial to the efficient operation of these machines is that they can operate at high speed and hence they require control systems that will ensure vibrations generated in the system do not degrade the quality of the operations. To enable this measurements are taken where the machine is actively vibrated and its frequency response determined. In addition a detailed simulations of the vibrations are generated based on the CAD design of the machine. These simulations can then be used to enable high quality control of the vibrations to be made.

# 1.2 **Problem Description**

The problem of interest is how to make the computational simulations replicate, as best as possible, the measurements of the frequency response. To do this it is necessary to identify the values for the numerous parameters in the model used for the simulations. Some of these parameters can be measured by careful procedures (such as the mass of any particular part) but other parameters cannot be measured independently and must be identified by fitting the results of the model to the data. This report discusses procedures for performing such fitting.

An image of the CAD model can be seen in Figure 1.3. Here the large grey lower region is the chassis, the green is the base, the red is the x-stage, which can move horizontally in the x-direction, the purple is the y-stage, which can move horizontally in the y-direction, and the blue is the z-stage (also called the  $\theta$ -stage), which is a rotating arm that performs the wire bonding. In addition to these main features the system has some small flexibility in the joints and guides that enable motion of each of the stages, out of their specific x, y, or z plane. This flexibility is described by "parasitic elements" in the model. Some of these parasitic phenomena can be





Figure 1.1: An ASMPT wire bonder machine.

Figure 1.2: "Gold wire ball-bonded on a silicon die" by Mister rf, licensed under CC BY-SA 4.0.



Figure 1.3: Details of the CAD design.

quantified by measurements, however there is great uncertainty in these measurements



Figure 1.4: Outline of the data and simulations used in this report.

and, in addition some parasitic behaviour cannot be quantified independently in any easy manner.

The overall system that we are interested in can be summarised in Figure 1.4. Here the measured data from experiments is given by  $D_0$ , while S(p) denotes the full nonlinear numerical model built in Simulink with the Simscape Multibody toolbox. The MATLAB program linmod is applied to the S(p) to produce a linearised model  $S_{\text{lin}}(p)$  which can then generate  $\hat{D}$ , the computed frequency response of the linearised system. Finally some norm of the error between the simulated and measured data,  $\hat{D} - D_0$  will be exploited to determine how well a specific set of parameters  $p_0$  fits the measurements so that optimal parameters values can be identified.

## 1.3 Approach

This report sets out the approaches that were considered for finding the optimal set of parameters to make the data from the simulations fit the measured data. The full model is very complicated and so we will start by considering two highly simplified models in order to gain some insight into the difficulties that might arise in fitting parameters to such models of oscillations. We will then look at possible cost functions that might be used to assess the quality of the fit between the simulated and measured data. Subsequently we focus our attention on how to seek optimal parameters conditions when the number of parameters is very large, as this is a serious barrier to using conventional methods of optimisation.

## 1.4 Modelling

In this section two toy models of the full dynamic model are investigated, and an alternative model for the machine base is considered.

#### 1.4.1 Toy Models

We consider two very simple models of mass-spring-damper systems to see what type of behaviour occurs and how parameters might be fit. Additional details and code for



Figure 1.5: Diagram of simple one-mass oscillator.

these models can be found online<sup>10</sup>.

#### **One-Mass Oscillator**

A single point mass attached to a stationary structure by a single spring and damper is studied. The mass can be subjected to some external force, and the position, x(t), of the mass can be observed. This simple model is demonstrated in Figure 1.5.

The mass  $m_1$  is assumed to be attached to a reference frame with a spring with spring constant  $k_1$  and damper with damping constant  $d_1$ . The mass will thus oscillate horizontally around its equilibrium point. We assume that the mass can be determined a-priori using a scale. This allows us to reduce the design problem from three to two design variables and to consider the scaled stiffness constant  $k_1/m_1$  and the scaled damping constant  $d_1/m_1$ .

The equation of motion is

$$m_1 \ddot{x}(t) + d_1 \dot{x} + k_1 x(t) = F(t), \tag{1.1}$$

where F(t) is an external force. Initial conditions complete the problem.

In the following, we study the frequency response of the system. The response is the ratio of the position to the forcing, and depends on the parameter values  $m_1$ ,  $k_1$ and  $d_1$ .

The state-space representation of the mechanical system allows to determine the transfer function. We therefore consider this representation here. The state-space representation is written in terms of the state vector X(t) defined as

$$X(t) = \begin{pmatrix} x(t) \\ \dot{x}(t) \end{pmatrix}.$$
 (1.2)

The output and control are written as y(t) and u(t), respectively, where y(t) = x(t), u(t) = F(t). Unlike in more realistic applications, the equation governing the motion of the single point-mass is a linear ordinary differential equation. This allows us to recover the classical results obtained using Laplace transforms. We furthermore consider to have a single control variable only. This control parameter u has no part

<sup>&</sup>lt;sup>10</sup>https://github.com/ziolai/software/blob/master/swi-groningen2023.ipynb

in the output. The state-space representation then reads

$$\dot{X} = AX + Bu, \tag{1.3}$$

$$y = CX + Du, \tag{1.4}$$

where

$$A = \begin{pmatrix} 0 & 1 \\ -k_1/m_1 & -d_1/m_1 \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$
  

$$C = \begin{pmatrix} 1 & 0 \end{pmatrix}, \quad \text{and} \quad D = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$
(1.5)

where u = F(t) is the external force exciting the system equally at all frequencies with an amplitude  $F_0$ .

The observability of the system depends on the rank of the matrix

$$\begin{bmatrix} C\\CA \end{bmatrix} = \begin{pmatrix} 1 & 0\\0 & 1, \end{pmatrix}$$
(1.6)

which in this case is equal to 2, and therefore the system is fully observable and controllable.

The transfer function of the system is defined in the Laplace domain with the variable s, given by

$$H(s) = \frac{1}{m_1 s^2 + d_1 s + k_1}.$$
(1.7)

After setting  $s = j \omega$ , where  $\omega$  is the frequency, we obtain the transfer function in the frequency domain

$$H(\omega) = \frac{1}{-m_1 \,\omega^2 + j \,d_1 \,\omega + k_1} \,. \tag{1.8}$$

A numerical example using the parameter values of the z-stage of the wire bonder machine highlights the insight we may gain from the transfer function. The mass of the z-stage was assumed as  $m_1 = 0.1363$  kg, the spring constant and damping with respect to the connection to the y-stage was assumed as  $k_1 = 1.4429$  Nm/rad and  $d_1 = 0.0031$  Nms/rad. Regarding the z-stage as a single mass-spring-damper system, we get the frequency response displayed in Figure 1.6.

The graph in the frequency domain shows three characteristics. First, for small frequencies we observe the limit case for  $\omega \to 0$ . This limit allows due to  $\log(|H(\omega)|) = \log(1/k_1)$  to determine the spring constant  $k_1$ .

Second, in the limit  $\omega \to \infty$  we have  $H(\omega) = -1/(m_1\omega^2)$ . Thus in this limit  $\log(|H(\omega)|) = \log(1/(m_1)) - 2\log(\omega)$ . This limit allows to determine the value of  $m_1$ , and we note the slope is 40 dB per decade.

Finally, the value of the damping coefficient  $d_1$  can be determined from the value of  $H(\omega)$  at the resonant frequency  $\omega_0 = \sqrt{\frac{k_1}{m_1}}$ , where

$$H(\omega_0) = \frac{1}{jd_1\sqrt{k_1/m_1}}.$$
(1.9)



Figure 1.6: Bode diagram of simple one-mass oscillator using the parameter values of the z-stage.

The Bode plot for the amplitude therefore allows to determine all three parameters of the single mass-oscillator.

Even though the transfer function is linear in forcing  $F_0$ , it is non-linear in the parameters  $k_1$ ,  $d_1$  and  $m_1$ . The optimization problem we wish to solve is thus non-linear in these parameters.

#### **Two-Mass Oscillator**

The second toy model has a single point mass attached to a stationary structure by a single spring and damper, which is subjected to some external force. However, we also consider that there is a second point mass which is connected by springs and dampers, both to the first mass and also to the stationary structure. The second mass and its connections are an attempt to model the parasitic behaviour that occurs in the real system. Hence our only observations are on the position, x(t) of the first mass, and the response is the ratio of the motion of this first mass to its forcing.

We want to investigate the influence of adding more spring-dampers to a two masses system. Therefore, we compare the behavior of the system in Figure 1.7 with the system behavior without the spring and damper connecting mass  $m_2$  with the stationary structure.

The dynamical system reads

$$M\begin{pmatrix} \ddot{x}(t)\\ \ddot{y}(t) \end{pmatrix} + D\begin{pmatrix} \dot{x}(t)\\ \dot{y}(t) \end{pmatrix} + K\begin{pmatrix} x(t)\\ y(t) \end{pmatrix} = \begin{pmatrix} F(t)\\ G(t) \end{pmatrix},$$
(1.10)



Figure 1.7: Diagram of simple two-mass oscillator.

with the diagonal mass matrix

$$M = \begin{pmatrix} m_1 & 0\\ 0 & m_2 \end{pmatrix},$$

the stiffness matrix

$$K = \begin{pmatrix} k_1 + k_{12} & -k_{12} \\ -k_{12} & k_2 + k_{12} \end{pmatrix},$$

and the damping matrix

$$D = \begin{pmatrix} d_1 + d_{12} & -d_{12} \\ -d_{12} & d_2 + d_{21} \end{pmatrix}.$$

We regard the spring  $k_2$  and damper  $d_2$  to be parasitic. In the following comparison we investigate the cases  $k_2 = d_2 = 0$  and  $k_2, d_2 > 0$ .

Transferring the ordinary differential equation in the frequency domain gives the representation

$$F(\omega) = Z(\omega) X(\omega), \qquad (1.11)$$

where  $Z(\omega)$  is the impedance matrix given by  $Z(\omega) = -\omega^2 M - j \omega D + K$ . Solving for the unknown displacement can thus be done as  $X(\omega) = H(\omega)F(\omega)$ , where

$$H(\omega) = Z^{-1}(\omega) \tag{1.12}$$

is the transfer function.

For our system, we can see that the influence of  $k_2$  and  $d_2$  affects the whole system behavior. The transfer function then reads

$$H(\omega) = \frac{1}{\det(Z(\omega))} \begin{pmatrix} H_{11} & -(j\omega d_{12} - k_{12}) \\ -(j\omega d_{12} - k_{12}) & H_{22} \end{pmatrix},$$
 (1.13)

where  $\det(Z)$  depends on  $d_2$  and  $k_2$  in almost all coefficients of the monomials of  $\omega$  and the diagonal elements are

$$H_{11} = -\omega^2 m_2 - j\omega(d_{12} + d_2) + k_{12} + k_2, \qquad (1.14)$$

and

$$H_{22} = -\omega^2 m_1 - j\omega(d_1 + d_{12}) + k_1 + k_{12}.$$
 (1.15)



Figure 1.8: Bode plot of the two-mass oscillator. Systems response of mass  $m_1$  on a force on mass  $m_1$ . Left: Additional connection with  $k_2, d_2 \neq 0$  to the stationary structure. Right: No connection with  $k_2, d_2 = 0$ .

The change in the system behavior is displayed in Figure 1.8, where the external force acts on mass  $m_1$ , and Figure 1.9, where the external force acts on mass  $m_2$ . The parameter values are chosen accordingly to the *y*-stage and the *z*-stage of the wire bonder machine.

While the change in the system response by a force on mass  $m_1$  is negligible, the dynamics of mass  $m_1$  on a force acting on mass  $m_2$  are larger. Figure 1.9 shows, for the model with parasitic connections of mass  $m_2$  to the stationary structure, an oscillating behavior for large frequencies. The decay of the magnitude is smaller for the model including the additional parasitic spring and damper.

The two changes in the slope of the magnitude in Figure 1.9 roughly correspond to the resonance frequencies of the single one-mass-system for the z-stage, see Figure 1.6, and the y-stage.

#### **1.4.2** Alternative Model of the Chassis

The chassis of the machine is currently described in the model by a single rigid body at a fixed position with appropriate mass and inertial parameters. This is reasonable for any part of the machine where the speed of wave propagation in the part is sufficient



Figure 1.9: Bode plot of the two-mass oscillator. Systems response of mass  $m_1$  on a force on mass  $m_2$ . Left: Additional connection with  $k_2, d_2 \neq 0$  to the stationary structure. Right: No connection with  $k_2, d_2 = 0$ .

that it moves quasi-statically. However, for the chassis, this might not be an adequate approximation since it has dimensions of around 1 m and shear waves travel at around 3 km/s, so that frequencies of over 1 kHz may create waves that are as short as the part. A more complex model of this could be created, using the finite element method (FEM) for example, however here we describe an alternative approach that retains the simplicity and number of parameters currently exploited in the rigid body model but accounts for wave propagation.

We consider an idealised base consisting of a uniform block of elastic material of height L. On top of the block the other parts of the machine are placed and these dictate a known displacement in the horizontal plane x(t), y(t), and vertical displacement z(t). We are interested in determining the response by quantifying how the resulting horizontal and vertical forces acting on this surface depend on the displacements.

Our simplification will be that the motion of the block is uniform at each height z and that the horizontal motion (u, v) induces shear waves and the vertical motion, w, induces compressive waves. The bottom of the base is typically on legs and these are quite weak so for the purposes of modelling it is adequate to assume the bottom

surface is stress-free. We now give details of the behaviour in the x direction but the other two directions are similar.

The governing equation is

$$\rho \frac{\partial^2 u}{\partial t^2} = \mu \left( \frac{\partial^2 u}{\partial z^2} - d_s \frac{\partial u}{\partial t} \right), \qquad (1.16)$$

where  $\rho$  is the material density,  $\mu$  its shear modulus and  $d_s$  the shear damping coefficient. Note the equation for v is identical, and for w is the same except  $\mu$  is replaced by  $\lambda + 2mu$ , where  $\lambda$  is the first Lame coefficient, and  $d_s$  replaced by the compressive dissipation  $d_c$ . Boundary conditions for the problem are that

$$u(t,0) = x(t)$$
 and  $\mu \frac{\partial u}{\partial z}(t,L) = 0.$  (1.17)

If we move to frequency space, by either taking Fourier transforms or simply considering  $x(t) = x_{\omega} \exp(-j\omega)$  with  $u(t) = u_{\omega}(z) \exp(-j\omega)$  we can find that the solution is given by

$$u_{\omega}(t) = x_{\omega} \frac{\exp\left((2L/c)\sqrt{\omega^2 - jd_s\omega}\right)}{1 + \exp\left((2L/c)\sqrt{\omega^2 - jd_s\omega}\right)},$$
(1.18)

where  $c = \sqrt{\mu/\rho}$  is the shear wave speed.

Finally we are interested in the relation between the displacement and the stress on the top surface, so that we can incorporate it into the more general modelling framework. Here we find

stress = 
$$\mu \frac{\partial u}{\partial z}$$
  
=  $x_{\omega} \left( (2L/c) \sqrt{\omega^2 - jd_s \omega} \right)$   
 $\cdot \frac{1 - \exp\left( (2L/c) \sqrt{\omega^2 - jd_s \omega} \right)}{1 + \exp\left( (2L/c) \sqrt{\omega^2 - jd_s \omega} \right)}.$  (1.19)

We can therefore interpret this expression as giving a transfer function in the form

$$-\sqrt{\frac{k}{m(\omega^2 - jd_s\omega)}} \coth\left(\sqrt{mk}\sqrt{\omega^2 - jd_s\omega}\right),\tag{1.20}$$

where we have introduced a mass, m, and a stiffness, k. This puts the transfer function in the same form as that for a single mass connected with stiffness and damping where we would expect

$$\frac{1}{-m\omega^2 + jd\omega + k},\tag{1.21}$$

and observe that the two agree, with the frequency behaviour proportion to  $1/(m\omega^2)$ , when the frequency is low in the shear wave and high in the single mass.

Implementation of this model of the chassis may not be straightforward in Sim-Scape. An alternative approach would be to consider the chassis as a small number of individual masses linked by springs and dampers that represent the elastic behaviour of the region. Such a simplified model would go some way to describing the wave-like behaviour in the region.

#### 1.5 Measured Data

The system has been measured using three sources of excitation and three senors of the motion with one of each on the x-stage, the y-stage and the z- (or  $\theta$ -) stage. After analysis this measured data consists of nine complex numbers (the xx, yy, zz, xy, xzand yz representing the response, where mn corresponds to response in direction mdue to excitation in n) at each frequency. We denote the measured data by  $Y_{\text{mes.}xx}$ and the simulated data from models by  $Y_{\text{sim.}xx}(\theta)$  where  $\theta$  is the vector of model parameters. Experiment data was supplied by the project owner.

The data can be presented in the form of Bode plots where the magnitude (measured typically in dB) of each response and the phase (measured typically in degrees, and restricted to the range [-180; 180]) of each response is plotted against the logarithm of the frequency.

## **1.6** Optimization Methods

We considered the full model of the wire bonder as developed by ASMPT and explored approaches to finding the parameter values that would fit the given data to the computations of the full model. A first issue to address was what cost function should be used to quantify the fit between the data and the computations. We discuss this is in Section 1.6.1. Next we explore approaches that will allow optimal parameter values to be estimated when the number of parameters is very large. Here we take two directions. The first is to look at the sensitivity of the cost function to the entire parameter set. Results for this are discussed in Section 1.6.2. The second approach was to do a limited optimisation where part of the parameter set was considered to be fixed at its nominal value while a smaller set was considered using an optimisation process. Three examples of this are presented. Each example optimises different parts of the parameter space. The first example studies the inertial parameters of the system, the second example we vary the parasitic parameters for the complete system, and in the third example we study the parasitic parameters in relation with the z-stage. These three examples are presented in Section 1.6.3, 1.6.4, and 1.6.5, respectively.

In our examples we have concentrated on local optimisation methods but this problem does require methods that will seek the global optimum amongst many local optima. A discussion and overview of current state of derivative free global optimisation methods can be found in Cartis, Fiala, et al. 2019 and Cartis, Roberts, and Sheridan-Methven 2022.

#### 1.6.1 Cost Function

The cost function is a crucial part of an optimization problem. Multiple error measures can be considered.

One strategy is to concentrate on just the magnitude of the complex function and only consider a small subset (e.g. just one) of the various responses. Hence one possible cost function is:

$$C_{\rm mag}(\theta) = \frac{1}{2} 20 \log_{10} \left( \frac{1}{N} \sum^{N} \left( |Y_{\rm mes}| - |Y_{\rm sim}(\theta)| \right)^2 \right) \, \mathrm{dB}, \tag{1.22}$$

where N is the number of frequency bins. This function has the property that it is dominated by those frequencies with large responses. For example the measured data shows approx 100 dB variation across the expected frequency range so any fitting will ignore large parts of the curve.

An alternative, that addresses the large variations, is to make the cost function directly applicable to the typical response curve where we seek to minimize the distance between the measured and simulated curves. This corresponds to taking

$$C_{\text{mag}}(\theta) = \left(\frac{1}{N} \sum_{n=1}^{N} (20 \log_{10} |Y_{\text{mes}}| - 20 \log_{10} |Y_{\text{sim}}(\theta)|)^2\right)^{1/2} dB.$$
(1.23)

A similar account of large variations can be made by normalising each simulated response by the magnitude of the corresponding measured response. It is then possible to simply consider the magnitude of the simulated response

$$C_{\rm mag}(\theta) = \frac{1}{N} \sum^{N} \left| \frac{Y_{\rm mes}}{|Y_{\rm mes}|} - \frac{Y_{\rm sim}(\theta)}{|Y_{\rm mes}|} \right|.$$
(1.24)

Note that this last cost function has the property that it does not significantly penalise simulations that greatly under-predict (in the sense of dB) the magnitude of response of the system but does penalise over-prediction of the magnitude.

The issue of the nine different responses can be addressed in a number of ways but the vast difference in the size of the various responses (e.g.  $Y_{\text{mes},xx}$  is more than 50 dB smaller than  $Y_{\text{mes},zz}$ ) makes it preferable to extend the ideas in (1.23) and (1.24) but to consider a sum across the nine different responses, perhaps with some weighting if certain directions are perceived as being of greater physical importance.

It is worth noting here that the norms used in the expressions for the cost functions have been assumed to be the  $L^2$ -norm. However, there are benefits to considering the  $L^1$ -norm. A general optimisation problem minimizing a cost function with respect to the parameters subject to bounds and constraints looks like:

$$\min_{\theta} \quad C_{\max}(\theta), \\ \text{s.t.} \quad \frac{\theta}{d} \leq \theta \leq \bar{\theta}, \\ m_i > 0, \\ I_i \succ 0,$$
 (1.25)

where  $\underline{\theta}$  and  $\overline{\theta}$  are the lower and upper parameter bound, respectively,  $m_i$  is the *i*th stage mass, and  $I_i$  its corresponding inertia matrix around the center of mass. The constraints on the inertial parameters of each rigid body in a robotics arm are studied in Sousa and Cortesão 2014; the so-called physical feasibility. We advise the company partner to incorporate this into a solution.

#### 1.6.2 Sensitivity Analysis

To reduce the dimensionality of the problem, we study which parameters affect the cost function the most. Therefore, we conduct a sensitivity analysis on the parameters, which provide an insight on how the change of parameter values affect the value of the cost function. This is performed by changing one parameter at a time, comparing with the experiment data, and computing variation to cost function. Each parameter has been varied in the range of 50% - 150% around its nominal value. For this sensitivity analysis we have used the cost function given in (1.22), but these results could equally be generated for the other cost functions described earlier.

In summary the important parameters appear to be the inertial parameters, especially masses of x-stage and y-stage, while damper and spring parameters and masses of chassis do not influence the cost much. Note that besides physical constraints on mass and inertia, the rigid body center of mass (CoM) could also be considered so that it remains in the convex hull of the rigid body to be realistic.

The specific sensitivity procedure is described in Algorithm 1.

In this algorithm, we set  $r_i = 0.5$ ,  $r_u = 1.5$ , and  $r_{\Delta} = 0.1$ . In each loop, only one parameter is changing, while the others are fixed at their initial values. The values of the cost function are computed with different parameter values, which is illustrated in Figure 1.10. We can also evaluate the cost function in each direction x, y, and z, when changing the parameter values. Then, Figure 1.11 shows the results for the cost function for each direction.

Combining the data obtained from Algorithm 1 and the Figures 1.10, 1.11, 1.12, we have the following observations.

- There are two significant parameters:
  - The mass of the x- and the y-stage.
- There are other parameters that affect the cost function:

#### Algorithm 1 Sensitivity Analysis

 $\begin{array}{ll} \textbf{Require: nominal values of parameters } \theta_0 \in \mathbb{R}^{n_\theta}; \\ \text{range } [r_l, r_u], \text{ step size } r_\Delta \\ \textbf{Ensure: cost function matrix } \mathcal{C} \\ \theta \leftarrow \theta_0 \\ \textbf{for do } i = 1, 2, ..., n_\theta \\ \textbf{b traverse all parameters} \\ j \leftarrow 1, r \leftarrow r_l + jr_\Delta \\ \textbf{while } r \neq 1 \text{ and } r \leq r_u \text{ do} \\ \theta \leftarrow \theta_0 \\ \theta(i) \leftarrow \theta(i)r \\ \text{Compute } C_{\text{mag}}(\theta) \text{ of the cost function (1.22).} \\ \mathcal{C}_{ij} \leftarrow C_{\text{mag}}(\theta) \\ j \leftarrow j + 1, r = r_l + jr_\Delta \\ \textbf{end while} \\ \textbf{end for} \end{array}$ 



Figure 1.10: Value of the cost function (1.22) as the 70 parameters vary in the range 50% to 150%.

- The center of mass and inertia for the x-, the y-, and the z-stage, and the mass of the z-stage.
- The rest of parameters do not affect the cost function; the parasitic spring and



Figure 1.11: Values of the cost functions computed for x-, y-, and z-directions as the 70 parameters vary in the range 50% to 150%.



Figure 1.12: The maximum change in the magnitude of the cost function vs. parameter when the parameters are changed individually.

dampers have little effect on the overall cost.

### 1.6.3 Experiment 1: Varying the Inertial Parameters

Sensitivity analysis suggests a prominent role of the masses in the model, which is reflected in a significant role in optimization. We will allow masses to deviate, perhaps unrealistically, up to  $\pm 20\%$  from the initial parameter values. Quite surprisingly, we see that a better overall fit is obtained for masses close to the maximum deviation allowed.

The experiment details are as follows:

- Frequency-range: 200–2000 Hz
- Parameters varied (17 total):
  - Inertial parameters of the x-stage; mass, center of mass, inertia.
  - Parasitic torsional spring between the base and x-stage.
  - Kinematic position of the joint between the base and the x-stage.
- Maximum deviation allowed:  $\pm 20\%$  for masses,  $\pm 50\%$  for all other parameters
- Optimization method: fmincon with numerically computed derivatives
- We restrict to fitting the  $Y_{xx}$ -response of the x-stage

Overall, the optimization algorithm leads to a better fit. In particular, we obtain a better fit of the 0.245 dip, see Figure 1.13.



Figure 1.13: Optimizing the mass gives a better fit of the 0.245 dip. The frequency range was normalized by dividing with the Nyquist frequency.

However, we notice that the one optimized parameter, namely the mass of the x-stage, coincides with the maximum value allowed. That is, a better fit has been obtained by increasing the x-stage mass by 20%.

This shows that allowing masses to change leads to a better fit, but probably to an unrealistic model. We suggest some possible causes: First, the "real" masses might not correspond to the global minimum of the cost function; second, large deviations might have brought us on a slope that minimizes the cost function but leads to a region of non-physical parameters (see Section 1.6.1 for a possible solution); third, the algorithm in use might not be suitable for the problem at hand.

#### 1.6.4 Experiment 2: Varying the Parasitic parameters

As the masses are quite accurately measured and the inertial parameters can be well estimated from CAD drawings the full model was explored with these parameters set at their nominal values. The parasitic behaviour of the system is much less well understood and hence in this section we explore how the simulations fit to the data when only parasitic effects are altered. At the same frequency range we vary the parasitic parameters. These are the details:

- Frequency-range: 200-2000 Hz
- Parameters varied: Parasitic parameters
- Parameter bounds:  $\theta > 0$
- Optimization method: fmincon with numerically computed derivatives
- Objective: Minimize diagonal costs (related to  $Y_{xx}, Y_{yy}, Y_{zz}$ )

In contrast to the case of masses, the optimization process here goes through many plateaus, see Figure 1.14, taking a lot of time. The result is an overall better fit in the relevant range 200–2000 Hz, see Figure 1.15.

Notice that one parameter, the parasitic torsional damping between the base and the x-stage, is 40 times its original value, see Figure 1.16. This does not come as a huge surprise. Indeed, sensitivity analysis already shows that modifying these parameters in the 50%-150% range has little to no effect on the system.

#### 1.6.5 Experiment 3: Exploring the z-Stage

Instead of optimizing the complete cost of all three axes, in this simulation we only optimize the behaviour in the z-axis. According to the project owner, the measurement of the z-axis is more precise in the whole range from 1-4000 Hz than the other two axes. We optimize 12 parameters characterising the parasitic joints between the three stages, see Figure 1.17. The details are as follows:

• Frequency-range: 1–1000 Hz



Figure 1.14: Decrease of the cost-functional during the optimization process. The convergence process exhibits various plateaus.



Figure 1.15:  $F_x$ -response of x-stage with optimized parasitic parameters and original parameters. The frequency range was normalized by dividing with the Nyquist frequency.

- Parameters varied: Parasitic parameters
- Parameter bounds:  $\theta > 0$
- Optimization method: fmincon with numerically computed derivatives



Figure 1.16: Absolute relative difference in percentage for the parasitic spring and damper coefficients before and after optimisation.

• Objective: Minimize z-stage cost (related to  $Y_{zz}$ )

Unsurprisingly, it can be seen that the two last parameters, the spring and damper between the y- and the z-stage, change the most as a result of the optimization; around 30%, see Figure 1.17. The fit itself can be seen on Figure 1.18, where the fit of the bump at  $\sim 12$  Hz is better with the new parameters compared with the nominal model.

# 1.7 Some Ideas on Data Analysis

In this section, we discuss some issues related to inaccurate data, which can generate incorrect models. Considering these factors may help to produce rich datasets, especially in low frequencies. Noticeably, discovering good datasets is important in system identification.

• The effect of the noise on the training of models should be considered. In studies like Xue, Whitecross, and Mirzasoleiman 2022, Dhifallah and Lu 2021,



Figure 1.17: Absolute relative change in percentage for the parasitic spring and damper coefficients after optimization.

the effect of noisy data on the performance of the models trained on them is shown. However, real-world datasets often contain a significant fraction of noisy labels and uncertainties. In Dhifallah and Lu 2021, the authors suggest robust training methods to learn from noisy-labeled data.

- To identify a model for a system it is important to check 'sufficient excitation' conditions. Persistent Excitation (PE) typically results from sufficiently rich reference inputs, and parameter convergence is achieved only in the presence of PE. In Bittanti, Campi, and Guo 2000, Lee, Tan, and Nešić 2014 and Song, Zhao, and Krstic 2016, more information about PE conditions can be found.
- As studied in Jolliffe and Cadima 2016, Principal component analysis (PCA) is a technique for reducing the dimensionality of such datasets, increasing interpretability but at the same time minimizing information loss.



Figure 1.18: Frequency response of the z-stage. The frequency range was normalized by dividing with the Nyquist frequency.

# 1.8 Discussion and Recommendations

The key insights in to the behaviour of the simulation model and how predictions might be fitted to experimental measurements are as follows. The sensitivity of each parameter depends on the frequency range being considered, i.e. dominant parameters change with respect to different frequency ranges. In particular we observe that the low frequencies are dominated by stiffness parameters, at near resonance frequencies the damping parameters dominate, and at the high frequencies the inertial parameters dominate. Exploring a very limited range of frequencies may allow improved estimation of parameters for starting values for subsequent fitting to the large range of relevant frequencies.

The choice of the cost function has a great impact on the performance of the optimisation. Finding a global minimum is plagued by local minima, even in (very simple) toy models. The scale of the discrepancy (measured in the  $L^2$  norm) in z-direction is  $e^{50}$  orders of magnitude higher than those of the x- and y-directions and needs to be accounted for carefully.

From our preliminary explorations we make the following recommendations. It may be beneficial to extend the model to more carefully account for shear waves in the chassis. This is because in the relevant frequency range the chassis may be larger than the shear wave length and hence not well described by a single point mass. As a preliminary scoping of parameters values it may be beneficial to focus on the local optimization of parameters rather than global. It may also help to divide the frequency domain into a number of smaller ranges and consider initially optimising in each range. Some parameters can be kept constant, for example some damping parameters at low frequencies. The results of the sensitivity test results might guide which parameters to consider and which to hold constant in the different frequencies ranges.

The sensitivity analysis in this report shows that the inertial parameters of the three stages dominate the cost function. The problem with this is that these are typically values are quite accurately known from independent measurements; these essentially create the nominal model. Instead, the optimisation could initially only be performed on the parasitic parts of the model. More work is needed to look at the sensitivities that occur for different cost functions. It is necessary to adopt strategies that can overcome the curse of dimensionality.

We suggest the following might be appropriate methods:

- Perform a sensitivity analysis for different frequency ranges separately,
- Optimize considering *only* the parasitic parameters as variables,
- Apply physical constraints to the inertial parameters of the rigid bodies including considering them as constant,
- Apply model order reduction, methodology.

Because of the vast range of responses in the x, y, and  $\theta$  directions these should either be normalised or their logarithm taken when considering what cost function to use. Because of the need to find global optimal, state-of-the-art global optimization software, e.g. Py-BOBYQA, should be explored.

## References

- Bittanti, Sergio, Marco C Campi, and Lei Guo (2000). "Persistence of excitation properties for the identification of time-varying systems". In: Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No. 00CH37187). Vol. 1. IEEE, pp. 680–684.
- Cartis, Coralia, Jan Fiala, et al. (2019). "Improving the flexibility and robustness of model-based derivative-free optimization solvers". In: ACM Transactions on Mathematical Software (TOMS) 45.3, pp. 1–41.
- Cartis, Coralia, Lindon Roberts, and Oliver Sheridan-Methven (2022). "Escaping local minima with local derivative-free methods: a numerical investigation". In: *Optimization* 71.8, pp. 2343–2373. DOI: 10.1080/02331934.2021.1883015. URL: https://doi.org/10.1080/02331934.2021.1883015.
- Dhifallah, Oussama and Yue Lu (2021). "On the inherent regularization effects of noise injection during training". In: International Conference on Machine Learning. PMLR, pp. 2665–2675.
- Jolliffe, Ian T and Jorge Cadima (2016). "Principal component analysis: a review and recent developments". In: *Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences* 374.2065, p. 20150202.

- Lee, Ti-Chung, Ying Tan, and Dragan Nešić (2014). "Stability and persistent excitation in signal sets". In: *IEEE Transactions on Automatic Control* 60.5, pp. 1188– 1203.
- Song, Yongduan, Kai Zhao, and Miroslav Krstic (2016). "Adaptive control with exponential regulation in the absence of persistent excitation". In: *IEEE Transactions* on Automatic Control 62.5, pp. 2589–2596.
- Sousa, Cristóvão D. and Rui Cortesão (May 2014). "Physical feasibility of robot base inertial parameter identification: A linear matrix inequality approach". en. In: *The International Journal of Robotics Research* 33 (6), pp. 931–944. DOI: 10.1177/ 0278364913514870. URL: http://dx.doi.org/10.1177/0278364913514870.
- Xue, Yihao, Kyle Whitecross, and Baharan Mirzasoleiman (2022). "The Final Ascent: When Bigger Models Generalize Worse on Noisy-Labeled Data". In: *arXiv preprint arXiv:2208.08003*.

# Chapter 2

# Frequency Domain Characteristics in Nonlinear Trimmer Dynamics

JAMES P. HARRIS<sup>1</sup>, JOSÉ MUJICA<sup>2</sup>, VIVI ROTTSCHÄFER<sup>3</sup>, ERIC SANDIN<sup>4</sup>

**Abstract** We analyse a dynamical system describing the movement of the blade of a hair trimmer. Using nondimensionalisation and perturbation methods, we systematically analyse this model, with the goal of explaining a dip in the current amplitude observed in numerical simulations. We find that the motion of the blade is controlled by seven dimensionless parameters, and the dip in amplitude of the current corresponds to one of the dimensionless parameter groups taking the value of one. There is excellent agreement between our asymptotic solutions and long-time behaviour of the model revealed by numerical simulation.

# 2.1 Introduction

At the Study Group Mathematics with Industry in Groningen in 2023, we worked on a problem posed by Philips. Philips manufacture, amongst many other products, hair and beard trimmers and their question was related to these. A picture of such a trimmer is given in Figure 2.1(a).

<sup>&</sup>lt;sup>1</sup>Mathematical Institute, University of Oxford, Oxford, UK

<sup>&</sup>lt;sup>2</sup>Department of Mathematics, Vrije Universiteit Amsterdam, Amsterdam, The Netherlands

<sup>&</sup>lt;sup>3</sup>Mathematical Institute, Leiden University, Korteweg–de Vries Institute for Mathematics, University of Amsterdam, Amsterdam, The Netherlands

<sup>&</sup>lt;sup>4</sup>Department of Mathematics, Vrije Universiteit Amsterdam, Amsterdam, The Netherlands



Figure 2.1: A typical trimmer head (a) and a schematic representation of of its functioning (b). Figures provided by Philips.

Philips use dynamical models to simulate and analyse the dynamic motion of the trimmer, in order to optimise performance. A simple but relevant model is one in which a dc-motor is coupled to a mass spring damper system (the cutting element). Individually, each system has linear dynamics. However, the interconnection between them is nonlinear which makes the whole system nonlinear.

The mechanical part of the shaver head consists of a toothed blade, attached via a small piece of plastic (the driving bridge) to an eccentric pin on a motor. The motion of the blade is restricted by two springs attached to the casing of the shaver head. The blade cuts hair which passes between itself and a comb mounted to the shaver head casing. A diagram of a shaver head is shown in Figure 2.1(b).

The model we analyse is described by equations (2.1)-(2.4) with typical parameter values given in Table 2.1. The motor current through the electric part consists of a dc-component and a sinusoidal component. When performing numerical simulations, Philips observed that when varying the voltage  $\hat{V}$ , the amplitude of the sinusoidal part of the motor current attains a minimum for a certain value of  $\hat{V}$ . See Figure 2.2 for the results of a numerical simulation where parameter values are taken as in Table 2.1 for two different choices of one of the parameters,  $\hat{k}_s$ . Philips would like to know whether this point can be determined using analytical methods. We provide insight in this behaviour using the techniques of nondimensionalisation and asymptotic analysis. These methods are widely applicable and can also be used for other (larger) systems.

First, in Section 2.2, we formulate the model, introduce the concept of nondimensionalisation and make the system dimensionless. Then, in Section 2.3, we use asymptotic analysis to study the solutions to the dimensionless system. Finally, in Section 2.4, we conclude with a discussion and recommendations.

## 2.2 Model

We study a physical model of the shaver head. The model is illustrated in Figure 2.3. We model the shaver head casing as a rigid wall, to which the blade is connected via a linearly-damped Hookean spring with spring constant  $\hat{k}_s$ , and damping coefficient



Figure 2.2: The amplitude of the oscillatory part of the motor current as a function of voltage from numerical solution of (2.1)–(2.3) for  $\hat{k}_s = 1000N/m$  (blue) and  $\hat{k}_s = 100N/m$  (red). All other parameters are taken from Table 2.1.



Figure 2.3: Physical set up for the coupled system (2.1)-(2.4).

 $\hat{d}_s$ . The blade, of mass  $\hat{m}$ , is attached to the eccentric pin of the motor via the driving bridge which we model as a Hookean spring with stiffness  $\hat{k}_b$ . We assume that the driving bridge is connected to the motor pin at all times, and slides without friction in such a way that the blade moves only in one line, radially from the centre of the motor. We denote the displacement of the blade, at time  $\hat{t}$ , by  $\hat{x}(\hat{t})$ . We assume the motor centre is fixed relative to the shaver head casing, and we denote the distance between the centre of the motor and the eccentric pin by  $\hat{A}$ . We denote the angle of the pin from its starting position by  $\theta(\hat{t})$ . We assume the motor has inertia  $\hat{J}$ , self-inductance  $\hat{L}$ , resistance  $\hat{R}$ , motor constant  $\hat{K}$ , and a linear damping coefficient of  $\hat{d}_m$ . We denote the current through the motor by  $\hat{I}(\hat{t})$ . We assume the system is at rest before a constant voltage,  $\hat{V}$ , is applied across the motor for times  $\hat{t} > 0$ . Typical values for each of the parameters are given in Table 2.1.

Using the principles of conservation of momentum, angular momentum and (elec-

$\hat{m}$	$\hat{k}_s$	$\hat{d}_s$	$\hat{k}_b$	Â	Ĵ
kg	$kg s^{-2}$	$kg m s^{-1}$	$\rm kg~s^{-2}$	m	$\rm kg \ m^2$
$3.5 \cdot 10^{-3}$	$10^{3}$	0.025	$4 \cdot 10^3$	$1.5 \cdot 10^{-3}$	$3 \cdot 10^{-7}$
Ŕ	$\hat{d}_m$	$\hat{L}$	$\hat{R}$	$\hat{V}$	
$kg m^2 A^{-1} s^{-2}$	$\rm kg \ m \ s^{-1}$	$kg m^2 s^{-2} A^{-2}$	$V A^{-1}$	V	
$4 \cdot 10^{-3}$	0	$0.5 \cdot 10^{-3}$	0.7	1	

Table 2.1: Model parameters and their units, together with typical values provided by Philips.



Figure 2.4: Solution to (2.1)–(2.4), with parameters values taken from Table 2.1. Shown are time series for  $\hat{I}$  (left) and  $\hat{x}$  (right). Note that after a short transient, the current of the motor stabilizes following periodic motion.

trical) energy, we write down a system of three coupled ODEs for  $\hat{x}, \theta, \hat{I}$ ,

$$\hat{m}\frac{\mathrm{d}^2\hat{x}}{\mathrm{d}\hat{t}^2} = -\hat{k}_s\hat{x} - \hat{d}_s\frac{\mathrm{d}\hat{x}}{\mathrm{d}\hat{t}} - \hat{k}_b\left(\hat{x} + \hat{A}\sin(\theta)\right),\tag{2.1}$$

$$\hat{J}\frac{\mathrm{d}^{2}\theta}{\mathrm{d}\hat{t}^{2}} = \hat{K}\hat{I} - \hat{d}_{m}\frac{\mathrm{d}\theta}{\mathrm{d}\hat{t}} - \hat{k}_{b}\hat{A}\cos(\theta)\left(\hat{x} + \hat{A}\sin(\theta)\right),\qquad(2.2)$$

$$\hat{L}\frac{\mathrm{d}\hat{I}}{\mathrm{d}\hat{t}} = -\hat{K}\frac{\mathrm{d}\theta}{\mathrm{d}\hat{t}} - \hat{R}\hat{I} + \hat{V}.$$
(2.3)

To specify a particular solution of this model, we impose five initial conditions,

$$\hat{x} = \frac{\mathrm{d}\hat{x}}{\mathrm{d}\hat{t}} = \theta = \frac{\mathrm{d}\theta}{\mathrm{d}\hat{t}} = \hat{I} = 0, \quad \text{at} \quad \hat{t} = 0, \tag{2.4}$$

corresponding to the system initially being at rest.

The model described by equations (2.1)-(2.4) is well-posed, and is readily solved numerically using off-the-shelf ODE solvers, such as MATLAB ode45. A numerical solution to (2.1)-(2.4) for the parameter values in Table 2.1 is shown in Figure 2.4.

To investigate the behaviour of this model, we *could* use a numerical approach such as repeated integration of the full coupled second-order system while sweeping through different values of the dimensional parameters or try to use a numerical continuation approach of the equivalent first-order vector field representation of the model. However, we are able to reduce the size of the parameter space by first *nondimensionalising*.

#### 2.2.1 Nondimensionalisation

When faced with a model such as (2.1)–(2.4), there are a number of benefits to rescaling the variables to remove their dimensions:

- We are able to identify dimensionless groups of parameters which control the behaviour of the model, reducing the size of the parameter space.
- It does not make sense to compare quantities with different dimensions; who is to say a kilogram is small compared to a metre when they are fundamentally different quantities? By removing dimensions we are able to compare the size of different terms in a model, which may allow us to identify which processes are dominating the behaviour of the model.
- The size of the dimensionless variables can be chosen so they are not too large or too small, meaning that the dimensionless version of the model is often less susceptible than the dimensional model to stability issues when solving numerically.

The dimensionless model captures exactly the same physics as the dimensional model, and we can always rescale the solutions to the dimensionless model to put them back in dimensional terms.

We will present how to nondimensionalise the model described by equations (2.1)–(2.4) in enough detail to understand how to nondimensionalise other similar models. Note that some of the choices we make along the way are not unique. However, if we had made alternative choices we would still find that that set of dimensionless parameters is similar to those we derive here. The technique of nondimensionalisation is a useful first step in evaluating any physical mathematical model. More examples and discussion of nondimensionalisation are given in Tayler 1986, §1, where it is referred to as normalization, and in Haefner 2005.

To start with, we rescale each of the *dimensional* variables, denoted with hats, with a typical *dimensional* scale, which we denote by square brackets, as so:

$$\hat{x} = [x]x, \quad \hat{I} = [I]I, \quad \hat{t} = [t]t.$$
 (2.5)

The variables x, I and  $\theta$  are dimensionless dependent variables, and t is the dimensionless independent variable. Note that  $\theta$  denotes an angle measured in radians, so it is already dimensionless. At this stage, we are free to choose the dimensional scalings, [x], [I] and [t]. We may have an idea of what sensible scalings could be from experiments or numerical simulations, but for now, we leave them as unknown constants. We will choose them later to simplify the model as much as we can.

Substituting (2.5) into the governing equations<sup>5</sup> (2.1)-(2.3) we find

$$\frac{\hat{m}[x]}{[t]^2}\frac{\mathrm{d}^2 x}{\mathrm{d}t^2} = -\hat{k}_s[x]x - \frac{\hat{d}_s[x]}{[t]}\frac{\mathrm{d}x}{\mathrm{d}t} - \hat{k}_b\left([x]x + \hat{A}\sin(\theta)\right),\tag{2.6}$$

$$\frac{\hat{J}}{[t]^2}\frac{\mathrm{d}^2\theta}{\mathrm{d}t^2} = \hat{K}[I]I - \frac{\hat{d}_m}{[t]}\frac{\mathrm{d}\theta}{\mathrm{d}t} - \hat{k}_b\hat{A}\cos(\theta)\left([x]x + \hat{A}\sin(\theta)\right),\tag{2.7}$$

$$\frac{\hat{L}[I]}{[t]}\frac{\mathrm{d}I}{\mathrm{d}t} = -\frac{\hat{K}}{[t]}\frac{\mathrm{d}\theta}{\mathrm{d}t} - \hat{R}[I]I + \hat{V}.$$
(2.8)

To make these equations dimensionless we divide through each of the equations so one of the terms is dimensionless (has no hats or square brackets). It does not really matter how we choose to do this, but one possibility is

$$\frac{\hat{m}}{\hat{k}_s[t]^2} \frac{\mathrm{d}^2 x}{\mathrm{d}t^2} = -x - \frac{\hat{d}_s}{\hat{k}_s[t]} \frac{\mathrm{d}x}{\mathrm{d}t} - \frac{\hat{k}_b}{\hat{k}_s} \left( x + \frac{\hat{A}}{[x]} \sin(\theta) \right),\tag{2.9}$$

$$\frac{\mathrm{d}^2\theta}{\mathrm{d}t^2} = \frac{\hat{K}[I][t]^2}{\hat{J}}I - \frac{\hat{d}_m[t]}{\hat{J}}\frac{\mathrm{d}\theta}{\mathrm{d}t} - \frac{\hat{k}_b\hat{A}[x][t]^2}{\hat{J}}\cos(\theta)\left(x + \frac{\hat{A}}{[x]}\sin(\theta)\right),\qquad(2.10)$$

$$\frac{\tilde{L}[I]}{\tilde{V}[t]}\frac{\mathrm{d}I}{\mathrm{d}t} = -\frac{\tilde{K}}{\hat{V}[t]}\frac{\mathrm{d}\theta}{\mathrm{d}t} - \frac{\tilde{R}[I]}{\hat{V}}I + 1.$$
(2.11)

Finally, because we are still free to choose [x], [I] and [t], we are able to set three further coefficients in (2.9)–(2.11) to be equal to one, reducing the number of parameters. Again, it does not really matter which terms we choose, but if we have some physical insight into the system this can be helpful. For instance, one could imagine that the amplitude of displacement of the mass might be similar to the radial distance,  $\hat{A}$ , between the centre of the motor and the eccentric pin. In which case, it would be sensible to set  $\hat{A}/[x] = 1$ . We further choose the time-scale to balance the acceleration of the blade with the force due to the spring, and choose the current-scale to balance the voltage and resistance, so that our dimensional scalings are

$$[x] = \hat{A}, \quad [I] = \frac{\hat{V}}{\hat{R}}, \quad [t] = \sqrt{\frac{\hat{m}}{\hat{k}_s}}.$$
 (2.12)

The resulting dimensionless model is

$$\frac{\mathrm{d}^2 x}{\mathrm{d}t^2} = -x - d_s \frac{\mathrm{d}x}{\mathrm{d}t} - \lambda(x + \sin\theta), \qquad (2.13)$$

$$\frac{\mathrm{d}^2\theta}{\mathrm{d}t^2} = \alpha KI - d_m \frac{\mathrm{d}\theta}{\mathrm{d}t} - \lambda \mu \cos\theta (x + \sin\theta), \qquad (2.14)$$

$$L\frac{\mathrm{d}I}{\mathrm{d}t} = -K\frac{\mathrm{d}\theta}{\mathrm{d}t} - I + 1, \qquad (2.15)$$

 $<sup>{}^{5}</sup>$ In general, the initial conditions should be considered as part of the nondimensionalisation process. In this case, since the initial conditions (2.4) are homogeneous, rescaling the variables does not result in any additional dimensionless parameter groups.
$d_s$	$\lambda$	α	K	$\mu$	L	$d_m$
$rac{\hat{d_s}}{\sqrt{\hat{k}_s\hat{m}}}$	$\frac{\hat{k}_b}{\hat{k}_s}$	$\frac{\sqrt{\hat{m}^3}\hat{V}^2}{\sqrt{\hat{k}_s^3}\hat{R}\hat{J}}$	$\frac{\hat{K}\sqrt{\hat{k}_s}}{\hat{V}\sqrt{\hat{m}}}$	$\frac{\hat{m}\hat{A}^2}{\hat{J}}$	$\frac{\hat{L}\sqrt{\hat{k}_s}}{\hat{R}\sqrt{\hat{m}}}$	$\frac{\hat{d}_m \sqrt{\hat{m}}}{\hat{J} \hat{k}_s}$
$1.3 \cdot 10^{-2}$	40	$3.1 \cdot 10^{-2}$	2.1	$2.9\cdot 10^{-2}$	0.38	0

Table 2.2: Definitions of the dimensionless parameters, and typical values based on the values in Table 2.1.

where the dimensionless parameters are defined in terms of the dimensional parameters in Table 2.2. The dimensionless initial conditions are

$$x = \frac{\mathrm{d}x}{\mathrm{d}t} = \theta = \frac{\mathrm{d}\theta}{\mathrm{d}t} = I = 0, \quad \text{at} \quad t = 0.$$
(2.16)

Based on the typical values provided in Table 2.1, the dimensional scalings (2.12) are approximately  $[x] = 1.5 \times 10^{-3} \text{ m}$ , [I] = 1.4 A and  $[t] = 1.9 \times 10^{-3} \text{ s}$ .

# 2.3 Asymptotic analysis

A number of the dimensionless parameters in Table 2.2 are small. There is a suite of mathematical techniques, called *perturbation methods*, used for finding approximate solutions to mathematical models with small parameters. The resulting solutions are *asymptotic approximations* to the solution. We will not precisely define what this means here, rather referring the reader to Hinch 1991, §1–2 and Holmes 2013.

The typical values for  $\lambda^{-1}, d_s, \alpha, \mu$  in Table 2.2 are of a similar size, and all much less than one. We therefore seek a solution valid in the limit  $\lambda \to \infty$  and  $d_s, \alpha, \mu \to 0$ , such that

$$\lambda \psi = O(1) \quad \text{as} \quad \lambda \to \infty,$$
 (2.17)

where  $\psi = d_s, \alpha, \mu$ , hence,  $\lambda^{-1}, d_s, \alpha$  and  $\mu$  are all of the same order. Furthermore, we will assume that  $d_m \equiv 0$  from here on in.

Formally, we seek a solution in the form

$$x \sim x_0 + \alpha x_1 + \cdots, \qquad (2.18a)$$

$$\theta \sim \theta_0 + \alpha \theta_1 + \cdots,$$
 (2.18b)

$$I \sim I_0 + \alpha I_1 + \cdots . \tag{2.18c}$$

The expressions (2.18) are asymptotic series. The key here is that each term is asymptotically smaller than the previous. For example, in (2.18a) we insist that  $\alpha x_1/x_0 \to 0$  as we take the limit (2.17).

Substituting (2.18) into (2.13)-(2.15) and collecting the leading order terms, the

equations are

$$x_0 + \sin(\theta_0) = 0, \tag{2.19}$$

$$\frac{\mathrm{d}^2\theta_0}{\mathrm{d}t^2} = 0, \tag{2.20}$$

$$L\frac{\mathrm{d}I_{0}}{\mathrm{d}t} = -K\frac{\mathrm{d}\theta_{0}}{\mathrm{d}t} - I_{0} + 1.$$
(2.21)

The general solution to these differential equations is

$$x_0 = \sin(c_1 + \omega t), \tag{2.22}$$

$$\theta_0 = c_1 + \omega t, \tag{2.23}$$

$$I_0 = c_2 \exp\left(-\frac{t}{L}\right) + 1 - K\omega, \qquad (2.24)$$

where  $\omega, c_1, c_2$  are constants to be determined. Unfortunately, if we try to apply the initial conditions (2.16), we find that  $\omega, c_1, c_2$  are all identically zero. This tells us that the balances we have chosen by assuming (2.12) are not the dominant balances during the initial motion of the blade.

We did not address this early time behaviour during the study group. Instead, we determined  $\omega$  assuming that the scalings we have chosen represent the longer time behaviour of the system, assuming  $c_1, c_2$  would come from analysing the, as yet unknown, early time behaviour.

To find the frequency,  $\omega$ , we must consider the second order approximation,

$$\frac{\mathrm{d}^2 x_0}{\mathrm{d}t^2} = -x_0 - d_s \frac{\mathrm{d}x_0}{\mathrm{d}t} - \alpha \lambda (x_1 + \theta_1 \cos(\theta_0)), \qquad (2.25)$$

$$\frac{\mathrm{d}^2\theta_1}{\mathrm{d}t^2} = KI_0 - \lambda\mu\cos(\theta_0)(x_1 + \theta_1\cos(\theta_0)), \qquad (2.26)$$

$$L\frac{\mathrm{d}I_1}{\mathrm{d}t} = -K\frac{\mathrm{d}\theta_1}{\mathrm{d}t} - I_1. \tag{2.27}$$

These equations describe the balance between the largest terms in the limit (2.17) which we neglected in deriving (2.19)– $(2.21)^6$ . We can eliminate  $x_1$  from (2.26) using (2.25) to find

$$\frac{\mathrm{d}^2\theta_1}{\mathrm{d}t^2} = KI_0 + \frac{\mu}{\alpha}\cos(\theta_0)\left(\frac{\mathrm{d}^2x_0}{\mathrm{d}t^2} + x_0 + d_s\frac{\mathrm{d}x_0}{\mathrm{d}t}\right).$$
(2.28)

<sup>&</sup>lt;sup>6</sup>It would be consistent with our asymptotic assumptions (2.17) to leave the  $d_s dx_0/dt$  term out of equation (2.25). However, leaving it in does not make the system of equations more difficult to solve, and means our analysis is valid for  $d_s = O(1)$ .

Substituting the leading order solutions (2.22)-(2.24) into (2.28) we find

$$\frac{\mathrm{d}^2\theta_1}{\mathrm{d}t^2} = K\left(c_2 \exp\left(-\frac{t}{L}\right) + 1 - K\omega\right) - \frac{\mu}{\alpha}(1 - \omega^2)\sin\left(c_1 + \omega t\right)\cos\left(c_1 + \omega t\right) - \frac{\mu}{\alpha}d_s\omega\cos^2\left(c_1 + \omega t\right).$$
 (2.29)

Integrating twice, we deduce

$$\theta_{1} = Kc_{2}L^{2}\exp\left(-\frac{t}{L}\right) + t^{2}\left(\frac{K(1-K\omega)}{2} - d_{s}\frac{\mu\omega}{4\alpha}\right) + t\left(c_{3} - \frac{\mu}{4\alpha\omega}\left(1-\omega^{2}\right)\right) + \frac{\mu}{8\alpha\omega^{2}}\left(1-\omega^{2}\right)\sin(2\theta_{0}) + d_{s}\frac{\mu}{8\alpha\omega^{2}}\cos(2\theta_{0}) + c_{4}.$$
 (2.30)

where  $c_3, c_4$  are constants of integration.

For our asymptotic solution (2.18b) to be valid for large t, the perturbation,  $\theta_1$ , can not grow faster than  $\theta_0$ . In particular, this means that the coefficient of the quadratic term must be zero<sup>7</sup>, so  $\omega$  must satisfy

$$\frac{K(1-K\omega)}{2} - d_s \frac{\mu\omega}{4\alpha} = 0.$$
(2.31)

Hence,

$$\omega = \frac{2K}{2K^2 + d_s \mu/\alpha}.\tag{2.32}$$

As  $d_s \mu / \alpha \to 0$ , which is consistent with (2.17), we find that approximately

$$\omega \sim \frac{1}{K}.\tag{2.33}$$

Recall that for certain voltages Philips observed that the amplitude of the oscillatory part of the current drops to close to zero. To understand this phenomenon, we solve for the second order correction,  $I_1$ , to the current. We solve (2.27) using variation of parameters, seeking a solution of the form  $I_1 = C(t) \exp(-t/L)$ . The equation for C(t) is

$$\frac{\mathrm{d}C}{\mathrm{d}t} = -\frac{K}{L} \exp\left(\frac{t}{L}\right) \frac{\mathrm{d}\theta_1}{\mathrm{d}t}.$$
(2.34)

<sup>&</sup>lt;sup>7</sup>This implies that there exist multiple timescales in the problem. There are methods to analyse problems with multiple timescales, see Holmes 2013, §3.

Differentiating (2.30), and integrating (2.34), we find

$$C = c_5 + K^2 c_2 t - K \left( c_3 - \frac{\mu}{4\alpha\omega} \left( 1 - \omega^2 \right) \right) \exp\left(\frac{t}{L}\right) + \frac{K\mu \exp(t/L)}{4\alpha\omega \left( 1 + 4L^2\omega^2 \right)} \left[ \left( d_s - 2\omega L (1 - \omega^2) \right) \sin(2\theta_0) - \left( 1 - \omega^2 + 2\omega L d_s \right) \cos(2\theta_0) \right], \quad (2.35)$$

where  $c_5$  is a constant of integration. Therefore the correction to the current is

$$I_{1} = \exp\left(-\frac{t}{L}\right) \left(c_{5} + K^{2}c_{2}t\right) - K\left(c_{3} - \frac{\mu}{4\alpha\omega}\left(1 - \omega^{2}\right)\right) + \frac{K\mu}{4\alpha\omega\left(1 + 4L^{2}\omega^{2}\right)} \left[\left(d_{s} - 2\omega L(1 - \omega^{2})\right)\sin(2\theta_{0}) - \left(1 - \omega^{2} + 2\omega Ld_{s}\right)\cos(2\theta_{0})\right].$$
 (2.36)

The oscillatory part of  $I_1$  is written in square brackets.

From our analysis, parameter  $d_s$  is  $O(\alpha)$  in the limit (2.17). If also  $1 - \omega^2 = O(\alpha)$ , then the amplitude of the oscillatory part of  $I_1$  in (2.36) is  $O(\alpha)$  as well. Considering the asymptotic expansion of the current,  $I \sim I_0 + \alpha I_1 + \cdots$ , this implies that the oscillatory part of the current is  $O(\alpha^2)$ . Therefore, we find the dip in current observed by Philips occurs when  $\omega$  is approximately one, which, from (2.33), we conclude occurs when K is also approximately one. Looking at the definition of K in Table 2.2, we see that the current drops when

$$\frac{\hat{K}\sqrt{\hat{k}_s}}{\hat{V}\sqrt{\hat{m}}} \approx 1. \tag{2.37}$$

Physically this means that the natural rotational speed of the motor matches the natural frequency of the spring-blade system. If there was no damping,  $d_s = 0$ , then the motor would approach a steady speed where it would not need to do work to keep the blade in motion.

# 2.4 Discussion and recommendations

We investigated a model for the dynamic motion of a trimmer proposed by Philips in order to understand and optimize its performance. The model is based on a classical mass-spring system and can be interpreted as two oscillators that are nonlinearly coupled.



Figure 2.5: Comparison between numeric (blue) and leading-order dimensionless (red) solutions. Panels (a), (b) and (c) show time series of the angle, the displacement of the blade and the current through the motor, respectively.

Our approach is based on nondimensionalisation of the model, which reduces the number of parameters for further analysis. Some of the dimensionless parameters in the nondimensionalised system are small, and therefore, we use the methods of asymptotic analysis and look for asymptotic approximations to solutions of the system. This way we find explicit leading-order representations for the (dimensionless) displacement of the blade, x, the angle of the pin,  $\theta$ , and the current through the motor, I, which show excellent agreement with the numerical simulation of the system described by equations (2.1)–(2.4). A comparison between the numerical solutions (blue) and the (dimensional) asymptotic solutions (red) is displayed in Figure 2.5. Panel (a) shows the long-term concordance of the angle of the pin, while panels (b) and (c) show agreement of the displacement and the current. Despite being slightly out of phase due to the choice of initial conditions, both solutions behave in the same way and within the same order of magnitude.

The trimmer dynamics model studied here provides interesting synchronization properties between two nonlinearly coupled oscillators. Nondimensionalisation together with asymptotic analysis has proven to be a useful tool to simplify the system and obtain a mathematical explanation for the dip in the motor current observed in simulations. Our recommendation is to use this approach in more extended versions of the given model, as it provides an organized way to obtain insights on the behaviour of relevant quantities. Alternatively, one could combine this analytic approach with a systematic numerical investigation of the reduced dimensionless parameter space, using, for instance, state of the art methods based on the numerical continuation of periodic solutions to the system under consideration; see, for instance, Krauskopf, Osinga, and Galán-Vioque (eds) 2007. Though it would be worth exploring that path, we do not pursue that method in this report.

# 2.5 Acknowledgements

The authors would like to thank the organisers of SWI 2023, particularly Stephan Trenn, for all their hard work to make the study group a success. They would also like to thank Daniel Dirksz of Philips for presenting this interesting problem at the study group. J.P.H. acknowledges the support provided by the Engineering and Physical Sciences Research Council (EPSRC) Centre for Doctoral Training in Industrially Focused Mathematical Modelling (EP/L015803/1). J.M. acknowledges support from the EPSRC through the grant EP/T013613/1.

# References

Haefner, J. W. (2005). Modeling Biological Systems. Springer.

Hinch, E. J. (1991). Perturbation Methods. Cambridge University Press.

Holmes, H. H. (2013). Introduction to Perturbation Methods. Springer.

- Krauskopf, B., H. M. Osinga, and J. Galán-Vioque (eds) (2007). Numerical Continuation Methods for Dynamical Systems: Path following and boundary value problems. Springer Netherlands.
- Tayler, A. B. (1986). Mathematical Models in Applied Mathematics. Oxford University Press.

# Chapter 3

# Modelling the heat transition

Muhammad Ahsan, Leander van der Bijl, Shreehari Bodas, Fabio Buccoliero, Yanfei Chen, Joaquim M.C. Correia, Tsega Kahsay Gebretekle, Leila Hashemi, Joris van der Hijden, Francesca Leonardi, Gabriel Lord, Juan E. Machado Martinez, Zhirui Tang, Leoni Winschermann, Jialing Yu

**Abstract** The Dutch government is committed to reducing its dependence on fuel imports and promoting sustainable practices, which has led to plans to move away from gas-powered heating systems towards alternative heat sources such as district heating or electric heat pumps. In this heat transition, DHM provides advice and leadership for investment decisions, particularly at the municipal level. Key considerations include the transition costs, choice of heat commodity, and necessary infrastructure. This project explores three decision-making models that aim to minimize social costs in an objective manner: Monte Carlo simulations, energy hubs, and mathematical programs. The results of this study can be used as a starting point for further research to support a sustainable and cost-effective heat transition.

KEYWORDS: heat transition, cost minimization, Monte Carlo methods, energy hubs, mixed integer program

# 3.1 Problem statement - The heat transition

DHM advises and leads processes and projects in mobility, infrastructure, water and energy. Sustainability, climate, smart cities and digital innovation are their core business.

Municipalities in the Netherlands are responsible of transitioning to an energy system that does not rely on fossil resources. One of the goals is to transition from a heating system that is highly reliant on natural gas to one that is using sustainable resources for the heating of homes and buildings. This process is called the *heat transition*. Due to resource scarcity and limited technical solutions, solutions have been limited to 3 alternatives (see SWI 2023):

- District heating: in highly concentrated areas with excess heating from for example industrial processes;
- Electrical heating systems: for individual homes and buildings with higher isolation levels and more recent building years (1990 and younger);
- Hybrid solutions: in rural areas with older buildings and buildings in areas with no access to district heating and can not be technically isolated to levels needed for electrical systems.

Policymakers have difficulties choosing where to use which alternative and individuals struggle with this insecurity. In this project, we explore models that can help decision makers (municipalities, government officials, energy companies) and potentially individuals to strategically allocate scarce resources and technical solutions based on price and technical availability.

The challenge posed was

- to derive a mathematical model for the current and potential future heating configurations of (a part of) a municipality which is able to describe the relevant decision variables (heat production, energy consumption, investment costs, ... );
- to verify the model with data from the municipality of Rotterdam; and
- to formulate suitable optimization problems based on the derived models.

The models could serve as a tool to strategically allocate financial resources in heat transition processes and to make financial investment plans for the Rotterdam municipality, specifically in relation to the sustainable development goals for 2030 and further.

In the following, we present three models, each taking another perspective on the challenge. In Section 3.2, a general framework for resource allocation is introduced that can help a governmental unit decide on how to distribute resources. The model introduced in Section 3.3 considers energy hubs modelling electricity and heat flows and their interrelation, offering a framework for multi-energy systems to provide decision support for e.g., infrastructure capacity scaling. Lastly, Section 3.4 discusses a local cost minimization model that, given various heat sources, decides what fraction of the heat demand in a district should transition to which heat source, and assigns fractions of the demand to specific facilities for district heating.

# 3.2 Graph Theory and the Monte Carlo Method

In this section, we propose a theoretical framework that utilizes graph theory and the Monte Carlo (MC) method to address the heat transition problem. Although no code has been implemented yet, the framework offers several advantages, which are discussed in Section 3.2.3.

For new strategies, policymakers need to decide how to distribute resources, or which strategy should be used in which cases, in other words, the heat configuration which is the foundation for decision variables like heat production, energy consumption, investment costs, and so on. The purpose of the model mentioned is to offer methods that can assist policy makers in constructing these heat configurations.

We first build a graph model based on the geometrical information and then use the MC method to calculate the relevant decision variables (heat production, energy consumption, investment costs, etc.). It is noted that the graph model perhaps involves the optimization problem. This is because graph theory provides helpful tools and algorithms for optimization. These techniques allow us to find the best solutions for decision-making, optimizing heat production, energy consumption, investment costs, and other important factors. This improves decision-making and helps us identify the optimal heat configurations that policymakers need. An illustrative example is provided in Section 3.2.2.

## 3.2.1 Graph

Let us start by representing our objects of interest with points. These points can represent various entities such as provinces, districts, towns, and more. We make this representation because policy makers operate at different levels, ranging from national or regional bodies to city councils. This abstraction enables us to use this framework to be applied in more general settings, as we will discuss in Section 3.2.3. To illustrate, let us take the city of Groningen as an example.



Figure 3.1: A district of Groningen (Source: Planbureau voor de Leefomgeving)

Figure 3.1 displays the administrative map of Groningen. Each region, such as the one highlighted in red, can be considered as a point on a graph. We connect two regions with an edge in the graph if they are adjacent on the map. This approach allows us to create a graph representation for Groningen. Figure 3.2 provides a sketch illustrating the concept of this graph. Please note that it is not the actual graph representation of Groningen.



Figure 3.2: A Geometrical Topology Graph

With this graph, we can for example define a new heat transition strategy. At the same time, we need infrastructure, like pipelines, to transport this energy. The topological graph can help us choose the scheme of pipeline laying. Assume that the new energy comes from one source and that it must reach every point we have since we should guarantee each region gets energy. We can formulate the problem as finding a minimum spanning tree within this graph (similar to the example shown in Figure 3.3).



Figure 3.3: The minimum spanning tree

Naturally, the construction of pipelines depends on different factors, e.g., the length, feasibility, and expected construction time. This should be considered by policymakers. Mathematically, these constrained conditions can be represented by the weights of the edges. Thus, we end up with a weighted graph, or a directed weighted graph. Note that the network can be not only built by the geometrical relation of cities but also by other topological relations.

The heat transition problem mainly considers three alternatives (see Section 3.1): district heating, electrical heating, and hybrid solutions. In the graph model, we could label the classification of each point with these three alternatives according to reality, such as in Figure 3.4.



Figure 3.4: Labeled classification of the network.

So far, we have introduced a framework to derive a graph modeling the topological structure of a region. In the next section, we discuss how to use the graph to calculate quantities of interest, e.g., heat demand, energy consumption, and investment costs. In the following, we assume the graph to be given.

## 3.2.2 Monte Carlo Simulation

The Monte Carlo (MC) method refers to a method involving random sampling processes. More frequently, it refers to a stochastic method used to calculate the expectation:

$$E[f(x)] = \int_{\Omega} f(x)q(x)dx, \qquad (3.1)$$

in which  $\Omega$  is the domain,  $x \in \Omega$  is a random variable following the distribution that is specified by the density function q(x), and f(x) is the physical quantity of interest. According to the law of large numbers, the simulated expectation will converge to the real mean as the sample size goes to infinity. Its convergence rate is  $\mathcal{O}(N^{-1/2})$ , which is slower than many other methods. The MC method is powerful in various scenarios, including:

- (1) High-dimensional problems, where other methods may become computationally expensive or impractical.
- (2) Problems involving stochastic variables, where the MC method can handle the uncertainty and randomness effectively.
- (3) Situations where no other practical options or alternatives are available.

#### Application to Heat Transition Problem

Previously, we assumed that there is a reasonable network in place. However, policymakers are interested in more than just building the network. In particular, they need to know about the current heating setups and possible future configurations to make important decisions. We observe that these quantities are a summation of each individual's behavior (in our case, represented as points) and specific district conditions with stochastic factors. We cannot control individual behavior, both from a computational point of view and from the policymakers' point of view. Under these circumstances, the MC method is a suitable approach.

Let us consider electricity consumption as an example. From historical data, we can get statistical information on each point. For instance, the electricity consumption of a house should follow a normal distribution. That means, based on analysis of historical data, the distribution of electricity consumption in residential houses tends to exhibit a normal distribution. This implies that the majority of houses will have average consumption levels, with fewer houses falling into higher or lower consumption extremes. Similarly, in the context of 24-hour factories, electric energy consumption often follows a uniform distribution.

Furthermore, the consumption of electricity by individual households depends on price, security, and climate conditions, and it is random. Their mean and variance can be obtained via statistical methods. If the time variable is considered, we end up with a stochastic process, e.g., the Gaussian process for household electricity consumption. Afterward, the random samples can be generated by these stochastic processes. With random samplings, any relevant quantities can be calculated. Now, let us express the statement above as mathematical formulas.

Suppose  $h = \{h_1, h_2, ..., h_H\}$  are the points of the highly concentrated district,  $n = \{n_1, n_2, ..., n_N\}$  be those of new buildings district, and  $r = \{r_1, r_2, ..., r_R\}$ , be isolated districts. Then, the expectation of the quantity of interest for the whole region can be expressed as

$$E[f(x)] = \int_{\Omega} f(x)q(x)dx, \qquad (3.2)$$

$$= \int_{\Omega_h \cup \Omega_n \cup \Omega_r} f(x)q(x)dx, \qquad (3.3)$$

where x is the random variable of electricity consumption. The whole region  $\Omega$  is the non-overlapping union of the highly concentrated region  $\Omega_h$ , new buildings region  $\Omega_n$ , and the isolated region  $\Omega_r$ .

According to the principle of the MC method, we have

$$E[f(x)] = \frac{1}{M} \lim_{M \to \infty} \left( \sum_{i}^{H} \sum_{h_{ij}}^{M} f(x_{h_{ij}}) + \sum_{i}^{N} \sum_{n_{ij}}^{M} f(x_{n_{ij}}) + \sum_{i}^{R} \sum_{r_{ij}}^{M} f(x_{r_{ij}}) \right)$$

where  $x_{h_{ij}}$  represents the *j*-th random sample of the *i*-th point in the highly concentrated region. Similarly,  $x_{n_{ij}}$  is the *j*-th random sample of the *i*-th point in the region of the new building, and  $x_{r_{ij}}$  is the *j*-th random sample of the *i*-th point in the isolated region.

For simplicity, here we assume that M random samples are generated for each point in our graph. However, the number of random samples could be different per point.

If f(x) = x, we are calculating the average of electricity consumption for the electrical heating system in, for example, the new building district.

If  $f(x) = \alpha(s, t)x$ , where  $\alpha$  is the electricity price depending on space s and time t, we calculate the average cost for the electrical heating system. Even though the cost, in reality, is not only the electricity price, for clarity, we only consider the price of electricity.

It is worth mentioning that if new energy is used, its consumption should follow the same statistical behavior as the old one since electricity habits will not change. With this understanding, the model can be enhanced by incorporating prediction methods to estimate future quantities.

#### 3.2.3 Advantages of The Framework

In this section, we outline the benefits of the considered framework. It provides a versatile problem-solving approach.

#### 1. Reuse of model and algorithm:

The same model and algorithm can be used for other problems. For instance, if Figure 3.2 is the graph for the Netherlands, each point is a city. Suppose that one of the points represents Groningen. If we zoom in at this point, we see that the case for Groningen is again a graph with a topology. Hence, the same algorithm can be applied to Groningen.

#### 2. Flexibility in variable consideration:

The heat transition problem requires investigation of potential future heat configurations, and the considered variable should be changeable. With this framework, we can change the way of constructing graphs (maybe a new optimization problem) and the integral of Monte Carlo. In many cases, we just need to change samples x and the concerned quantity f(x). By doing this, the framework can adapt to different scenarios. It seems like we still alter many things, but in terms of programming, it is much easier than a completely new model.

## 3. Suitability for complicated problems:

In conclusion, the Monte Carlo method is a good choice for complicated problems. For instance, when we consider each individual building in a region (each point is a building), resulting in millions of data points. Additionally, x may be a vector with several components, which can be handled by the Monte Carlo method more effectively than deterministic methods.

# 3.3 Framework of Energy Hubs

In this section, we present a framework for the optimal design of multi-energy systems, which can assist in the elaboration of a plan for a sustainable heat transition in local and regional energy systems. The framework is known in the literature as *energy hubs*; see Cao et al. 2020; Geidl, Koeppel, et al. 2007; Geidl and Andersson 2007. Most of the contents of this section are based on Cao et al. 2020, but they are slightly tailored to meet the specifications by DHM.

## 3.3.1 Introduction to Energy Hubs

An energy hub is an item which exchanges heat and/or power with the three main energy networks, namely, the power, heat and gas grids. Each energy hub may contain up to four elementary devices: a heat pump (HP), a heat exchanger (HX), a combined heat and power unit (CHP) and a furnace (F). These devices will be described by the efficiency at which they transform energy. In particular, heat pumps transform power in heat, furnaces transform gas in heat and combined heat and power plants consume gas to generate power and heat. See the schematic diagram in Figure 3.5.

Due to time constraints, we bring the following standing assumption:

## Assumption 1.

- (i) Heat exchangers will not be considered. They can be assumed to be of high efficiency or to be of a much lower cost than a CHP or a HP.
- (ii) We will not take into consideration the gas grid. Therefore, energy hubs will only be able to exchange energy through the electric distribution grid and/or district heating network.
- (iii) The overall multi-energy systems is viewed as the networked interconnection of several single energy hubs.

# 3.3.2 Mathematical Modeling

## A Single Energy Hub

In this part we introduce the basic model to describe the interaction between the devices that are part of a same energy hub. Based on Figure 3.5 and using Kirchhoff's



Figure 3.5: Schematic diagram of an energy hub based on Cao et al. 2020. It is composed of four basic devices, namely, a heat pump (HP), a heat exchanger (HX), a combined heat and power unit (CHP) and a gas furnace (F). Each of these devices are described by input-output relationships.

laws for power and heat balance, we get the following relationships:

$$p_i^{\rm in} + p_{{\rm chp},i}^{\rm out} - p_{{\rm hp},i}^{\rm in} = p_i^{\rm out} \quad (\text{power})$$
(3.4a)

$$h_i^{\text{in}} + h_{\text{chp},i}^{\text{out}} + h_{\text{hp},i}^{\text{out}} + h_{\text{f},i}^{\text{out}} = h_i^{\text{out}} \quad \text{(heat)}$$
(3.4b)

where we use the variables p and h to denote electric power and heat, respectively. For both p and h, the subscripts identify the devices and the superscripts if whether it represents an inlet our outlet with respect to a specific device.

Next, we identify the input-output relationships for each of the devices in a given energy hub. Such relationships are essentially the efficiencies at which the devices transform energy, potentially from one domain to another. See:

$$m_{\mathrm{chp},i}^{\mathrm{in}} = \frac{p_{\mathrm{chp},i}^{\mathrm{out}}}{\eta_{\mathrm{chp},i}^{\mathrm{p}}} + \frac{h_{\mathrm{chp},i}^{\mathrm{out}}}{\eta_{\mathrm{chp},i}^{\mathrm{h}}} \quad (\mathrm{CHP})$$
(3.5a)

$$m_{\mathrm{f},i}^{\mathrm{in}} = \frac{h_{\mathrm{f},i}^{\mathrm{out}}}{\eta_{\mathrm{f},i}^{\mathrm{h}}} \quad (\mathrm{F})$$
(3.5b)

$$h_{\mathrm{hp},i}^{\mathrm{out}} = \mathrm{COP}_i p_{\mathrm{hp},i}^{\mathrm{in}} \quad (\mathrm{HP}).$$
(3.5c)

Note that for a CHP unit, there are two efficiencies.  $\eta^{\rm p}_{{\rm chp},i}$  represents the efficiency at which electric power is produced, and  $\eta^{\rm h}_{{\rm chp},i}$  represents the efficiency at which heat is produced. For heat pumps, we use the standard coefficient of performance (COP) to describe the rate at which heat is produced from a given input of electric power.

The devices at a given energy hub are subject to capacity constraints. These can be written as follows:

$$\begin{cases}
\frac{p_{chp,i}^{\text{odd}}}{\eta_{chp,i}^{p}} + \frac{h_{chp,i}^{\text{odd}}}{\eta_{chp,i}^{h}} &\leq C_{chp,i} \\
R_{chp,i}^{p}C_{chp,i} &\leq \frac{p_{chp,i}^{\text{out}}}{\eta_{chp,i}^{p}} \\
\frac{h_{chp,i}^{\text{out}}}{\eta_{chp,i}^{h}} &\leq R_{chp,i}^{h}C_{chp,i}
\end{cases} \quad (CHP) \quad (3.6a)$$

$$h_{\mathrm{f},i}^{\mathrm{out}} \le C_{\mathrm{f},i} \quad (\mathrm{F}) \tag{3.6b}$$

$$h_{\mathrm{hp},i}^{\mathrm{out}} \le C_{\mathrm{hp},i}$$
 (HP), (3.6c)

where  $C_{\text{chp},i}$  represents the capacity, and  $R^{\text{p}}_{\text{chp},i}$  and  $R^{\text{h}}_{\text{chp},i}$  are constants Cao et al. 2020. For the CHP, the above inequalities represent the operation region of the CHP and include the maximum fuel import rate, the minimum electric output and the maximum thermal output.

The final part of the model to describe a single energy hub corresponds to a number of non-negativity constraints as well as to some upper bounds on the capacities of the devices. These are the following:

$$C_{\mathrm{chp},i}, C_{\mathrm{f},i}, C_{\mathrm{hp},i} \ge 0, \forall i$$

$$(3.7a)$$

$$(C_{\mathrm{chp},i}, C_{\mathrm{f},i}, C_{\mathrm{hp},i}) \preceq (C_{\mathrm{chp},i}^{\max}, C_{\mathrm{f},i}^{\max}, C_{\mathrm{hp},i}^{\max}), \forall i$$
 (3.7b)

$$m_{\mathrm{chp},i}^{\mathrm{in}}, \ m_{\mathrm{f},i}^{\mathrm{in}} \ge 0, \ \forall i$$
 (3.7c)

$$h_{\mathrm{hp},i}^{\mathrm{out}}, h_{\mathrm{chp},i}^{\mathrm{out}} \ge 0, \ \forall i.$$
 (3.7d)

**Remark 1.** The energy hub model described in this section can be extended to include electric or heat storage devices. See, for example, Cao et al. 2020. Storage tanks are omitted in this report for simplicity.

### Power Grid

In this subsection we provide a model of the power grid. Since we are interested in describing mostly local energy systems—at municipality levels—the power grid we consider is restricted to the distribution level of the electric power transport system.

The power grid is modeled as a graph  $G_{\rm p} = (\mathcal{N}_{\rm p}, \mathcal{E}_{\rm p})$ , where its nodes  $\mathcal{N}_{\rm p}$  are the set of power buses, its edges  $\mathcal{E}_{\rm p}$  are the set of power lines. Moreover, throughout the remaining of the document we will make reference to the following subset:

$$S_i = \text{ subset of nodes adjacent to the } i\text{th node}$$
$$= \{ j \in \mathcal{N}_{p} : (i, j) \in \mathcal{E}_{p} \text{ or } (j, i) \in \mathcal{E}_{p} \}.$$
(3.8)

With the above considerations, we write the power grid model as follows Geng,

Vrakopoulou, and Hiskens 2020:

$$p_i^{\text{out}} = \sum_{i \in S_i} B_{i,k}(\theta_i - \theta_k), \quad i \in \mathcal{N}_p$$
(3.9a)

$$B_{i,k} \ge 0, \quad (i,k) \in \mathcal{N}_{\mathbf{p}} \times S_i$$

$$(3.9b)$$

$$|\theta_i - \theta_j| \le \delta_i^{\max}, \quad i, j \in \mathcal{N}_{\mathrm{p}}.$$
 (3.9c)

$$-C_{\mathbf{p},i} \le B_i(\theta_j - \theta_k) \le C_{\mathbf{p},i}, \ \forall (j,k) = i \in \mathcal{E}_{\mathbf{p}}.$$
(3.9d)

Equation (3.9a) represents the power balance at node  $i \in \mathcal{N}_{p}$ : in the left-hand side, we have the electric power the *i*-th energy hub injects into the electric power distribution network; on the right-hand side, the sum of the power that flows through the power lines that are incident to the *i*-th energy hub appears. The power that flows through a given power line is (approximately) proportional to the difference in the voltage phase angles, denoted as  $\theta$ , between the source and target power buses of the line (with a tolerance  $\delta_i^{\max} > 0$ ). The constant of proportionality is the power line *admittance* B. Inequality (3.9b) is simply a nonnegativity constraint for the admittance of all power lines. Inequalities (3.9c) restricts the the domain of feasible values of the voltage phase angles at the power buses of the electric grid. Such a constraint stems from stability considerations. Lastly, inequality (3.9d) limits through the lines according to their respective capacity.

#### Remark 2.

(i) If we assume that a DC power distribution network is in place, instead of a conventional AC network, the power flow equations will be the following:

$$p_i^{\text{out}} = \sum_{j \in S_i} \frac{1}{r_{ij}} (v_i - v_j)^2,$$

where  $v_i$  denotes the DC voltage of a given power bus, and  $r_{i,j}$  the resistance of the power line  $(i, j) \in \mathcal{E}_p$ .

(ii) One way to simplify the model (3.9) consists in introducing the variable  $\phi_{p,j}$  for each  $j \in \mathcal{E}_p$  representing the actual power that flows through the line j. Then, (3.9a) would become

$$p_i^{\text{out}} = \sum_{j \in \mathcal{I}_i} \phi_{\mathbf{p},j}, \quad i \in \mathcal{N}_{\mathbf{p}}$$
(3.10)

where  $\mathcal{I}_i$  denotes the set of power lines that are incident to the *i*-th energy hub. Moreover, the constraints on the capacities of the power lines would now be given by

$$-C_{\mathbf{p},i} \le \phi_{\mathbf{p},i} \le C_{\mathbf{p},i}, \quad i \in \mathcal{E}_{\mathbf{p}}.$$
(3.11)

The above simplified model can be used to do a preliminary analysis on how electric power should be flowing in the network, without considering the power losses. This could also simplify the process of topology optimization. For the sake of simplicity, we use this model in our simulations.

Assumption 2. Our model can further be simplified if we assume that the topology, but not the capacity, of the heat and power grids has already been defined. By topology we mean the location of the energy hubs, the existence of district heating pipelines and the length of such lines.

#### Heat Grid

This section describes a model for the heat grid (or district heating). As analogously done for the power grid, we describe the heat grid as a graph  $G_{dh} = (\mathcal{N}_{dh}, \mathcal{E}_{dh})$ , where the set of nodes  $\mathcal{N}_{dh}$  represents hydraulic junctions (connection points between two or more pipes) and  $\mathcal{E}_{dh}$  is the set of pipes and heat exchangers. The interface between any energy hub and the heat grid is done through a heat exchanger; however, we assume their cost is low and their efficiency high compared to the other devices at any given energy hub.

Before presenting the model, we introduce the following sets:

$$\mathcal{E}_{dh} = \mathcal{E}_{\pi} \cup \mathcal{E}_{hx}$$
: set of pipes and heat exchangers, (3.12a)

- $S_i^{\rm s}$ : set of edges starting at  $i \in \mathcal{N}_{\rm dh}$ , (3.12b)
- $S_i^{e}$ : set of edges ending at  $i \in \mathcal{N}_{dh}$ . (3.12c)

Then, following Cao et al. 2020, we model the heat grid as follows:

$$h_i^{\text{out}} = c_{\text{p}} \dot{m}_i \left( \tau_i^{\text{out}} - \tau_i^{\text{in}} \right), \ i \in \mathcal{E}_{\text{hx}}$$
(3.13a)

$$\tau_i^{\text{out}} = \left(\tau_i^{\text{in}} - T_{\text{a}}\right) e^{-\frac{\lambda_i L_i}{c_{\text{p}} \dot{m}_i}} + T_{\text{a}}, \ i \in \mathcal{E}_{\pi}$$
(3.13b)

$$\tau_k = \frac{\sum_{i \in S_k^{\mathrm{s}}} m_i \tau_i^{\mathrm{scar}}}{\sum_{j \in S_k^{\mathrm{s}}} \dot{m}_j}, \ k \in \mathcal{N}_{\mathrm{dh}}$$
(3.13c)

$$\tau_i^{\rm in} = \tau_k, \ \forall i \in S_k^{\rm s}, \forall k \in \mathcal{N}_{\rm dh}, \tag{3.13d}$$

$$0 = \mathcal{B}\dot{m}.\tag{3.13e}$$

Equation (3.13a) represents the heat balance at the heat exchanger of the *i*-th energy hub: in this equation,  $\dot{m}_i$  represents the mass flow rate through the heat exchanger;  $c_p$  is the heat capacity, which is assumed to be constant, and  $\tau_i^{\text{in}}$  and  $\tau_i^{\text{out}}$  represent the temperature at the inlet and outlet of the heat exchanger respectively. Equation (3.13b) describes the temperature  $\tau_i^{\text{out}}$  at the outlet of a given pipe of the heat grid. Note that this temperature depends on the pipe's inlet temperature  $\tau_i^{\text{in}}$ , the ambient temperature  $T_a$ , the pipe's length  $L_i$  and the heat transfer coefficient  $\lambda_i$ . Equation (3.13c) models the temperature of any stream of water coming out of a given node  $k \in \mathcal{N}_{dh}$ , which represents the weighted average of the temperatures of the streams entering the node. Equation (3.13d) models the temperature at the inlet of any pipe or heat exchanger. Equation (3.13e) represents the mass balance equation at each node of the heat grid. Here,  $\mathcal{B}$  represents the incidence matrix of the heat grid's graph. The above model is complemented with the following inequalities, which respectively restrict the values of the mass flow rates, temperatures and heat transfer coefficients to physically sensible subsets:

$$\dot{m}_i \le |m_i^{\max}|, \quad i \in \mathcal{E}_{dh}$$
(3.14a)

$$0 < \tau_k^{\min} \le \tau_k \le \tau_k^{\max}, \quad k \in \mathcal{N}_{dh}$$
(3.14b)

$$0 \le \lambda_i, \quad i \in \mathcal{E}_{\pi}. \tag{3.14c}$$

$$c_{\rm p}\dot{m}_i\tau_i^{\rm in} \le C_{{\rm dh},i}, \ i \in \mathcal{E}_{{\rm dh}}.$$
 (3.14d)

Note that inequality (3.14d) limits the heat that can flow through the pipes in accordance to the respective pipe capacity.

**Remark 3** (Linearization pipe model). We can substitute the pipe model described above by Equation (3.13b) with the following Li et al. 2018:

$$c_{\mathrm{p}}\dot{m}_{i}\left(\tau_{i}^{\mathrm{out}}-\tau_{i}^{\mathrm{in}}\right)=\lambda_{i}L_{i}(T_{\mathrm{a}}-\tau_{i}^{\mathrm{in}}),\;\forall i\in\mathcal{E}_{\pi},$$

where we linearized the exponential using its first order expansion.

**Remark 4.** Analogous to the simplified model (3.10) to describe the electric power system, we introduce for each  $i \in \mathcal{E}_{dh}$  the variable  $\phi_{dh,i}$  to represent the heat flow the goes from a given energy hub to another. Further simplifications come from only considering the supply (warm) layer of the district heating network and taking the energy hubs directly as the set nodes  $\mathcal{N}_{dh}$  and not as edges (HXs). Therefore, a simplified heat flow model can be written as follows:

$$h_i^{\text{out}} = \sum_{j \in \mathcal{I}_i} \phi_{\text{dh},j}, \quad i \in \mathcal{N}_{\text{dh}}.$$
(3.15)

Nonetheless, a significant drawback of this simplified model is that it cannot readily incorporate heat losses. This will lead to imprecise computations of the operational costs. Then, we can impose the following constraint to determine the appropriate capacity of a given pipeline:

$$-C_{\mathrm{dh},i} \le \phi_{\mathrm{dh},i} \le C_{\mathrm{dh},i}, \ i \in \mathcal{E}_{\mathrm{dh}}.$$
(3.16)

Due to time constraints, we use the above model in our numerical simulations.

#### Infrastructure and operational costs

Having defined the model of each energy hub and of the electric power and heat grids, we are in position to introduce the total infrastructure costs and the total operational costs as follows:

total cost = total installation costs + total operation costs  
= 
$$f_{\rm C} + f_{\rm O}$$
,

where

$$\begin{split} f_{\rm C} &= I_{\rm chp} \sum_{\forall i} C_{\rm chp,i} + I_f \sum_{\forall i} C_{\rm f,i} + I_{\rm hp} \sum_{\forall i} C_{\rm hp,i} \\ &+ I_p \sum_{j \in \mathcal{E}_{\rm p}} L_j C_{{\rm p},j} + I_{\rm dh} \sum_{k \in \mathcal{E}_{\rm dh}} L_k C_{\pi,k} \\ f_{\rm O} &= \sum_{y=1}^{\rm year \ 30} 365 \Biggl( \omega_{\rm chp} \left( y \right) \sum_{h=1}^{\rm hour \ 24} \sum_{\forall i} m_{\rm chp,i}^{\rm in} \left( h, y \right) \\ &+ \omega_f(y) \sum_{h=1}^{\rm hour \ 24} \sum_{\forall i} m_{f,i}^{\rm in} (h, y) \\ &- \omega_{\rm p}^{\rm out}(y) \sum_{h=1}^{\rm hour \ 24} \sum_{\forall i} p_i^{\rm out} - \omega_{\rm h}^{\rm out}(y) \sum_{h=1}^{\rm 24} \sum_{\forall i} h_i^{\rm out} \Biggr) \end{split}$$

For defining  $f_{\rm C}$ , we have assumed that the installation costs of CHP, HP and Furnace units are directly proportional to their capacities. Also, in  $f_{\rm C}$  we have the installation costs per kilometer associated to installing power lines or pipelines for the heat distribution network. The proportionality constants  $I_{\rm chp}$ ,  $I_{\rm f}$  and  $I_{\rm hp}$  should be expressed in Eur/MW, whereas  $I_{\rm p}$  and  $I_{\rm dh}$  in Eur/MW·km. Note also, that for the power lines and the district heating pipes, we assume that the infrastructure costs are proportional to the sum of the capacities  $C_{{\rm p},j}$  and  $C_{{\rm dh},j}$  of the power lines in the electric grid and the pipes in the district heating network, respectively.

For the operational costs, we assume out of simplicity that the energy costs are constant through each year, yet they can vary from year to year. Also, the daily profile for mass consumption (gas or biomass) or the heat or power injection/extraction to/from the grids is repeated 365 days. For a more detailed model, we can incorporate typical days of different seasons throughout the year (spring, summer, autumn, winter) and repeat the profile of those typical days to whole seasons. Linear combinations of all the typical days are also possible (see Cao et al. 2020).

**Remark 5.** For simplicity's sake, we will not aim at minimizing the operational costs of the hydraulic layer of the district heating system. This can be incorporated at a later stage straightforwardly, but at the expense of introducing strong nonlinearities into the optimization problem.

**Remark 6.** The considered modeling framework is suitable to also include terminal constraints that would restrict, for example, the amount of  $CO_2$  emissions per year, decade, etc. Also, we can impose a monotonicity constraint ensuring that  $CO_2$ emissions should be reduced by a certain percentage each year.

## 3.3.3 An optimal capacity design problem (OCDP)

Let us first define the following variables:

$$\begin{aligned} x &= \operatorname{stack} \left( C_{\operatorname{chp},i}, C_{\mathrm{f},i}, C_{\mathrm{hp},i}, \lambda_{j}, \kappa_{k}, \right)_{\forall i,j,k} \\ y &= \operatorname{stack} \left( m_{\mathrm{f},i}, m_{\mathrm{chp},i}, p_{i}^{\operatorname{out}}, h_{i}^{\operatorname{out}} \right)_{\forall i} \\ z &= \operatorname{stack} \left( p_{\mathrm{hp},i}^{\operatorname{in}}, p_{\mathrm{chp},i}^{\operatorname{out}}, h_{\mathrm{hp},i}^{\operatorname{out}}, h_{\mathrm{chp},i}^{\operatorname{out}}, h_{\mathrm{f},i}^{\operatorname{out}}, v_{i}, \dot{m}_{i}, \tau_{i}^{\operatorname{out}}, \tau_{i}^{\operatorname{in}}, \tau_{i} \right) \\ \xi &= \operatorname{stack} \left( p_{i}^{\operatorname{in}}, h_{i}^{\operatorname{in}} \right)_{\forall i} \\ d &= \left( \omega_{\mathrm{chp}}, \omega_{\mathrm{f}}, \omega_{\mathrm{p}}^{\operatorname{out}}, \omega_{\mathrm{h}}^{\operatorname{out}} \right) \\ q &= \left( \operatorname{COP}_{i}, \eta_{\mathrm{chp}}^{\mathrm{p}}, \eta_{\mathrm{chp}}^{\mathrm{h}}, \eta_{\mathrm{f}}^{\mathrm{h}}, c_{\mathrm{p},i}, T_{\mathrm{a},\mathrm{h}}, L_{i}, R_{\mathrm{chp}}^{\mathrm{p}}, R_{\mathrm{chp}}^{\mathrm{h}}, \rho_{0} \right) \\ c &= \left( I_{\mathrm{chp}}, I_{\mathrm{f}}, I_{\mathrm{hp}}, I_{\pi}, I_{\mathrm{p}} \right). \end{aligned}$$

Then, the optimization problem of interest is the following:

$$\min_{(x,y)\in\mathcal{K}} f_{\rm C} + f_{\rm O} = c^{\top} x + d^{\top} y, \qquad (3.17)$$

where

 $\mathcal{K} = \{(x, y) : \text{ all defined (in)equality constraints are met}\}.$ 

### 3.3.4 Case study: municipality of Rotterdam

In this section we describe, perhaps in a simplified manner, how to numerically solve problem (3.17) using MATLAB and Yalmip Lofberg 2004 considering a specific case study, namely, the heat transition in Rotterdam, the Netherlands.

**Remark 7.** The data used in this section is mostly artificially generated and solely to illustrate the use of a framework of energy hubs and our MATLAB code to identify suitable designs for the energy systems of the municipality of Rotterdam. More accurate and realistic data can be incorporated straightforwardly to get more realistic designs from our code.

Taking a macroscopic approach, in Figure 3.6.(a) we identify the neighborhoods of the municipality of Rotterdam as nodes of a simple graph  $\mathcal{G}$ . Each node represents an independent energy hub and the set of all nodes is denoted by  $\mathcal{N}$ . The edges  $\mathcal{E}$  of this graph correspond to the distances between any pair of nodes (distances between the centroids of any two neighborhoods). The graph  $\mathcal{G}$  sets the basis for optimally identifying the topologies (and capacities) of the electric power and heat grids. Indeed, as a starting point we will assume that the graph  $\mathcal{G}_p$  describing the power grid and the graph  $\mathcal{G}_{dh}$  describing the heat grid are topologically equivalent to  $\mathcal{G}$ . Then, the solution to problem (3.17) will specify which edges should be removed from  $\mathcal{G}_p$  and  $\mathcal{G}_{dh}$  so that the overall installation and operation costs are minimized. Potential topologies for the power and heat grids are shown in Figure 3.6.(b) and (c).

To solve problem (3.17), it is necessary to specify the numerical values of a number of external parameters. These parameters include the power and heat demand profile



Figure 3.6: (a) The municipality of Rotterdam is represented by a graph. Districts with abbreviations: Overschie (Ov), Hillegersberg-Schiebroek (Hi), Prins-Alexander (Pr), Noord (No), Kralingen-Crooswuk (Kr), Delfshaven (De), Centrum (Ce), Feijenoord (Fe), Waalhaven (Wa), Charlois (Ch), Ijsselmonde (Ij). The nodes are the neighborhoods and the edges are the distances between any two of them. (b) Possible configuration of the power grid. (c) Possible configuration of the heat grid.



Figure 3.7: Block diagram illustrating how our MATLAB code processes input data to return a solution to problem (3.17).

of each neighborhood. Other parameters that should be specified beforehand are the efficiencies of CHP units, furnaces and heat pumps. In addition, in the model some decision variables must comply with constraints in the form of lower or upper bounds. Such bounds should be specified beforehand too. All these data are then fed into out Matlab code, which will return as an output the solution to problem (3.17); see Figure 3.7.

To gain more insights about what type of data is needed to execute our MATLAB code, consider Tables 3.1–3.4.

In Table 3.1 we include geographic data about the (x, y)-coordinates of the centroids of every neighborhood of Rotterdam, taking as the origin the neighborhood Centrum. This information is needed to compute the distances between any pair of neighborhoods. In the same table, the column "Temp. Level" identifies the capabilities of an individual neighborhood to receive or supply high temperature or low temperature heat into a heating grid; however, due to time constraints we did not incorporate this information into our design yet. The column about population should specify the number of inhabitants in each neighborhood. For simplicity we took the total number of inhabitants in Rotterdam and we equally distributed them in the neighborhoods. In our simulations, we use population data to give a rough estimate of the electricity and heat demand. Indeed, using information about the yearly average energy consumption per person for The Netherlands shown in Figure 3.8.(a), we were able to produce the synthetic electricity and heat demand profiles of each neighborhood in Rotterdam as respectively shown in Figs. 3.8.(b) and 3.8.(c). Note that we have used historic data in the period 1990–2020, nonetheless, for the purposes of our simulations, we will just simply assume the same profiles correspond to the period 2020-2050.

In Table 3.2 we display synthetic data about the efficiencies of CHP units, furnaces, and the COP of heat pumps. For simplicity, we assume that these are constant values, however, the efficiencies can be state dependent and even weather dependent. More intricate efficiency curves can be approximated by piecewise-linear functions of the devices states (see Cao et al. 2020) but this is not pursued here. Finally, Tables 3.3 and 3.4 show the costs associated to infrastructure installation and operation.

The solution returned by out MATLAB code is shown in Table 3.5 and Figure 3.9. Table 3.5 represents a sample of the data produced by our code and stored into an excel file titled **solution\_energy\_hub.xlsx**. The first sub-table displays the total costs as well as the infrastructure costs and operational costs, obtained by numerically

Neighborhood	x-coord. (km)	y-coord. (km)	Temp. Level	Population	CHP	F	$_{\mathrm{HP}}$
Or	-4	8	$_{\rm HT}$	52,500	1	1	0
Hi	0	8	-	52500	0	0	1
$\Pr$	4	8	LT	52500	1	1	0
No	0	4	-	52500	0	0	1
Kr	4	4	-	52500	0	0	1
De	-4	0	-	52500	0	0	1
Ce	0	0	-	52500	0	0	1
Wa	-8	-4	LT	52500	1	1	0
$\mathbf{Ch}$	-4	-4	-	52500	0	0	1
Fe	0	-4	-	52500	0	0	1
Ij	4	-4	LT	52500	1	1	0

Table 3.1: Geographic and device feasibility data

Table 3.2: Device efficiencies

$\eta_{\rm chp}^{\rm p}$	$\eta^{ m h}_{ m chp}$	$R_{\rm chp}^{\rm p}$	$R^{\rm h}_{ m chp}$	$\eta_{\mathrm{f}}^{\mathrm{h}}$	COP	
1	2	3	4	5	6	

Table 3.3: Infrastructure (unitary) costs

$I_{ m chp}~({ m Eur}/{ m MW})$	$I_{\rm f}~({\rm Eur}/{ m MW})$	$I_{ m hp}~({ m Eur}/{ m MW})$	$I_{\rm p}~({\rm Eur}/{\rm MW}{\cdot}{\rm km})$	$I_{ m dh}~({ m Eur}/{ m MW\cdot km})$
1	2	3	4	5

Table 3.4: Operational (unitary) costs

$\omega_{ m chp}~({ m Eur}/{ m MWh})$	$\omega_{ m f}~({ m Eur}/{ m MWh})$	$\omega_{ m p}~({ m Eur}/{ m MWh})$	$\omega_{ m h}~({ m Eur}/{ m MWh})$
1	2	3	4

solving the optimization problem (3.17). The second sub-table summarizes some of the known information that should be provided as input to the MATLAB code, but the three last columns display the optimal values of the capacities that should be installed at each energy hub concerning CHP units, furnaces, and heat pumps. Similarly, the last two columns of the third and last sub-table display a sample of the optimal capacities that electric power lines and district heating pipelines should have in order to optimally distribute the generated power and heat throughout the municipality of Rotterdam. The plots displayed in Figure 3.9 are the graphical representations of the optimal topologies of the electric power grid and of the district heating system. The values displayed next to the edges of each of these graphs represent their capacities.



Figure 3.8: (a) Yearly energy demand per person in the Netherlands. (b) Electricity consumption of each neighborhood of Rotterdam. (Synthetic data.). (c) Heat consumption of each neighborhood of Rotterdam. (Synthetic data.)

		tota	$l_{cost}$	s infrast	tructu	$re_{-}$	cost	ts operat	$tion\_cost$	s	
		998	118296.2	2	2338264	486		764	291810.2		
neighborhood	x	У	temp	population	CHP	F	HP	C chp (M	W) C f (N	MW) C hp (MV	N)
Ov	-4	8	HT	52500	1	1	0	4.8044611	53 0	0	
Hi	0	8		52500	0	0	1	0	0	7.36859237	76
$\Pr$	4	8	LT	52500	1	1	0	9.9887858	07 0	0	
No	0	4		52500	0	0	1	0	0	5.11079735	<i>i</i> 1
Kr	4	4		52500	0	0	0	0	0	0	
De	-4	0		52500	0	0	1	0	0	2.6749854	6
Ce	0	0		52500	0	0	0	0	0	0	
Wa	-8	-4	LT	52500	0	0	0	0	0	0	
Ch	-4	-4		52500	0	0	1	0	0	5.18706151	9
Fe	0	-4		52500	0	0	1	0	0	7.40370511	1
Ij	4	-4	LT	52500	1	1	0	12.112080	66 0	0	
		Edge	Source	e Target	Distanc	e (k	m)	<b>C p</b> (MW)	C dh (MW	7)	
	_	1	Ov	Hi	4			0.332306523	2.513271456	3	
		2	Ov	$\mathbf{Pr}$	8			0	0		
		3	Ov	No	5.6568	5424	9	0.328661978	0.017608603	3	
		4	Ov	Kr	8.9442	27191	L	0	0		
		5	Ov	De	8			2.211971215	0		
		6	Ov	Ce	8.9442	27191	L	0	0		
		7	Ov	Wa	12.649	1106	4	0	0		
		8	Ov	Ch	12	2		0	0		
		9	Ov	Fe	12.649	1106	4	0	0		
	_	10	Ov	Ij	14.422	2051	L	0	0		

Table 3.5: Sample of the Excel file produced by our MATLAB code.

# 3.4 Cost model - a MIP formulation

In this Section, we present a model that supports decision-making on a municipality or district level. Using a mixed integer programme (MIP) formulation of a cost minimization problem, we determine which percentage of the population's energy demand should be covered by which heat source. Hereby, costs are assumed to be social costs, i.e., to include investments on municipal and household levels, maintenance costs, and operational energy costs. An overview of all parameters and decision variables used in the model can be found in Table 3.6.

Based on data available on the websites of Leefongeving 2023; PBL 2020; Expertise Centrum Warmte 2023; Omgeving 2023 to obtain both test data and derive basic model assumptions.

The model examines a district where four different sources of heat are available.

- Low temperature heat distribution (LT)
- Mid temperature heat distribution (MT)
- Electric heating using heat pumps (HPs) or boilers
- Gas heating (G)



Figure 3.9: (a) Representation of the optimal topology of the electric power grid returned by our MATLAB code. (b) Representation of the optimal topology of the district heating network returned by our MATLAB code.

Table 3	3.6:	Nomenclature
---------	------	--------------

Variable	Definition
#H	Total number of houses
$lpha_i$	Fraction of houses connected to LT supplier i
$lpha_{it}$	Fraction of houses connected to LT supplier i at time $t$
$\beta_j$	Fraction of houses connected to MT supplier j
$\beta_{jt}$	Fraction of houses connected to MT supplier j at time $t$
$\gamma$	Fraction of houses with heat pump
$\gamma_t$	Fraction of houses with heat pump at time $t$
$\delta$	Fraction of houses using gas for heating
$\delta_t$	Fraction of houses using gas for heating at time $t$
$ ho_{ij}$	Density factor around facility $ij$ in km per household
D	Total heat demand during one time period in GJ
$F_{ m LT/MT/HP}$	CaPex connecting supplier to grid in euro
$H_{ m LT/MT/HP}$	CaPex connection of household to grid in euro
$k_{LT/MT/HP}(\alpha/\beta)$	CaPex network as a function of $\alpha/\beta$ ,
	the population density and price per km for $LT/MT/HT$
$m_{ m LT/MT/HP}$	Maintenance cost as percentage of Capex
$P_{ m LT/MT/HP}$	Price of heat in euro per GJ
$P^{km}_{ m LT/MT/HP}$	Price of LT/MT/HP network infrastructure in euro per km
Pel	Price of electricity in euro per unit of energy
$R_t^{max}$	Maximum transition realized by time $t$
$R_t^{min}$	Minimum transition realized by time $t$

### 3.4.1 Model description

In the following, we discuss the base cost model for heat transition within a district or area. We start by discussing decision variables and components of the cost function, after which we present the full model in Section 3.4.1.

If the suppliers of LT are indexed by i and the suppliers of MT by j, the decision variables within the model are the fractions  $\alpha_i, \beta_j, \gamma, \delta$ . Here  $\alpha_i$  for the fraction of households using LT from source i and  $\beta_j$  the fraction of households using LT from source j. Next to that,  $\gamma$  and  $\delta$  are the fractions of households using electric heating and gas boilers respectively. To supply all households with heat we impose  $\sum_i \alpha_i + \sum_j \beta_j + \gamma + \delta = 1$ . The output is a total social cost  $c(\alpha_i, \beta_j, \gamma, \delta)$  for the district over a given time period. The objective function of the model is to minimize the cost. The social cost includes both investments to be made by the utility companies and installation and energy costs for the consumer. The total social cost is the sum over the cost for each different energy source

$$c = c_{\rm LT} + c_{\rm HT} + c_{\rm HP} + c_{\rm G}$$
 (3.18)

where  $c_{\text{LT/MT/HP/G}}$  is the total social cost associated with LT, MT, HP, and G respectively. In the following sections, each energy source will be examined to define their respective social costs.

#### Low/Medium temperature heat distribution

Given LT sources indexed by *i* the cost for fractions  $\alpha_i$  is given by:

$$c_{\rm LT}(\alpha) = \sum_{i} (1+m(t))(\alpha_i H + F_{\rm LT} + k_{\rm LT}(\alpha_i)) + \alpha_i Dp_{\rm LT}$$
(3.19)

where m(t) is the cost of maintenance as a fraction of the total CaPex;  $F_{\rm LT}$  is the cost of connecting an LT facility to the grid;  $k_{\rm LT}(\alpha_i)$  is the cost of expanding the network such that facility  $LT_i$  supplies some  $\alpha_i$  of the total heat demand. Assuming a uniform population distribution, one can approximate that function to be linear in  $\alpha_i$ , say  $k_{LT}(\alpha_i) = \alpha_i \cdot P_{LT}^{km} \cdot \#H \cdot \rho_i$  where  $P_{LT}^{km}$  is the price per km of LT network infrastructure, #H is the total number of households, and  $\rho_i$  is a density factor describing the kilometers of infrastructure needed to connect a new house. Due to the uniformity of the distribution, the area covered with some km of the network is linearly related to  $\alpha_i$ . One approximation is  $k_d$  as depending on the radius and population density around facility  $LT_i$ . This kilometer function is crucial to the model and will be discussed in more detail in Section 3.4.4.

For the medium temperature the same variables are considered, but the values differ.

#### Heatpumps and electric boilers

For the electricity network, we do not consider the costs of the network, since it will be also expanded for other purposes and we were not able to separate these costs. For a municipality, it would not be complicated to add to the model. The cost function corresponding to electric heating is

$$c_{\rm HP}(\gamma) = \gamma H_{\rm HP} + \gamma D p_{\rm HP}. \tag{3.20}$$

#### Gas Network

For gas, i.e. the status quo, we assume no further costs have to be made to expand the network since (almost) all households currently have a gas-based boiler and are already connected to the gas network. The cost function therefore purely depends on the operational costs of the network. For coherence, we model the maintenance term in the same fashion as for other heat sources, as a yearly fraction m(t) of some initial investment  $H_{\rm G}$ 

$$c_{\rm G}(\delta) = \delta m(t) H_{\rm G} + \delta D p_{\rm G}$$

#### Mixed integer programme formulation

Based on the cost functions presented above, we derive a simple cost minimization model:

$$\min_{\alpha,\beta,\gamma,\delta} c_{\rm LT}(\alpha) + c_{\rm HT}(\beta) + c_{\rm HP}(\gamma) + c_{\rm G}(\delta)$$
(3.21)

subject to 
$$\alpha = \sum_{i} \alpha_{i}$$
 (3.22)

$$\beta = \sum_{j} \beta_{j} \tag{3.23}$$

$$\alpha_i \cdot D \le LT_i^{cap} \quad \forall i \tag{3.24}$$

$$\beta_j \cdot D \le HT_j^{cap} \quad \forall j \tag{3.25}$$

$$\alpha + \beta + \gamma + \delta = 1 \tag{3.26}$$

$$0 \le \alpha_i, \beta_j, \gamma, \delta \le 1 \quad \forall i, j \tag{3.27}$$

Next to the cost functions, we add various constraints. Equations (3.24) and (3.25) ensure that the heat demand covered by LT and HT facilities does not exceed their respective capacities. Equation (3.26) sums the fractions of the demand per heat source, specifying that all demand is met (sum to one). The group of constraints in Equation (3.27) force decision variables to be between zero and one. Given that they model fractions, this is a reasonable assumption. Note that by pre-defining  $\delta$ , one can model phase-out targets. For example, setting  $\delta = 0$  would minimize costs for a full transition away from gas.

## 3.4.2 **Proof of concept implementation**

As a proof of concept, we implemented a version of the base model presented in Section 3.4 and ran numerical experiments. For simplicity, we model only a single time step, removing time-dependency. Furthermore, we fix  $\delta = 0$ , modelling a full transition away from gas. The MATLAB code used for the experiment can be found in Appendix 3.7

#### Data

The numerical experiment has been done for five selected districts in the municipality of Groningen. The corresponding used parameters in the model are presented in Table 3.7.

neighborhood	BU00140300	BU00140402	BU00140403	BU00140404	BU00140500
name	De Hoogte	Bloemenbuurt	Florabuurt	Damsterbuurt	De Linie
#H	2,416	1,789	807	702	411
$H_{ m LT}$	1,015,213	$834,\!439$	306,541	319,201	$121,\!054$
$F_{\rm LT}$	50,000	50,000	50,000	50,000	50,000
$P_{\rm LT}$	31	31	31	31	31
$H_{\rm MT}$	377,505	292,075	124,939	69,858	49,294
$F_{\rm MT}$	500,000	500,000	500,000	500,000	500,000
$P_{\rm MT}$	36	36	36	36	36
$M_{textG}$	1,072,704	794,316	358,308	$311,\!688$	$182,\!484$
$P_{ m G}$	22	22	22	22	22
Pel	31	31	31	31	31

Table 3.7: Neighborhood Statistics (PBL 2020 and DHM)

The figures that differ per neighborhood are from the "Startanalyse aardgasvrije buurten" PBL 2020 which provides an overview of the costs per neighborhood to transition to different energy sources. The other costs and prices were provided by DHM.

#### Results

The result of the numerical model for De Hoogte is shown in Figures 3.10, 3.11, 3.12, 3.13, 3.14, 3.15 and 3.16. They indicate the costs per heating method over multiple time spans and electricity prices. The optimal method is based on minimized cost and is a combination of individual heating methods. Note that in the figures the optimal method is always equal to a pure method due to the linear kilometer function and therefore taking linear combinations of multiple methods will always result in a higher cost than picking the method with the lowest cost. For these Figures, energy prices in the Netherlands are used from before the war in Ukraine.



Figure 3.10: Calculated cost for each Strategy of LT, MT, HP, and the optimal solution, in District 1. The price of electricity is assumed to be 30 cents per kWh on average, which is a high estimate.



Figure 3.11: Calculated cost for each Strategy of LT, MT, HP, and the optimal solution, in District 1. The price of electricity is assumed to be 30 cents per kWh on average, which is a high estimate.



Figure 3.12: Calculated cost for each Strategy of LT, MT, HP, and the optimal solution, in District 1. The price of electricity is assumed to be 30 cents per kWh on average, which is a high estimate.



Figure 3.13: Calculated cost for each Strategy of LT, MT, HP, and the optimal solution, in District 1. The price of electricity is assumed to be 30 cents per kWh on average, which is a high estimate.



Figure 3.14: Calculated cost for each Strategy of LT, MT, HP, and the optimal solution, in District 1. The price of electricity is assumed to be 30 cents per kWh on average, which is a high estimate.



Figure 3.15: Calculated cost for each Strategy of LT, MT, HP, and the optimal solution, in District 1. The price of electricity is assumed to be 30 cents per kWh on average, which is a high estimate.


Figure 3.16: Calculated cost for each Strategy of LT, MT, HP, and the optimal solution, in District 1. The price of electricity is assumed to be 30 cents per kWh on average, which is a high estimate.

From these figures, it can be seen that a 100% heat pump is always the best strategy. However, it should be noted that the relative difference gets smaller as the time horizon increases. Furthermore, a heat pump will usually only last around 15 - 20 years but this is only a small difference in costs and does not affect the outcome. Another thing to note is that a heat pump is only economically profitable after about 10 - 15 years and because the homeowners have to pay for it themselves, they are forced to keep their home until they gained their money back if they do not want to lose some invested money, though this loss is possibly compensated by an increase in home value. It is also possible that not all homes are suited for a heat pump such as certain apartments or buildings that are poorly isolated, in which case even more investments would be necessary. Finally, the results could differ highly if another (more realistic) kilometer function k was used, more on this can be found in section 3.4.4. The results from these figures were similar for the other districts that were tested.

### 3.4.3 Adding time dependency to the model

For policy makers, various decisions have to be made with regard to time. Firstly, decision-makers are interested in a short-term answer on what first steps to take in the transition. What and where should they invest right now? Secondly, minimizing the cost of the entire heat transition over a number of years, while also knowing when to make which investments, is crucial in the long run. Those two perspectives differ in practice. For example, looking at a one-year horizon where we need to restructure the heat supply for any one district in a municipality, it might be the cheapest to choose a small and sparsely populated district and install heat pumps in all buildings. In the long run, some of those buildings might be close to facilities that in the coming years will be added to a district heating network. Then, the costs of connecting those buildings to the district heating network might be less than installing heat pumps. We adapt the base model presented in Section 3.4.1 accordingly to facilitate that decision-making.

To this end, we add time dependency to the decision variables. For example, instead of  $\alpha_i$ , we now consider  $\alpha_{it}$ , the fraction of the demand supplied by LT facility i at time t. Similarly for other decision variables. A time step could for example be a (financial) year. The decision variables now need to be initialized at time 0. A reasonable scenario would be to initialize all decision variables for LT, HT, and HP to be zero, and assume a heat network with all households having gas-boilers,  $\delta_0 = 1$ .

If run like that, the solution for times  $t = 1, \cdots$  will be all the same, converging to the most cost effective model within the first time step. Depending on the prices, the result may be an all-gas infrastructure, or the entire heat demand should be covered by district heating after the first year. In practice, a transition away from gas over the course of multiple years is desirable. We model the first requirement by introducing intermediate transition goals. Be  $R_t^{min}$  the minimum fraction of the demand that should be non-gas by time t. The following constraint ensures that the heat transition in the region happens at a minimum rate:

$$R_t^{min} \le \alpha_t + \beta_t + \gamma_t \quad \forall t$$

In particular, for t = 0 this approach can support decision-making that requires a certain amount of direct action (e.g. connect at least a certain amount of households to electric or waste-heat sources within the next year) while also minimizing costs over the entire transition.

Similarly, we can impose a maximum transition rate by introducing a constraint

$$R_t^{max} \ge \alpha_t + \beta_t + \gamma_t \quad \forall t$$

where  $R_t^{max}$  describes the maximum fraction that can be supplied by non-gas powered heat by time t. This allows us to model for example yearly budget constraints, construction times, and other resource-related constraints. Alternatively,  $R_t^{max}$  could be modelled as the maximum growth rate bounding  $\alpha_t + \beta_t + \gamma_t - (\alpha_{t-1} + \beta_{t-1} + \gamma_{t-1})$ . We include time dependency in the cost function as follows:

$$c_{\rm LT}(\alpha, t) = (1 + m(t)) \sum_{i} ((\alpha_{it} - \alpha_{it-1}) H_{\rm LT}$$

$$+ \mathbf{1}_{\{ \substack{\alpha_{it} > 0, \\ \alpha_{it-1} = 0 \}}} F_i + k(\alpha_{it}) - k(\alpha_{it-1}))$$

$$+ \sum_{i} \alpha_{it} \cdot D \cdot p_{\rm LT}$$

$$c_{\rm HT}(\beta, t) = (1 + m(t)) \sum_{j} ((\beta_{jt} - \beta_{jt-1}) H_{\rm HT}$$

$$+ \mathbf{1}_{\{ \substack{\beta_{jt} > 0, \\ \beta_{jt-1} = 0 \}}} F_j + k(\beta_{jt}) - k(\beta_{jt-1}))$$

$$+ \sum_{j} \beta_{jt} \cdot D \cdot p_{\rm HT}$$

$$c_{\rm HP}(\gamma, t) = (1 + m(t)) \cdot ((\gamma_t - \gamma_{t-1}) H_{\rm HP}) + \gamma_t \cdot D \cdot p_{\rm HP}$$

$$c_{\rm G}(\delta, t) = \delta_t \cdot D \cdot p_{\rm G}$$

$$(3.30)$$

As opposed to the base model, the time-dependent cost functions discount investment costs per time step. That is to say, investments made in the previous time step won't have to be made in the following time steps.

A full formulation of the time-adapted model is presented below:

$$\min_{\alpha,\beta,\gamma,\delta} \sum_{t} c_{\rm LT}(\alpha,t) + c_{\rm HT}(\beta,t) + c_{\rm HP}(\gamma,t) + c_{\rm G}(\delta,t)$$
(3.32)

subject to 
$$\alpha_t = \sum_i \alpha_{it} \quad \forall t$$
 (3.33)

$$\beta_t = \sum_j \beta_{jt} \quad \forall t \tag{3.34}$$

$$\alpha_{it} \cdot D \le LT_i^{cap} \quad \forall i, t \tag{3.35}$$

$$\beta_{jt} \cdot D \le HT_j^{cap} \quad \forall j, t \tag{3.36}$$

$$0 \ge \delta_t - \delta_{t-1} \tag{3.37}$$

$$\alpha_t + \beta_t + \gamma_t + \delta_t = 1 \tag{3.38}$$

$$R_t^{min} \le \alpha_t + \beta_t + \gamma_t \quad \forall t \tag{3.39}$$

$$R_t^{max} \ge \alpha_t + \beta_t + \gamma_t \quad \forall t \tag{3.40}$$

$$0 \le \alpha_{it} \le 1 \quad \forall i, t \tag{3.41}$$

$$0 \le \beta_{jt} \le 1 \quad \forall j, t \tag{3.42}$$

 $0 \le \gamma_t \le 1 \quad \forall t \tag{3.43}$ 

$$0 \le \delta_t \le 1 \quad \forall t \tag{3.44}$$



Figure 3.17: Illustrative example of the shortcomings of a purely radius-based kilometer function.

### 3.4.4 Notes on the kilometer function

The cost functions for LT and MT networks (Section 3.4.1) depend on a kilometer function  $k_{\text{LT}}(\alpha_i)$  (or similarly for MT and  $\beta_j$ ). It specifies the cost of expanding the heat network such that facility  $\text{LT}_i$  supplies a fraction  $\alpha_i$  of the total energy demand. This is excluding the cost of connecting the facility to the heat network. Therefore, the cost depends strongly on how many kilometers of pipelines are needed for transport, and we refer to the function as the kilometer function.

Above, we pointed out that, given a uniform population distribution, one can approximate that function to be linear in  $\alpha_i$ , say  $k_{\text{LT}}(\alpha_i) = \alpha_i \cdot P_{\text{LT}}^{km} \cdot \#H \cdot \rho_i$  where  $P_{\text{LT}}^{km}$  is the price per km of LT network infrastructure, #H is the total number of house-holds, and  $\rho_i$  is a density factor describing the kilometers of infrastructure needed to connect a new house. There are various limitations that come with this assumption. We will address them in this section.

Firstly,  $\rho_i$  represents the density of the population around the facility which in practice is likely not uniformly distributed. If the heat source is in an industrial area, for example, connecting it to the closest household requires more kilometers per household than connecting it to the closest household and its neighbor. Depending on the scale of the problem, a facility might be between a densely and sparsely populated area for which the radial approximation with constant density is a poor modelling choice.

Secondly, the setup assumes a single-facility dependency. In practice, facilities might be close to each other, and their supply radius might overlap. See Figure 3.17 for an illustrative example of overlapping radii, assuming that the plots are the distance to the nearest facility. Due to this overlap, the investment in heat networks around a certain facility to cover a given fraction of the heat demand might be higher than approximated by the function proposed above. A similar effect can be observed if a facility is near the border of the considered district.

Finally, we note that in developing the kilometer function, one should consider

the insulation of households. Especially in densely populated areas, it may be costefficient to choose heat pumps for houses that are within the radius of a facility and instead expand the heat network to houses slightly farther away.

To conclude this discussion, the kilometer function  $k_{\text{LT}}(\alpha_i)$  needs to be considered carefully. We recommend further research into what approximation fits the real context best. One approach could be a function depending on the population density, or based on the size of the existing gas infrastructure of an area.

# 3.4.5 Recommendations and Future work

The improvements to the model can be divided into two categories. Firstly, we will discuss recommendations to improve the accuracy of the model. Secondly, we share ideas on how the model can be adapted to answer further questions arising on a municipal level.

### Accuracy

**Energy price** In reality the electricity price is elastic, depending on market prices which in turn depend on the total electricity demand. Based on the prognoses of the growth of (green) energy sources, this could be implemented quickly into the model.

**KM-function** On top of the suggestions given in section 3.4.4 connections can also be made with the two models described earlier. Both the energy hub formulation and the graph theoretical method could calculate the cost for a given partitioning of the energy consumption.

### Further questions

**Bankruptcy** If the municipality decides on a particular strategy, it is important to know how robust this solution is. If the district is dependent on HT heating from a single facility, the bankruptcy of this facility could have a big impact on the total cost. To study this the capacity per heat source could change over time.

**Cost of infrastructure** Something raised by the problem owner, is that municipalities hope for the future decrease in infrastructure costs. If such a decrease in F, H over time is added, the effect could be studied to determine whether waiting could decrease the total costs.

# 3.5 Conclusion

In this project, we have presented three modeling approaches for the heat transition, addressing the challenge posed by DHM. Firstly, we discussed graphical representations and Monte Carlo simulations as a means of modelling heat and energy demands, as well as network capacities. Secondly, we introduced an energy hub model that allows for modelling the interrelationship between various energy commodities, such as electricity and heat. Using this model, we formulated an optimal capacity design problem, which we applied to the case of the municipality of Rotterdam.

Both the Monte Carlo approach and the energy hub model are adaptable to various use cases and scales, making them useful tools for addressing different challenges in the heat transition.

Lastly, we presented a high-level mixed integer programme that determines the optimal fraction of heat demand to supply per heat source and commodity, while minimizing social costs. This model focuses more directly on modelling costs, rather than energy flows, distinguishing it from the other models presented in this project.

In conclusion, these three models offer valuable insights for supporting decisionmaking processes in the heat transition, providing a solid starting point for further research towards a sustainable and cost-effective solution.

## 3.5.1 Recommendations and future work

Our study has highlighted multiple open questions regarding the heat transition, and the models presented in this chapter offer some solutions while being adaptable to other challenges. We suggest combining the models into an integrated approach to create a more holistic cost model.

Furthermore, we recommend defining use cases and objectives to tailor the models to specific applications. This approach will increase transparency and likely acceptance from a stakeholder perspective, benefiting both dissemination at municipalities and explainability to private households.

A careful definition of the objective function is also crucial. In our study, we assumed the minimization of social costs as the objective function, but in practice, questions of energy security and access, as well as issues of energy justice, need to be considered.

Lastly, we provide proof of concept code for two of the models, but validation using real data and realistic assumptions is still required. Future research could focus on validating the models and developing new models to address the challenges of heat transition.

# 3.6 Appendix: Matlab code for optimizing energy hub

The code presented here corresponds to the model presented in Section 3.3.

```
1
2 close all
3 clc
4
```

```
_{5} %%% Load the geographic and energy data about the region of
      → interest. In
  %%% our example about Rotterdam, each neighborhood represents an
6
      \hookrightarrow Energy
  %%% Hub. Each has energy generation and demand capabilities.
7
8
  EH_nodes=readtable('EH_simplified_data_rotterdam.xlsx','Range','
9
      \hookrightarrow C4:J15'); "Each EH is viewed as a node of a graph.
10
  n_hubs=size(EH_nodes,1); %number of energy hubs.
11
12
  %%% Synthetic generation of the electric power and heat demand
13
      \hookrightarrow profiles of
  %%% each energy hub (neighborhood).
14
15
  %%% First, we import the data (from the IEA) about the
16
     \hookrightarrow approximate yearly energy
17
  %%% consumption per capita in the Netherlands.
18
19
  capita_energy=table2array(readtable('EH_per_capita_energy_nl.csv'
      years=capita_energy(:,1); %number of years
20
  avg_energy_capita=capita_energy(:,2); % average yearly energy
21
      \hookrightarrow demand per person.
22
  %%% The energy is now expressed in MWh, not KWh as in the Excel
^{23}
     \hookrightarrow file.
  avg_energy_capita=1e-3*avg_energy_capita;
24
25
26
  %%% Plot of the energy demand per person in the Netherlands.
27
28 figure()
  plot(years,avg_energy_capita,'-b','LineWidth',2)
29
  xlabel('year')
30
  ylabel('Energy_per_person_(MWh)')
31
  grid on
32
  exportgraphics(gcf, 'EH_fig_energy_capita.pdf');
33
34
35
  %%% Number of inhabitants in Rotterdam
36
  no_inhab_rotterdam=623652;
37
38
  %%% Number of inhabitants in each neighborhood (equalized)
39
  no_inhab_neighborhoods=623652/n_hubs*ones(n_hubs,1);
40
41
42 XXX Yearly (total) energy consumption per neighborhood. We take
      \hookrightarrow the yearly
43 XXX data per person and multiply it by the number of inhabitans
```

```
\hookrightarrow in each
  %%% energy hub.
44
45
  yearly_cons_neigh=zeros(length(years),n_hubs);
46
  for i=1:n_hubs
47
      yearly_cons_neigh(:,i)=diag(0.9+0.2*rand(length(years),1))*
48
          → avg_energy_capita*no_inhab_neighborhoods(i,1);
49
  end
50
  %%% Assumption: The total energy consumption per neighborhood is
51
  %%% distributed 33% for electric power and 66% for heat.
52
53
54
  %%% Yearly (total) electric power consumption/production per
55
      \hookrightarrow neighborhood.
  yearly_power_neigh=0.33*yearly_cons_neigh;
56
57
  %%% Plot:
58
59
  figure()
  for i=1:n_hubs
60
      plot(years, yearly_power_neigh(:,i),'LineWidth',1)
61
      legend
62
      hold on
63
      grid on
64
      xlabel('year')
65
      ylabel('Electric_energy_consumption_(MWh)')
66
      xlim([years(1) years(end)+10])
67
  end
68
  legend(EH_nodes.neighborhood)
69
  exportgraphics(gcf, 'EH_fig_heat_consumption.pdf');
70
71
72
  %%% Yearly (total) heat consumption/production per neighborhood.
73
  yearly_heat_neigh=0.66*yearly_cons_neigh;
74
75
76 %%% Plot:
  figure()
77
  for i=1:n_hubs
78
      plot(years, yearly_heat_neigh(:,i),'LineWidth',1)
79
      hold on
80
      grid on
81
      xlabel('year')
82
      ylabel('Heat_consumption(MWh)')
83
      xlim([years(1) years(end)+10])
84
  end
85
86 legend(EH_nodes.neighborhood)
  exportgraphics(gcf, 'EH_fig_electricity_consumption.pdf');
87
88
```

```
89
90
91
   N=32; %size of data (we run for 30 years)
92
93
94
95
   \frac{1}{2} We assume that each node is connected to all the others.
                                                                            This
96
           is useful
       \rightarrow
   %%% to define the distance between nodes. The capacity costs of
97
       \hookrightarrow power lines
   %%% and district heating piping is dependent on the distance
98
       \hookrightarrow between nodes.
   %%% Also, it will help us define incidence matrices whose
99
       \hookrightarrow topologies we
   %%% will design optimally.
100
101
   %%% Note also that we identify the source and target nodes of
102
       \hookrightarrow each edge. To
   %%% each edge we assign a weight representing the distance
103
       \hookrightarrow between energy
   %%% hubs.
104
105
   EH_edges=table();
106
   EH_edges.Edge=(1:n_hubs*(n_hubs-1)/2);
107
   Source=[];
108
109 Target = [];
   Distance=zeros(n_hubs*(n_hubs-1)/2,1);
110
   count=0;
111
112
   for i=1:n_hubs
        for j=i+1:n_hubs
113
            count = count + 1;
114
            Source=[Source;{EH_nodes.neighborhood{i}}];
115
            Target = [Target; {EH_nodes.neighborhood { j } }];
116
            Distance(count)=norm([EH_nodes.x(i);EH_nodes.y(i)]-[
117
                 \hookrightarrow EH_nodes.x(j);EH_nodes.y(j)],2);
        end
118
   end
119
120
   EH_edges.Source=Source;
121
   EH_edges.Target=Target;
122
   EH_edges.Distance=Distance;
123
124
   %%% Having defined the edges, we can define an incidence matrix.
125
       \hookrightarrow Note that
126 %%% this incidence matrix still codifies the preliminary
       \hookrightarrow interconnection
127 XXX among all the Energy Hubs. This is mostly used to identify a
```

```
\hookrightarrow subgraph
   %%% formed by removing interconnection points. Such a removal is
128
       \hookrightarrow done by
   %%% enforcing the constraint that if the capacity of a given
129
       \hookrightarrow power line or
   %%% district heating pipe is zero, then the associated flow
130
       \hookrightarrow should also be
   % % % zero.
131
132
   calB_p=zeros(n_hubs,size(EH_edges,1));
133
   for i=1:n_hubs
134
        for j=1:size(EH_edges,1)
135
             if strcmp(EH_nodes.neighborhood{i},EH_edges.Source{j})
136
             calB_p(i,j) = -1;
137
             elseif strcmp(EH_nodes.neighborhood{i},EH_edges.Target{j
138
                 \rightarrow \rightarrow \rightarrow
                  calB_p(i,j)=1;
139
             end
140
141
        end
142
   end
143
   %%% Adjacency matrix of G_p.
144
   calA_p=-calB_p*calB_p'+(n_hubs-1)*diag(ones(n_hubs,1));
145
146
   %%% We temporarily enforce the condition that the power system
147
       \hookrightarrow and
   %%% the district heating system graphs are equal. Recall that
148
       \hookrightarrow this is
   %%% changed when the actual capacities of each line/pipe are
149
       \hookrightarrow computed.
   calB_dh=calB_p;
150
   calA_dh=calA_p;
151
152
153
154
155
156
157
   %%% FIXED PARAMETERS OF THE SYSTEM
158
159
   %%% Devices efficiencies.
160
   dum=table2array(readtable('EH_simplified_data_rotterdam.xlsx','
161

→ Range', 'H23:H28'))';

   eta_chp_p=dum(1);
162
163 eta_chp_h=dum(2);
164 R_chp_p=dum(3);
|_{165}| R_chp_h=dum(4);
166 eta_f_h=dum(5);
```

```
COP = dum(6):
167
168
169
170
   %%% Parameters for the infrastructure costs
171
172
  dum=table2array(readtable('EH_simplified_data_rotterdam.xlsx','
173
      → Range', 'L23:L27'))';
  I_chp=dum(1); % Installation costs CHPs (X EUR/MW)
174
175 I_f=dum(2); % Installation costs Furnaces (X EUR/MW)
  I_hp=dum(3); % Installation costs Heat Pumps (X EUR/MW)
176
177 I p=dum(4): % Installation costs power lining (X EUR/(MWh))
  I_dh = dum(5);
                  % Installation costs District Heating Piping (X EUR
178
      \hookrightarrow /(MWh))
179
180
181
   %%% Parameters for the operational costs
182
   dum=table2array(readtable('EH_simplified_data_rotterdam.xlsx','
183
      → Range', 'P23:P26'))';
   omega_chp=dum(1)*ones(N,1); % Price of natural gas (EUR/MWh)
184
   omega_f=dum(2)*ones(N,1); % Price of natural gas OR of biomass (
185
       → EUR/MWh)
   omega_p_out=dum(3)*ones(N,1); % Price of selling/buying
186
      \hookrightarrow electricity (EUR/MWh)
   omega_h_out=dum(4)*ones(N,1); % Price of selling/buying heat (EUR
187
      \rightarrow /MWh)
188
189
190
   %%% DECISION VARIABLES
191
192
193
  %%% capacities
  C_chp=sdpvar(n_hubs,1); % of CHPs
194
  C_f=sdpvar(n_hubs,1); % of Furnaces
195
196 C_hp=sdpvar(n_hubs,1); % of Heat Pumps
197 C_p=sdpvar(n_hubs*(n_hubs-1)/2,1); % of power lines
198 C_dh=sdpvar(n_hubs*(n_hubs-1)/2,1); % of district heating piping
  B_vec=sdpvar(n_hubs*(n_hubs-1)/2,1); % Admittance vector of power
199
          lines
      \rightarrow
200
201
202
  %%% mass-related variables (gas or biomass)
203
  m_chp_in=sdpvar(repmat(n_hubs,1,N),ones(1,N));
204
  m_f_in=sdpvar(repmat(n_hubs,1,N),ones(N));
205
206
207 %%% Interaction variables (between EHs and power or heat grid)
```

```
p_out=sdpvar(repmat(n_hubs,1,N),ones(1,N));
208
   h_out=sdpvar(repmat(n_hubs,1,N),ones(1,N));
209
210
   %%% power inlet to HPs
211
   p_hp_in=sdpvar(repmat(n_hubs,1,N),ones(1,N));
212
213
   %%% Power output of CHPs
214
   p_chp_out=sdpvar(repmat(n_hubs,1,N),ones(1,N));
215
216
   %%% Heat output of CHPs
217
   h_chp_out=sdpvar(repmat(n_hubs,1,N),ones(1,N));
218
219
   %%% Heat output of Furnaces
220
   h_f_out=sdpvar(repmat(n_hubs,1,N),ones(1,N));
221
222
   %%% Heat output of HPs
223
   h_hp_out=sdpvar(repmat(n_hubs,1,N),ones(1,N));
224
225
226
227
   %%% Power edges
228
229
   phi_p=sdpvar(repmat(n_hubs*(n_hubs-1)/2,1,N),ones(1,N));
230
231
232
233
   %%% Number of edges in the district heating system's graph
234
   n_e_dh=2*size(EH_edges,1)+n_hubs;
235
236
   %%% Mass flow edges district heating system
237
   phi_dh=sdpvar(repmat(n_hubs*(n_hubs-1)/2,1,N),ones(1,N));
238
239
240
241
242
243
   %%% Definition of the constraints and cost function
244
245
246
   %%% non-negativity constraints on capacities:
247
   cap_cons = [0 \le C_chp, 0 \le C_f, 0 \le C_hp, 0 \le C_p, 0 \le C_dh];
248
249
250
251
  %%% Constraints associated with the non-feasibility of devices:
252
       \hookrightarrow recall that
  %%% there are EHs for which we could know a priori if it will
253
       \hookrightarrow have an
```

```
%%% specific energy transforming device.
254
   exis_cons=[diag(EH_nodes.CHP==0)*C_chp==0,diag(EH_nodes.F==0)*C_f
255
       \hookrightarrow ==0,diag(EH_nodes.HP==0)*C_hp==0];
256
257
258
   %%% Next we run a for loop to compute the yearly, monthly or
       \rightarrow hourly
   %%% operation costs of the system. Some operational constraints
259
       \hookrightarrow should also
   %%% be satisfied in the same time stamps.
260
   f o=0:
261
   op_cons=[]; %%% Operational constraints
262
263
264
265
   for k=1:N
266
267
       p_in=-yearly_power_neigh(k,:)';
268
269
       h_in=-yearly_heat_neigh(k,:)';
270
271
272
        %%% Balance equations at each Energy Hub
273
        power_balance_EH=p_in+p_chp_out{k}-p_hp_in{k}==p_out{k};
274
        heat_balance_EH=h_in+h_chp_out{k}+h_hp_out{k}+h_f_out{k}==
275
            \hookrightarrow h_out{k};
276
        %%% Constituve relationships of the devices present in each
277
            \hookrightarrow Energy Hub.
        consti_CHP=m_chp_in{k}==p_chp_out{k}./eta_chp_p+h_chp_out{k
278
            \hookrightarrow }./eta_chp_h;
        consti_F=m_f_in{k}==h_f_out{k}./eta_f_h;
279
        consti_hp=h_hp_out{k}==COP.*p_hp_in{k};
280
281
282
        %%% Power flow equations:
283
        power_flow=p_out{k}+calB_p*phi_p{k}==0; %Kirchhoff's law at
284
           \hookrightarrow nodes.
285
        %%% Heat flow equations.
286
287
        heat_flow=[h_out{k}+calB_dh*phi_dh{k}==0];
288
289
290
291
        %%% Operational constraints for the CHPs
292
        constraint_capacity_CHP=[p_chp_out{k}./eta_chp_p+h_chp_out{k
293
            \hookrightarrow }./eta_chp_h<=C_chp,...
```

294	R_chp_p*C_chp<=p_chp_out{k}./eta_chp_p,h_chp_out{k}./
	$\hookrightarrow$ eta_cnp_n<=R_cnp_n*C_cnp];
295	
296	1/1/1 Uperational constraints for the Furnaces
297	constraint_capacity_F=h_f_out{k}<=C_f;
298	
299	1/1/2 Uperational constraints for the Heat Pumps
300	constraint_capacity_HP=h_hp_out{k}<=C_hp;
301	
302	XXX The power flow through a given line is constrained by the
	$\rightarrow$ capacity
303	XXX of the line. If the capacity is zero, then the flow
	Should be zero
304	//// through that line.
305	constraint_capacity_p=[pni_p(k)<=0_p,pni_p(k)>=-0_p];
306	444 The constructed for the best flow through the district
307	XXX The constraint for the near from through the assirict
	$\rightarrow$ neutring 1/2 minima follows the same measuring as for the new minima
308	ARA piping joillows the same reasoning as jor the power times
	$\gamma$ uescribeu
309	NNN u000e. constraint canacity dh=[nhi dh{k}<=C dh nhi dh{k}>=_C dh].
211	constraint_capacity_an_[phi_an(k) <=0_an, phi_an(k) >==0_an],
212	
212	111 Non-negativity constraints: most of the innus to the
313	$\rightarrow$ devices should
314	%%% be non-negative
315	non negativity cons=[m chp in{k}>=0.m f in{k}>=0.h hp out{k
	$\rightarrow$ }>=0.h chp out{k}>=0]:
316	
317	
318	
319	%%% We combine all operational constraints into a single one.
320	
321	op_cons=[op_cons,non_negativity_cons,constraint_capacity_HP,
322	<pre>constraint_capacity_CHP,power_flow,heat_flow,consti_hp,</pre>
	→ consti_F,consti_CHP,
323	heat_balance_EH,power_balance_EH,constraint_capacity_dh,
	<pre> → constraint_capacity_p]; </pre>
324	
325	
326	%%% Operational costs:
327	$f_o=f_o+omega_chp(k,1)*sum(m_chp_in\{k\})+omega_f(k,1)*sum($
	$\hookrightarrow$ m_f_in{k})
328	$-$ omega_p_out(k,1)*sum(p_out{k})-omega_h_out(k,1)*sum(
	$\hookrightarrow$ h_out{k});
329	

```
end
330
331
   %%% Infrastructure costs
332
   f_C=I_chp*sum(C_chp)+I_f*sum(C_f)+I_hp*sum(C_hp)+I_p*sum(EH_edges
333
      → .Distance.*C_p)+I_dh*sum(EH_edges.Distance.*C_dh);
334
   %%% Objective function to be minimized.
335
   objective=f_o+f_C;
336
337
   %%% All constraints.
338
   cons=[cap_cons, exis_cons, op_cons];
339
340
   \%\%\% We solve the posed optimization problem using YALMIP
341
   optimize(cons,objective)
342
3/13
344
   disp('total__costs')
345
   value(objective)
346
347
   disp('Infrastructure__costs')
348
   value(f_C)
349
350
   disp('Operational_costs')
351
   value(f o)
352
353
   %%% We export the data about costs into an excel file
354
355
   \%\%\% We add the obtained solution to tables, for easy reading and
356
      \hookrightarrow for exporting.
357
   costs_table=table();
358
   costs_table.total_costs=value(objective);
359
   costs_table.infrastructure_costs=value(f_C);
360
   costs_table.operational_costs=value(f_o);
361
  EH_nodes.C_chp_MW=1e-6*value(C_chp); % capacities of CHP units
362
  EH_nodes.C_f_MW=1e-6*value(C_f); % capacities of gas furnaces
363
   EH_nodes.C_hp_MW=1e-6*value(C_hp); % capacities of heat pumps
364
   EH_edges.C_p_MW=1e-6*value(C_p); % (indicative) capacities of
365
       \hookrightarrow power lines
   EH_edges.C_dh_MW=1e-6*value(C_dh); % (indicative) capacities of
366
      \hookrightarrow heating pipelines
367
   %%% We export the solution of the capacities into an excel file:
368
369
   system('taskkill_/F_/IM_EXCEL.EXE');
370
371
   delete 'EH_solution_energy_hub.xlsx'
372
373
```

```
writetable(costs_table,'EH_solution_energy_hub.xlsx','Sheet',1,'
374
       \hookrightarrow Range', 'C2:F4')
375
   writetable (EH_nodes, 'EH_solution_energy_hub.xlsx', 'Sheet', 1, '
376
       \hookrightarrow Range', 'C5:M20')
377
   writetable (EH_edges, 'EH_solution_energy_hub.xlsx', 'Sheet', 1, '
378
       ↔ Range', 'C20:M100')
379
380
381
   %%% Plot of the electric power network: topology and line
382
       \hookrightarrow capacities.
383
   figure()
384
   adj_p=calB_p*diag(value(C_p))*calB_p';
385
386 C_p_val=value(C_p);
  C_p_val_red=C_p_val(C_p_val^{\sim}= 0);
387
388 L_widths_p=C_p_val_red/(sum(C_p_val_red)/length(C_p_val_red));
  test_p=graph(adj_p,'omitselfloops');
389
  plot_p=plot(test_p,'g','LineWidth',1.5,'XData',EH_nodes.x,'YData'
390
       → ,EH_nodes.y,'NodeLabel',...
       EH_nodes.neighborhood,'EdgeLabel',1e-6*C_p_val_red,'LineWidth
391
           \hookrightarrow ',L_widths_p);
   plot_p.Marker='s';
392
   plot_p.NodeColor = 'k';
393
   plot_p.MarkerSize = 7;
394
   grid on;
395
   xlabel('horizontal_distance(km)')
396
   ylabel('vertical__distance__(km)')
397
   exportgraphics(gcf, 'EH_solution_power_lines.pdf');
398
399
400
401
   %%% Plot of the district heating network - topology & pipe
402
       \hookrightarrow capacities.
403
   figure()
404
   adj_dh=calB_dh*diag(value(C_dh))*calB_dh';
405
   test_dh=graph(adj_dh,'omitselfloops');
406
  C_dh_val=value(C_dh);
407
   C_dh_val_red=C_dh_val(C_dh_val^{\sim}= 0);
408
409
  L_widths_dh=C_dh_val_red/(sum(C_dh_val_red)/length(C_dh_val_red))
410
       \rightarrow +eps;
411 plot_dh=plot(test_dh,'r','LineWidth',1.5,'XData',EH_nodes.x,'
       YData', EH_nodes.y, 'NodeLabel', EH_nodes.neighborhood,'
       → EdgeLabel', 1e-6*C_dh_val_red, 'LineWidth', 1.5*L_widths_dh);
```

```
412 plot_dh.Marker='s';
413 plot_dh.NodeColor = 'k';
414 plot_dh.MarkerSize = 7;
415 grid on;
416 xlabel('horizontal_distance_(km)')
417 ylabel('vertical_distance_(km)')
418
419 exportgraphics(gcf, 'EH_solution_district_heating.pdf');
```

# 3.7 Matlab code for cost model

The code presented here corresponds to the model presented in Section 3.4.

```
function cost = cost_function(x)
1
2
3 % What variables are given
4 nf_lt = 2; % Number of lt factories
  nf_mt = 2; % Number of mt factories
5
6
  alpha = x(1:nf_lt);
7
  beta = x(nf_lt+1:nf_lt+nf_mt);
8
  gamma = x(end);
9
10
11 Demand_per_house = 16; % Demand in GJ
12 Number_of_Houses = 3000;
13 Area = 30; % Area in km^2, not used atm
  D = Number_of_Houses * Demand_per_house; % Total heat demand in
14
      \hookrightarrow GJ
15
  p = 40; % The price of pipeline building per kilometer
16
  t = 1; % The time horizon in years
17
18
  Hlt = 12723 * Number_of_Houses; % Price of all households
19
      \hookrightarrow connected for lt
  Hmt = 12723 * Number_of_Houses; % Price of all households
20
      \hookrightarrow connected for mt
  Hhp = 22687.50 * Number_of_Houses; % Price of all households
^{21}
      \hookrightarrow connected for hp
22
23 Flt = 50000; % Price per LT facility
  Fmt = 500000; % Price per MT facility
^{24}
^{25}
26 Plt = 31.51; % Price per energy (GJ) for lt
27 Pmt = 36.17; % Price per energy for mt
28 Php = 31.51; % Price per energy for hp
^{29}
```

```
maintenance = 0.01; % The maintenance cost in terms of percentage
30
      \hookrightarrow
          of CAPEX
  maintenance_hp = 0; % The maintenance factor for heat pumps
31
32
  \% \ km = p * a^2;
33
34
  % Define model and set constraints
35
  \% delta = 1;
36
37
  %% constraint
38
39
  capacity_lt = 2* ones(1, nf_lt);
40
  capacity_mt = 3 * ones(1, nf_mt);
^{41}
42
43 %% LT
  infracost = 0; % capex
11
  for j = 1:nf_lt
45
       infracost = infracost + (1+maintenance*t)*(j*Hlt + Flt + p*j
46
          \rightarrow ^2);
47
  end
^{48}
49 operationcost = sum(alpha)*D*Plt;
  cost_LT = infracost + operationcost;
50
51
  %% MT
52
  infracost = 0; % capex
53
  for j = 1:nf_mt
54
       infracost = infracost + (1+maintenance*t)*(j*Hmt + Fmt + p*j
55
          \rightarrow ^2);
  end
56
57
  operationcost = sum(beta)*D*Pmt;
58
  cost_MT = infracost + operationcost;
59
60
  %% HP
61
  gamma = 1 - sum(alpha) - sum(beta)
62
63
  infracost = 0; % capex
64
  for j = 1:length(gamma)
65
       infracost = infracost + (1+maintenance_hp*t)*(j*Hhp);
66
  end
67
68
  operationcost = sum(gamma)*D*Php;
69
  cost_HP = infracost + operationcost;
70
71
72 cost = cost_LT + cost_MT + cost_HP;
```

1

```
global nf_lt nf_mt Demand_per_house Number_of_Houses D
     ← capacity_lt capacity_mt plt pmt t Hlt Hmt Hhp Flt Fmt Plt
     ↔ Pmt Php maintenance maintenance_hp HDensity
TableData = zeros(28, 23);
9 time_id = 0;
 price_id = 0;
 for time = 0:10:60
      time_id = time_id+1;
     for price = 30:10:60
         price_id = price_id+1;
15 nf_lt = 1; % Number of lt factories
16 nf_mt = 1; % Number of mt factories
17 alpha_init = ones(1,nf_lt);
18 beta_init = ones(1,nf_mt);
  gamma_init = ones(1,1);
21 Demand_per_house = 16; % Demand in GJ
22 Number_of_Houses = 2416;
23 Area = 30; \% Area in km<sup>2</sup>, not used atm
HDensity = 1/500; % people per Km^2
 D = Number_of_Houses * Demand_per_house; % Total heat demand in
 capacity_lt = 150000* ones(1, nf_lt);
```

```
capacity_mt = 3000000 * ones(1, nf_mt);
28
29
30
  plt = 0.936E6; % The price of pipeline building per kilometer
31
  pmt = 1.736E6; % The price of pipeline building per kilometer
32
  t = time; % The time horizon in years
33
34
  Hlt = 1015213; % * Number_of_Houses; % Price of all households
35
      \hookrightarrow connected for lt
  Hmt = 377505; % * Number_of_Houses; % Price of all households
36
      \hookrightarrow connected for mt
  Hhp = 1305667; % * Number_of_Houses; % Price of all households
37
      \hookrightarrow connected for hp
38
39 Flt = 50000; % Price per LT facility
40 Fmt = 500000; % Price per MT facility
41
42 Plt = 31; % Price per energy (GJ) for lt
```

2 clc: 3 clear; 4 close all;

5

6

8

10

11

12

13

14

19 20

25

26

27

 $\hookrightarrow GJ$ 

```
Php = price; % Price per energy for hp
44
^{45}
  maintenance = 0.02; % The maintenance cost in terms of percentage
46
      \hookrightarrow
          of CAPEX
  maintenance_hp = 0.005; % The maintenance factor for heat pumps
47
48
49 % Cost
_{50} % PS constraints: xa, xb, xq
51 % initial quess, lower & upper bond
52 % init_guess=[alpha_init, beta_init, gamma_init];
  init_guess=[alpha_init.*0, beta_init.*0, gamma_init.*0];
53
54
  lb=[alpha_init.*0, beta_init.*0, gamma_init.*0];
55
  ub=[alpha_init, beta_init, gamma_init];
56
57
  % init_guess=0;
58
  A = [diag(alpha_init), zeros(nf_lt, nf_mt), zeros(nf_lt, 1);
59
      → zeros(nf_mt, nf_lt), diag(beta_init), zeros(nf_mt, 1)];
                                                                         %
          Linear inequality constraint: A*X <= b
      \rightarrow
  bin= [capacity_lt/D, capacity_mt/D]; % Linear inequality
60
      ↔ constraint: A*X <= b
  Aeq = [ones(1, nf_lt), ones(1, nf_mt), 1]; % Linear equality
61
      \hookrightarrow constraint: Aeg * X = beg
              % Linear equality constraint: Aeg*X = beg
  beq = [1];
62
_{63} % A = [7]:
_{64} % bin = [];
65 % Aeq = [];
66 % beq = [];
  nonlcon = [];
                   % A function returning the nonlinear inequality (C
67
      \hookrightarrow (x) <= 0) and equality (Ceq(x)=0) vectors
68
  % Setting Options as Specified by the User: Defining pattern
69
      \hookrightarrow search (hybrid function) options
70 Hybrid_Options = psoptimset(@patternsearch);
  Hybrid_Options = psoptimset(Hybrid_Options,'CompletePoll','on'
71
      \hookrightarrow , . . .
       'TolMesh',1e-10, 'TolFun',1e-10, 'TolX',1e-10,...
72
      'Display', 'iter', 'PlotFcn', {@psplotbestf @psplotmeshsize
73
          \hookrightarrow @psplotfuncount @psplotbestx});
74
75
  % Running the pattern search
76
77
  % Running the pattern search algorithm with start point(1,1)
78
  % [Coefficients] = patternsearch(@cost_function,init_guess,A,bin,
79

→ Aeq, beq, lb, ub, nonlcon, Hybrid_Options);

80 % [Coefficients] = patternsearch(@cost_function,init_guess);
81 [Coefficients] = fmincon(@cost_function, init_guess, A, bin, Aeq, beq,
```

```
\hookrightarrow lb,ub,nonlcon);
82
   Coefficient = sum(Coefficients)
83
   cost = cost_function (Coefficients)
84
85
86
   % Saving
   cost_solution =zeros(1,9);
87
   [cost_solution(1), cost_solution(2), cost_solution(3),...
88
       cost_solution(4), cost_solution(5), cost_solution(6), ...
89
       cost_solution(7), cost_solution(8), cost_solution(9)] =
QU
           \hookrightarrow cost_function_saving (Coefficients);
91
   cost_LT_MT_HP =zeros(1,9);
92
   [cost_LT_MT_HP(1), cost_LT_MT_HP(2), cost_LT_MT_HP(3), ...
93
       cost_LT_MT_HP(4), cost_LT_MT_HP(5), cost_LT_MT_HP(6), ...
94
       cost_LT_MT_HP(7), cost_LT_MT_HP(8), cost_LT_MT_HP(9)] =
95
           \hookrightarrow cost_function_saving ([1 1 1]);
96
97
   %%
98
   %% Plotting
99
  % operationcost_co_LT, operationcost_customer_LT, infcost_LT, ...
100
          operationcost_co_MT, operationcost_customer_MT, infcost_MT,
101
   %
          operationcost_co_HP, operationcost_customer_HP, infcost_HP
   %
102
103
   fig = figure;
104
   legendName ={'CAPEX', 'OPEX_customer', 'OPEX_company'};
105
106
107
   %% cost solution
108
109
   % Categorizing data for Pressure-Temperature
110
   TestNo = 0;
111
   Test = [];operationcost_co=[];operationcost_customer=[];infcost
112
      \rightarrow =[];
113
            operationcost_co(1,1) = cost_LT_MT_HP(1);
114
            operationcost_co(2,1) = cost_LT_MT_HP(4);
115
            operationcost_co(3,1) = cost_LT_MT_HP(7);
116
            operationcost_co(4,1) = cost_solution(1)+ cost_solution
117
                \hookrightarrow (4)+ cost_solution(7);
            operationcost_customer(1,1) = cost_LT_MT_HP(2) +
118
                \hookrightarrow operationcost_co(1,1);
            operationcost_customer(2,1) = cost_LT_MT_HP(5)+
119
                \hookrightarrow operationcost_co(2,1) ;
            operationcost_customer(3,1) = cost_LT_MT_HP(8)+
120
```

```
\hookrightarrow operationcost_co(3,1);
```

```
operationcost_customer(4,1) = cost_solution(2) +
121
                \hookrightarrow cost_solution(5)+ cost_solution(8)+
                \hookrightarrow operationcost_co(4,1) ;
            infcost(1,1) = cost_LT_MT_HP(3)+operationcost_customer
122
                \hookrightarrow (1,1);
            infcost(2,1) = cost_LT_MT_HP(6)+operationcost_customer
123
                \rightarrow (2,1);
            infcost(3,1) = cost_LT_MT_HP(9)+operationcost_customer
124
                \leftrightarrow (3,1);
            infcost(4,1) = cost_solution(3) + cost_solution(6) +
125
                \hookrightarrow cost_solution(9)+operationcost_customer(4,1);
126
127
   %Bar chart
128
       x = [1, 2, 3, 4];
129
       b1 = bar(x, infcost(:,:)/1E6);
130
       b1(1).FaceColor = '#B3E5FC';
131
       hold on
132
133
       b2 = bar(x, operationcost_customer(:,:)/1E6);
134
       b2(1).FaceColor = '#0099FF';
135
       hold on
136
137
       b3 = bar(x, operationcost_co(:,:)/1E6);
138
       b3(1).FaceColor = '#673AB7';
139
          b2(2).FaceColor = '#673AB7';
   %
140
   %
          b2(3). FaceColor = '#FF3300';
141
   %
          b2(4). FaceColor = '#FFCC00';
142
   %
          b2(5). FaceColor = '#00CC00';
143
       hold on
144
145
  legend(legendName,'NumColumns',1 ,'Location','northeast','
146
       \hookrightarrow Fontsize', 12);
  xticks(1:4);
147
148 xticklabels({'LT','MT','HP','OPT'})
149 % xlim([0,110])
  ylim([0,200])
150
  ylabel('\bfCost_(Million_Euros)');
151
   grid on
152
153 newT =['Cost_per_strategy_for_', num2str(t), '_year(s)'];
154 title(newT):
155 set(gca,'FontSize',12)
  saveName = ['District_1_','Year',num2str(time),'_Price', num2str(
156
       \rightarrow price)];
157 saveas(fig, saveName,'jpg')
158 set(fig,'Units','Inches');
159 pos = get(fig, 'Position');
160 set(fig, 'PaperPositionMode', 'Auto', 'PaperUnits', 'Inches', '
```

```
\hookrightarrow PaperSize', [pos(3), pos(4)])
   close(fig)
161
162
   TableData(price_id, 1) = time;
163
   TableData(price_id, 2) = price;
164
   TableData(price_id, 3:11) = cost_solution;
165
   TableData(price_id, 12:20) = cost_LT_MT_HP;
166
   TableData(price_id, 21:23) = Coefficients;
167
168
       end
160
   end
170
171
172
173
174
   %%
175
   function cost = cost_function(x)
176
177
   global nf_lt nf_mt D plt pmt t Hlt Hmt Hhp Flt Fmt Plt Pmt Php
178
       \hookrightarrow maintenance maintenance_hp HDensity Number_of_Houses
179
   % What variables are given
180
181
   alpha = x(1:nf_lt);
182
   beta = x(nf_lt+1:nf_lt+nf_mt);
183
   gamma = x(end);
184
185
186
   \% \ km = p * a^2;
187
188
   % for the identifier function
189
   c1 = 100;
190
   c2 = 0.5;
191
192
   %% LT
193
   infracost = 0; % capex
194
   for j = 1:nf_lt
195
        infracost = infracost + (1+maintenance*t)*(alpha(j)*Hlt +
196
           \hookrightarrow 2*(1/(1+exp(-c1*alpha(j))) - c2) * Flt + plt*alpha(j))
           ↔ *HDensity*Number_of_Houses;
   end
197
198
   operationcost = sum(alpha)*D*t*Plt;
199
   cost_LT = infracost + operationcost;
200
201
202
203 %% MT
204 infracost = 0; % capex
```

```
for j = 1:nf_mt
205
       infracost = infracost + (1+maintenance*t)*(beta(j)*Hmt
206
           \hookrightarrow +2*(1/(1+exp(-c1*beta(j))) - c2) * Fmt + pmt*beta(j))
           ↔ *HDensity*Number_of_Houses;
   end
207
208
   operationcost = sum(beta)*D*t*Pmt;
209
   cost_MT = infracost + operationcost;
210
211
212
   %% HP
213
   % gamma+ sum(alpha) + sum(beta)=1
214
215
   infracost = 0; % capex
216
   for j = 1:length(gamma)
217
       infracost = infracost + (1+maintenance_hp*t)*(gamma(j)*Hhp);
218
   end
219
   operationcost = sum(gamma)*D*t*Php;
220
221
   cost_HP = infracost + operationcost;
222
223
   cost = cost_LT + cost_MT + cost_HP;
224
225
   end
226
227
   function [operationcost_co_LT, operationcost_customer_LT,
228

→ infcost_LT, operationcost_co_MT, operationcost_customer_MT

       \hookrightarrow , infcost_MT, operationcost_co_HP,
       \hookrightarrow operationcost_customer_HP, infcost_HP] =
       \hookrightarrow cost_function_saving (x)
229
   global nf_lt nf_mt D plt pmt t Hlt Hmt Hhp Flt Fmt Plt Pmt Php
230
       ← maintenance maintenance_hp HDensity Number_of_Houses
231
   % What variables are given
232
233
   alpha = x(1:nf_lt);
234
   beta = x(nf_lt+1:nf_lt+nf_mt);
235
   gamma = x(end);
236
237
238
   \% \ km = p * a^2;
239
240
   % for the identifier function
241
  c1 = 100;
242
  c2 = 0.5;
243
244
245 XX LT: operationcost_co_LT, operationcost_customer_LT, infcost_LT
```

```
infracost = 0; % capex
246
   operationcost_co_LT = 0;
247
  for j = 1:nf_lt
248
       operationcost_co_LT = operationcost_co_LT + (maintenance*t)*(
249
           \rightarrow alpha(j)*Hlt + 2*(1/(1+exp(-c1*alpha(j))) - c2) * Flt
           infracost = infracost + (1+maintenance*t)*(alpha(j)*Hlt +
250
           \hookrightarrow 2*(1/(1+exp(-c1*alpha(j))) - c2) * Flt + plt*alpha(j))
           ↔ *HDensity*Number_of_Houses;
   end
251
   infcost_LT = infracost - operationcost_co_LT;
252
253
   operationcost_customer_LT = sum(alpha)*D*t*Plt;
254
255
256
  %% MT
257
  infracost = 0; % capex
258
   operationcost_co_MT = 0;
259
260
   for j = 1:nf_mt
       operationcost_co_MT = operationcost_co_MT + (maintenance*t)*(
261
           \hookrightarrow beta(j)*Hmt +2*(1/(1+exp(-c1*beta(j))) - c2) *
                                                                  Fmt +

→ pmt*beta(j))*HDensity*Number_of_Houses;

       infracost = infracost + (1+maintenance*t)*(beta(j)*Hmt
262
           \rightarrow +2*(1/(1+exp(-c1*beta(j))) - c2) * Fmt + pmt*beta(j))
           ↔ *HDensity*Number_of_Houses;
   end
263
264
   infcost_MT = infracost - operationcost_co_MT;
265
266
   operationcost_customer_MT = sum(beta)*D*t*Pmt;
267
268
269
   %% HP
270
   % gamma+ sum(alpha) + sum(beta)=1
271
272
   infracost = 0; % capex
273
   operationcost_co_HP = 0;
274
   for j = 1:length(gamma)
275
       operationcost_co_HP = operationcost_co_HP + (maintenance_hp*t
276
           \rightarrow )*(gamma(j)*Hhp);
       infracost = infracost + (1+maintenance_hp*t)*(gamma(j)*Hhp);
277
   end
278
279
   infcost_HP = infracost - operationcost_co_HP;
280
281
  operationcost_customer_HP = sum(gamma)*D*t*Php;
282
283
284 end
```

# References

- Cao, Yang et al. (2020). "Capacity Planning of Energy Hub in Multi-Carrier Energy Networks: A Data-Driven Robust Stochastic Programming Approach". In: *IEEE Transactions on Sustainable Energy* 11.1, pp. 3–14. DOI: 10.1109/TSTE.2018. 2878230.
- Expertise Centrum Warmte, ECW (2023). URL: https://www.expertisecentrumwarmte. nl/default.aspx (visited on 01/31/2023).
- Geidl, Martin and Göran Andersson (2007). "Optimal Power Flow of Multiple Energy Carriers". In: *IEEE Transactions on Power Systems* 22.1, pp. 145–155. DOI: 10. 1109/TPWRS.2006.888988.
- Geidl, Martin, Gaudenz Koeppel, et al. (2007). "Energy hubs for the future". In: *IEEE Power and Energy Magazine* 5.1, pp. 24–30. DOI: 10.1109/MPAE.2007.264850.
- Geng, Sijia, Maria Vrakopoulou, and Ian A Hiskens (2020). "Optimal capacity design and operation of energy hub systems". In: *Proceedings of the IEEE* 108.9, pp. 1475– 1495.
- Leefongeving, PBL Planbureau voor de (2023). *PBL*. URL: https://www.pbl.nl/ (visited on 01/31/2023).
- Li, Rui et al. (2018). "Participation of an energy hub in electricity and heat distribution markets: An MPEC approach". In: *IEEE Transactions on Smart Grid* 10.4, pp. 3641–3653.
- Lofberg, Johan (2004). "YALMIP: A toolbox for modeling and optimization in MAT-LAB". In: 2004 IEEE international conference on robotics and automation (IEEE Cat. No. 04CH37508). IEEE, pp. 284–289.
- Omgeving, Energietransitie Gebouwde (2023). Energietransitie Gebouwde Omgeving. URL: https://www.expertisecentrumwarmte.nl/default.aspx (visited on 01/31/2023).
- PBL, Thema websites (2020). URL: https://themasites.pbl.nl/leidraadwarmte/2020/ (visited on 01/31/2023).
- SWI (2023). STUDY GROUP MATHEMATICS WITH INDUSTRY. URL: https:// www.swi-wiskunde.nl/swi2023/problems/the-heat-transition-mathematicalmodels-and-optimization/ (visited on 02/05/2023).

# Chapter 4

# Stereo 3D cloud motion from tandem satellites

Yang Zhou<sup>1</sup> (yz3259@bath.ac.uk) Sophie Abrahams<sup>2</sup> (sophie.abrahams@maths.ox.ac.uk), Rik Verbiest<sup>4</sup> (r.verbiest@rug.nl) Maddalena Donà<sup>4</sup> (m.dona@rug.nl), Francisco Richter<sup>3</sup>, (richtf@usi.ch)

### Affiliations:

- <sup>1</sup> University of Bath
- <sup>2</sup> University of Oxford
- $^{3}$ Università della Svizzera Italiana
- <sup>4</sup> University of Groningen

Study Group: SWI2023, University of Groningen, Feb 2023

**Partner**: The European Space Agency (ESA) and The Royal Netherlands Meteorological Institute (KNMI)

Presenter: Alexandre Payez (KNMI), Björn Rommen (ESA)

**Team members**: Bas van't Hof, VORtech BV; Dhruva Kelkar; Francisco Richter, Università della Svizzera Italiana;Maddalena Donà, University of Groningen;Oliver Nagy;Rik Verbiest, University of Groningen;Sophie Abrahams, University of Oxford;Yang Zhou, University of Bath

**Abstract** In this study, we present an innovative software framework developed during the SWI 2023 Study Group for the accurate geolocation and 3D reconstruction of cloud observations. This project, a joint proposal by the European Space Agency (ESA) and the Royal Netherlands Meteorological Institute (KNMI), tackles the challenge of extracting relevant information from cloud observation data and transforming it into actionable insights.

The framework is primarily implemented in Python3, leveraging basic libraries such as numpy, matplotlib, scipy, and others. It offers a robust pipeline for extracting relevant arrays from netCDF formatted data, such as cloud observation and location details, and subsequently updating these back into the original data.

Our software processes, extracts, and leverages data in three main branches: reconstruction, motion extraction, and shape analysis. Within these branches, various procedures like 3D reconstruction, bulk motion extraction, shape analysis, cloud selection and smoothing, and optical flow are executed. The focus is on the accurate extraction of real cloud latitude and longitude from observed positions, considering the true height of the cloud surface.

KEYWORDS: 3D reconstruction, optical flow, shape analysis, Fourier bulk motion detection, cloud observation

# 4.1 Introduction

The Harmony mission was selected by the European Space Agency to be implemented as the 10th Earth Explorer mission. This mission was mainly implemented to resolve physical processes in the marine atmospheric boundary layer at high resolution, thus aiming for a better understanding of the Earth System. One of the main scientific objectives of this mission is to advance the knowledge of cloud processes, as this is required in weather and climate modelling. The two Harmony satellites, both carrying a multiview thermal-infrared instrument, supply images of the same scene taken approximately at a time difference of 100 seconds. By exploiting the stereo views of these multiview cameras, it is possible to extract 3D information, thus providing cloud motion in particular. For a detailed problem description and the used data, we refer to Payez 2023.

First, assuming that the cloud height is known, we extracted the real cloud latitude and longitude from the apparent latitude and longitude given by the data in section 4.3.1. The reconstruction of 3D images is discussed in section 4.3.2. The information about the true locations of the pixels was used to develop an algorithm for the calculation of cloud motion. This is described in section 4.3.3. The report ends with a brief conclusion.

# 4.2 Framework Implementation

During this SWI 2023 Study Group, our team has developed software for tackling the problem described above. Codes and data can be found here on Payez 2023. Our code includes data handling methods in jupyter notebook files and .py files.

The framework was implemented mainly in Python3 using basic Python libraries numpy, matplotlib, scipy, xarray, datetime, glob, dill and ctypes. Parts of the program use implemented codes written in C and needed to be adapted for use in Python.

Our code provides the pipeline to extract relevant arrays (cloud observation and locations etc.) from netCDF formats. Those arrays are input data for our models and go through different calculation stages depending on the purpose of calculation. Codes are provided to have the real longitude and latitude locations updated back in the original netCDF data.

This automatic data-extraction and update pipeline was developed and tested on Windows and Linux systems, whereas it might have problems on macOS. Additionally, details regarding the computation branches will be discussed in the next section 4.3.

# 4.3 Methodology

This section describes the pipeline of the software (excluding data processing and cleaning which are described in subsection 4.2.) developed during the Study Group. This includes 3 main branches (see flowchart 4.1): reconstruction, motion extraction, and shape analysis:



Figure 4.1: Flowchart of our software pipeline. Structures marked with black font are finished whereas structures with light gray font are in progress by the time of finishing this paper.

- 1 data processing
- 2 location extraction
- 3a branch a: 3D reconstruction
- 3b branch b: motion extraction: bulk motion
  - 4c shape analysis
  - 4b cloud selection and smoothing
  - 4c Optical flow

### 4.3.1 Extract positions from cloud observations

This subsection describes the method used to extract real cloud latitude and longitude from apparent cloud latitude, longitude with given true height.

As demonstrated in Figure 4.2, each camera on a satellite observes the target block and collects their observed locations of cloud surfaces. The angle formed from the camera to object along the vertical direction is  $\theta$  and the angle formed from camera to the North is  $\phi$ . The angles  $\phi$  and  $\theta$  are given:  $\theta = 0^{\circ}$  means 'straight up';  $\theta = 180^{\circ}$ means 'straight down';  $\phi = 0^{\circ}$  means 'straight ahead to the North,  $\phi = 90^{\circ}$  means 'to the East'.

A correction had to be applied to the data, since the definition of  $\phi$  is different to the documents found inside the .nc data. In the data, the original  $\phi$  is defined to be the angle formed from the observed object to the camera and to the path orbit where the satellite is travelling. The orbit forms a 13° with the North. However when we tested our model, we noticed the data has been processed to have  $\phi$  measured against the North of the orbit. This is illustrated in Figure 4.3.



cloud surface

Figure 4.2: 3D view of camera to cloud:  $\theta$ , north to cloud:  $\phi$ .



Figure 4.3: Two observed locations are shown (orange and blue dots), which are the same cloud surface with different viewing angles. When we apply our model to extract the real location of the clouds, we noticed that when  $\phi=13^{\circ}$  as described in the documentation, the projected locations are not close to each other. However  $\phi=0^{\circ}$ , gives correct results.

The extraction can be processed by considering the 3D vector diagram in Figure 4.4. Each camera forms a triangle of one point on the cloud surface. If we take the same point on the cloud at time  $t_1$ , the observed positions from A3 and B1 will be different due to the projection. The observed locations are marked as A3 cloud and B1 cloud. The true height of the observed point on the cloud is given. Therefore, to find the true position we need to find the shifted vectors from B1 cloud and A3 cloud, then  $(lon, lat) - (\delta x, \delta y)$  will give us the true position  $(lon_*, lat_*)$ , where (lon, lat) is the observed position shifted vector.



Figure 4.4: Simultaneous observations from camera A3 and B1 of the same cloud surfaces from different viewing angles.

This calculation only requires one observation point at each time to find the exact location, due to the fact that we know the true height of each cloud-surface point. However simultaneous observations can provides 3D reconstructions of the same cloud. The reconstruction is described in subsection 4.3.2.

As shown in Figure 4.5, each measurement provides  $\phi$ ,  $\theta$  and h, as well as the latitude lat and longitude lon. The measurement ( $\phi$ ,  $\theta$ , h, lat, lon) therefore corresponds to the actual location at height h, and pointing to longitude direction and latitude direction:

$$(\delta x, \delta y) := (\sin\left(\phi - 13^{\circ}\right), \cos(\phi - 13^{\circ})) \frac{h}{\tan(\theta - 90^{\circ})}$$

So we have to go  $(\delta x, \delta y)$  to the (west, north), starting from (lat,long). This gives the equation of the true location (lon<sub>\*</sub>, lat<sub>\*</sub>, h):

$$\begin{aligned} (\texttt{lon}_*,\texttt{lat}_*) &= \left(\texttt{lon} - \frac{180^o \delta x}{\pi R \cos(\texttt{lat})},\texttt{lat} - \frac{180^o \delta y}{\pi R}\right) \\ &= \left(\texttt{lon} - \frac{180^o \sin(\phi - 13^o)h}{\pi R \tan(\theta - 90^o)},\texttt{lat} - \frac{180^o \cos(\phi - 13^o)h}{\pi R \cos(\texttt{lat}) \tan(\theta - 90^o)}\right) \end{aligned}$$

where  $R = 6.371 \cdot 10^6 [m]$  is the radius of the Earth. Note that the equation assumes a spherical shape of the Earth.



Figure 4.5: Side view (left): this represents the view parallel to the satellite orbit. The vertical direction is the one perpendicular to ground and pointing upwards. Theta ( $\theta$ ) is the angle formed by the camera, pointing to the top of the observed object. The red dot is the real location of a cloud surface. h is the height of the cloud with respect to the sea level.  $h/\tan(\theta - 90)$  is the magnitude of the projection shift vector along longitude and latitude directions. Top view (right): the vertical line points to the North and the direction formed by  $13^{\circ}$  is the direction of the stereo-satellite's orbit. The red dot is the true location of the cloud surface (lon\*, lat\*). The observed location is (lon, lat).

### 4.3.2 Cloud reconstruction

Subsection 4.3.1 explained how to calculate the true location on one cloud surface, and this subsection continues to use the above algorithm to reconstruct from each observation image and from simultaneous images.



Figure 4.6: (The right) Image from camera A2. (The left) Image is the reconstruction from the right image, where empty pixels mean that there is no data for those locations.

As demonstrated in Figure 4.6, cloud surfaces observation from A2 camera (the right image) are processed and then produce the left side graph. These two are 3D images of positions and heights of cloud surfaces. There are some shifted pixels on the left image and other pixels are left out. This is because the A2 camera, while taking this image, can only observe some sides of the clouds. Indeed the cloud, seen by a fixed angle, hides areas behind or under it. Therefore when we have two simultaneous

images with different viewing angles, we can combine them after extracting the real positions and using different information contained in the two simultaneous images to fill in missing pixels and composite cloud images. Figure 4.7 gives an example of the reconstruction with images from A2 and B2. By combining these two measurements, we can fill in a big percentage of missing pixels.



Figure 4.7: (Top left) Image is the reconstruction from A2; (Top right) Image is the reconstruction from B2. (Bottom) The reconstruction by combining the reconstructed images from A2 and B2 has filled in missing pixels. Note that empty pixels on all of these 3 images mean that there is no data for those locations.

Data from tandem observations (A3-B1, A2-B2, A1-B3) can be combined, since each pair of observations is taken at different angles at the same time step. In general, when there are multiple height measurements for the same location, it is possible to reconstruct 3D shapes which will give a better results than using only 2 measurements. In our project we only have 3 pair measurements which are simultaneous at 3 different time steps. We did not implement reconstruction with more than two images. However, the methods remains the same for more than 2 measurements.

### 4.3.3 Cloud motion - bulk motion

After the true locations have been extracted from the original measurements, the information can be used to calculate cloud motion using the 6 different time steps. The unit for a time step used here is one minute.

The first step is to calculate the bulk motion using Fourier transform Arking, Lo, and Rosenfeld 1978.

Let us define:

 $f_1(x, y) :=$  height function of measurement 1,

 $f_2(x, y) :=$  height function of measurement 2,

 $\Delta x = \text{horizontal shift},$ 

 $\Delta y = \text{vertical shift.}$ 

This means that  $f_2(x, y) \approx f_1(x - \Delta x, y - \Delta y)$ . Let **F** denote the fast Fourier transform:

$$F_1(u, v) = \mathbf{F}(f_1)$$
$$F_2(u, v) = \mathbf{F}(f_2)$$

Then we have:

$$\delta(x + \Delta x, y + \Delta y) = \mathbf{F^{-1}} \left( \frac{F_1(u, v) F_2^*(u, v)}{|F_1(u, v) F_2^*(u, v)|} \right)$$

This gives us the bulk displacement of a shape with a time step  $\Delta t$  as demonstrated in Figure 4.8:

An intuitive way to capture the velocity of each block of clouds is to smooth the image and calculate the bulk motion of each captured block of clouds. This is implemented in the code however the result still requires tuning. Additionally, the calculation is carried out by looping across each measurement in the array which is time consuming. This process will be repeated in calculating both vertical and horizontal velocities. One way to optimise is by finding cloud clusters. We implemented one algorithm to first capture the boundary of each block of clouds using a bounding box as demonstrated in Figure 4.9. Then this algorithm provides the location of interested blocks of clouds in addition to the velocity calculations and other improvement algorithms described below.

However this method has two drawbacks:

- This method only computes a "bulk velocity".
- It is therefore not precise enough for computing accurate vertical velocities.

Thus, we need a more precise matching algorithm to calculate accurate velocities. In scenes where we have two different blocks of clouds moving at different velocities, we need to "zoom in" on these blocks and run the matching algorithm separately in


Figure 4.8: Bulk movement of the entire measurements.



Figure 4.9: Cloud clustering by bounding box.

both frames to get more accurate information regarding the displacement and hence obtain accurate velocities. We can extract information from images by subtraction:

$$f_2(x,y) - f_1(x - \Delta x, y - \Delta y).$$

This detects unmatched points and allows to "zoom in" on the clouds with a "large" fraction of unmatched points and recalculate a refined match. If we want to take into account both rotational and scaling effects we can compute the rotational matching by the following steps. Assume there is rotational around the angle  $\theta$ :

$$f_2(x,y) \approx f_1(x\cos\theta_0 + y\sin\theta_0, -x\sin\theta_0 + y\cos\theta_0)$$

We do a change of coordinates as in Table 4.1.

Cartesian coordinate	Polar coordinate		
(x,y)	$((x^2+y^2)^{1/2}, \tan^{-1}(y/x))$		
$(r\cos\theta, r\sin\theta)$	(r, heta)		

Then  $f_2(r,\theta) \approx f_1(r,\theta-\theta_0)$ , which gives us the rotational matching.

To compute scaling matching, let us assume that there is a horizontal stretch of factor a in  $\mathbf{e}_x$  and a vertical stretch of factor b in  $\mathbf{e}_y$  direction (see Graph 4.2). This means that:

$$f_2(x,y) \approx f_1(\frac{x}{a},\frac{x}{b}).$$

We perform a change of coordinates as in Table 4.2.

Cartesian	Log Cartesian
(x,y)	$(\log x, \log y)$
$(e^c, e^d)$	(c,d)

Table 4.2: Change of Coordinates from Cartesian to Log Cartesian.

We have  $f_2(c, d) = f_1(c - \log a, c - \log b)$ . The implementation of this scaling and rotational matching algorithm is still left for future work.

#### 4.3.4 Statistical shape analysis

Statistical shape analysis (SSA) is a general term for a range of methods used to determine differences in the shape of objects in two images. SSA is widely used in medical imaging and could be used here to determine the movement or deformation in clouds between time steps. This method could be used alone or in conjunction with other methods – for example, bulk motion could be calculated and removed, after which SSA could be applied to determine localised deformation or motion.

#### Registration

In particular, "registration", a branch of SSA, attempts to find a deformation field or "velocity field" that most closely maps one image to another. We will here refer to a deformation field but in the application of cloud movement, this can be seen as the velocity of the clouds between two images taken at different timesteps. A key advantage of registration, as opposed to other methods such as optical flow is that it can be implemented without the need for feature detection. Python software Deformetrica is one option for implementing registration. The Deformatrica website states: 'Deformetrica is a software for the statistical analysis of 2D and 3D shape data. It essentially computes deformations of the 2D or 3D ambient space, which, in turn, warp any object embedded in this space, whether this object is a curve, a surface, a structured or unstructured set of points, an image, or any combination of them' Durrleman, Bône, et al. n.d. Registration can also be implemented manually.

In brief, registration takes the following steps. Details of the process can be found in Bône, Louis, et al. 2018 and Durrleman, Prastawa, et al. 2014.

1. Input data of template image, T, and target image, S. The image can be in a variety of forms including a greyscale image or a pointcloud (i.e. a list of vector points in 3D space) Bône 2021.

2. Shapes in the image are reconstructed as a discretized surface with cell centres,  $c_k$  and normals  $n_k$ .

3. The template image is deformed by a deformation field  $\Phi(T)$ . A cost function is defined to be dependent on the difference between  $\Phi(T)$  and S. A key feature is that the cost function is defined between every point in  $\Phi(T)$  and every point in S, hence points on the two images do not need to correspond 1-to-1. The cost function will be of the form

$$C = \frac{d(\Phi(T), S)^2}{\sigma^2} + R(\mu),$$

where  $d(\Phi(T), S)$  is the distance between the deformed image and the target image,  $R(q(\mu))$  is a regularisation term that acts to minimise the magnitude of the deformation vector at points  $\mu$ , with  $\mu$  being chosen 'test' or 'control' points across the domain. In the literature, the regularisation term refers to minimising the 'kinetic energy' of the system Bône, Louis, et al. 2018 and Durrleman, Prastawa, et al. 2014. The term  $\sigma$  is chosen to select the relative weighting of the distance and regularisation terms.

4. Optimisation methods are applied (e.g. gradient descent) to find  $\Phi$  that minimises C and hence yields the optimal deformation field. Deformetrica has three different optimisation methods Bône and Martin 2021.

#### Custom algorithm

To find the value  $V_{i,j,k}(t_0)$ 

which solves the minimization problem

$$\min V_{i,j,k}(t) | [Img(t + \Delta t)]_{i,j,k} - [Img(t)]_{i',j',k'} |$$

we have developed a genetic-type algorithm to find the optimal 'velocity' of pixels to match the image of the next timestep. Img is the realization of the cloud at time t. Values in Img correspond to the intensity of the pixel. See 2 for algorithm details; in Figure 4.10 it is possible to see the algorithm's performance in a simulation.

#### Algorithm 2 Cloud Velocity Estimation

**Require:** Two images: Img(t) and  $Img(t + \Delta t)$  **Ensure:** Velocity  $V_{i,j,k}(t)$ 1: Initialize population for  $V_{i,j,k}(t)$ 2: **while** termination criteria not met **do** 3: **for** each individual in the population **do** 4:  $V_{i,j,k}^{\text{new}}(t) \leftarrow \text{Mutation and crossover operations on individual pixels}$ 5: Calculate fitness  $f(V_{i,j,k}^{\text{new}}(t)) = |[Img(t + \Delta t)]_{i,j,k} - [Img(t)]_{i',j',k'}|$ 6: **end for** 7:  $V_{i,j,k}(t) \leftarrow \text{Individual with the minimum fitness value}$ 8: **end while** 



Figure 4.10: As a proof-of-concept we performed a simulation study with the goal to match a simulated picture with 8000 pixels. The simulation reached convergence after 500000 iterations. The results of the simulation showed that most of the picture was accurately matched, suggesting that the simulation was successful in its image processing or comparison task.

### 4.3.5 Optical Flow

Optical Flow (see OpenCV n.d.) is a method used to estimate the motion of an object, based on comparing pixels of two image frames taken at times t and  $t+\Delta t$ . We assume that:

1. The pixel intensities of an object do not change between consecutive frames.

2. Neighbouring pixels have similar motion.

Let us indicate by I(x, y, t) the intensity of a pixel (x, y) at time t, that represents a specific part of an object. After the time step  $\Delta t$ , it moves a distance  $(\Delta x, \Delta y)$  and, by the first assumption, we have

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t).$$

$$(4.1)$$

By expanding the intensity  $I(x + \Delta x, y + \Delta y, t + \Delta t)$  on the right hand side of (4.1) with Taylor series around (x, y, t) and neglecting second order terms, it follows that

$$\frac{\partial I}{\partial x}\frac{\Delta x}{\Delta t} + \frac{\partial I}{\partial y}\frac{\Delta y}{\Delta t} + \frac{\partial I}{\partial t} = 0.$$

This leads to what is called the Optical Flow equation, in which the unknowns are the velocity components  $v_x$  and  $v_y$ .

OpenCV provides an algorithm to find the dense Optical Flow, which computes the Optical Flow for all the points in the frame, based on Gunnar Farneback's algorithm. For our purposes, Optical Flow was used to obtain the velocity field of a cloud by comparing two pictures taken from the satellites at two consecutive time steps, from the same angle.

For example, using input images A5 and B1 (see Figure 4.11), the algorithm yields the motion detection (see Figure 4.12), which is not yet fully satisfactory and needs to be refined in future work.



Figure 4.11: Images taken from the same angle, at two different time steps.

## 4.4 Conclusion

In this paper, we presented an innovative software framework for the accurate geolocation and 3D reconstruction of cloud observations which was developed during the SWI 2023 Study Group. First, we developed a method to extract real cloud latitude and real cloud longitude from apparent cloud latitude and apparent cloud longitude



Figure 4.12: Output of the Optical Flow algorithm with images A5 and B1 as input.

with given true height. Hereby, we discovered an inconsistency in the data, namely that the definition of the angle formed from camera to north, denoted by  $\phi$  in this report, did not agree with the actual data. The method succeeded in extracting the real cloud latitude and longitude. By using this method and combining two simultaneous images with different viewing angles, we were able to fill in a big percentage of missing pixels and to reconstruct 3D images.

Next, the information about the true locations of the pixels was used to calculate cloud motion. The bulk motion of the clouds was calculated using the fast Fourier transform. This method only computes a bulk velocity and is therefore not precise enough for computing accurate vertical velocities. Therefore, we described a more precise matching algorithm by taking into account rotational shifts and scaling effects. However, the algorithm was not implemented by the time this paper was written. Further, we developed an algorithm based on Statistical Shape Analysis (SSA) to determine differences in the shape of objects in two images. To be more precise, a genetic-type algorithm was used to find the optimal velocity of pixels to be matched in two images. A simulation study of a test image showed that most of the image was accurately matched. Finally, we tested the optical flow algorithm to obtain the velocity field of a cloud by comparing two pictures taken from the satellites at two consecutive time steps, from the same angle. However, the results of this approach were not yet satisfactory.

#### 4.4.1 Limitation

This study has potential limitations. First of all, we used simulated data and not real cloud data. In our method to calculate the real latitude and longitude, we assumed that the true height is known. However, in real observations, it is not easy to calculate the true height. Further, only certain types of clouds are presented in this report and it is not clear whether or not the accuracy of the algorithms depends on the type of cloud.

Next, there is room for improvement of the 3D cloud reconstruction method. From some angles it might be possible to see multiple height readings for the same geographic coordinates. We only keep the data points with maximal height in this study. Theoretically we could keep the other values as well and obtain a better reconstruction of the 3D shape of the cloud.

Finally, the algorithm for the velocity extraction also has its limitations. First of all, only the bulk velocities are calculated accurately. Secondly, the vertical velocity calculations have large errors. Next, the cluster clouds only considered boundaries, further classification algorithms are required to improve the accuracy. Lastly, the scaling and rotational shifting algorithms have not been implemented, yet.

## References

- Arking, Albert, Robert C Lo, and Azriel Rosenfeld (1978). "A Fourier approach to cloud motion estimation". In: Journal of Applied Meteorology (1962-1982), pp. 735–744.
- Bône, A. (2021). Deformetrica GitLab, 3.1\_available\_objects. https://gitlab. com/icm-institute/aramislab/deformetrica/-/wikis/3\_user\_manual/3.1\_ available\_objects. Online; accessed: 2023-04-21.
- Bône, A., M. Louis, et al. (2018). "Deformetrica 4: An Open-Source Software for Statistical Shape Analysis". In: *International Workshop on Shape in Medical Imaging*. Ed. by M. Reuter et al. Vol. 11167. Lecture Notes in Computer Science. Springer, Cham, pp. 3–13.
- Bône, A. and B. Martin (2021). Deformetrica GitLab, 3.4\_optimization\_parameters\_xml\_file. https://gitlab.com/icm-institute/aramislab/deformetrica/-/wikis/3\_ user\_manual/3.4\_optimization\_parameters\_xml\_file. Online; accessed: 2023-04-21.
- Durrleman, S., A. Bône, et al. (n.d.). Deformetrica. https://www.deformetrica. org/. Online; accessed: 2023-04-21.
- Durrleman, S., M. Prastawa, et al. (2014). "Morphometry of anatomical shape complexes with dense deformations and sparse parameters". In: *NeuroImage* 101, pp. 35–49.

Payez, Alexandre (Jan. 2023). Files, tutorial, and technical documents prepared for SWI 2023. DOI: 10.5281/zenodo.8154740. URL: https://doi.org/10.5281/ zenodo.8154740.

OpenCV (n.d.).

## Chapter 5

# Top Dutch Solar Racing A novel strategy model approach for the Bridgestone World Solar Challenge

JEREMY BUDD<sup>1</sup>, RICARDO ENGUIÇA<sup>2</sup>, YITAEK KIM<sup>3</sup>, JULIAN KOELLERMEIER<sup>4</sup>, SIMON VAN MOURIK<sup>5</sup>, ARIJIT SARKAR<sup>4</sup>, FILIPA SOARES<sup>2</sup>, NELE THOMSEN<sup>6</sup>

Abstract This project considers the derivation of strategies for a solar car race through Australia, posed by TopDutch Solar Racing. The task is to optimize the arrival time of an electric car powered by solar energy by changing the velocity along the track taking into account the characteristics of the car, external influences such as wind and sun, as well as constraints like the maximal and minimal battery charge. To address this challenge, the problem is modeled and four different optimization strategies are devised, focusing on different time horizons and methods. We explain a databased long-term strategy, an optimisation-based long-term strategy, a dynamic programming long-term strategies require more work, the results of the long-term strategies give very realistic results and are powerful tools to increase the winning chance for the TopDutch Solar Racing team. We outline several possibilities to improve the

<sup>&</sup>lt;sup>1</sup>Universität Bonn

<sup>&</sup>lt;sup>2</sup>Instituto Superior de Engenharia de Lisboa & CEMAT-CIÊNCIAS

<sup>&</sup>lt;sup>3</sup>University of Southern Denmark

 $<sup>^5\</sup>mathrm{Wageningen}$  University & Research

<sup>&</sup>lt;sup>6</sup>Katholieke Universiteit Leuven

models based on additional data, modeling and methods.

KEYWORDS: optimization, long-term, short-term

## 5.1 Problem Formulation

In this report, we aim to optimize the velocity of a solar car so as to finish a long race as fast as possible and without running out of power as shown in Fig. 5.1. It is important to consider the minimum charge of a battery and the power consumption for the whole racing period. To this end, we use the comprehensive information of forecasts provided by the previously collected data, which can be used as inputs for a high-level planner to generate the optimal velocity. Simultaneously, we also need to take into account unexpected dynamic disturbances such as cloud and wind changes during racing. To address these issues, we leverage Model Predictive Control (MPC) method and take the approach with a long-term planner and short-term ones proposed by Mocking 2006.



Figure 5.1: Road map for solar car racing in Australia.

#### 5.1.1 Details of the race and available data

The Bridgestone World Solar Challenge (BWSC) is an international race for solar power cars of 3020 km, through the Australian outback. The event takes place in October, and it is held every two years (since 1987), though last edition took place in 2019 (due to the pandemic restrictions), with the participation of 53 teams, from 24 different countries.

The objective is, naturally, to cover the distance in the shortest time possible, but several restrictions are imposed to the participating teams, namely:

- the battery, with the capacity of 5 kWh, can start with full charge, but cannot be charged during the race by any other means than using solar energy;
- the car must be less than 5 m long and 2.2 m wide;
- the maximum amount of solar panels is 4 m<sup>2</sup>;
- as the challenge utilises public roads, the cars have to adhere to the normal traffic regulations;
- driving time is between 8:00 and 17:00 and in order to select a suitable place for the overnight stop (alongside the highway) it is possible to extend the driving period for a maximum of 10 minutes, for which the extra driving time will be compensated by a starting time delay the next day;
- there are 9 mandatory control stops (also called checkpoints) of 30 minutes along the race, which we will call checkpoints. These stops must occur during the driving time and their locations are at the following kilometers:

0, 332, 588, 987, 1210, 1493, 1766, 2179, 2432, 2720.

Each team has a convoy on the road that provides insight information so that the team can improve their strategy (for example to have solar and wind information from a location in advance, to anticipate traffic delays and slow drivers, etc).

The road is not very bumpy overall, but the influences of slopes will be taken into account in our model. The track has a cumulative elevation gain of 18000 m, which corresponds to an average 0.6% of inclination. The total elevation loss is of the same magnitude. Analysing the road every kilometer, we have a maximum slope of 4% and about 100 km above 1%.

The solar data from several years in Darwin, Alice Springs and Adelaide (corresponding to the start, middle and end of the race) is available, and we used the usual profile of solar energy for the month of October in our long-term strategy data-based model. Wind data is harder to analyse and systematize, so we opted to consider this factor in our model in a simpler way, considering a random daily impact percentage of the wind on the power consumption.

Some data regarding the power consumption of the vehicle with respect to velocity was available from the problem owner and is depicted in Fig. 5.2. From this data, the power consumption depending on the velocity v is modeled by the trendline

$$0.17v^2 - 6.6v + 252.95.$$

However, no acceleration data is provided.

## 5.2 Optimization strategies

To address the problem, we propose a high-level decision framework, which consists of two optimization parts as illustrated in Fig. 5.3. Long-term strategy optimization



Figure 5.2: Consumption vs velocity Geling 2021, Figure 12. Trendline:  $0.17v^2 - 6.6v + 252.95$ .

can provide the desired velocity profile and the expected power profile, for example when given a desired "winning time" based on the forecast inputs. The horizon of long-term prediction is for the whole racing period. Afterwards, the short-term strategy optimizes the final desired velocity to not only track the desired velocity profile from the long-term strategy, but also save power consumption during racing. The prediction frequency depends on the preference of a user.

We give four different strategies: two examples for long-term strategies include a data-based variant and an optimisation-based variant. In addition we provide an example for an optimisation-based short-term strategy and a short-term strategy based on dynamic programming.

#### 5.2.1 Data-based long-term strategy

We first describe an approach to estimate an optimal velocity for the whole race more precisely, for the remainder of the race from any point during its unfolding based on the data that is assumed to be available, namely, the characteristics of the vehicle, the route, and the weather forecast. Consequently, the estimates achieved can be improved with more accurate information, and the algorithm is designed so that every update is automatically taken into account.



Figure 5.3: The proposed framework architecture for solar car racing including a long-term strategy and a short-term strategy.

#### The data

With respect to the vehicle, this approach resorts to the data regarding consumption vs velocity of the solar vehicle that took part in the BWSC 2019 (cf. Fig. 5.2). Using the trend line determined by these data, which yields  $P(v) = 0.17v^2 - 6.6v + 252.95$ , where v is the velocity (in km/h) and P the power consumption per hour (in W), we thus have an estimate of the power consumption (converted to kW for simplicity) of the vehicle at constant velocity for  $v \in V = \{60, 61, \ldots, 95\}$ . At the start of the race, the vehicle battery is at its full capacity of 5 kWh.

Regarding the route, we consider the checkpoints—distance of each checkpoint to the start the race (cf. Subsection 5.1.1), length of the control stops (30 minutes) and the possibility of tilting the solar panels with which the car is equipped in order to achieve an optimal charging rate—and the incline of the road, given every 14.66 m. The extra charging efficiency during stops is considered to be of 20%, based both on the information provided by TDSR and by the Solcast Solar API (https://solcast.com/).

Finally, the data concerning the weather conditions, namely the sun and the wind, is given as vectors,  $\mathbf{s}$  and  $\mathbf{w}$ , respectively. Based on the weather forecast, each coordinate of  $\mathbf{s}$  represents the total energy produced by the solar panels in each day of the race and each coordinate of  $\mathbf{w}$  represents the daily average of the impact factor on the consumption due to the wind.

As with the data concerning consumption,  $\mathbf{s}$  and  $\mathbf{w}$  should also be tuned in view of the real performance of the new vehicle for the particular race.

In our simulations, we considered

$$\mathbf{s} = [5.5, 6.0, 5.8, 6.0, 5.7, 5.0, 5.5, 5.5],$$
$$\mathbf{w} = [0.01, 0.02, -0.01, 0.005, 0.03, 0.01, 0.02, 0.0]$$

so the only day where the wind had a decreasing effect on the consumption is the third day, reducing the overall consumption by 1%, while the solar production for that same day was 5.8 kWh. The considered values of the vector  $\mathbf{s}$  are based on historic data for solar power production during October, where the average production with horizontal panels is about 6.5 kWh. The average of the values used are below this value, to avoid over optimistic estimates, but we also provide conclusions for sunnier scenarios (with averages of 6.4 kWh and 7 kWh).

#### The model

The algorithm partitions the race into 30 minute time units, with  $t_u = 0$  corresponding to the interval between 08:00 am and 08:30 am of the first day of the race,  $t_u = 17$ corresponding to the interval between 16:30 and 17:00 am of the first day of the race, and so on. The optimal velocity for the remaining of the race may thus be estimated at every 30 minutes during the race.

The core of the algorithm relies in a procedure that, given:

- the distance and hours (including the time spent at checkpoints) elapsed from the start of the race,
- the number of checkpoints fulfilled, and
- the current battery,

computes, for every velocity in V and at each subsequent time unit:

- the position,
- the battery level,
- the power demand, and
- the power available.

Notice that the distance elapsed alone, when coinciding with the location of a checkpoint, does not provide information as to that checkpoint having already been completed or not, and so, as to whether the vehicle is about to start moving or, on the contrary, will have to stop for 30 minutes. For this reason, the number of checkpoints fulfilled is an information that must be provided to the algorithm. Evidently, the optimal velocity is the highest velocity for which the power demand does not surpass the power available.

To accomplish the necessary computations, for each  $v \in V$ , we begin by initializing the following variables:

- the position, *p*, initialized as the distance elapsed;
- the battery level, b, initialized as the current battery;
- the power demand,  $P_d$ , initialized as 0;
- the power available,  $P_a$ , also initialized as the current battery, and
- the time unit,  $t_u$ , initialized as twice the hours elapsed.

Then, for each time unit starting at  $t_u$ , we determine if the vehicle is moving or will fulfill a checkpoint. (Without loss of generality, we will consider that the checkpoint is made in a single time unit, the one where the car reaches the checkpoint.) In case the car is moving, the updates are:

- the position is updated to p + 0.5v;
- the power demand is updated to

$$P_d \to P_d + \underbrace{0.5P(v) \times (1 + w_{t_u}) \times (1 + ri_{t_u})}_{\Delta P_d},$$

where  $w_{t_u}$  is the appropriate coordinate of **w** and  $ri_{t_u}$  the road incline component, computed as the the sum of the elevation gain (in  $t_u$ ) multiplied by 12, together with the elevation loss (in  $t_u$ ) multiplied by 6, all divided by the kilometers made in  $t_u$  (this way, positive slopes are twice as influential as negative slopes and the range of the impact of the road incline on the consumption yields realistic values according to the BWSC 2019 data); the power available, by adding the amount of solar power produced during the time unit ( $\Delta Pa$ );

• the battery status is updated to

$$b \to \min\{b - \Delta P_d + \Delta P_a, 5\}$$

The solar power produced in a given time unit is the corresponding fraction of the total solar power produced that day. Fig. 5.4 shows how, on average, the total solar power produced in a day distributes over every 30 minute intervals of that day, in percentage. For example, from 8:00 to 8:30, 2.78% is produced, whereas from 12:00 to 12:30, 6.76% is produced (naturally, at the middle of the day, the angle of the sun enables a larger portion of the sunlight to be converted to electrical energy for the battery). In our computations, the first time unit of any day except the first one (that is, for  $t_u = 18, 36, \ldots$ ) not only considers the power produced at that time unit, but also the one produced after 17:00 of the previous day and before 8:00 of the given day (these last two with tilted solar panels).



Figure 5.4: Distribution of the solar power production throughout the day.

In case the current time unit corresponds to a checkpoint, the updates are:

• the position and power demand keep the previous value because the car is not moving or using energy;

• because of the extra power production due to the tilting of the solar panels during checkpoints, 120% of the amount of solar power produced during the time unit  $(\Delta P_a)$  is added to the power available, so that the battery status is updated with

$$b \to \min\{b + 1.2 \cdot \Delta Pa, 5\}$$

In either case, the computations are interrupted whenever the 3020 km of the race are completed or the car runs out of battery (we considered a minimum battery level of 3%).

#### The outputs

The optimal velocity estimated by the algorithm for the whole race is 81 km/h, in a total of 4.67 days, that is, finishing at approximately 14:00 of day 5, and with the battery at 6.34%. Fig. 5.5 displays the power demand and the power available, for the whole race and at every velocity, showing 81 km/h to be the highest constant velocity for which the power available surpasses the power demand. At this constant



Figure 5.5: Estimating the optimal velocity for the whole race: 81 km/h is the highest velocity for which the power available surpasses the power demand.

velocity, the battery level throughout the whole race will evolve as shown in Fig. 5.6. As expected, there are several local maxima, corresponding to the charging of the battery during checkpoints, after 17:00 and before 08:00, all of which with the vehicle stopped and the solar panels tilted.

For comparison, Figs. 5.7 and 5.8 display various conditions regarding the solar input and the wind conditions, respectively. The solar input has an impact on the amount of power available, whereas different wind conditions give rise to different power demands. We note that, although the optimal velocity and the duration of the race are the same in the baseline scenario and in the (unlikely) scenario where the is



Figure 5.6: The battery level throughout the whole race at the constant optimal velocity.

no wind, in the first case the vehicle's battery would finish the race with 6.34%, as already mentioned, and, in the second, with 12.56%.

Assuming that the vehicle that will race the BWSC 2023 to have better performance with respect to power consumption, one considers also estimates for two scenarios: 5% and 10% less power consumption (cf. Fig. 5.2). Fig. 5.9 shows how these scenarios compare with the baseline, that is, using the data from the vehicle that took part in the previous edition of the BWSC.

The outputs for the trials regarding the solar input, the wind impact and the vehicle's power consumption are summarized in Table 5.1. In this table, the "duration in hours" includes 4.5 hours of checkpoints and "+5% economic" denotes the scenario in which the power consumption is 5% less than with the BWSC 2019 vehicle. In the BWSC 2019, the wining team completed the race in 39.36 hours (including the time spent at checkpoints) with an average velocity of 86.6 km/h, whereas TDSR finished  $4^{\text{th}}$  in 43.01 hours, having driven at an average velocity of 78.4 km/h. This means that the simulated values here are very realistic.

	Optimal velocity	Duration in hours	Duration in days	Finishing time	% battery left
Baseline	81	42	4.76	14:00 of day 5	6.34
+10% solar input	84	40.5	4.5	12:30 of day 5	14.81
+20% solar input	88	39	4.33	11:00 of day 5	5.53
No wind	81	42	4.67	14:00 of day 5	12.87
+100% wind	80	42.5	4.72	14:30 of day 5	14.87
+200% wind	79	43	4.78	15:00 of day 5	16.6
+5% economic	83	41	4.56	13:00 of day 5	9.74
+10% economic	85	40	4.5	12:00  of day  5	14.89

Table 5.1: Summary of the outputs for the different trials using the data-based long-term strategy.

As previously mentioned, the algorithm allows for an estimate every 30 minutes,



Figure 5.7: Different optimal velocities (and different finishing times) for distinct solar inputs.

in order to accommodate updates of any kind—namely, regarding the vehicle's performance and the weather forecast—in the course of the race and the most natural event that parts of the race have not elapsed as initially intended. For illustration, Fig. 5.10 shows what the new optimal velocity would be, at the start of the third day of the race, as the result of three different scenarios: a first scenario where the distance completed falls short of the ideal one, but the battery level is high; a second, where the distance completed is satisfactory, but the battery is significantly low; and a third, where both the distance completed and the battery level are quite satisfactory. The code runs in a matter of seconds, as was the case with the estimate for the entire race.

#### 5.2.2 Optimisation-based long-term strategy

The primary goal of the optimisation-based long-term strategy is to compute a (nonconstant) velocity profile which will form a benchmark for the short-term strategising. This velocity profile we will represent as a function  $v : [t_0, T_{\max}] \rightarrow [0, V_{\max}]$  which, for each moment in time, prescribes a velocity for the car at that time. Here,  $T_{\max}$  is the maximum time a car is allowed to finish the race within, which is 60 hours, and  $V_{\max}$  is the maximum speed attainable by the car. This velocity profile is subject to



Figure 5.8: Different optimal velocities (and different finishing times) for distinct wind conditions.

the following constraints:

$$v(t) \in [0, \text{SpeedLimit}(D(t))],$$
(5.1a)

$$|\dot{v}(t)| \le MaxAcceleration,$$
 (5.1b)

$$D(T_{\rm max}) \ge 3000 \text{ km},$$
 (5.1c)

$$v(t) = 0$$
 if  $t \in [t_i, t_i + 30 \text{ min}].$  (5.1d)

where D(t) is the distance travelled along the track at time t, given by

$$D(t) := D(t_0) + \int_{t_0}^t v(s) \, ds,$$

and  $t_i := \min\{t \mid D(t) = D_i\}$ , where the  $D_i$  are the locations of the checkpoints (measured in distance along the track). In words, these constraints insist that:

- a. The velocity abides by the road speed limit at the current location.
- b. The car never accelerates or decelerates too quickly.
- c. The car finishes the race within the maximum time.



Figure 5.9: Different optimal velocities (and different finishing times) for distinct power consumption.

d. The car stops during the control stops (checkpoints).

We now define two key quantities of interest. First, we define T(v), the time to finish the race with strategy v, given by

$$T(v) := \min \left\{ T \left| \int_{t_0}^T v(t) \, dt = 3000 \, \mathrm{km} \right\} \right\}.$$

Second, we define  $\mathbb{P}(v)$ , the probability of finishing the race given strategy v. This is given by  $\mathbb{P}(\min_t S(t) \ge S_{\min})$  where S(t) is the state of charge in the car's battery (between 0 and 5000 Wh) given by

$$\begin{split} S(t) &= S(t_0) + \int_{t_0}^t \text{SolarInput}(s, D(s)) \ ds \\ &- \int_{t_0}^t \text{Consumption}(v(s), \dot{v}(s), \theta(D(s)), W(s, D(s))) \ ds, \end{split}$$

where  $\theta$  is the road incline at a given location and W describes the wind speed and direction at a given time and location, and  $S_{\min}$  is a threshold below which the charge should never drop. This minimum power threshold should incorporate two factors. First, there will be a certain power level below which the battery may not safely operate. However,  $S_{\min}$  should in fact be taken to be above that threshold, to give a safety margin for the short-term strategy to work within.



Figure 5.10: Update at the start of day 3 from three different scenarios.

#### Formulations of the optimisation problem

Given the constraints above there are three related optimisation problems one might desire for the long-term strategy to solve. The choice between these is not mathematical, but instead depends on the overall strategic goals of the team.

The first formulation is "maximise the probability of finishing 'fast'":

 $\max_{v} \qquad \mathbb{P}(v) \qquad s.t. \qquad v \text{ obeys } (5.1), \qquad T(v) \le T^*, \tag{5.2}$ 

where  $T^*$  is some target finishing time.

The second formulation is "finish as soon as possible within a fixed risk of failure":

$$\min \qquad T(v) \qquad s.t. \qquad v \text{ obeys } (5.1), \qquad \mathbb{P}(v) \ge \mathbb{P}^*, \tag{5.3}$$

where  $\mathbb{P}^*$  is some minimum probability of finishing.

Finally, the third formulation is "minimise the expected finishing time, where running out of power is treated like finishing very slowly":

$$\min_{v} \qquad \mathbb{P}(v)T(v) + (1 - \mathbb{P}(v))T_{\max} \ s.t. \qquad v \text{ obeys } (5.1). \tag{5.4}$$

#### Solving these optimisation problems

We first consider how to compute the key quantities involved. Computing T(v) is straightforward. To compute  $\mathbb{P}(v)$ , we run *n* simulations of the form:

- i. Sample  $S(t_0)$  from a normal distribution with mean the measured value and variance the known variance in measurements of battery state.
- ii. From the weather forecast, sample the solar input and wind at each (s, D(s)) (or in practice, at each discrete time step).
- iii. Compute S(t) at each t (or at each time step).
- iv. Record the simulation as a success if  $\min_t S(t) \ge S_{\min}$ .

The probability can then be estimated as the number of successes divided by n.

Then, to solve any of equations (5.2) to (5.4), we have two major concerns. The first is that we really seek a *global* optimiser. The second is that we may not have access to the derivatives of our objective functions.

That is, we need to perform derivative-free global optimisation. Previous work by Top Dutch employed *genetic algorithms* for this task. These algorithms, first introduced in Holland 1992,<sup>7</sup> attempt to mimic natural selection, with the "fitness" of a given "organism" determined by how well it minimises the objective function. In this work, we will employ particle swarm methods, introduced in Kennedy and Eberhart 1995.

However, this choice of method was guided more by our use of the MATLAB Global Optimization Toolbox than any systematic considerations. There is now considerable literature on algorithms for these sorts of problems, and the choice of an effective algorithm is a major direction of future work for Top Dutch. As some suggestions for further investigation, we refer to: Conn, Scheinberg, and Vicente 2009 and Audet and Hare 2017, textbooks on derivative-free optimisation; Rios and Sahinidis 2013, a comprehensive review and comparison of derivative-free global optimisation methods; and Cartis, Roberts, and Sheridan-Methven 2022, a recent investigation of the use of the Py-BOBYQA method (introduced in Cartis, Fiala, et al. 2019) for global optimisation.<sup>8</sup>

#### Numerical experiments for optimisation-based long-term strategy

In the following experiments, we will be solving (5.4), but we will be a bit looser with our modelling than in Section 5.2.1.

First, as in the previous experiments we will be neglecting acceleration, though here we will be allowing v to be piecewise constant (with potentially more than one piece). We will be modelling time as consisting of 8 hour race days, ignoring any charging that can be performed at night or during the control stops. We will also

<sup>&</sup>lt;sup>7</sup>The date here refers to the most recent edition, the first edition was published in 1975.

<sup>&</sup>lt;sup>8</sup>See https://numericalalgorithmsgroup.github.io/pybobyqa/build/html/index.html for details and code for the Py-BOBYQA method.

be ignoring wind and road inclination, and treating the solar input as if it depended only on time and not location. Thus we simplify the state of charge formula to

$$S(t) = S(t_0) + \int_{t_0}^t \text{SolarInput}(s) - \text{Consumption}(v(s)) \ ds,$$

where we take (denoting by [x] the fractional part of x)

SolarInput(t) = max 
$$\left(\frac{6500 + \mathcal{N}(0, 500^2)}{8} \times 6[t/8](1 - [t/8]), 0\right)$$
 W,  
Consumption(u) =  $0.17u^2 - 6.6u + 252.95 + \mathcal{N}(0, 100^2)$  W.

That is, we model the sun as delivering on average 6500 Wh per 8 hour day, fluctuating with Gaussian error with standard deviation 500 Wh, depending quadratically on time of day. We model power consumption according to the data from Fig. 5.2, with Gaussian error with standard deviation 100 W.

We split the 60 hours of possible racing time into 1000 time-steps, corresponding to a time step of 3 minutes and 36 seconds. We set  $S_{\min} = 500$  Wh, and compute  $\mathbb{P}(v)$  using 100 simulations. We modelled the speed limit as being 110 km/h for the entire race.

For our first experiment, we follow the previous section and require that v take a constant velocity. In Fig. 5.11 we show how the objective function and probability of finishing depend on this constant velocity. We see that there is a very sharp transition where the probability drops from almost 1 to almost 0. A consequence of this, if it holds more generally, would be that equations (5.2), (5.3), and (5.4) would have almost identical minimisers.

The optimum constant velocity strategy from Fig. 5.11 comes out as 84.5 km/h, giving a finishing time of 35.5 hours, which yields very realistic values also corresponding to the values in the previous section. In Fig. 5.12 we show the corresponding behaviour of the state of charge under this strategy.

In our second experiment, we allowed the velocity profile to be piecewise constant on 2.4 hour blocks, i.e. v can be encoded as a 25 dimensional vector. We solved (5.4) in this setting using the particle swarm method from the MATLAB Global Optimisation Toolbox version 4.6 (see Kennedy and Eberhart 1995; Mezura-Montes and Coello 2011; Pedersen 2010 for details on the particle swarm algorithm employed). Fig. 5.13 shows the resulting velocity strategy as well as the behaviour of the state of charge under this strategy.

The finishing time for this strategy is 35.4 hours, so is only a mild improvement over the constant velocity strategy. However, this may be due to the simplicity of this set-up. We expect that as more details are added in the disparity will increase between the constant velocity strategies and the strategies where the velocity is permitted to vary. Overall this strategy gives very realistic values.



Figure 5.11: Objective function value for (5.4) (left) and  $\mathbb{P}(v)$  (right), for v a constant velocity with the specified value. The minimum value is attained at 84.5 km/h.



Figure 5.12: Expected value for S(t) (solid line)  $\pm$  standard deviations (dashed lines) for the velocity profile  $v(t) \equiv 84.5$  km/h.



Figure 5.13: Particle swarm solution for (5.4) where v is constant on 2.4 hour intervals (left) and corresponding expected value for S(t) (right, solid line)  $\pm$  standard deviations (right, dashed lines). Computation time: 77s with MATLAB Global Optimization Toolbox version 4.6.

#### 5.2.3 Optimisation-based short-term strategy

The following optimisation-based short-term strategy, which is still preliminary work, should resist dynamic disturbances during racing. We minimize the following two terms: 1) the power that has been consumed in a specific short period and 2) the error term between the desired velocity profile provided by long-term strategy and current velocity, when given constraints such as local velocity limit and acceleration/deceleration limits and power limits that we can use in a day or an hour. A short-term optimization problem to find the optimal velocity is formulated by the following:

$$\begin{split} \min_{u} & E(t) + \alpha(t_{0}) || v - v_{\text{profile}} ||^{2} \\ s.t. & \dot{x}(t) = f(x(t)) + g(x(t)) u(t) \\ & y(t) = x(t) \\ & v \leq v_{\text{limit}} \\ & v_{\text{limit}} = \begin{cases} \frac{130}{3.6}, & \text{if } p(t) \geq 1700 \times 10^{3} \text{ m}, \\ \frac{110}{3.6}, & \text{if } 1700 \times 10^{3} \text{ m} < p(t) \leq 3027 \times 10^{3} \text{ m}, \\ 0, & \text{otherwise.} \end{cases} \\ & |\dot{v}(t)| \leq 6 \text{ m/s}^{2} \\ & u(t) \leq \text{Power}_{\text{limit}}(t) \end{split}$$

Here,  $x(t) = [p(t) \quad v(t) \quad E(t)], p(t) \rightarrow \text{position}, v(t) \rightarrow \text{velocity and } E(t) \rightarrow \text{energy}$ consumed,  $f(x(t)) = \begin{bmatrix} v(t) & -\frac{F_{\text{total}}}{m} & 0 \end{bmatrix}^{\top}, g(x(t)) = \begin{bmatrix} 0 & \frac{\phi(v(t))}{mv(t)} & 1 \end{bmatrix}^{\top}$ , where  $\phi(v(t))$ is the motor efficiency, y(t) is the output/measured variables.  $F_{\text{total}}$  is the total force acting on the car.  $u(t) \rightarrow$  is the input power to the motor and  $\alpha(t)$  is a parameter that can be computed based on the solar input and wind at time t. When the solar input is high and there are no clouds, the value of  $\alpha$  can be kept higher.  $\alpha$  has been computed only at  $t_0$  as it is been assumed that the solar input and wind are going to be the same during the short-term prediction horizon  $t_f - t_0$ . This assumption is not conservative as the short-term strategy can be considered for a time span of 30 minutes to an hour, for which consideration of constant weather conditions is good enough and saves the effort of calculating  $\alpha(t)$  at multiple time instants.  $v_{\text{profile}}(t)$  is the velocity based on the velocity profile generated by the long-term strategy.



Figure 5.14: The optimal power input for the short-term strategy with minimum consumption of energy

The following figures provide the outcome of the optimization problem while minimizing only the consumed energy. The nonlinear MPC toolbox in MATLAB has been used to solve the optimization problem. However, the solver associated with the toolbox is not reliable enough to search for the feasible set. So, when velocity regularization has been introduced in the cost function no solution could be achieved even with relaxed set of constraints. A better solver/ optimization algorithm should be considered in future to cater to this problem. This strategy is thus clearly still preliminary work but might reveal its potential after future improvements.

#### 5.2.4 Distance maximization under stochastic uncertainty

In this subsection an optimal control algorithm is employed for maximizing the distance travelled of a solar car, when state dynamics are uncertain due to stochastic noise from model errors and uncertain solar input. The control objective is to maximize the expected travelled distance after a fixed time interval of a day.

#### System dynamics

For times  $k \in \mathbb{K} := \{0, 1, 2, 3, ..., T-1\}$  we define the stochastic scalar dynamic system

$$x_{k+1} = x_k + f(x_k, u_k, d_k) \,\Delta t + \sigma \sqrt{\Delta t} \,\epsilon_k \tag{5.5}$$

$$u_k = g_k(x_k). \tag{5.6}$$

Here  $x \ [kWh]$  is the battery state, with time index k (time is discretized as  $t_k = k\Delta t$ ). The function  $f : \mathbb{R}^2 \to \mathbb{R}$  describes the dynamic interactions between state and input. The scalar  $u_k \ [kW]$  is the control input that represents the power flow from the battery to the engine. The scalar  $d_k \ [kW]$  is the solar radiation that acts as an external, uncontrollable input. The inputs are determined by a state feedback control policy, which is a sequence of input functions  $g_k : \mathbb{R} \to \mathbb{R}$  for all time instances k. The stochastic variables  $(\epsilon_k)_{k\in\mathbb{K}}$  are iid with standard Gaussian distributions. These variables represent the prediction uncertainty in the battery status dynamics due to uncertainty in solar input, and unforeseen fluctuations in energy use due to variation in wind resistance, and rolling resistance. We use the notation  $x_k^g$  for battery states that are found if a policy g is applied for a battery with an initial state  $x_0^g = x_0$ .

Due to the stochastic state dynamics, it cannot be predicted a priori how the state will evolve exactly, and therefore it is also not possible to a priori determine an optimal sequence of inputs. Therefore, for each point in time, k, and for each possible state,  $x_k$ , an optimal input needs to be determined.

#### Control problem

The goal is to determine a policy that maximizes value, defined by the expected revenues  $J(x_T^g)$  for a battery at finish time T, minus the running costs  $L_k$ . The associated control problem is to find a policy that maximizes

$$\mathbb{E}\left[J(x_T^g) - \sum_{k=0}^{T-1} L_k(g_k(x_k^g)))\Delta t\right]$$
(5.7)

for any given state  $x_k$ , at any time k, over all control laws  $(g_k)_{k \in \mathbb{K}}$  such that  $g_k$  is in an admissible set  $\mathcal{G}_k$  (defined in section Computational Settings). We use  $\mathcal{G}$  to denote the collection of admissible sets  $(\mathcal{G}_k)_{k \in \mathbb{K}}$ 

The control problem was solved using dynamic programming based on a discrete state and dynamics form of system 5.5, as described in Mourik, Ooster, and Vellekoop 2023.

#### Solar car model

We consider the case of a solar car where the battery power converted to car velocity, is controlled. The model describes the state dynamics of the battery status in terms of a function  $f : \mathbb{R}^2 \to \mathbb{R}$  that is given by:

$$f(x_k, u_k, d_k) = c_\alpha d_k \mathbf{1}_{x_k \le x_{max}} - u_k \mathbf{1}_{x_k > x_{min}}.$$
 [kW] (5.8)

Here  $c_{\alpha} = 1 \ [m^2]$  is the effective surface of the solar panels (there are  $4 \ m^2$  of panels with an assumed efficiency of 0.25 [-]),  $d_k = I_k$  is the solar irradiance  $[kWm^{-2}]$ ,  $x_{max}$  is the battery capacity (5 kWh), and  $x_{min}$  is the minimum battery status for power use for the engine (0.375 kWh).

The relation between velocity and input,  $u(v) = 0.17v^2 - 6.6v + 252.95$ , was inverted to v(u) for control purposes. For that, the relation was approximated to  $u(v) = 0.13v^2 + 0.1v + 50$  to allow a straightforward inversion using the ABC formula (the negative coefficient -6.6 prohibits this). The requirement that  $v \ge 0$  results in

$$v(x,u) = \frac{-b + (b^2 - 4a(c-u))^{0.5}}{2a} \mathbf{1}_{x > x_{min}} \mathbf{1}_{u \ge c}$$
(5.9)

with a = 0.13, b = 0.1, and c = 50.

#### Running costs

The running costs on time k are defined as minus the expected distance traveled between time k and k + 1. Hence, the cost function is minus the velocity:

$$L_k(u, x) = -v(u, x).$$
(5.10)

The revenues associated with battery status at the end of the day is in this case assumed to be zero: J(x) = 0. That is, for this particular day the amount of battery left at the end of the day has no value. For the more general case where this has value, as linear reward function could be employed:

$$J(x) = c_{\beta}x. \tag{5.11}$$

#### Computational settings

- To avoid extremely large numbers in the transition probability computations, some S.I. units have been scaled: the distance is in km, the power in kW, and the time in hours.
- To avoid infinitely steep slopes in the optimization process, the constraint function  $\mathbf{1}_{x>c}$  was approximated by a smoothing function  $\frac{x^6}{x^6+c^6}$ , and  $\mathbf{1}_{x\leq c}$  approximated by  $\frac{c^6}{x^6+c^6}$ . The exception is  $\mathbf{1}_{u\geq c}$  in equation (5.9), where a hard constraint was enforced to avoid complex valued solutions.

- The computational grid was  $N \times T = 300 \times 96$ . The battery state was thus discretized into N = 300 parts. The time was discretized in 5 minute intervals. Solving the dynamic feedback control problem took around 5 minutes on a standard laptop using parallel computing on 4 cores of each 2.1 GHz.
- The state space ranges from  $[0, 5] \times [0, 8]$  ( $[kWh] \times [h]$ ), and the initial state in the dynamic simulations was  $x_0 = 5 kWh$ , i.e., full battery.
- The set of admissible control values is  $\mathcal{G} = [0, 2.5]$ , in kW.
- The solar input was modelled as  $I(t) = 0 \vee 1.0 \sin(2\pi t/T)$  in kW.
- The expected distance travelled was retrieved by evaluating  $V_1^*(x_0)$ .
- The nominal stochastic noise level was  $\sigma^2 = 210^{-3} [kW^2]$ .
- The fixed input controller input  $u = c_{opt}$  was retrieved by directly solving equation (5.7), using the 'fmincon' routine.

#### **Results for deterministic dynamics**

Fig. 5.15 shows the dynamics of the battery, travelled distance, and irradiance under two choices of steady input: an arbitrary value of  $u_c = 1.2$  kW, and the optimized value of  $u_{c,opt} = 0.83$  kW.

The arbitrary input results in a relatively large velocity, however after approximately 5.5 hours the battery gets depleted, resulting in a sub-optimal travelled distance. The peak in solar input after about 2 hours reduces the rate of battery depletion somewhat, resulting for both choices of input in a wiggle in battery status dynamics.

#### Results for control policy and performance

Fig. 5.16 shows 1) the stochastic control strategy u(x, t), 2) resulting battery depletion rates, 3) solar input, and 4) the associated value V (expected travelled distance) for all  $x_k$  and k in four heat maps, where the color scales with the intensity. The optimal input map shows for each battery state and at each time instance the optimal input value. There is a distinctive border under which little or no input should be given (dark blue area), to avoid running out of battery. The line is a bit wavy due to the solar input dynamics. There is also a distinctive line above which maximum power should be delivered (yellow area) since there is no risk of running out of battery. The resulting depletion rate of the battery mirrors the input map. The value plot shows for each state and time instance the expected distance that will be travelled at the end time. The value is maximal at the left top corner, as could be expected. The gradient of V at t = 0 might be used to design a value function J(x), that rewards battery status at the end of the day according to the value this status has at the start of the next day.



Figure 5.15: The optimal constant input  $u_c = 1.2$  kW results in a battery that is depleted precisely at the end of the day.

#### Results for comparison between fixed input and dynamic feedback control

The stochastic control laws  $u = c_{opt}$  and u = u(x, t) are compared with each other, for initial value  $x_0 = 5$  kWh. The expected travelled distance was 606 km for the fixed input controller, and 621 km for the dynamic feedback controller. The dynamics of the probability of battery status are shown in Fig. 5.17.

The dynamic feedback controller uses more power in the beginning compared to the fixed input controller. The feedback controller also narrows the variance in predicted dynamics compared to fixed input control, thereby decreasing risk of large deviations from the anticipated battery status. Especially near the end the standard deviation of predictions is small  $\sigma(x(T)) = 0.02$  compared to a fixed input controlled system:  $\sigma(x(T)) = 0.06$ .

#### Discussion

This case study demonstrates that a dynamic feedback control may substantially improve performance in terms of expected travelled distance, and prediction uncertainty. As follow up research it would be interesting to investigate the relation between uncertainty and performance. Subsequently, this might provide insights into which way of improving prediction accuracy (e.g. by improving the model, or weather predictions, or by repetitively updating the state and re-solving the control problem in a model predictive control scheme, or any combination thereof). To reduce the influence of



Figure 5.16: Optimal control policy, external input, and associated performance. Top left: optimal input. Top right: solar input. Bottom left: battery depletion rate. Bottom right: value function V (expected travelled distance in km).



Figure 5.17: Dynamics of state probabilities for fixed input and dynamic feedback control. Solid line:  $u = c_{opt}$ , dashed line: u(x,t). The feedback controller uses more power in the beginning causing the battery to deplete faster. The prediction variance is smaller for the feedback controller, especially near the end of the day.

model errors, a learning method could be employed to optimize parameter values or infer their probability distributions (to be used for inferring parameter based prediction uncertainty). A challenge regarding model refinement would be the extension to more than one state. For with a  $300 \times 96$  grid,  $300 \times 96$  optimizations need to be carried out, which took about 5 minutes. Extrapolating this to two states would result in  $300^2 \times 96$  optimizations, and thus 25 hours of computing time. To completely solve the control problem of finishing the race as fast as possible, might require at least two extensions of the methodology described here. For the first four days, the control problem is to find a control policy that maximizes distance. That could be possible by extending the horizon to four days instead of one. However, the weather predictions could become unreliable, so regular state updates and recalculation of g in a model predictive control fashion would be favorable. The last day poses a different type of control problem, namely to find a control policy that minimizes finishing time. This problem is known as minimal time optimal control.

## 5.3 Conclusions and future work

The outcomes of the data-based strategy indicate that with a constant velocity, a maximum possible velocity of 81 km/h would be feasible. This is considerably faster compared to the results of last year of this team (78 km/h), however still significantly lower than that of the winning team (87 km/h). The results of the optimisation-based long-term strategy suggest that a constant velocity of 84.5 km/h would be feasible. The optimisation-based short-term strategy unfortunately only led to preliminary results and needs to be explored further before reliable results can be shown. The outcomes of the dynamic programming strategy indicate that changing from a static to a dynamic velocity profile could increase the average velocity from 76 km/h to 78 km/h. Altogether, the results of the different strategies suggest that small but significant improvements can be made using velocity optimization. Whether this alone will be sufficient to beat all competition remains to be seen.

All of the results should be interpreted with caution, due to the numerous assumptions on weather, terrain, car mechanics, and uncertainty sources, that were needed to come to a solution within the study week. The influence of the above mentioned assumptions deserves further exploration in follow-up research. An interesting approach would then be to synthesize the optimization problem with respect to control and design parameters in order to optimize velocity control and car design simultaneously. Another interesting approach would be an adaptive method to fine-tune the system dynamics of the car on the fly.

## References

Audet, Charles and Warren Hare (2017). Derivative-free and blackbox optimization. 1st. Springer. ISBN: 978-3-319-68913-5. DOI: 10.1007/978-3-319-68913-5.

- Cartis, Coralia, Jan Fiala, et al. (2019). "Improving the Flexibility and Robustness of Model-Based Derivative-Free Optimization Solvers". In: ACM Trans. Math. Softw. 45.3. ISSN: 0098-3500. DOI: 10.1145/3338517.
- Cartis, Coralia, Lindon Roberts, and Oliver Sheridan-Methven (2022). "Escaping local minima with local derivative-free methods: a numerical investigation". In: *Optimization* 71.8, pp. 2343–2373. DOI: 10.1080/02331934.2021.1883015. eprint: https://doi.org/10.1080/02331934.2021.1883015.
- Conn, Andrew R., Katya Scheinberg, and Luis N. Vicente (2009). Introduction to Derivative-Free Optimization. Society for Industrial and Applied Mathematics. DOI: 10.1137/1.9780898718768.
- Geling, H.B.R. (July 2021). Energy management optimization for a solar electric vehicle.
- Holland, John H. (1992). Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. The MIT Press. ISBN: 9780262275552. DOI: 10.7551/mitpress/1090.001.0001.
- Kennedy, James and Russell Eberhart (1995). "Particle swarm optimization". In: Proceedings of ICNN'95-international conference on neural networks. Vol. 4. IEEE, pp. 1942–1948.
- Mezura-Montes, Efrén and Carlos A Coello Coello (2011). "Constraint-handling in nature-inspired numerical optimization: past, present and future". In: Swarm and Evolutionary Computation 1.4, pp. 173–194.
- Mocking, Ceriel (Jan. 2006). Optimal design and strategy for the SolUTra. URL: http: //essay.utwente.nl/56520/.
- Mourik, Simon van, Bert van't Ooster, and Michel Vellekoop (2023). "Plant Performance in Precision Horticulture: Optimal climate control under stochastic uncertainty". In: arXiv. DOI: https://doi.org/10.48550/arXiv.2303.14678. eprint: 2303.14678 (math.OC).
- Pedersen, Magnus Erik Hvass (2010). "Good parameters for particle swarm optimization". In: Hvass Lab., Copenhagen, Denmark, Tech. Rep. HL1001, pp. 1551–3203.
- Rios, Luis and Nikolaos Sahinidis (2013). "Derivative-free optimization: A review of algorithms and comparison of software implementations". In: *Journal of Global Optimization* 56, pp. 1247–1293. DOI: 10.1007/s10898-012-9951-y.