Proceedings of the 84th European Study Group
Mathematics with Industry

SWI 2012

Eindhoven, January 30 – February 3, 2012

Cover design by Manuel Davila Delgado
ISBN: 978-90-6464-630-0

Contents

Introduction

There are a few well-defined moments when mathematicians can get in contact with relevant unsolved problems proposed by the industry. One such a moment is the so-called "Study Group". The concept of the Study Group is rather simple and quite efficient: A group of mathematicians (of very different expertise) work together for one week. As a rule, on a Monday the industrial problems are presented by their owners, then few research groups self-organize around the proposed problems and work intensively until Friday, when the main findings are presented. The insight obtained via mathematical modeling together with the transfer of suitable mathematical technology usually lead the groups to adequate approximate solutions. As a direct consequence of this fact, the problem owners often decide to benefit more from such knowledge transfer and suggest related follow-up projects.

In the period January 31– February 3, 2012, it was the turn of the Department of Mathematics and Computer Science of the Eindhoven University of Technology to organize and to host the "Studiegroep Wiskunde met de Industrie/Study Group Mathematics with the Industry" (shortly: SWI 2012, but also referred to as ESG 84, or as the 84th European Study Group with Industry). This was the occasion when about 80 mathematicians enjoyed working on six problems. Most of the participants were coming from a Dutch university, while a few were from abroad (e.g. from UK, Germany, France, India, Russia, Georgia, Turkey, India, and Sri Lanka).

The open industrial problems were proposed by Endinet, Philips Lighting, Thales, Marin, Tata Steel, and Bartels Engineering. Their solutions are shown in this proceedings. They combine ingenious mathematical modeling with specific mathematical tools like geometric algorithms, combinatorial optimization of networks, identification of parameters and model structures, probability theory, and statistical data analysis.

It is worth mentioning that this scientific proceedings is accompanied by a popular proceedings, written by Ionica Smeets, containing layman's descriptions of the proposed problems and of the corresponding results.

A Case Study in the Future Challenges in Electricity Grid Infrastructure

Marjan van den Akker (Utrecht University), Herman Blok (University of Leiden), Chris Budd (University of Bath), Rob Eggermont (Eindhoven University of Technology), Alexander Guterman (Moscow State University), Domenico Lahaye (TU Delft), Jesper Lansink Rotgerink (University of Twente), Keith W. Myerscough (CWI Amsterdam), Corien Prins (Eindhoven University of Technology), Thijs Tromper (University of Twente), Wander Wadman (CWI Amsterdam)

Abstract

The generation by renewables and the loading by electrical vehicle charging imposes severe challenges in the redesign of today's power supply systems. Indeed, accommodating these emerging power sources and sinks requires traditional power systems to evolve from rigid centralized unidirectional architectures to intelligent decentralized entities allowing a bidirectional power flow. In the case study proposed by ENDINET, we investigate how the penetration of solar panels and of battery charging stations on large scale affects the voltage quality and loss level in a distribution network servicing a residential area in Eindhoven, NL. In our case study we take the average household load during summer and winter into account and consider both a radial and meshed topology of the network. Our study results for both topologies considered in a quantification of the levels of penetration and a strategy for electrical vehicle loading strategy that meet the voltage and loss requirements in the network.

keywords: power systems, load flow computations, distributed generation, electrical vehicle charging

1 Introduction

The problem brought to SWI2012 by ENDINET is the hugely important question of the future performance, stability and integrity of the power supply network. The issues facing power generation are changing rapidly. Until recently we have had a situation of a small number of large suppliers of electricity (typically power stations delivering 100MW or more of power to consumers with high demand during the day and low demand at night. In the future, and with the coming of the smart grid, this will change. In particular we will see a large number of small scale generation (and storage) of power (in the range of 1-10kW) from households, typically through solar cells or batteries, combined with a much

larger drain on the network at night due to the charging of electrical vehicles (at a rate of 3kW per vehicle). The increase in solar power usage for instance is illustrated in Figure 1. Both the generation and the new loads on the system substantially change the dynamics of the grid. Various questions then arise, in particular, (i) will the grid be able to cope with the new loads imposed on it without a significant change in the quality of the voltage, (ii) what are the optimal strategies for charging electrical vehicles, and (iii) will the grid supply remain stable under the unpredictable situation of variable power generation and load.

Some of these issues have been considered in previous study groups. For example the 2010 Study Group in Amsterdam, NL, looked into the optimal distribution of decentralized power generation under network constraints [5]. The 2011 Study Group in Cardiff, UK, addressed a stability question and studied in particular the behavior of the inverter units which couple the solar cells to the grid [1]. It attempted to match the phase of the locally generated AC with that of the grid.

In the SWI 2012 meeting we considered the quasi-static problem in which the grid is assumed to be in equilibrium at each time, with a slowly evolving load and generation profile (see e.g. [3]). In this context, the study group considered the question of voltage quality under varying load and supply, and looked into strategies for the optimal charging of electrical vehicles (EVs). The group examined the effect of different loading patterns by considering each household to have a time-varying stochastic load. It also looked into how either adding or deleting an important conducting line affects the voltage quality. To this end repeated load (or power) flow simulations of the voltage amplitude and phase at each nodes (or bus) in the network were performed. Computations were performed using the package MATPOWER [8] on a relatively small model network with less than 100 households connected. However, in practice all of the techniques considered could be easily scaled up to a much larger network. For the small network considered a reasonably complete answer can be given for all of the questions raised above, and this is the subject of this report.

This report is structured as follows. In Section 2 we describe the mathematical model for load flow computations resulting in the voltage amplitude and phase in a power network. We also describe the distribution network considered as well the requirements on the voltage, currents and losses imposed. In Section 3 we describe the stochastic distribution of household load both within the network as over a year. In Section 4 we describe how the penetration of solar panels affects the voltage quality. In Section 5 we describe a strategy that prevents the network to be overloaded by the charging of electrical vehicles. In Section 6 we present numerical results on of simulations of the distribution generation by solar panels and the loading of electrical vehicles. We end this report by giving conclusion and recommendations for future work in Section 7.

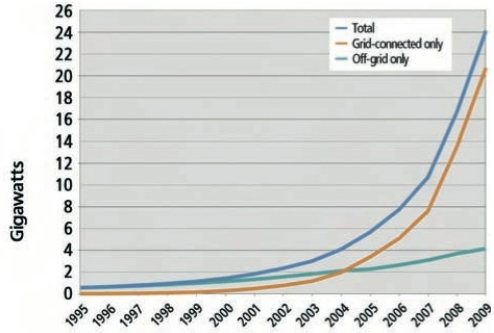


Figure 1: The increase of solar energy usage [4].

2 Power Flow Problem

The power flow problem is the problem to determine the voltage at each bus of a power system, given the supply at each generator and the demand at each load in the network (see e.g. [3]). The network we will consider is a low voltage network supplying a residential area consisting of a few streets. The power is fed into a network by a connection to medium voltage network through as transformer that can be regarded as an infinite source of power. The solar panels installed will be taken into account as decentralized power sources. Apart from the household loads, we will also take the loads of the charging of vehicles into account.

Let $Y = G + jB$ denote the network admittance matrix of the power system. Then the power flow problem can be formulated as the nonlinear system of equations

$$\sum_{k=1}^{N} |V_i|\,|V_k|\,(G_{ik}\cos\delta_{ik} + B_{ik}\sin\delta_{ik}) = P_i, \tag{1}$$

$$\sum_{k=1}^{N} |V_i|\,|V_k|\,(G_{ik}\sin\delta_{ik} - B_{ik}\cos\delta_{ik}) = Q_i, \tag{2}$$

where $|V_i|$ is the voltage magnitude, δ_i is the voltage angle, with $\delta_{ij} = \delta_i - \delta_j$, P_i is the active power, and Q_i is the reactive power at bus i. The current, voltage and power are measured in Ampère (A), Volts (V) and Watts (W), respectively. For details see again e.g. [3].

Define the power mismatch function as

$$\vec{F}(\vec{x}) = \begin{bmatrix} P_i - \sum_{k=1}^{N} |V_i|\,|V_k|\,(G_{ik}\cos\delta_{ik} + B_{ik}\sin\delta_{ik}) \\ Q_i - \sum_{k=1}^{N} |V_i|\,|V_k|\,(G_{ik}\sin\delta_{ik} - B_{ik}\cos\delta_{ik}) \end{bmatrix} \tag{3}$$

where $\vec{x}$ is the vector of voltage angles and magnitudes. Then the power flow problem (1), (2) can be reformulated as finding a solution vector $\vec{x}$ such that

$$\vec{F}(\vec{x}) = \vec{0}. \tag{4}$$

This is the system of non-linear equations that we solve to find the solution of the power flow problem. In our experiments we will make use of the MatPower package [8].

In our study we perform repeated load flow computations to simulate the load profile over the course of a week in either summer or winter. We seek to understand to what extend the solar panels can penetrate in the required power generation and to what affect the EVs can be loaded with out introducing malfunctions in the network.

2.1 Distribution Network Considered

The distribution network considered is the network with 14 busses and 14 lines shown in Figure 2. It consists of two branches. The upper branch in this figure has a meshes structure as customers in this branch are fed from more than one source. The lower branch is intentionally kept radial to make the case study more interesting. The busses are numbered consecutively from 1 to 9 and from 10 to 14 by first traversing the upper and then the lower branch in from top left to bottom right. The dotted line in the lower right part of the figure is a hypothetical that ENDINET considers building to convert the radial topology into a meshed one. The line data of the network considered is given in Table 1. Households are connected to the network by 3x25 A connections.

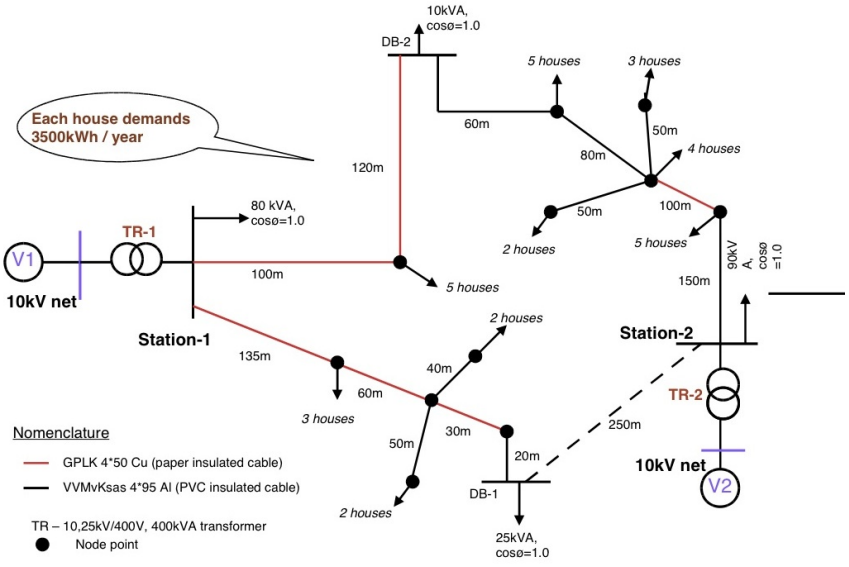


Figure 2: Diagram of the distribution network considered [2].

Cable Type	Resistance (Ohm/km)	Reactance (Ohm/km)	Maximum Current capacity (A)
Copper	0.387	0.072	170
Aluminum	0.333	0.082	195
Transformer	-	-	725

Table 1: Data on component in the network considered [2].

2.2 Network Requirements

To assess the performance of the network considered under loading, both the nodal voltages $|V_i|$ at node i and the line currents $|I_{ij}|$ between node i and node j need to be taken into account. Key performance indicators are:

1. the nodal voltages should be within 10% of the nominal value of 230 V, i.e.,

$$207\,\mathrm{V} < |V_i| < 253\,\mathrm{V};$$

2. the nodal voltage at any node should not vary by more than 3% between consecutive 10 minute intervals;

3. the line current $|I_{ij}|$ on any one line should not exceed the nominal rating given in Table 1;

4. the real power losses $|I_{ij}|^2 R_{ij}$ on any line should be sufficiently low.

In our computations, we found that the most significant reason for a loss of quality in the network was the effect of voltage variation. The limits on the current and real power loss can all be relaxed either by inserting more power sources or by avoiding radial networks with long lines. The latter indeed require higher values of current at their start node which immediately implies larger voltage drops and greater power loss. In this context radial networks are outperformed by different network topologies.

3 Household Load

In order to solve the power flow problem it is necessary at each load bus, to specify the real power P_i (we took the reactive power drain $Q_i = 0$). We therefore have that

$$P_i = P_i^S - P_i^L - P_i^{EV}$$

where P_i^S is the power *generated* by the solar cells, P_i^L is the general household load and P_i^{EV} is the load due to the electrical vehicle charging. Each of these terms has a different form and we consider each separately.

3.1 Distribution of Household Load over Network

The power drain P_i^L due to consumer load is a time varying variable, that depends stochastically upon the customer. A typical household will consume around 400 W on average, with a peak load of around 1 kW. This load varies during the day (and is highest in the early evening) and we will consider the time variation in the next subsection. Similarly, the power drain varies from one household to the next, dependent, for example, on the number of people living in each house. Data for annual usage (in kW hours) presented by ENDINET is shown in Figure 3. This figures indicates that over one year the total consumption P_i follows a *log-normal* distribution so that $\log(P_i)$ has a mean of 7.904 kWh and a variance of 0.5607 kWh, i.e.,

$$\log(P_i) \sim N(7.904, 0.5607)\,.$$

When simulating the performance of the network the households were each assumed to follow this distribution, and the time varying household load scaled accordingly, so that the values of P_i^L in the power equation were each treated as stochastic variables with the distribution as above.

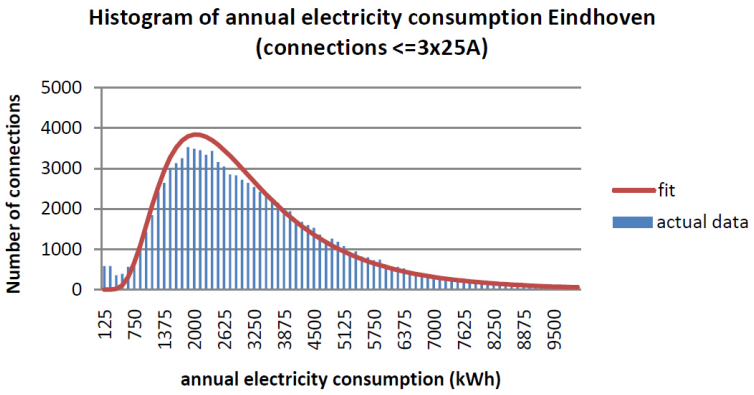


Figure 3: Annual household load distribution accross the network and log-normal fit [2].

3.2 Time Evolution of Single Household Load

Figure 4 gives the electricity consumption of an average household as the percentage of the annual electricity consumption. The data consist of a winter and summer time series, both of one week length and a the step size of 15 minutes. Assuming a total annual electricity consumption of 3.5 kWh, we immediately transformed the data to load time series, with each data point denoting the average power consumption over 15 minutes. The winter load time series is shown in

Figure 5. An obvious observation is the diurnal periodicity of the load pattern.

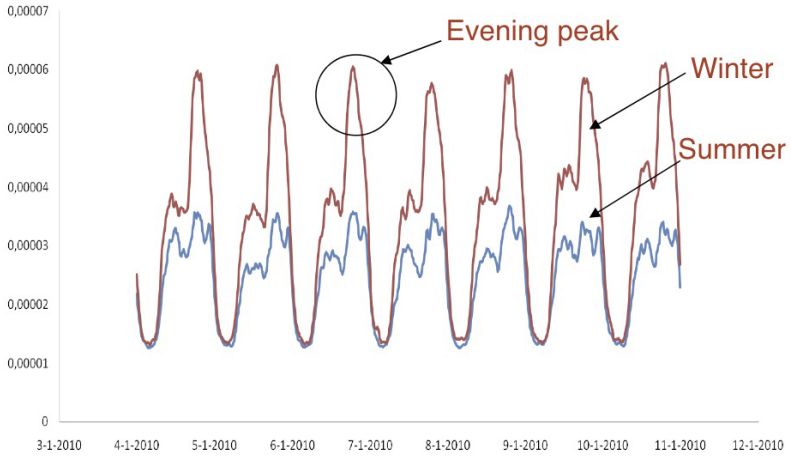


Figure 4: Summer and winter weekly profile of household load [2].

Weekdays see to follow a comparable pattern, whereas weekend days seem to follow a slightly different pattern. In what follows, we try to capture the periodicities to obtain a *load function*. This function may be more of use than the data set itself: in our test cases e.g., we will require a load data time series on a 10 minute basis, which can be easily extracted from the load function.

Discrete Fourier Transform By use of the discrete Fourier transform (DFT), we find the frequency components of the winter load time series. We define an equidistant frequency grid with step size $1/7$ per day, starting at zero frequency up till 96 per day. The double-sided amplitude spectrum is displayed in Figure 6.

To be more specific, the six most dominant frequencies (that is, those corresponding to the largest absolute values) of the single-sided amplitude spectrum are displayed in Table 2. The largest absolute amplitude corresponds to zero frequency and reflects the mean of the time series. Note that only the largest five amplitudes have an absolute value larger than 10 W. This *spikiness* suggests that the time series can reasonably well be approximated by a weighted sum of harmonics with these frequencies, which is illustrated in Figure 5.

4 Distributed Generation by Solar Panels

A first key questions posed to the Study Group was to investigate the possible penetration of the use of solar panels. This required looking into the usage

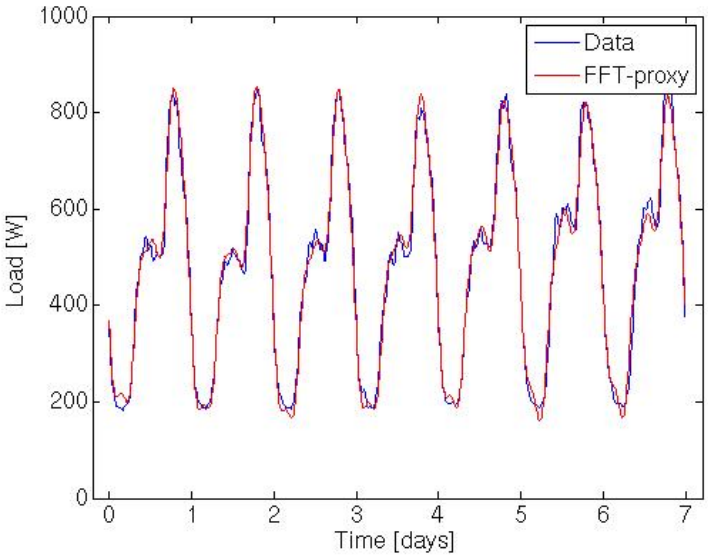


Figure 5: Time series and discrete Fourier transform approximation of load at an average Dutch household, from Monday January 4th 2010, 00:00 a.m. untill Sunday January 10th 2010, 11:45 p.m.

and performances of such panels. On a good sunny day a panel can deliver a maximum of 3 kW at mid-day, but the performance of this panel depends greatly upon both the level of cloud and the the time of day and the time of the year. On an average sunny day the greatest power output is at mid-day with a rise from dawn and a fall to dusk. On a cloudy day the output will vary rapidly. This rapid variation is a potential problem as it can lead to large voltage fluctuations which can then lead to a degrading of the network as a whole. This effect will be illustrated further in this report. The variation in the illumination level of a panel, termed the *insolation* is tabulated. For example in Eindhoven, NL, it is 4.24kWh/m^2 per day in August, and 0.74 kWh/m^2 per day in the January. These figures allowed us to scale the values given by ENDINET for both summer and winter.

Accordingly in the simulations we assumed that a percentage p of the households were using solar panels and that the usage of such panels per household was a uniformly distributed random variable. Four cases were considered (using the data supplied by ENDINET) namely sunny and cloudy days in both summer and winter.

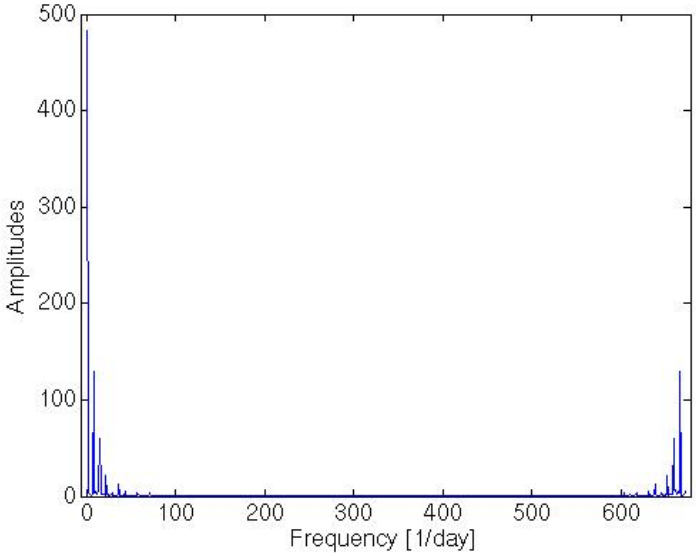


Figure 6: Double-sided amplitude spectrum of the winter load time series.

5 Loading by Charging of Electrical Vehicles

A second key questions posed to the Study Group was to investigate whether or not the given network can cope with the charging of EVs. The difficult part in modeling the loading by charging of EVs in the network, is the modeling of the behaviour of people in the sense of when they charge their cars and of course how much they have been driving that day. For simplicity we assume that everybody in our network gets home at 18.00 hours and immediately plugs in their car, which might be assumed as some kind of worst case scenario, because of course at 18.00 there is also a peak in household load. Another assumption is

frequency [day^{-1}]	|amplitude| [W]	arg(amplitude) [rad]
0	482.3	0
1	129.3	1.78
2	59.8	1.73
3	20.9	−2.52
5	13.2	1.53
1 + 6/7	7.2	−1.94

Table 2: Six most dominant frequencies of the single-sided amplitude spectrum of the winter load time series.

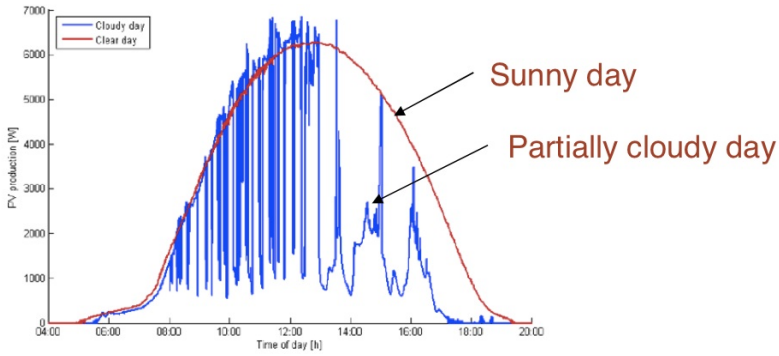


Figure 7: Typical daily power production profile from solar panels [2].

that every car actually drives the average amount of kilometers driven per person per day on a single day. This amount is about 50 kilometers and this assumption, together with the given efficiency of EVs of 5 km/kWh gives us a charging time as will be stated below.

From previous research [7, 6], we know that simply charging when people want to is not optimal at all and thus will not be a realistic case in the future. Indeed, when all cars are charged at the same time, immediately at 18.00, the network will definitely fail due to overloading. We therefore need a strategy for charging EVs. A smart grid is then a grid that is able to implement such a strategy.

Of course future work might lead to even better strategies, which because we only look at a small piece of the network might be needed regarding large networks with a lot of EVs. Our strategy is not optimal at all because it does not allow for vehicles charging at loads lower then the maximum one nor for partial charging in case of necessacity.

5.1 Greedy Control Strategy for EV Charging

If a customer j arrives at home and wants to charge his car we are given:

t_j	the time at which the customer puts the request
E_j^{rest}	the energy still left in the car
c_j	the time by which the customer wants the charging to be completed.
$E_j^{charged}$	(optional) the amount of energy that the customer wants to have after charging. The default value is that the customer wants to have the car charged fully
D_j	the energy requested by the customer, $E_j^{charged} - E_j^{rest}$

We assume that the customer plugs in his car on the network at time t_j. In a *smart grid* the network can decide when the car is actually charged. The

question is now to assign to each customer j an interval that is a subset of $[t_j; c_j]$ during which the the car is charged. This might be generalized to a collection of intervals, if we are allowed to preempt the charging of a car. In the future, it probably will be possible to charge cars at different speeds.

The problem then becomes a complex scheduling problem. At time t_j we have to decide if the car of customer j is charged or if he has to wait. If we decide to charge the car, we have to make sure that the voltage drop is at most 3 % and that the network constraints on the voltage and current are met during the charging period. Moreover, if there are waiting requests and the power consumption in the network decreases, we have to decide whether we start charging another car. Solving this problem requires an intelligent strategy.

In our case study, we restricted ourselves to a basic case. We assume that EV-charging always takes place at 3.5 kW and takes exactly 3 hours requiring therefore 10.5 kWh in total. We also assume that charging cannot be preempted. Each household owns one car, which arrives at 18.00 and wants to be charged by 07.00 the next morning. We apply the following *greedy strategy*:

1. Initialization: set time $t = 18.00$, all customers get state 'Request'

2. Select from the customers with state 'Request' the one at the location with largest voltage and check if starting to charge the car of this customer is feasible subject to the network constraints.

    - If Yes, start charging this car, set state of this customer to 'Charging' and update voltage and current in the network for the charging time. If there are customers left with state 'Request' go to Step 2, otherwise we are Finished.
    - If No, set t to the next point in time were consumption decreases. If the decrease is caused by completing the charging of a car, set the state of the customer to 'Complete' Go to Step 2.

An alternative strategy is obtained by selecting a random customer from the set of 'Request' customers. If charging for this customer is infeasible in the network, we randomly select another customer. We repeat this until we have found a 'feasible' customer, or found out that no request can be fulfilled. In the latter case, we have to try again at the next point in time where energy consumption has been decreased. It is not hard to see that this approximates the situation where charging requests arrive in a random order.

6 Numerical Results

In this section we present numerical results illustrating the impact of distributed generation by solar panels and loading by the charging of electrical vehicles.

6.1 Distributed Generation by Solar Panels

Impact on Single Node Voltage Figure 8 shows the simulated evolution in time of the voltage of the node with four neighbours in the lower branch of the network for seven consecutive days. In this simulation, the power usage of all houses in the network was chosen to be the average power usage on a typical summer day. The bottom smooth line represents the voltage at the chosen node when no solar panels are present. The voltage at this node is significantly lower than the nominal value of 230 V, but still larger than the minimum required voltage of 207 V. The top oscillatory line represents the voltage at the same node when all houses have solar panels. The solar panels increase the voltage in the node up to 7 V, which is only a good thing, as the voltage was already quite low. The graph also shows steep jumps in the voltage. This could cause violation of voltage profile constraints. Figure 8 does not significantly change if one replaces the assumption of all house consuming an average load by a load according to a log-normal distribution.

Impact on Overall Network Performance For every combination of household loads and solar panel placement four characteristic days are simulated: a cloudy and a sunny day in both summer and winter. The network must satisfy all the requirements mentioned in Subsection 2.2 in all four conditions to be considered a success. We separately perform the experiment on the original network and on the network with the extra cable.

The scatter plot in Figure 9 displays the results of our Monte Carlo simulations. The solar panel penetration is plotted along the abscissa and the average household power demand is plotted against the ordinate. Both quantities disregard any information about the distribution within the network.

The household power demand has little influence on the network performance. On the other hand, solar panel penetration has a great impact on the network. For penetration levels below 70% the network is robust for any distribution of solar panels and household demand. Above 90% the network fails regardless of distribution, almost exclusively due to excessive jumps in the voltage level. For intermediate values the internal distribution of solar panels and household demand has an influence. But even then the average household usage is of little influence on network performance. par We studied the influence of adding a cable (the dotted line in Figure 2) to network performance by comparing the power loss in the original network to that in the network with the extra cable added. The household power usage and solar panel distribution were taken randomly in identical fashion to before, taking an average over the four typical conditions. However only the midday conditions are considered, saving considerable computational time.

The results are presented in Figure 10. It is immediately obvious the extra cable is beneficial to network performance. The power loss over the entire network is approximately halved, regardless of solar penetration.

The only influence of solar penetration on the network losses is that for intermediate values there is more variation in the distribution of solar panels throughout the neighbourhood and consequently there is some variation in the network losses. But this holds for both network configurations equally.

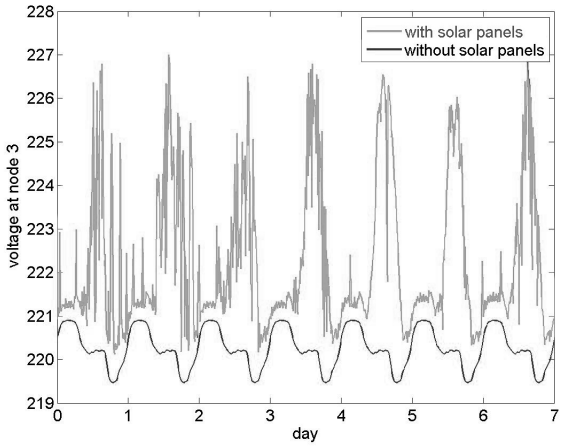


Figure 8: Voltage at the node with 4 neighbours in the lower branch of the network in Figure 2, where all houses have an average winter power usage pattern, with or without solar panels.

6.2 Loading by Charging of Electrical Vehicles

Charging with no Strategy We ran simulations for the charging of EVs between 06:00 and 22:00. We assumed average power usage on a winter day for every household. Subsequently, we randomly distributed different numbers of cars over the neighbourhood, where every household has at most one car charging at 3500W. The results are shown in Figure 11. The horizontal axis shows the number of cars in the neighbourhood. The vertical axis shows the percentage of distributions of cars that caused a network failure. For up to 10 cars, the network can still handle the load, but for larger numbers, the network may fail if these cars are wrongly distributed over the network. If all households have a car, the network will definitely fail. The same experiment was performed for simultaneous loading of cars between 24:00 and 06:00. In this simulation, no network failures occurred, for any number of cars in any distribution. Apparently the combination of high household power usage in the evening and the loading of a large number of electric cars at the same time is the cause of the problems. These results suggests that a control strategy that spreads the load

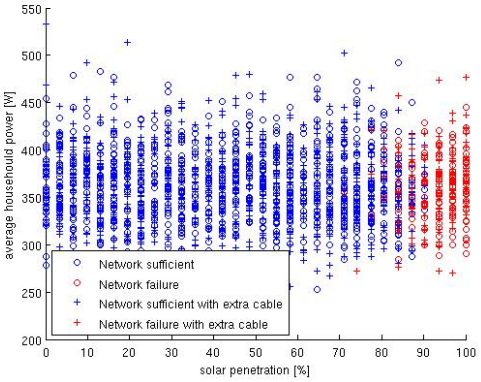


Figure 9: Scatter plot indicating network performance for combinations of solar power penetration and household power demand. Each case is run for one sunny and one cloudy day in both summer and winter. All network requirements must be satisfied in all conditions for a case to be considered a success (blue), otherwise it is considered a failure (red). The experiment was performed for the original network (o) and for the network with the extra cable (+).

due to carcharging will solve the network overload problems, while all cars can be fully charged in the morning.

Charging with Greedy Strategy We run simulations for the charging of EVs between 18:00 en 06:00, where we apply the greedy charging strategy described in Section 5. We assumed that everybody gets home at 18:00 and immediately plugs in their car. Each car is charged nonstop at full capacity for three hours. Each car should be fully charged before 07:00 the next morning. For simplicity, we assumed that each house has the exact same load profile. We run two simulations. One for the case that each house owns one car (Figure 12(a)), and one for the case that each house owns two cars (Figure 12(b)). Note that this second case is not realistic in the current setting, since each household has a maximum connection capacity of 25 A, thus exceeding the limit of 12 A. Charging two cars simultaneously would by itself require a capacity of 30 A. It does provide insight in the way the network handles charging that much cars using our greedy control strategy.

In Figure 12 we see that if each household owns one car, the network can handle charging almost all cars at once. Only at node three one or more cars have to await their respective turns. It takes six hours before all cars are fully charged.

If the demand is doubled, i.e. each household now owns two cars, much more planning is asked from our strategy. Now it takes nine hours before all cars are

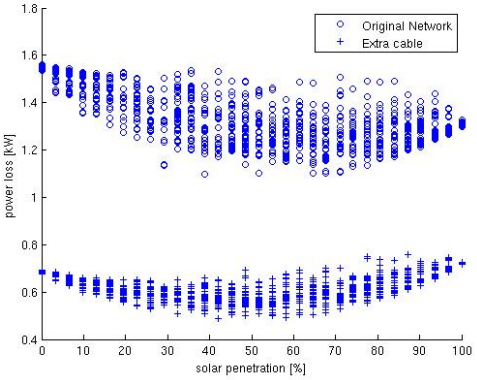


Figure 10: Network power loss in the original network compared to that in the network augmented with the extra cable.

fully charged, but still all cars are charged before 07:00.

7 Conclusions and Future Work

We examined the impact of the distributed generation by solar panels and of the load by the charging of electrical vehicles on the performance of a low-voltage distribution network servicing a residential area in Eindhoven, NL. Numerical results show that the penetration of solar panel usage is mainly limited by the requirement on the voltage variation between consecutive 10 minute intervals. For penetration levels above 70% the network ceases to be robust, independently of how the panels are are distributed over the households. The charging of electrical vehicles requires due care to prevent that the required power amplifies the evening peak in household loading. The greedy scheduling strategy we propose allows to fully charge up to two EVs per household before 07:00. Adding the extra cable in the network allows to approximately half the power loss in the entire network.

Further investigations can be directed to the improvements of our strategies by taking into account the possibility to use different speeds of EV charging; the possibilities for partial charging and/or break-down in charging; a pricing policy for households who desire to charge an EV during the peak time; the distances driven by a car per day; other types of decentralized generators; and finally the further development of smartness of the power grid.

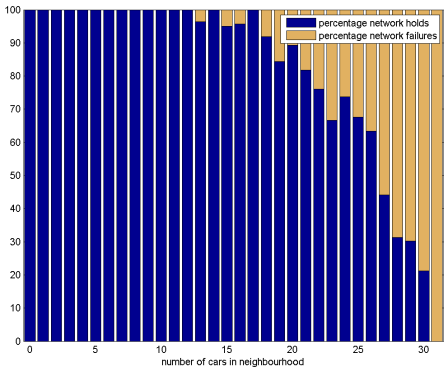


Figure 11: Percentage of distributions of cars that caused a network failure vs. number of cars attached to the network.

References

[1] http://www.cf.ac.uk/maths/subsites/esgi80/.

[2] S. Bhattacharyya. Future Challenges in Electricity Grid Infrastructure. Problem contributed by ENDINET to SWI 2012 Eindhoven, NL, Opening Presentation, 2012.

[3] L. Powell. *Power System Load Flow Analysis*. McGraw-Hill, 2004.

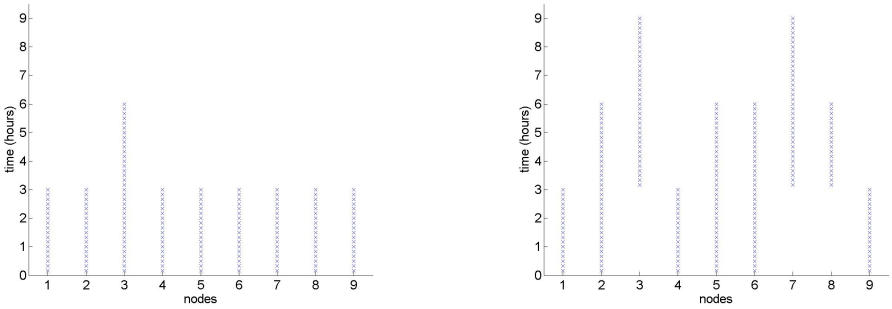
[4] REN21. Renewables 2010 global status report. Technical report, Renewable Energy Policy Network for the 21st Century, 2010.

[5] J. M. van den Akker, G. Bloemhof, J. Bosman, D. T. Crommelin, J. E. Frank, and G. Yang. Optimal Distributed Power Generation Under Network-Load Constraints. In *Proceedings of 72nd European Study Group Science with Industry 2010*, pages 25 – 38. CWI, 2010.

[6] R. A. Verzijlbergh, Z. Lukszo, J. G. Slootweg, and M. Ilic. The impact of controlled electric vehicle charging on residential low voltage networks. In *International Conference on Networking, Sensing and Control, 2011 IEEE*, pages 14 – 19, 2011.

[7] R. A. Verzijlbergh, Z. Lukszo, E. Veldman, J. G. Slootweg, and M. Ilic. Deriving electric vehicle charge profiles from driving statistics. In *Power and Energy Society General Meeting, 2011 IEEE*, pages 1 – 6, 2011.

[8] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas. MATPOWER: Steady-State Operations, Planning, and Analysis Tools for Power Systems



(a) One car per household.

(b) Two cars per household.

Figure 12: EV charging profiles. The zero on the y-axis corresponds to 18:00. The x-marks at node i denote that one or more EVs are charging at that node.

Research and Education. *Power Systems, IEEE Transactions on*, 26(1):12–19, Feb. 2011. ISSN 0885-8950. doi: 10.1109/TPWRS.2010.2051168. URL http://dx.doi.org/10.1109/TPWRS.2010.2051168.

Image Recognition of Shape Defects in Hot Steel Rolling

Evgeniya Balmashnova (Eindhoven University of Technology), Mark Bruurmijn (Eindhoven University of Technology), Ranjan Dissanayake (Rajarata University), Remco Duits (Eindhoven University of Technology), Mark Bruurmijn (Eindhoven University of Technology), Leo Kampmeijer (Tata Steel Research Development and Technology), Tycho van Noorden (Universität Erlangen-Nürnberg)

Abstract

A frequently occurring issue in hot rolling of steel is so-called tail pinching. Prominent features of a pinched tail are ripple-like defects and a pointed tail. In this report two algorithms are presented to detect those features accurately in 2D gray scale images of steel strips. The two ripple detectors are based on the second order Gaussian derivative and the Gabor transform, a localized Fourier transform, yielding the so-called rippleness measures. Additionally a parameter called tail length is defined which indicates to what extent the overall shape of the tail deviates from an ideal rectangular shape. These methods are tested on images from the surface inspection system at Tata Hot Strip Mill 2 in IJmuiden, it is shown that by defining a simple criterion in the feature space spanned by these two parameters a given set of strips can correctly be classified into pinched and non-pinched strips. These promising results open the way for the development of an automatic pinch detection system.

Keywords: Image analysis, Gabor transform, Gaussian scale space.

1 Introduction

One of the key installations on the IJmuiden steel production site is the hot strip mill. In this mill slabs with a length of between 5.5 and 12 m and a thickness of 225 mm, weighing between 8 and 34 tons, are heated up to a temperature of 1200 °C. Along a trajectory of about half a kilometer they are rolled in consecutive steps into strips of up to 2 km long and 2 mm thin. After the final reduction step they are cooled to a temperature in roughly the range 550-750 °C and coiled in generally less than 60 seconds. In this way 250 000 coils leave the hot strip mill each year for further processing and finally end up in construction, in lifting and excavating machines, in consumer goods such as white goods (refrigerators and stoves) and in the automotive and packaging industries.

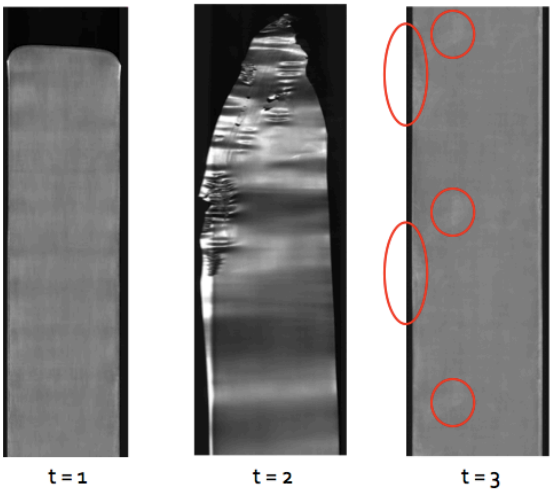


Figure 1: Pictures of parts of three consecutive strips. The first strip ($t = 1$) is without defects, the second strip ($t = 2$) is pinched and has damaged one of the mill rolls which is visible by the periodically occurring defects indicated on the next strip ($t = 3$).

The final reduction steps, where the strips can reach a speed over 20 m/s, are very critical. Errors in the gap settings of the mill may lead to defects in the shape of the strip. Especially at tailing out of the strip such shape defects may lead to pinches, damaging the strip as well as the rolls (see Figure 1), which then in turn need to be changed at significant costs.

Tail pinching is clearly visible by ripples in the strip and the pointed tail. In some case the strip surface is torn apart. It is not exactly known in what circumstances such pinches occur. To determine a statistical relation, and ultimately a causal relation, between certain process conditions and pinching, it is very useful to detect these pinches automatically. Once the mechanism is better understood, an online detection system might also be used to modify the process in order to prevent more pinches.

By means of a camera of the hot strip mill automatic surface inspection system, images of the strip tails are produced. In some of these images pinching is clearly visible. Automatic detection, however, is still problematic. Commercially available surface inspection systems need to be trained with categorized images. Once trained such a system computes the likelihood that a certain defect corresponds to a certain category on the basis of for example their dimensions, orientation on the strip surface and the gray scale distribution. In this way simple defects can be detected quite successfully. However, those systems are not able to detect complex defects such as tail pinches.

The goal of the assignment at hand is to create a method or algorithm that with a large collective of gray scale images can determine for individual strips:

- whether tail pinching has occurred or not;
- the location of the pinches relative to the image frame.

One of the challenges of this problem is that it is difficult to distinguish pinches from other ripple-like defects. It may even be that such shape defects and pinches occur simultaneously.

In this paper we show that for a given collection of strips it is possible to detect tail pinching successfully by considering the overall shape of the tail in combination with the presence of ripples with a certain wave length (Figure 2). For this purpose we have developed three different image analysis techniques to a collection of gray scale images of strip tails. First, we preprocess the images as discussed in Section 2. Subsequently, in Section 3.1, a simple technique, which proofs to be very effective, for estimating the tail shape counting background pixels is explained in more detail. For the detection of ripples we discuss two different methods in Sections 3.2.1 and 3.2.2. For both methods we make use of the fact that the ripples on the strips are always aligned horizontally. The first method tries to find the local maxima caused by the ripples in the vertical direction by looking at second order Gaussian derivatives [3] in that direction. The convolution kernel that is used to compute these second order derivatives contains a scale parameter that can be used to select a length scale at which the derivative information is computed and can be set using the expected bandwidth of frequencies at which the ripples occur. The second, alternative ripple detection method extracts local frequency information from horizontally averaged vertical bands of the images using a Gabor transform [1], which is a localized Fourier transform. Ripples are considered to be spots where high local frequencies are present, which can be assessed using the Gabor transform. In Section 4, we apply the introduced techniques to a collection of images of strip tails and we assess the effectiveness of the techniques for classification purposes. We build a feature space from the extracted features, which can then be used to to classify images into two useful categories. We end this paper in Section 5 with conclusions and recommendations.

2 Preprocessing

The upper image row is removed because it often contains gray artifacts. Since the images taken by the installed camera system are very large (around 8000×2000 pixels), we resize them to 10% of the original sizes (800×200). This allows us to speed up the process without loosing important features since in the rescaled images all the defects are still visible. Background is defined by setting a threshold. For the images we have set this threshold T to 0.15.

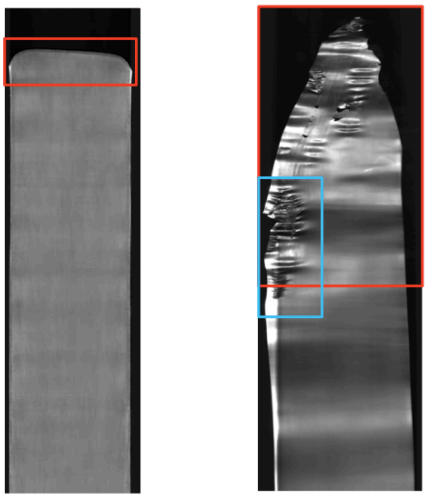


Figure 2: A strip without defects (left) and a pinched strip (right). The techniques discussed in this paper are designed to extract two features: the tail length, indicated by the red rectangles and the appearance of ripples, indicated in blue.

Note that we often consider an image as a continuous function in real space $\mathbb{R}^2$, and therefore we are able to integrate and differentiate. Since in reality we deal with discrete images, the integration is replaced with summation in the implementation of the discussed image analysis techniques.

3 Features

The aim of this project is to find features distinguishing pinched and normal strips. In this report we focus on two which we think are the most important ones - tail length and the presence of high frequency ripples.

3.1 Tail length

One of the main differences between normal and pinched strips is their tail shape. The normal ones have a more square shaped tail and the pinched ones have in most cases an elongated tail (see Figure 2). In order to distinguish these two basic shapes we introduce the parameter *tail length* defined as the distance between the top point of the tail and the place where strip reaches its full width. In order to find this length we compute number of background (black) pixels for each image row in the image. In the lower half of each image normally the width of the strips is almost constant and therefore we estimate from the lower half of

the image the full strip width R by taking the median value of the width in this part of the image (the median is less sensitive to outliers than the mean).

We start scanning the image (size $N \times M$) starting from the top row of pixels. Since in the top row all the pixels are black (gray value below T), the number of black pixels will be equal to N, the length of the image in $x-$dimension. The first row with less than N background pixels is the position of the top point of the tail. The position where the width exceeds R for the first time is the end point of the tail, see Figure 3.

1. For each row i count $B(i)$, the number of pixels with value lower than T, for $i = 1, \ldots, M$
2. i_1 - is the first row (from the top) where $B(i) \neq N$
3. i_2 - is the first row where $B(i) < N - R$.
4. Tail length $L := i_2 - i_1$

In some situations, for example when shadows appear in the image, there may be a lot of black pixels inside the strip, which leads to a wrong estimate of the tail length. The way to improve this feature is to compute the width for each row. By width we mean the distance between the first and the last point with nonzero (non-black) value in an image row. The rest of the procedure is the same. For most of the images it yields the same length (see Figure 4), but for some, with a lot of shadows, the second approach gives more accurate results (see Figure 5).

Although the second approach is more accurate and will be used as a feature for distinguishing between normal and pinched strips, the first approach still has some potential. For example, the difference between two approaches to estimate the tail length, based on the width and the number of non-black pixels, can be used for detecting strips with shadows or holes.

3.2 Rippleness

Tail length alone cannot completely separate pinched and good strips since there are cases when a strip has a normal shape but due to ripples is called a pinched one and there are cases then tail estimation algorithm fails due to shading, see Figure 6. Another feature that distinguishes between normal and pinched strips is the appearance of ripples with high frequency, which mainly occur on pinched strips. We would like to construct a measure of *rippleness* as well as to detect their locations. Ripples are characterized by fast periodic changes of gray-values in vertical direction.

3.2.1 Gaussian derivative

Fast changes of gray values lead to high second order derivative in vertical direction. Derivatives can be computed in a well-defined and well-posed way by using

scale space theory [3] Given a two-dimensional image f, $f : \mathbb{R}^2 \to \mathbb{R}$, its scale representation $u : \mathbb{R}^2 \times \mathbb{R}^+ \to \mathbb{R}$ is defined by the convolution product

$$u(\mathbf{x}, \sigma) = (g(\cdot, \sigma) * f)(\mathbf{x}) \ , \tag{1}$$

in which g is the Gaussian kernel given by

$$g(\mathbf{x}, \sigma) = \frac{1}{2\sigma^2\pi} \exp(-\|\mathbf{x}\|^2/2\sigma^2) \ . \tag{2}$$

Scale space theory allows one to interchange differentiation and convolution, viz. as follows:

$$\partial^\alpha (f * g(\cdot, \sigma))(\mathbf{x}) = (f * \partial^\alpha g(\cdot, \sigma))(\mathbf{x}) \ . \tag{3}$$

This property allows implementation to be fast and robust. We compute the second order Gaussian derivative in$y$-direction $u_{yy}(\mathbf{x}, \sigma)$ with reflective boundary conditions [2]. By choosing the scale σ we can actually tune to the frequencies we are interested in. A maximum filter is applied to the resulting two-dimensional Gaussian derived image, replacing every value by the maximum pixel value in a chosen neighbourhood (in our case 5 pixels).

In order to introduce a measure for the amount of vertical variation for the whole image, we compute the area in the image where the response is higher than some threshold (in our case 0.03). A drawback of this method is that the Gaussian derivative gives a response to any changes, not only periodic ones, therefore edges also give high values. However, the difference between those two is that edges occupy a small area compared to ripples. Images with ripples will have a larger area with responses above the threshold As can be seen from Figure 7 this gives a clear distinction between rippled and flat strips. In the next section we consider another approach, in which frequencies are taken into account in a more explicit fashion, namely the Gabor transform approach.

3.2.2 The Gabor transform

Because ripples occur only locally, the Fourier transform of an image does not not give a clear peak at the frequency of the ripples, i.e. detection is practically impossible. This is directly related to the fact that the ripples are local. The Gabor transform [1] is a windowed-Fourier transform, which is well suited to find local frequencies. We restrict ourself to the one-dimensional case, since we are looking for harmonic behavior in the vertical direction only. Applying the two-dimensional Gabor transform to the whole image is computationally too heavy. Therefore we cut the images into K bands (we used 20 for the visualization purposes, but it can be less), and for each band we construct a one-dimensional intensity profile f_i, $k \in [1, \ldots, K]$ by summing up all values in every row.

Continuous case Although images are always discrete, the theory we use is for the continuous case. Later we will translate it to the discrete case. The Gabor

transform $G_\psi(f) : \mathbb{R} \times \mathbb{R} \to \mathbb{C}$ of a signal $f \in \mathbb{L}_2(\mathbb{R})$ is given by

$$G_\psi(f)(y, \omega) = \int f(\xi)\overline{\psi(y - \xi)}e^{-2\pi i(y-\xi)\omega}\mathrm{d}\xi \,, \tag{4}$$

where ψ is the Gaussian kernel in $1D$,

$$\psi(y) = \frac{1}{2\sigma^2\pi}\exp(-y^2/2\sigma^2) \,. \tag{5}$$

The Gabor transform yields a complex-valued spectrum in each spatial position, which can now be used to find positions in which high frequencies are present (i.e. ripples). However, we are not interested in the phase of the frequency pattern, but only in its amplitude. Therefore, we define the probability density function of finding a certain frequency ω at position y as:

$$U_k(y, \omega) = |G_\psi(f_k)(y, \omega)|^2, \; k = 1, \ldots, K \,. \tag{6}$$

Discrete case For the images vertical coordinate y takes only integer values, $y \in [1, \ldots, M]$ and 4 has a summation instead of integration

$$U_k(y, \omega) = \left| \sum_{k=0}^{M} f(k)\overline{\psi(y - k)}e^{-2\pi i(y-k)\omega} \right|^2 , \tag{7}$$

$\omega \in [-M/2, \ldots, \ldots, M/2]$. Due to the symmetry around zero, for the computation it is enough to consider only positive side of the spectrum. Locations with ripples differ from normal ones by giving high response for certain ω values (Figure 8).

We define the first moment by

$$E_k(y) = \sum_{\omega=0}^{M/2} U_k(y, \omega)\omega, \; k = 1, \ldots, K \,. \tag{8}$$

It gives larger weight to high frequencies and allows to filter zero frequency response in a soft way. The first moment for locations in a flat area will be close to zero, since only frequencies in the neighborhood of zero are present. For a rippled area the first moment will be shifted to a higher value (Figure 9). Therefore the first moment is expected to be a reasonable measure of rippleness. We use the total response $\sum_{k=1}^{K}\sum_{y=1}^{M} E_k(y)$, where we sum over every cut band, of an image as a measure of severeness of the ripple defects.

4 Classification

We have discussed two measures: tail length and rippleness. For rippleness we introduced two methods, which are highly correlated and it is subject for further

experiments to decide which one is more favorable for classification purposes (we use the Gaussian derivative for further experiments in this paper). Note that both methods have their advantages. For example, the Gaussian derivative requires less computational power but with the Gabor transform specific frequencies can be detected. In Figure 10 a two-dimensional feature space is presented. As one can see, the defected strips (green dots) are, as expected, more scattered in feature space. The straight line in the figure represents a possible and simple manner to classify the images of the strips into two classes: normal strips and strips with a defect. We have a set of images with labels "normal" and "pinched", with 163 normal and 21 pinched strips. For the labeled training set we take the line $y = ax + b$ (for the collection we have $y = 8000 - \frac{8000}{125}x$) that separates the most normal strips (96.3% are true negative) from strips with a defect (for the 3.6% false positive results, see Figure 11), without creating false negatives (0% false negative). The classification has not been considered thoroughly and may be improved further. For new cases the decision is taken by determining on which side of the line the image is in feature space. We would like to stress that, apart from classification use, the introduced features could also indicate severeness of the pinching as shown in Figure 12-13.

5 Conclusions and recommendations

In this paper features of two different types have been considered: the tail length and the presence of ripples. For each of them two different approaches are given and compared. Although all of them have some advantages, only one of each type was chosen for the classification (the tail length based on width estimation and rippleness using Gaussian derivatives), and a simple, but effective, classification rule has been considered. As a result we have constructed an efficient algorithm with which one can:

- distinguish tail shape;
- detect and locate ripples;
- effectively classify pinching.

We propose several points for further research:

1. Based on the proposed features, a measure of severeness of the defect can be introduced.
2. Improvement of existing features can be done:
    - In the Gabor domain the important information is hidden by fluctuations. Applying (contextual) enhancements in the Gabor domain before extracting features such as the 1st order moment Equation (8) can clean up the signal and improve the results;

- Rippleness measure requires some parameters, like scale for the Gaussian window. It is an open question whether just one scale should be fixed, or several of them can bring more information. Therefore rippleness can be extended to a multi-scale framework: edge focussing, automatic scale selections, tracking critical path of the Laplacian in scale space.

3. Consider other features in the Gabor domain, e.g. other moments or responses to the certain range of frequencies, corresponding to the bandwidth of the actual ripples (this would correspond to changing the summation limits in Equation 8 on $[\omega_0, \omega_1]$, where the limit values are obtained from statistics of ripple frequencies).

4. Classification in feature space should be explored further. A basic linear discriminant analysis (as in Sec. 4) would probably be sufficient.

5. New features can be introduced:

   - Deformation modeling. Unfortunately, the process does not allow imaging of an evolving process. So use idealized flat image as reference for rough single step deformation estimates. The amount of deformation needed to obtain the image from the ideal can tell about the severeness of the defect.
   - 2D-filtering to detect endings of ripples and scratches.
   - Include contextual filters via the Gabor domain. It is important to check whether the surrounding local frequencies are coherent/aligned or not. This could distinguish between folding and ripples. If some frequency appear over the complete x-axis than it is a fold and it can be filtered out.
   - New applications. Ripple detection algorithms are developed in this work can also be applied to other wave-like shape defects over other parts of the strip, and hence can be used to distinguish wavy edges, quarter buckles and center buckles much more accurately than with the standard optical shape meter at the hot strip mill.

References

[1] D. Gabor. Theory of communication. part 1: The analysis of information. *Journal of the Institution of Electrical Engineers - Part III: Radio and Communication Engineering*, 93(26):429–441, november 1946. doi: 10.1049/ji-3-2.1946.0074.

[2] B. M. t. Haar Romeny. *Front-End Vision and Multi-Scale Image Analysis: Multi-Scale Computer Vision Theory and Applications, written in Mathematica*, volume 27 of *Computational Imaging and Vision Series*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2003. ISBN 1-4020-1507-0.

[3] T. Lindeberg. On the axiomatic foundations of linear scale-space. In J. Sporring, M. Nielsen, L. M. J. Florack, and P. Johansen, editors, *Gaussian Scale-Space Theory*, volume 8 of *Computational Imaging and Vision Series*, chapter 6, pages 75–97. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1997.

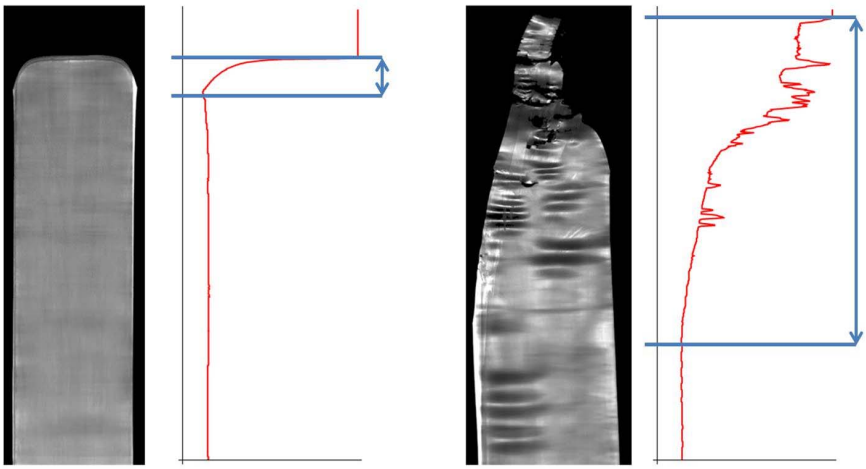


Figure 3: Tail length estimation for a strip without defects (left) and a pinched strip (right): red graph represents number of background pixels per row.

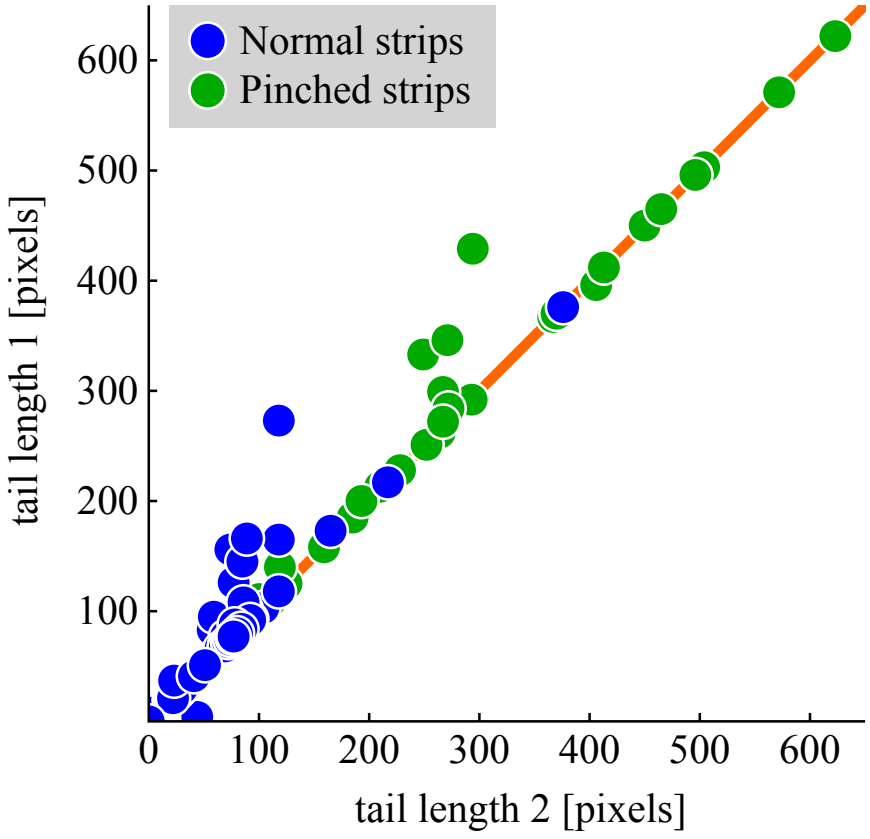


Figure 4: Two different tail length estimations, based on width estimation on x-axis and based on black pixels on the y-axis. Green dots represent pinched strips and blue dots normal strips.

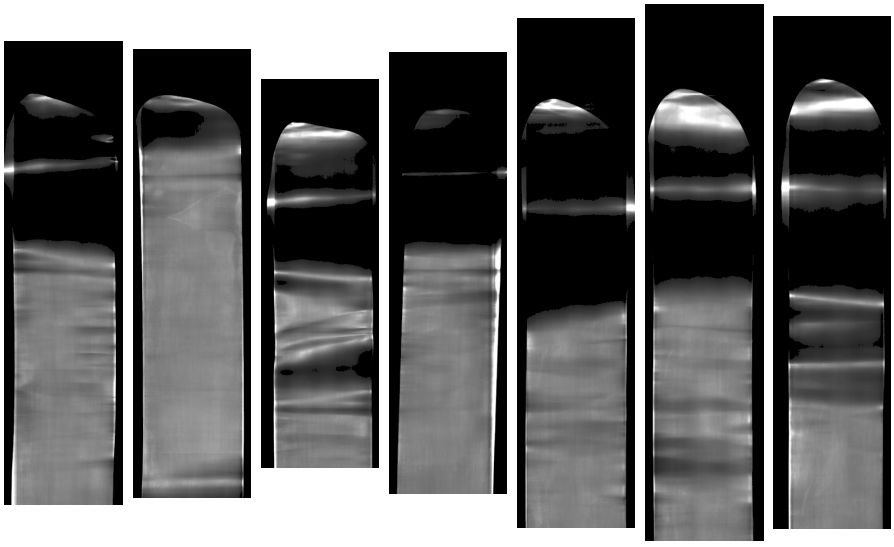


Figure 5: Cases when computing width directly leads to better tail length estimation.

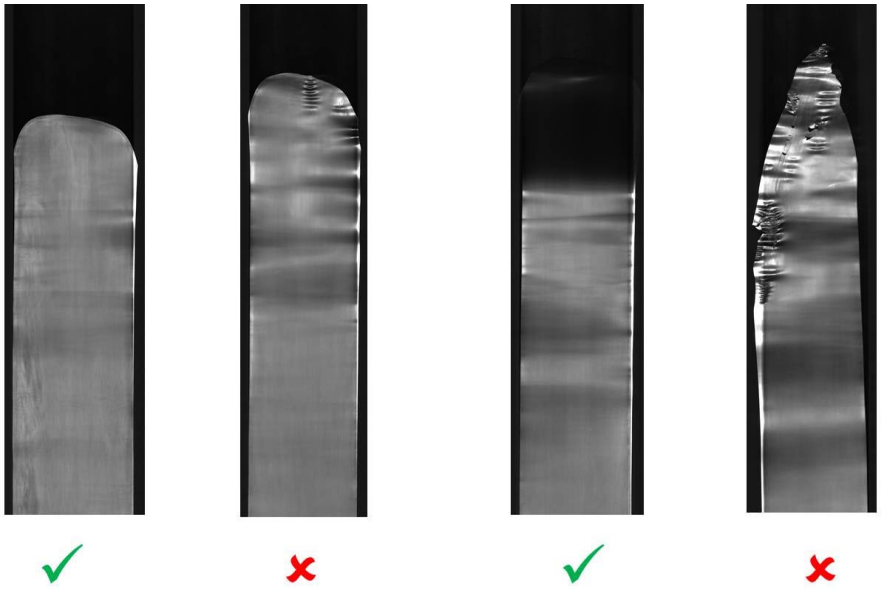


Figure 6: Tail length fails to separate pinched strips from the rest.

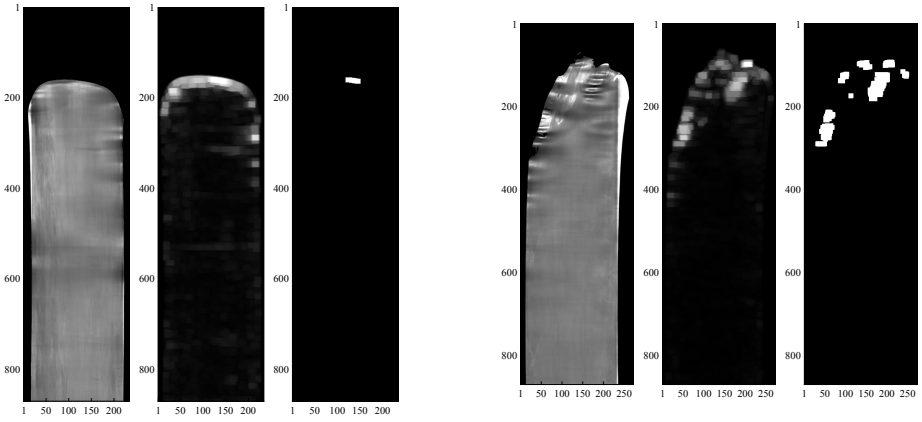


Figure 7: Results (before and after thresholding) of applying the Gaussian derivative to normal (left) and to pinched (right) strips.

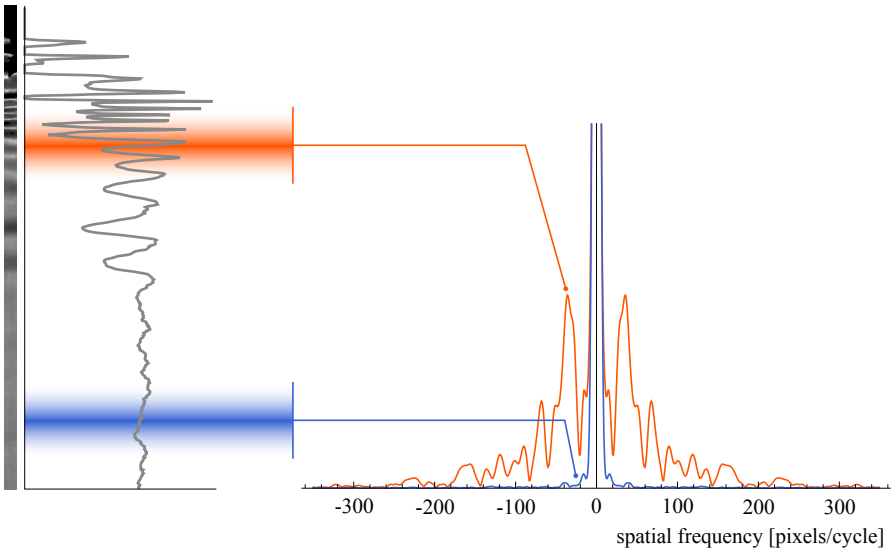


Figure 8: Left: band cut from pinched strip image (870×27 pixels) and its intensity profile by summing over rows. Right: local spectra from the two indicated regions in the intensity profile. The rippled region (orange) contains more high frequencies than the flat region (blue).

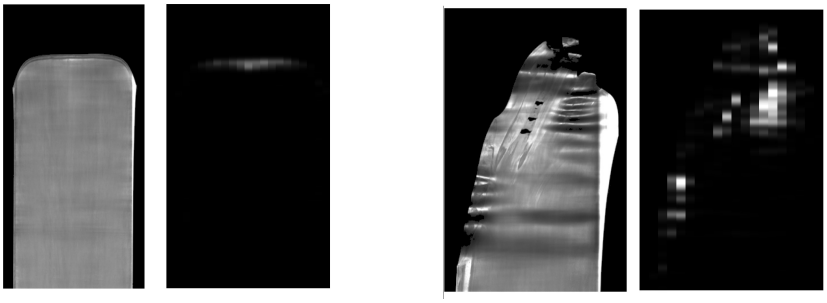


Figure 9: The first moment for a normal and a pinched strips.

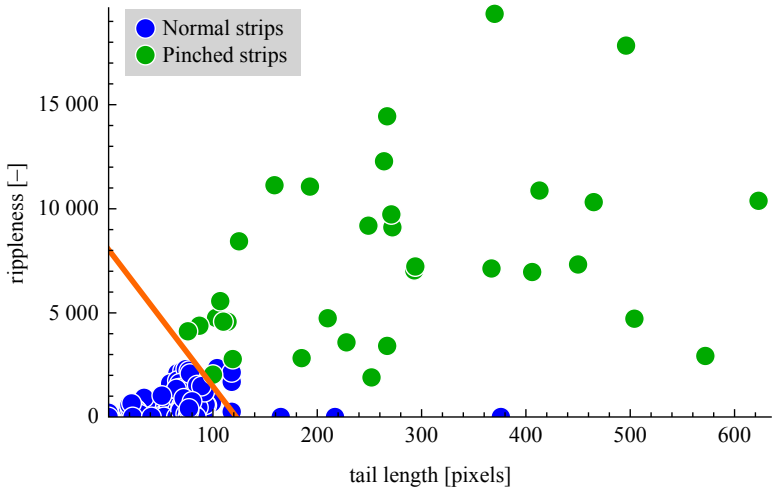


Figure 10: Feature space, tail length on x-axis and rippleness (using Gaussian derivatives) on y-axis. Green dots represent pinched strips and blue ones normal strips.

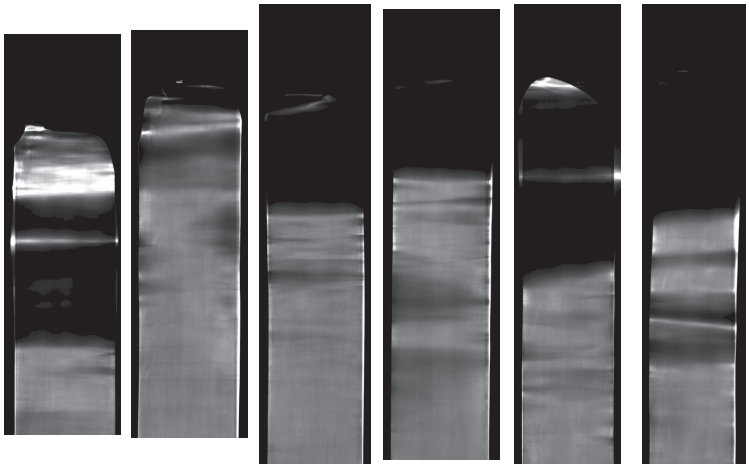


Figure 11: False positive classification results.

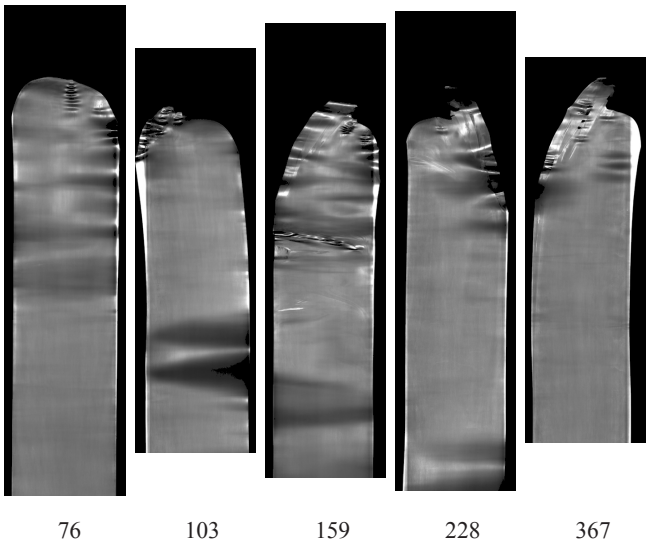


Figure 12: Pinched strips sorted according to their tail length.

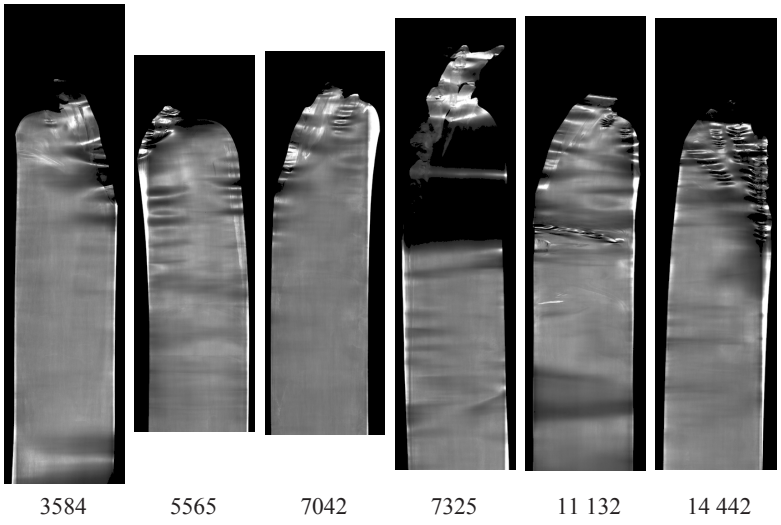


Figure 13: Pinched strips sorted according to the severeness of their ripples.

Optimization of Lifetime in Sensor Networks

Nikhil Bansal (Eindhoven University of Technology), David Bourne (Eindhoven University of Technology), Murat Firat (Eindhoven University of Technology), Maurits de Graaf (Thales), Stella Kapodistria (Eindhoven University of Technology), Kundan Kumar (Eindhoven University of Technology), Corine Meerman (Leiden University), Mihaela Mitici (University of Twente), Francesca R. Nardi (Eindhoven University of Technology), Björn de Rijk (Leiden University), Suneel Sarswat (National Institute of Securities Markets, India), Lucia Scardia (Eindhoven University of Technology)

Abstract

We consider the problem proposed by Thales Nederland at the SWI 2012 meeting. Thales Nederland is the Dutch branch of the international Thales Group. The company specializes in designing and producing professional electronics for defence and security applications, such as radar and communication systems. Moreover, Thales Nederland acts as a local point of contact for the complete portfolio of the Thales Group.

During the SWI 2012 meeting, on behalf of Thales Nederland, Dr. Maurits de Graaf posed several questions regarding the maximization of the lifetime of a wireless sensor network. We addressed these questions during the workshop and our most significant results are summarized as follows: We have proven that this lifetime maximization problem, even under the most simple constraints, is NP-complete, so it is not possible to find an algorithm that gives the optimal solution within polynomial time. Furthermore, we have constructed a counterexample to illustrate that the heuristic currently used by Thales can be asymptotically at least $\log n$ times worse than the optimal solution. Moreover, we have developed a new heuristic and have illustrated with several numerical examples that it performs better than the heuristic currently used by Thales. Finally, we have formulated the problem as a linear programming problem and used this to quantify how far the heuristic used by Thales is from being optimal.

KEYWORDS: ad hoc networks, network lifetime, multipoint relay selection, linear programming.

1 Introduction

One of the most important examples of wireless ad hoc networks are wireless sensor networks. Sensor networks can be dynamic, i.e. the topology of the wireless network can change over time (e.g. networks of wearable communication

systems), or static (e.g. sensors in a forest for detecting fires). Sensor networks occur in different real-life settings, for example, in the military, in emergency services or in radar systems (see, e.g., [2] and its references). In all the aforementioned situations it is of vital importance that all nodes in the network can always communicate with each other. We assume that each node in a wireless ad hoc network is equipped with an (omnidirectional) antenna and with a battery of limited capacity. The constraint of limited battery capacity is one of the most important features of sensor networks. Therefore it is essential to develop networking algorithms and protocols that are optimized for energy efficiency.

Usually these wireless ad hoc networks are spread over very large areas. Under this condition one has to employ routing techniques so that the out-of-range nodes can communicate with each other via intermediate nodes. The problem of routing in wireless networks is addressed using different routing protocols. These protocols are distinguished into reactive (finding the route to a destination when it is needed) and proactive (periodically exchanging control messages, which inform on the local or entire topology of the network). The proactive protocols immediately provide the required routes when they are needed, at the cost of bandwidth and battery consumption. Over the last decade the most commonly used proactive protocol is the Optimized Link State Routing (OLSR) (see, e.g., [3]). The OLSR protocol is an optimization strategy for wireless networks that reduces the number of control messages and minimizes flooding of this control traffic by using only a subset of the nodes to send the messages through the system. Each node in the network selects a set of nodes in its neighborhood to retransmit each message it broadcasts. This set of selected neighbor nodes is called a *multipoint relay set* (MPR-set) of this node. The neighbors of any node that are not in its MPR-set receive its messages but do not retransmit them. Each node v selects its MPR-set among its 1-hop neighbors in such a way that messages sent from v are relayed by its MPR-set to all nodes that are two hops away (in terms of radio range). The smaller the MPR-set, the more optimal the traffic control of the network. The MPR-set can change over time, in accordance to the changes of the network topology over time. For the sake of simplicity, in the rest of the manuscript we will consider a stationary network, i.e. we assume that the topology of the network does not change over time.

Many research papers aim at optimizing the selection of MPRs with a specific purpose in mind, e.g., to minimize the number of MPRs used, to keep paths with high Quality of Service, or to maximize the network lifetime (the time until the first node runs out of battery power or the first time at which a communication fails due to battery depletion). In this manuscript we will focus on the maximization of the network lifetime. We adopt the definition of network lifetime as the time until the first node fails due to battery depletion (see, e.g., [9] and its references). Our aim is two-fold: on the one hand to check the non-optimality of the existing algorithms that select relay nodes with the objective of maximizing

the network lifetime, and on the other hand to propose a better heuristic. We will carry out our analysis in two levels distinguishing whether we fall within the specifications of an OLSR network or not.

Consider a group of wireless static nodes randomly distributed in a region, where each node has its own battery supply used mainly for the transmission of messages. We assume that for each transmission (independently of whether this is the initialization of a transmission or a message forwarding) the battery level is reduced by a *fixed* amount. This linear battery model approach is a simplification compared to reality, where the batteries have a recovery time. The node initializing the data transmission is called the source. After the source has sent the message, which will be received first by its neighbors and eventually by all the other nodes in the network, another node becomes the source. We assume that this process continues until all the nodes have acted as a source, which means that a round has been completed. The order of the sources during a round may be prescribed or random.

This problem was brought to our attention during the SWI 2012 meeting on behalf of Thales Nederland, by Dr. Maurits de Graaf. The following directions of investigation were proposed regarding the maximization of the lifetime of a wireless sensor network:

Direction 1: What would be the optimal MPR selection algorithm? With a linear battery decrease model it should be possible to formulate this as a linear program. How much does the optimal solution differ from the known heuristics? Can we define easily a better heuristic than the Maximum Willingness heuristic algorithm used by Thales?

Direction 2: Assume additionally that a node can choose between different power levels. For a higher power level a node will have larger set of neighbors to choose its MPR-set from. Can we formulate the optimization problem and find some good heuristic to solve it (a solution being an assignment of transmit powers and MPR-sets)? What would be the impact on the network lifetime?

Direction 3: What is the effect on the network lifetime problem when using a battery model with a recovery effect?

The paper is organized as follows: In Section 2 we formulate the problem. In Section 2.1 we present the Maximum Willingness heuristic algorithm used by Thales. The rest of the paper is divided into two parts. In the first part (Section 3) we treat the problem of selecting the relays of a network having a general topology in order to maximize the network lifetime. In the second part (Section 4) we treat the same problem within the OLSR framework.

More specifically: In Section 3.1 we show that outside the OLSR framework

the problem under consideration is NP-complete, so it is not possible to find an algorithm that gives the optimal solution within polynomial time. We do this by reducing our problem to the Set-Cover problem, which is a known NP-complete problem. Furthermore, in Section 3.2 we construct a counterexample to illustrate that the Maximum Willingness heuristic does not provide the optimal solution to the problem, which was expected since the problem is NP-complete (unless P=NP). In Section 3.3 we present a new heuristic and in Section 3.4 we illustrate with several numerical examples that our algorithm performs better than the Maximum Willingness heuristic. Furthermore, in Section 3.5 we formulate the problem as a linear programming problem (LP). In Section 4 we work inside the OLSR framework. In Section 4.1 we show that the Maximum Willingness heuristic can be at least $\Omega(\log n)$ times worse than the optimal solution. Furthermore, in Sections 4.2 and 4.3 we present two LPs for optimizing the network lifetime. Finally, in Section 4.4, we conclude our analysis with some numerical results comparing the Maximum Willingness heuristic with the optimal solution. We close with discussions and extensions in Section 5.

2 Formulation of the Network Lifetime Problem

Let $G = (V, E)$ be a connected graph with n nodes, where V denotes the set of nodes and E the set of edges. A network transmission (a message circulated through the entire network) is defined as a time slot, in the sense that time is updated by 1 when a message is sent to all the $n-1$ nodes of the network starting from any given source, i.e. for every time r there is one single node acting as source, say $s_r \in V$, that sends the initial message. This initial message is then forwarded through the entire network. Hence, time $r \in \mathbb{N}_0$ represents the total number of different sources that have sent a message across to the entire network.

A battery level $B_v(r)$ is associated with every node v at time r. We assume that the battery levels are reduced by a fixed amount for every message sent. For simplicity this fixed amount is chosen to be 1.

If we denote by $R(r)$ the set containing the source at time r and all relay nodes that forward the message at time r, then the battery level evolution can be described as follows: for every $v \in R(r)$ we have $B_v(r+1) = B_v(r) - 1$. For all other nodes $B_v(r+1) = B_v(r)$.

Instead of only focusing on minimizing the battery consumption of the individual nodes, it is important, especially in emergency situations, to maximize the network lifetime l, i.e. the first time at which at least one of the batteries is flat:

$$l = \min\{t : B_v(r) = 0 \text{ for at least one } v \in V\}.$$

Note that l depends on the algorithm that is used to circulate the message through the entire network.

The problem can be formulated as follows: Given a connected graph G, a starting distribution of battery levels and a source assignment protocol, choose the relays used to circulate the messages so that the network lifetime is maximized:

$$L = \max\{\min\{r : B_v(r) = 0 \text{ for at least one } v \in V\}\}$$

subject to

$$B_v(r+1) = \begin{cases} B_v(r) - 1, & \text{if } v \in R(r) \\ B_v(r), & \text{otherwise.} \end{cases}$$

2.1 Thales Approach: Maximum Willingness Heuristic

In order to solve the problem stated above we need to have a heuristic that provides us with an optimal selection of relays at each time. Many algorithms address the network lifetime problem in general topology networks. The most relevant for this paper is the Maximum Willingness (MaxWill) algorithm. In this section we closely follow the notation and terminology of [9].

Let $G = (V, E)$ be a connected graph, where V denotes the set of nodes, E the set of edges and n is the number of nodes, i.e. $|V| = n$. Furthermore, let $N^m(v)$, $v \in V$, denote the strict m-hop neighborhood of node v, i.e. the set of nodes for which the shortest path to v has exactly m edges. A subset $M(v) \subset N^1(v)$ is called an MPR-set if $M(v)$ dominates $N^2(v)$, i.e. each node in $N^2(v)$ has a neighbor in $M(v)$. Furthermore, for a given MPR-set $M(v)$, we call each node in this set an MPR node of v. Finally we denote the set of all possible MPR-sets of v by $MPR(v)$.

The MaxWill MPR selection algorithm uses the following structure to calculate an optimal MPR-set $M(v)$ for node v:

1. Start with an empty MPR-set for node v and add nodes of $N^1(v)$ that are the only neighbors of some node in $N^2(v)$.

2. While there are still uncovered nodes in $N^2(v)$, select the nodes from $N^1(v)$ that cover at least one uncovered node and have the highest remaining battery level.

3. Optimize the MPR-set by attempting to remove a node from $M(v)$ and checking if $N^2(v)$ is still dominated. If this is the case, the node is removed from $M(v)$. Nodes are removed in the order lowest remaining battery level first.

Note that the MaxWill MPR-selection algorithm is a localized algorithm, since each node $v \in V$ selects an MPR-set $M(v)$ independently from the other nodes.

Thales uses the MaxWill algorithm and as demonstrated in [9] this algorithm performs better in most cases than other MPR-selection algorithms.

3 Our Approach: Outside the OLSR Framework

We investigate the impact of the selection of the relays on the lifetime of the sensor network. In this section we do not enforce the OLSR constraint that messages should be broadcast in layers (see Section 4 for more details about this constraint). In Section 3.1 we prove that the Network Lifetime problem is NP-complete. Our ideas are similar to those of [7] and [10], who consider related (but not exactly the same) problems. In Section 3.2 we emphasize the weaknesses of the MaxWill heuristic introduced by [9]. We show that for small networks the MaxWill heuristic is outperformed by a path-based heuristic, which is described in Section 3.3. The path-based heuristic attempts to maximize the network lifetime by avoiding using nodes with low battery level, if possible. In Section 3.4 we give simulation results for the path-based algorithm and the MaxWill algorithm. In Section 3.5 we formulate the relay selection problem as a LP.

3.1 Proof of NP-completeness

We will show that the problem of maximizing the network lifetime is not only NP-complete, but that it is also hard to approximate within a factor of $\Omega(\log n)$, where one writes $f(n) = \Omega(g(n))$ as $n \to \infty$ for two functions f and g if and only if there exists a positive real number M and positive integer n_0, such that $|f(n)| \geq M|g(n)|$, for all $n \geq n_0$.

We give a reduction to the so-called maximum domatic partition problem (see [5]), defined as follows. Given a graph $G = (V, E)$, a *dominating set* of G is a subset $S \subset V$ such that each node $v \in V$ is either in S or has a neighbor in S. The domatic number of G is the maximum number of dominating sets into which the vertices V can be partitioned. Let k be the maximum number of disjoint dominating sets contained in the graph G. [5] showed that for every $\epsilon > 0$, no polynomial times algorithm can approximate the domatic number problem within a $(1 - \epsilon) \log n$ factor, unless NP has a slightly superpolynomial-time algorithm, i.e. unless $NP \subseteq DTIME(n^{\log \log n})$. Hence, it is impossible to approximate the size of the maximum domatic partition to a factor better than $\Omega(\log n)$. In particular, for any n, there are instances such that no efficient algorithm can distinguish whether the domatic number is at least k or at most $O(k/\log n)$. In fact, [5] show the following stronger result (which is the one we need): No efficient algorithm can distinguish whether the domatic number is at

least k, or whether there is a dominating set of size $O((n \log n)/k)$. Note that this is a much stronger hardness result, as the domatic number must be at most $O(k/\log n)$ if the minimum dominating set has size $\Omega((n \log n)/k)$.

We will use the above problem to show the $\Omega(\log n)$ hardness of the network lifetime problem. Let G be a hard instance of the domatic partition problem, and n denote the number of vertices in G. We will construct an artificial graph $\hat{G}$ with $2n$ vertices, which is equivalent to G. The artificial graph is constructed as follows: For each vertex $i \in G$, we define two vertices $(i,1)$ and $(i,2)$. We think of the second index as defining a layer. So there are two layers. For every $1 \le i < j \le n$, we connect the vertices $(i,1)$ and $(j,1)$ by an edge (i.e. all vertices in the first layer form a clique). For each vertex $(i,2)$, we connect it to the vertex $(j,1)$ whenever (i,j) is an edge in G or if $j = i$. Note that a vertex in the second layer is not connected to any other vertex in the second layer. Hence, the graph $\hat{G}$ has k disjoint dominating sets. Let the initial battery levels of $\hat{G}$ be $(2n/k+3)B$ for layer 1 vertices and B for layer 2 vertices. When each vertex has had its turn being a source once, we call that a round.

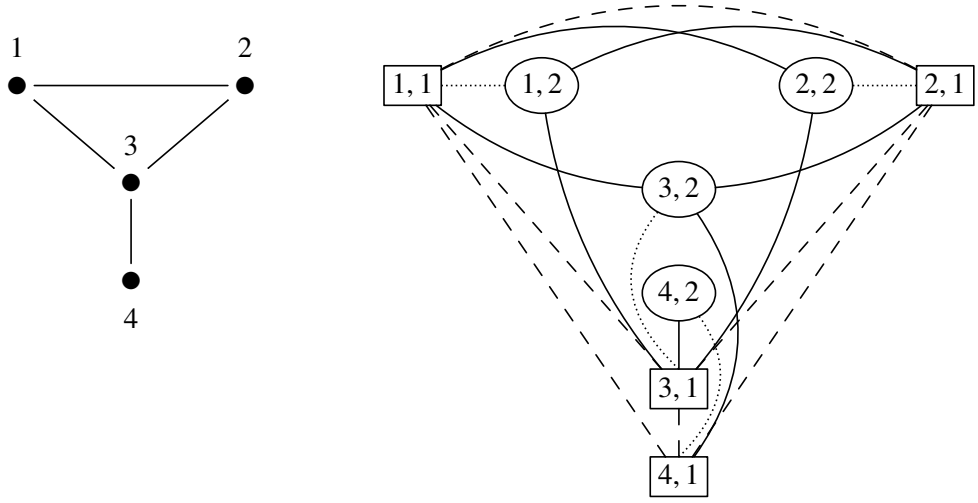


Figure 1: Example of the construction of $\hat{G}$. In this case $k = 2$ and $C_1 = \{3\}$, $C_2 = \{2,4\}$ are examples of disjoint dominating sets.

Claim 3.1. *If the domatic number is k, the source can transmit for B rounds (i.e. optimum lifetime is at least B).*

Proof. Let $C_1, \ldots, C_k$ denote the k disjoint dominating sets of $\hat{G}$. When the source is a layer 1 vertex, say $(i,1)$, it first directly transmits its message to all the layer 1 vertices. This can be done as layer 1 vertices form a clique. Then the nodes

$(j,1)$ such that $j \in C_{i'}$ where $i' = i \bmod k$ transmit the message to all the nodes in layer 2. Note that this is possible as each $C_{i'}$ is a dominating set. Next, the strategy when the source is a layer 2 vertex, say $(i,2)$, is to first transmit its message to $(i,1)$ and then the above strategy is followed.

Now note that during each round, each vertex $(i,2)$ in layer 2 transmits exactly once when it is the source. Each vertex $(i,1)$ in layer 1 transmits exactly once when it is the source, once when it is the relay for the corresponding node $(k,2)$ from layer 2, and at most $2n/k+1$ times as a relay in some dominating set. The latter is because the dominating sets are disjoint (they form a partition), and each collection $C_{i'}$ is used $\lceil 2n/k \rceil$ times. By the choice of our initial battery levels, it is easily checked that this strategy lasts for at least B rounds. □

We now show the other direction, i.e. if the lifetime is high, then there is a small dominating set. In particular,

Claim 3.2. *If the network lifetime is cB, for some constant $c \in [1, \lceil 2n/k \rceil]$, then the domatic number is at least $2n/(ck)$.*

Proof. Observe that a layer 2 vertex $(i,2)$ can receive a message only via some layer 1 node. Thus, no matter which node is the source, to transmit its message to all nodes, the relays on layer 1 must form a dominating set. So in each round, nodes from at least $2n$ dominating sets must transmit. Let s be the size of the minimum dominating set. If the network lifetime is at least cB, this means that the total battery power used by the layer 1 nodes is at least $2n \cdot s \cdot cB$. On the other hand, the total initial battery power of these nodes is $(2n/k+3)B \cdot n$. This implies that $2nscB \leq n(2n/k+3)B \leq 4n^2/kB$, and hence $s \leq 2n/(ck)$. □

Given the hardness result of [5] stated above (the stronger form) and the equivalence of the original graph G and the artificial graph $\hat{G}$, the two claims imply the following result.

Corollary 3.3. *The network lifetime cannot be approximated to within $O(\log n)$.*

3.2 Weakness of Maximum Willingness

Consider the example in Figure 2 of five nodes in a cyclic order, in which the batteries of nodes $1, 2, 4$ and 5 have initial capacity of 100, while the battery capacity of node 3 is only 10. Furthermore, consider the following deterministic transmission order protocol: node 1 is the first to transmit, then node 2 transmits and they continue in ascending order until all the nodes transmit a message. Then node 1 is again the first to transmit and they continue in this way until the first node fails due to battery depletion.

According to the MaxWill algorithm if node 1 is the source then nodes 2 and 5 are the selected MPRs, if node x, $x \in \{2,3,4\}$, is the source then nodes $x-1$

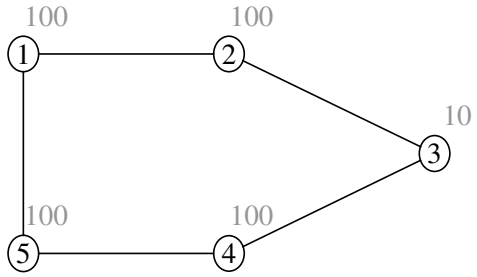


Figure 2: Example of the weakness of the MaxWill algorithm.

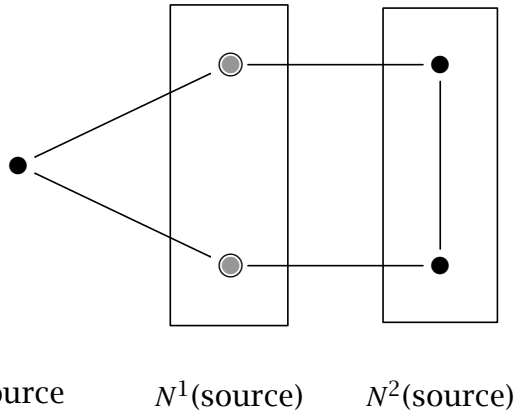


Figure 3: Selected MPRs according to MaxWill are indicated by a circled gray bullet.

and $x + 1$ are the selected MPRs, and finally if node 5 is the source then nodes 1 and 4 are the selected MPRs. Hence there will be in total 17 messages sent before the battery of node 3 is used up. However, one can easily verify that the best strategy to prolong the system's lifetime is to never use node 3, unless this node is the source. In this way there will be a total of 52 messages sent. The weakness of the MaxWill algorithm (and the OLSR framework in general) is that it only focuses on the 2-hop neighborhood, forcing in this example both nodes in the 1-hop neighborhood to be MPRs. A more efficient heuristic will be presented in the next section.

3.3 Path-Based Heuristic

We propose an algorithm that aims to prolong the network lifetime by *not* using the nodes with low battery levels as relays. The relay set determined by the heuristic chooses nodes with battery levels as high as possible.

Path selection. Given a source node $s \in V$, let the node $v' = \text{Argmin}_{v \in V \setminus \{s\}} \{B_v\}$. Our goal is to assign a neighbor node of v' as relay so that v' receives the transmission and hopefully is not assigned as relay. This is achieved by finding a path from s to v' (we know that such a path exists, since G is connected). In graph G we may have a number of such paths, so we choose one that uses relay nodes with the highest battery levels possible. The path selection continues by choosing a node with the second lowest battery level and by finding a path from s to it, and so on.

Relay assignments. Let $P_{s,v'} = \{s, \ldots, v'\}$ be the path found in the way explained above. We assign all nodes in $P_{s,v'} \setminus \{s, v'\}$ as relay nodes. Note that a low-battery level node may be assigned as relay if it is on a path from the source

Table 1: Some notation for the path-based heuristic

s	Source node, $s \in V$
B_v	Battery level of node v , $v \in V, B_v \in \mathbb{N}_0$
R_s	The set of relay nodes for source s, $R_s \subset V$
$G[V']$	Subgraph of G that is induced by $V', V' \subseteq V$
$N(R_s)$	Adjacent nodes to the set R_s of relay nodes
$SP(G, v_1, v_2)$	Shortest path from v_1 to v_2 in graph G

Note: We find all paths by assuming that all edges have same lengths (weights).

Path-Based Heuristic.

Input: See Table 1

$R_s \leftarrow \{s\}$;
while $R_s \cup N(R_s) \neq V$ **do**
  $v' \leftarrow \text{Argmin}_{v \in V \setminus (R_s \cup N(R_s))} \{B_v\}$;
  $V' \leftarrow \{s, v'\}$;
  $P_{s,v'} \leftarrow SP(G[V'], s, v')$;
  while $P_{s,v'} := null$ **do**
    $V' \leftarrow V' \cup \text{Argmax}_{v \in V \setminus V'} \{B_v\}$;
    $P_{s,v'} \leftarrow SP(G[V'], s, v')$;
  end
  $R_s \leftarrow R_s \cup P_{s,v'} \setminus \{s, v'\}$;
end

Output: The set R_s of relay nodes.

to another node. For example, cut-vertices must be used as relays no matter what their battery levels are. By a cut-vertex we mean one whose deletion makes the graph disconnected.

Path selection and relay assignment continues until each node in $V \setminus \{s\}$ is either a relay or is adjacent to a relay node. Table 1 introduces the necessary notation to describe our heuristic and the path-based heuristic is given afterwards.

In the path-based heuristic, in Step 1, we initialize the relay-set with the source s. Then we check, in Step 2, if the current relay set transmits to all nodes in G. If this is not achieved, we select, in Step 3, the lowest-battery level node, say v', among the ones not receiving any transmission. Next we try to find a (shortest) path from s to v' in the subgraph $G[V']$, where V' is defined in Step 4. If such a path does not exist, in Step 7 we update V' by adding the highest-battery level node in $V \setminus V'$ until the desired path is found. In Step 10 we update the set R_s of relay nodes by adding the intermediate nodes in the path $P_{s,v'}$. Finally, the algorithm terminates when all nodes receive transmissions from relay nodes.

Running time. Note that the while loop runs over all the nodes in the graph. So the path-based algorithm runs in $O(n^2 SP)$ time where SP denotes the time to find the shortest path in a graph with unit length (weight) edges. If the nodes in the inner while loop (Steps 6-9) are determined with a binary search, the running time of the algorithm can be improved to $O(n \log n SP)$.

3.4 Numerical Results: Comparing MaxWill with the Path-Based Heuristic

We denote by Algorithm 1 the MaxWill algorithm and by Algorithm 2 the path-based heuristic introduced above.

Let t_i be the number of messages sent using Algorithm i, for $i = 1, 2$, and let R denote the ratio t_2/t_1. The battery capacity B_v of each node v is uniformly distributed over $[\mu_B - \sigma_B, \mu_B + \sigma_B] \cap \mathbb{Z}$. For our simulation purposes we generated several graphs based on the binomial model of the Erdös-Rényi random graph. This model generates an edge between a pair of nodes with equal probability, say p, independently of the other edges. Therefore the lower p is, the sparser G is. If G turns out to be not connected after the construction, another graph is generated. Each time a message is sent, the source will be uniformly selected from the set of nodes V. For parameter values $n = 30, \mu_B = 15, \sigma_B = 10$ and $p = 0.1$, the results of 10.000 simulations in MATLAB are depicted in Figure 4.

Note that in all the simulations we obtained $R \geq 1$. This strongly suggests that, for the type of graph topologies that we have generated, $t_2 \geq t_1$ for all possible battery capacities $(B_v)_{v \in V}$ assigned to the nodes. Furthermore, we have

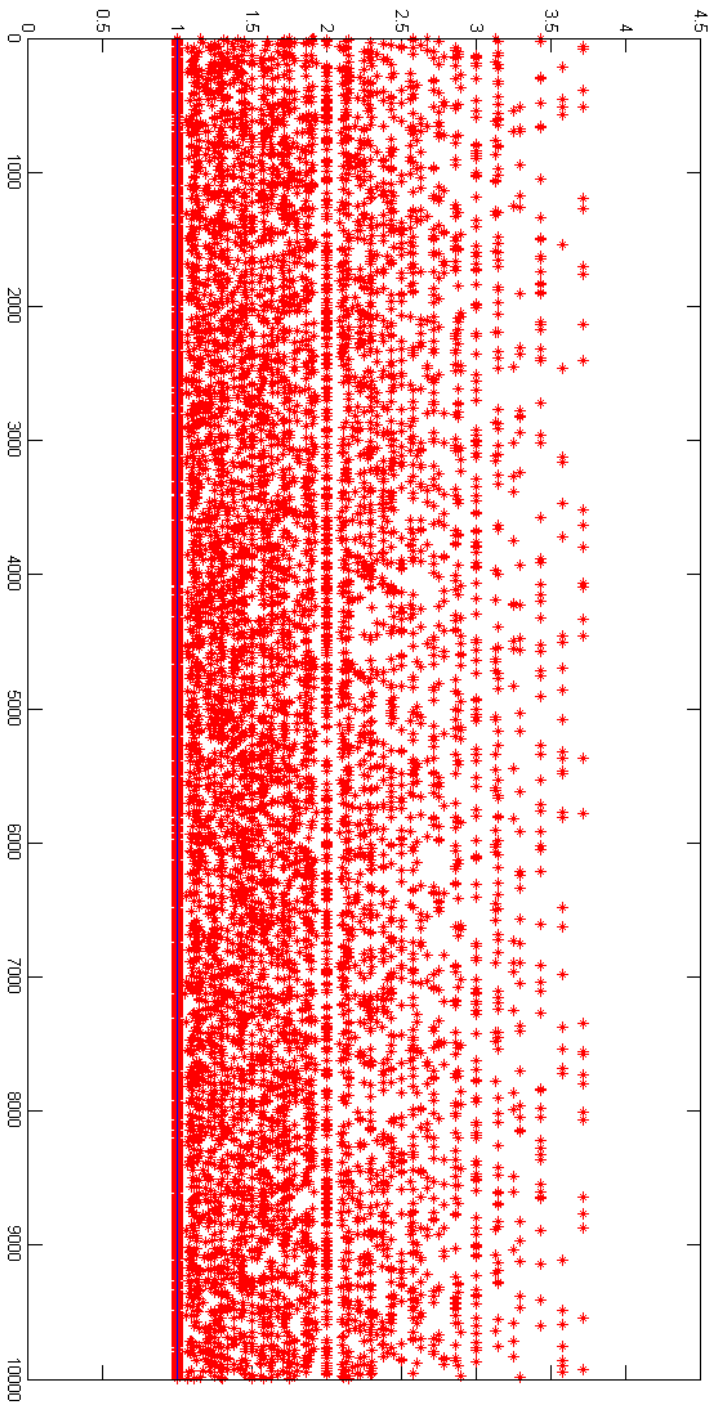


Figure 4: Comparison of the MaxWill and Path-Based algorithms for $n = 30, B_v \sim U[5, 25]$ and $p = 0.1$.

calculated the mean and standard deviation of R to be 1.642 and 0.639, respectively. So all together this suggests that Algorithm 2 performs overall better than Algorithm 1 and does this on average with ratio 1.642. Since at Thales they are dealing with various types of graphs we are interested in the effect of each of the variables n, μ_B, σ_B and p on the ratio R. The results are shown in Figures 5–8.

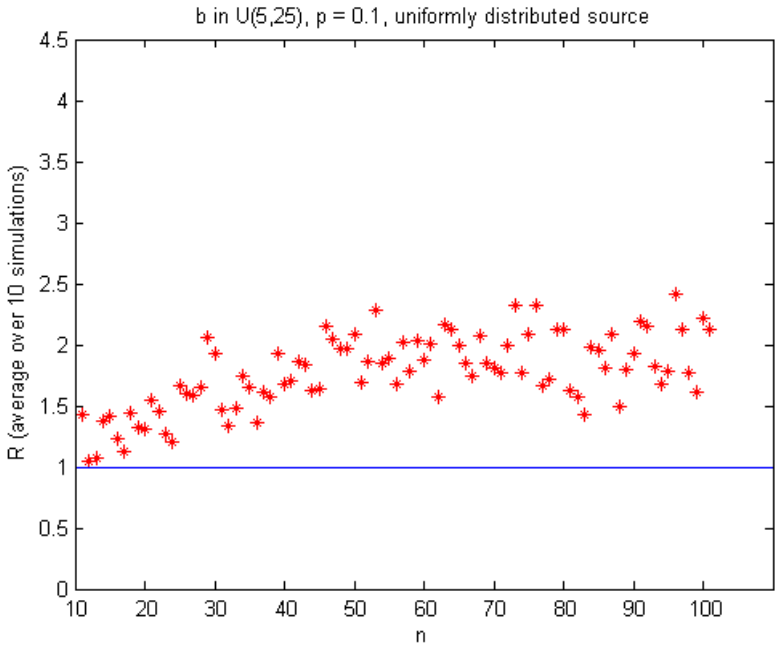


Figure 5: Effect of changing the number of nodes n on the ratio R.

In Figure 5 we see that if n is very small the value of R tends to be closer to one. This is due to the fact that for a randomly generated small graph the local approach of Algorithm 1 (looking at the 1-hop neighborhood) is in fact almost global.

In Figures 6 and 7 we see that R is particularly high if the initial battery values of the nodes differ a lot. Indeed, Figure 6 shows that when μ_B increases the ratio R decreases and Figure 7 shows that when σ_B increases the ratio R increases. So it is likely that Algorithm 2 performs significantly better if the variance of the initial battery values of the nodes is high. This might be due to the fact that Algorithm 2 tends to bring the battery values closer together with time, whereas, in an unlucky case, Algorithm 1 might use nodes with low battery capacities over and over again due to the local nature of the algorithm.

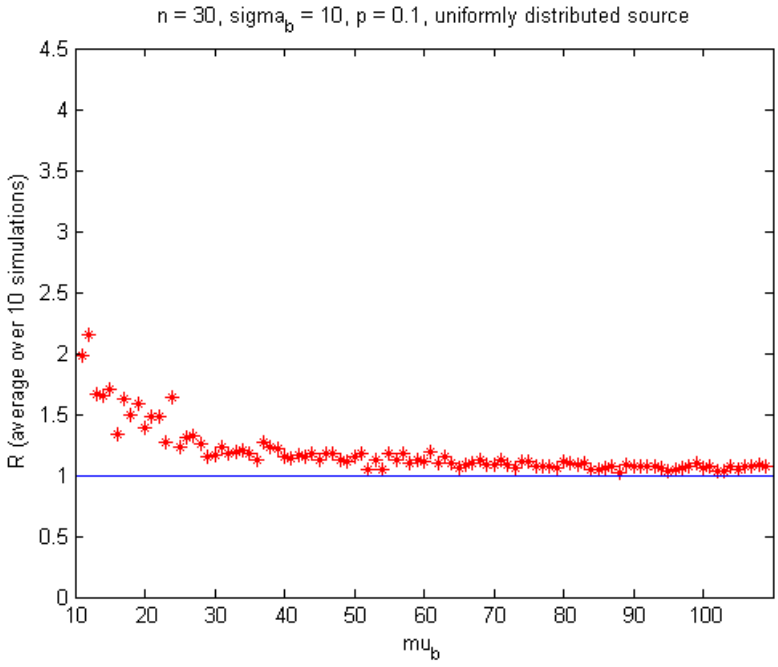


Figure 6: Effect of changing μ_B on the ratio R.

Finally, in Figure 8, we see that R is almost equal to 1 for both highly connected graphs and highly sparse graphs. This is because Algorithm 1 and Algorithm 2 perform optimally for both fully connected graphs and tree structured graphs (the most sparse graphs). Algorithm 2 does perform especially well if the graph is 'intermediately' sparse. This is because Algorithm 1 performs suboptimally when there are larger cycles in a non-fully connected graph (see Section 3.2).

In summary, it is likely that Algorithm 2 performs as well or better than Algorithm 1 for all graph topologies and all possible battery capacities assigned to the nodes. Furthermore, Algorithm 2 performs especially well in the case when the variance of the initial battery capacities is high and the graph is 'intermediately' sparse.

3.5 LP Formulation and Solution Approach

We can also design a linear programming formulation of the problem. We simply sketch the idea here since it is very similar to the one considered later in Sec-

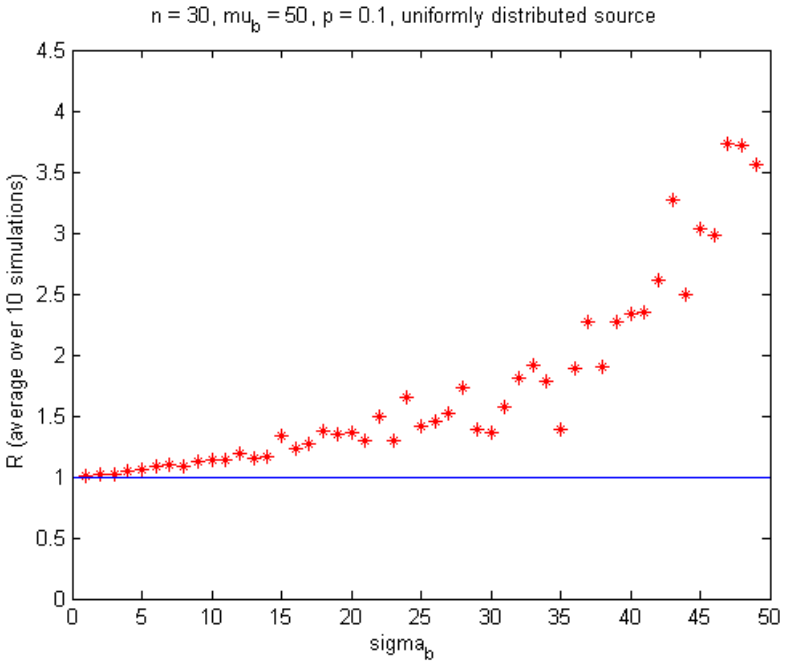


Figure 7: Effect of changing σ_B on the ratio R.

tion 4.2. Given a source s, we define variables $x_{s,S}$ for every subset S of nodes corresponding to a valid set of relays when s is chosen as the source (we always assume that S contains the source s). Given the battery levels B_v, one can thus write the following integer programming formulation:

$$\max r$$

$$\text{s.t.} \sum_s \sum_{S: v \in S} x_{s,S} \leq B_v \qquad \forall v$$

$$\sum_S x_{s,S} \geq r \qquad \forall s$$

$$x_{s,S} \in \{0, 1, 2, \ldots\} \qquad \forall s, S \text{ such that } S \text{ is a valid set of relays for source } s.$$

Let us see why this is a valid formulation. Here r denotes the number of rounds (i.e. where each node takes a turn being a source) that we wish to maximize. The first constraint says that the number of times vertex v is used as a relay is at most the initial battery level B_v. The second constraint says that for each source s we must determine at least r sets S that are valid sets of relays. Clearly, given any valid integer solution to this program, we can perform r rounds by choosing the set S as relays for $x_{s,S}$ rounds when s is the source.

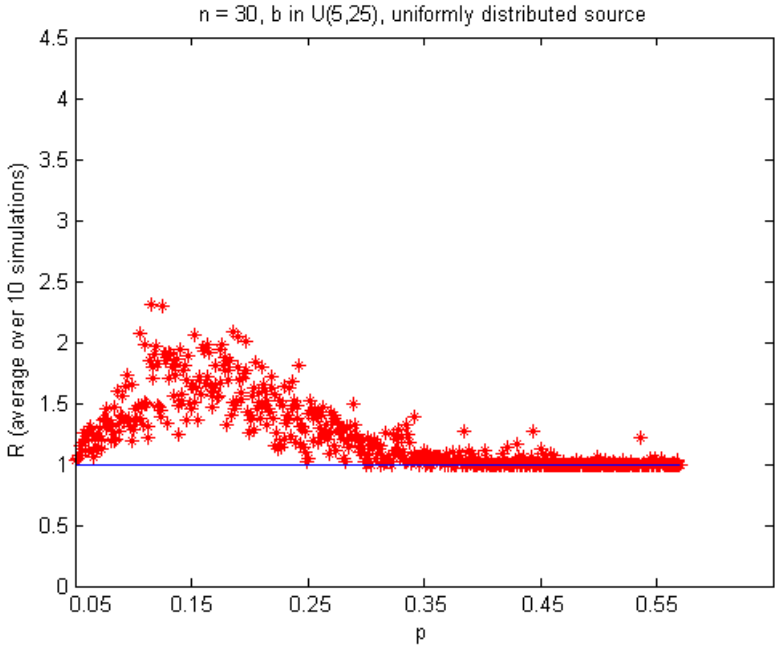


Figure 8: Effect of changing p on the ratio R.

As is standard in approximation algorithms (see [8, 11]), we relax the integrality constraints on the variables $x_{s,S}$ above, and consider the linear programming relaxation (LP) where we only require that $x_{s,S} \geq 0$. This is done because linear programs can be solved to optimality in polynomial time in the number of variables and constraints (albeit at the expense of allowing the variables $x_{s,S}$ to take non-integral values).

However it is not immediately clear how to use this in our setting since the LP above has exponentially many variables (corresponding to the exponentially many possible sets S). To get around this, we note that while this LP has exponentially many variables, it has only polynomially many constraints. So we can consider the dual of this LP (which has polynomially many variables, and exponentially many constraints). Even though there are exponentially many constraints, this is not an major problem, as we can solve it using the Ellipsoid method (by adding constraints on the fly as needed), provided there is an approximate separation oracle for the dual separation problem. Recall that a separation oracle is a subroutine that, given a candidate solution x as input, either outputs a violating constraint, or certifies that x satisfies all the constraints and is a valid

solution. By the standard equivalence between optimization and separation (see for example [6] for more details) an α approximation algorithm to solve the dual separation problem implies an α approximate solution for the primal LP. It can be easily checked that in our setting, the dual separation problem is a weighted set cover problem and hence has an $O(\log n)$ approximation. Given such a solution, one can now apply randomized rounding as explained in Section 4.2. Again, similar ideas are discussed in section 4 and hence we omit the details here.

4 Our Approach: Within the OLSR Framework

As we have already mentioned in the introduction we are interested in selecting the MPRs in order to maximize the network lifetime while satisfying the OLSR constraint. The OLSR protocol relies on the selection of MPRs and calculates its routes to all known destinations through these nodes, i.e. MPR nodes are selected as intermediate nodes in a path. To implement this scheme each node in the network periodically broadcasts the information about its 1-hop neighbors. Upon receipt of this message each node calculates and updates its routes to each known destination. For more details on the OLSR protocol the interested reader is referred to the paper of [3].

Within the OLSR framework each node of the network selects its own MPR-set. This set is a subset of its 1-hop neighbors that covers all its 2-hop neighbors, i.e. the union of the neighbor sets of all MPR nodes contains the entire 2-hop neighborhood. For a given source s this results in a layered network: The first layer is the set of nodes that can be directly reached from the source s, i.e. there exists a directed edge (s, j) if s can directly transmit to node j. We denote the first layer by $L_s(1)$ and we set $L_s(1) = N^1(s)$, where $N^1(s)$ is the set of 1-hop neighbors of s. We denote $L_s(0) = s$. Then the k-th layer is denoted by $L_s(k)$ and is constructed recursively as follows

$$L_s(k) = \bigcup_{v \in L_s(k-1)} \left\{ N^1(v) \cap \left\{ V \setminus \cup_{i=0}^{k-1} L_s(i) \right\} \right\},$$

until all nodes of the graph are classified into layers. Among the nodes in layer $L_s(k)$, MPRs are selected to forward the message to the next layer, $L_s(k+1)$, so that all the nodes in the $(k+1)$-th layer receive the message.

Keeping in mind that each broadcast depletes the battery level of the nodes that are transmitting the message, we are interested in selecting the MPRs in the layered network so that the network lifetime is maximized. In Viennot [10] it was shown that the problem of finding an optimal set of MPRs is NP-complete. The authors showed that the Dominating Set Problem, which is known to be NP-complete, can be reduced to the Multi-point Relay Problem. In Coenen et al. [4], the authors consider the selection of master nodes that relay to their neighbors.

In Verbree et al. [9] the authors analyze the impact of the network topology on the selection of MPRs.

The rest of the paper is organized as follows: In Section 4.1 we give a network instance for which the MaxWill heuristic performs at least $\Omega(\log n)$ times worse than the optimal solution. In Section 4.2 we formulate the relay problem as a linear programming (LP) problem and we use randomized rounding to convert the LP solution into an integral solution. In Section 4.3 we present a second LP formulation and we implement it in Section 4.4 to compare the Maximum Willingness heuristic with the optimal solution.

4.1 Example of the Non-Optimality of Maximum Willingness

We consider here the simpler case of a single source. We construct an example of a graph G for which the MaxWill heuristic is at least $\Omega(\log n)$ times worse than the optimal solution.

Consider the following layer-structured graph of $n = 2k + 2^k - 1$ nodes, with $2k$ nodes in the first layer, and $2^k - 2$ nodes in the second layer. For the rest of the analysis k will be treated as a parameter. We view each node in layer 1 as the pair (i, b) where $i \in \{1, \ldots, k\}$ and $b \in \{0, 1\}$. The nodes in layer 2 are labeled $1, \ldots, 2^k - 2$. It is convenient to consider the binary representation of these labels. The edges between the layers are defined as follows. A node (i, b) is adjacent to all nodes in layer 2 whose i-th bit is b.

Suppose the initial battery of each node is $B_v = B$, for all $v \in V$. We claim that

Lemma 4.1. *The optimum solution can transmit kB messages.*

Proof. Note that for any i, if we pick $(i, 0)$ and $(i, 1)$ to be MPRs, then all the nodes in layer 2 can be covered. This is because each node has either a 0 or 1 in the i-th position. Suppose we pick the above solution with $i = 1, \ldots, k$. If we call this one round, then we can repeat this for B rounds until the batteries are exhausted. □

We now show that the MaxWill heuristic can perform badly. Since the battery levels are the same, it is possible that the nodes $(1, 0), \ldots, (k, 0)$ are chosen. Note that this collection covers all the nodes in layer 2 (since each label has a 0 in at least one position). Moreover, none of these sets are redundant. Indeed if some $(i, 0)$ were dropped, then the point which has 0 in the i-th position and 1 everywhere else would not be covered. After picking these nodes as MPRs, the battery level of these nodes is down to $B - 1$, so in the next step the algorithm will pick the nodes $(1, 1), \ldots, (k, 1)$ as MPRs. At this point, all the battery levels are $B - 1$, and the process repeats again. In this way the batteries are exhausted in $2B$ steps. This gives a gap of $k/2$. Hence in terms of the number of nodes in

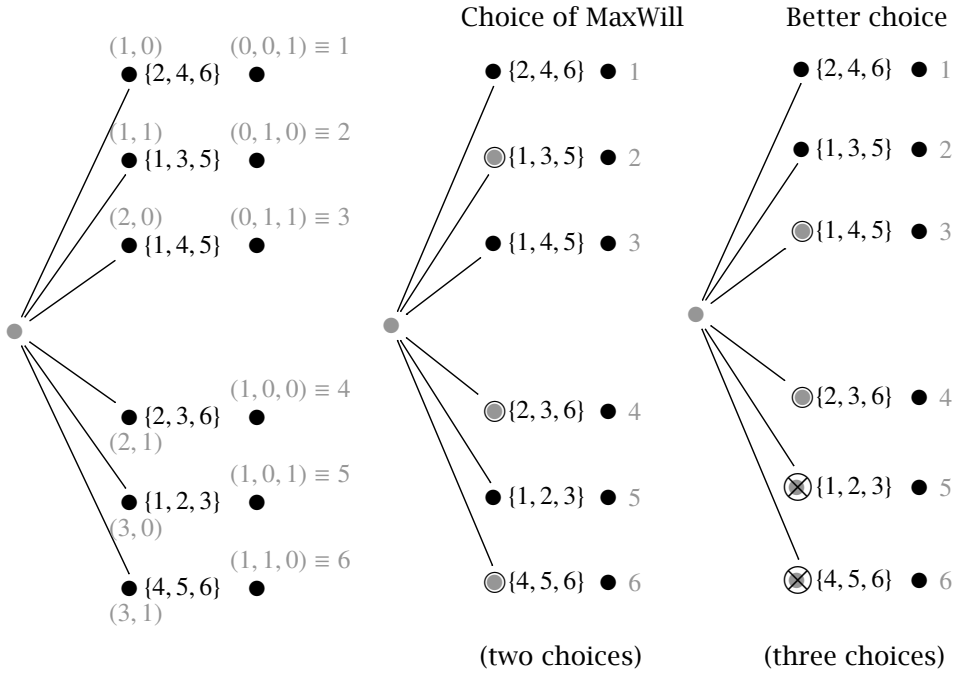


Figure 9: MaxWill heuristic can be at least $\Omega(\log n)$ times worse than the optimal solution.

the network, this gives a gap of $\Omega(\log n)$. See Figure 9.

Remark: It is possible that there are instances where the performance of the MaxWill algorithm is even worse.

4.2 First LP Formulation

In this section we present a linear program that might be useful when the number of nodes in each layer is not too large. We first give some notation. Let $L_s(i)$ denote the i-th layer in the graph when the source is s, $s \in V$. Call a subset $S \subset L_s(i)$ of nodes a valid MPR-set if it can cover all elements of $L_s(i+1)$. Let C_s^i denote the set of all valid MPR-sets in layer i when s is the source. Let $\ell_s = \max_{v \in V} d_G(s, v)$, where d_G is the graph distance in G. Furthermore, ℓ_S can be seen as the number of layers in the graph when s is the source.

Consider the following formulation. For each valid MPR-set S in C_s^i, we have

a variable $x_S \in \mathbb{R}^+$ which indicates how many times this set is chosen,

$$\begin{aligned} &\max r \\ \text{subject to } &\sum_{S \in C_s^i} x_S \geq r, \qquad \forall i = 1, \ldots, \ell_s - 1, \\ &\sum_{S : v \in S} x_S \leq B_v, \qquad \forall v \in V. \end{aligned}$$

Note that in an exact integer programming formulation of the problem we require that $x_S \in \{0, 1, \ldots\}$. Here, however, we have relaxed these integrality constraints to obtain a linear program which is tractable. The first constraint above says that for each source and each layer, we must choose at least one set per round (and hence the total number of sets chosen from a layer should be at least r). The second constraint says that the total number of times a node v is ever used as a MPR is at most B_v.

If the size of each layer is at most k, then this LP has a most $2^k \cdot n \cdot n$ variables (as there can be at most 2^k subsets per layer, at most n sources and at most n layers per graph). So, this LP might be solvable if k is not too large, say around 10 (of course, one needs to further explore this experimentally to determine up to what values of k this might scale to).

4.2.1 Rounding

The above LP gives a fractional solution for the number of times that the sets are chosen (because $x_S \in \mathbb{R}^+$). However, we need an integral solution $x_S \in \mathbb{N}_0$. To do this, we can use the standard randomized rounding approach (see for example [8, 11]). Here, given any accuracy parameter $\epsilon > 0$, we can convert the fractional LP solution into an integral one with $(1 - \epsilon)r$ rounds, provided every initial battery capacity B_v and the number of rounds r are $\Omega(\log n / \epsilon^2)$, where n denotes the number of nodes in the graph. That the battery and the number of rounds are modestly large, in the sense above, is perhaps a reasonable assumption in practice.

Before we state and prove the result formally, we recall some standard probabilistic tail bounds.

Lemma 4.2 (Chernoff Bounds, Theorems A.1.12 and A.1.13 in [1])**.** *Let X_i, $i = 1, \ldots, n$ be independent* 0-1 *random variables with mean $E[X_i] = \mu_i$. Let $X = \sum_i X_i$ and $\mu = E[X]$. Then for any $\epsilon > 0$, it holds that*

$$Pr[X \leq (1 - \epsilon)\mu] \leq e^{-\epsilon^2 \mu / 2},$$

$$Pr[X \geq (1 + \epsilon)\mu] \leq \left[\frac{e^\epsilon}{(1 + \epsilon)^{1+\epsilon}} \right]^\mu.$$

Theorem 4.3. *Given any $0 < \epsilon \leq 1/2$, the LP solution can be rounded to a feasible integral solution where the number of rounds is at least $(1-2\epsilon)r$, provided that $r \geq 10(\log n)/\epsilon^2$ and $B_v \geq 10(\log n)/\epsilon^2$ for each node v, where n is the number of nodes.*

Proof. For each set S such that $x_S > 0$ in the LP solution, let $y_S = (1-\epsilon)x_S$, and let $f_S = y_S - \lfloor y_S \rfloor$ be the fractional part of y_S (note that $f_S \in [0,1]$). Let z_S denote the 0-1 random variable that is 1 with probability f_S and 0 otherwise. To round the LP solution, for each set S, we independently (of other sets) choose $\lfloor y_S \rfloor + z_S$ copies of S. In other words, we choose $\lceil y_S \rceil$ copies of S with probability z_S and $\lfloor y_S \rfloor$ copies otherwise.

Let us bound the probability that the battery capacity is exceeded for any node v (this is precisely the event $\sum_{S:v\in S}(\lfloor y_S \rfloor + z_S) > B_v$, for some v). To this end, consider a particular node v,

$$\begin{aligned}
\Pr\left[\sum_{S:v\in S}(\lfloor y_S \rfloor + z_S) > B_v\right] &= \Pr\left[\sum_{S:v\in S} z_S > B_v - \left(\sum_{S:v\in S}\lfloor y_S \rfloor\right)\right] \\
&= \Pr\left[\sum_{S:v\in S} z_S > B_v - \sum_{S:v\in S}(y_S - f_S)\right] \\
&\leq \Pr\left[\sum_{S:v\in S} z_S > \epsilon B_v + \sum_{S:v\in S} f_S\right] \\
&= \Pr\left[\sum_{S:v\in S} z_S > (1+\epsilon')\sum_{S:v\in S} f_S\right] \text{ with } \epsilon' = B_v/\left(\sum_{S:v\in S} f_S\right) \\
&\leq \left[\frac{e^{\epsilon'}}{(1+\epsilon')^{1+\epsilon'}}\right]^{\mu} \text{ with } \mu = \sum_{S:v\in S} f_S.
\end{aligned}$$

Above, the first inequality follows since the LP satisfies, $\sum_{S:v\in S} x_S \leq B_v$ and hence $B_v - \sum_{S:v\in S} y_S \geq \epsilon B_v$.

Consider two cases: $\epsilon' < 2e-1$ and $\epsilon' \geq 2e-1$. If $\epsilon' \geq 2e-1$, then we can bound the expression above by

$$(1/2)^{\epsilon'\mu} = (1/2)^{B_v} \leq 1/n^{10}.$$

If $\epsilon' < 2e-1$, then by using Taylor expansions for $e^{1+\epsilon'}$, we can bound this by $e^{-\epsilon'^2/4\mu} \leq 1/n^2$. Thus, we see that the probability of this event is at most $1/n^2$. Taking the union over all the n nodes, the probability of the battery running out for any node is at most $1/n$.

Similarly, since the LP has r rounds, the expected number of rounds in the rounded solution is at least $(1-\epsilon)r$ since the original LP constraint is $\sum_{S\in C_s^i} x_{s,S} \geq$

r, and hence $\sum_{S \in C_s^i} y_{s,S} \geq r$ (and the expected number of copies of S that we choose in our rounded solution is precisely $\lfloor y_S \rfloor + E[z_S] = \lfloor y_S \rfloor + f_S = y_S$). Again, direct application of Lemma 4.2 implies that the probability that there are fewer than $(1-2\epsilon)r$ rounds is at most $1/n^2$. So the overall probability of any of the bad events happening is at most $1/n + 2/n^2 \leq 2/n$. This algorithm can be derandomized using standard techniques, see for example [1]. □

4.3 Second LP Formulation

Here we consider an alternative formulation of the linear program. The binary variable $x_{r,s,v} \in \{0,1\}$ is defined for every round $r \in \{1,\ldots,\mathcal{R}\}$, for every source $s \in V$ and for every node $v \in V$. We take $V = \{1,2,\ldots,n\}$. We set

$$x_{r,s,v} = \begin{cases} 1 & \text{when node } v \text{ is broadcasting in round } r \text{ and node } s \text{ is the source,} \\ 0 & \text{otherwise.} \end{cases}$$

For a node v we denote by B_v its battery level and p_v the battery depletion after the broadcasting of a message. This means that each time it transmits a message (as a MPR or source) its battery level reduces by p_v. So if a node originally has battery level B_v it can broadcast at most $\lfloor B_v/p_v \rfloor$ messages. Note that up until now we have always assumed that $p_v = 1$.

For a source s and for two nodes u and v we define $P_{uv}^s \in \{0,1\}$ as follows: We set $P_{uv}^s = 1$ if the node u is a *predecessor* of the node v when s is used as source, namely if the node u is in the layer before that of v, and if the nodes u and v are connected. Clearly $P_{us}^s = 0$, since the source has no predecessors. The matrix $\boldsymbol{P}^{(s)} = (P_{uv}^s)_{u,v}$, for a fixed source s, will be referred to as the *s-predecessor matrix*.

To better understand the meaning of the predecessor matrix, we compute it for a simple, concrete example, namely for the graph in Figure 10.

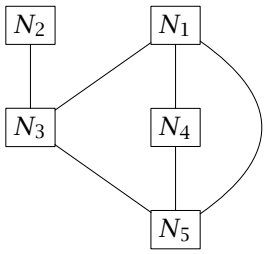


Figure 10: The Network.

Before we can compute P^s_{uv} we need to sort the graph into layers according to which node is the source. See Figures 11-15. Then we see, e.g., that matrix $\boldsymbol{P}^{(1)}$ (which corresponds to $s = 1$, Figure 11) is

$$\boldsymbol{P}^{(1)} = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}; \tag{1}$$

indeed since N_1 is the predecessor of N_3, N_4 and N_5 the corresponding entries P^1_{13}, P^1_{14} and P^1_{15} are equal to one; also $P^1_{32} = 1$ since N_3 is a predecessor of N_2.

We formulate the linear program as a feasibility problem with linear constraints. More precisely, we first fix a number of rounds $\mathcal{R}$ and we ask whether our network can transmit messages for $\mathcal{R}$ *whole* rounds. Then we optimize over $\mathcal{R}$ to find the maximum number of feasible rounds. When we implement the algorithm in the following section we take as a first guess of the number of feasible rounds the result given by the MaxWill algorithm, which is clearly a lower bound. Then we keep incrementing $\mathcal{R}$ by 1 until the problem becomes infeasible.

More precisely, for a fixed $\mathcal{R}$, we consider the following feasibility problem with linear constraints:

(IP) Find $x_{r,s,v} \in \{0,1\}$, where $r \in \{1,\dots,\mathcal{R}\}$, $s, v \in \{1,\dots,n\}$, such that

(1) $x_{r,s,s} = 1 \quad \forall\, s \in \{1,\dots,n\},\ \forall\, r \in \{1,\dots,\mathcal{R}\};$

(2) $\sum_{u=1}^{n} x_{r,s,u} P^s_{uv} \geq 1 \quad \forall\, s \in \{1,\dots,n\},\ \forall\, v \in \{1,\dots,n\},\ \forall\, r \in \{1,\dots,\mathcal{R}\};$

(3) $\sum_{r=1}^{\mathcal{R}} \sum_{s=1}^{n} x_{r,s,v} \leq \frac{B_v}{p_v} \quad \forall\, v \in \{1,\dots,n\}.$

Constraint (1) is the trivial constraint that node s transmits a message when it is the source; constraint (2) ensures that every message is received by every node; and constraint (3) ensures that the battery level of each node remains nonnegative. Note that since the inequality in constraint (3) is not strict, it is possible for the battery level of one or more nodes to be zero or to go down to zero (but not below zero) in a feasible round, provided that every node receives the message.

Problem (IP) is a binary integer program and so it is NP-complete. Note, however, that the number of variables equals $\mathcal{R}n^2$. In particular it grows polynomially in n, in contrast with the linear program presented in the Section 4.2, where the number of variables grows exponentially in n. Note that the number of constraints also grows polynomially in n and $\mathcal{R}$.

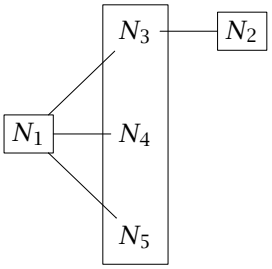


Figure 11: Source N_1.

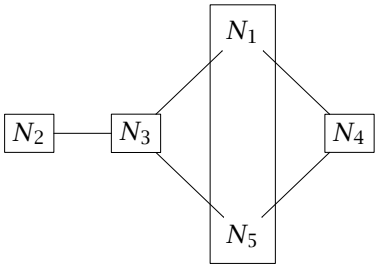


Figure 12: Source N_2.

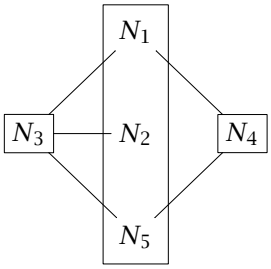


Figure 13: Source N_3.

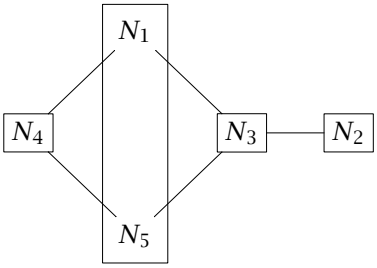


Figure 14: Source N_4.

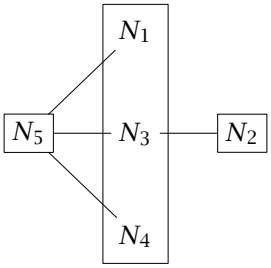


Figure 15: Source N_5.

4.4 Numerical Results: Comparing MaxWill with the Optimal Solution

In this section we implement the binary integer program (IP) introduced in the previous section and compare it with the MaxWill algorithm. The implementation was done in MATLAB using the function *bintprog* (with the cost vector taken to be zero).

Performance ratios. For a node $v \in V$ with battery level B_v and transmission power p_v, let R_{IP} denote the optimal number of rounds, i.e., let R_{IP} be the maximum value of $\mathcal{R}$ such that the binary integer program (IP) is feasible. Let R_{MW} denote the number of rounds that is possible if the MaxWill algorithm is used to select the relay nodes. In Figure 16 we plot the ratio $R_{\text{IP}}/R_{\text{MW}}$ for 100 different simulations, for $n = 5, 10, 15$. For each simulation we generated a binomial model of the Erdös-Rényi random graph, with probability $p = 0.5$ of two nodes

being connected. The battery levels B_v were also random, selected uniformly from the integers in the interval $[20, 30]$. We took the transmission battery depletion to be equal to one, $p_v = 1$, for all nodes in every simulation.

From Figure 16 we see that for $n = 5$ the MaxWill algorithm gives the optimum number of rounds in almost every simulation, 98 out of 100. As the number of nodes is increased the performance of the MaxWill algorithm decreases: For $n = 10$ the MaxWill algorithm is optimal 58% of the time, and for $n = 15$ only 45% of the time. Moreover, the ratios $R_{\text{IP}}/R_{\text{MW}}$ increase as n increases: For $n = 5$ the maximum ratio is 1.29, while for $n = 15$ it is 1.67.

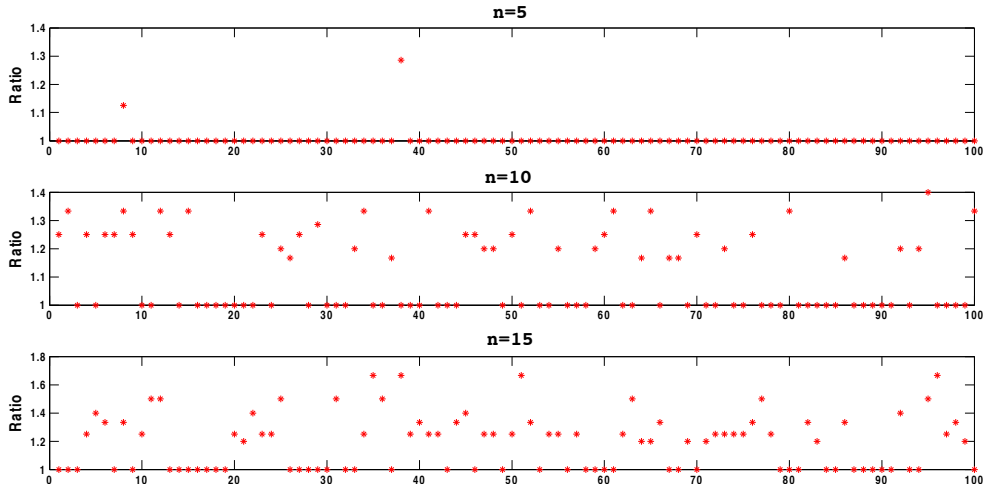


Figure 16: Performance ratios: The optimality of the MaxWill algorithm was tested for 100 different random graphs, with $n = 5, 10, 15$ nodes. The x-axes correspond to the simulation number. The y-axes correspond to the performance ratios $R_{\text{IP}}/R_{\text{MW}}$ of the optimal number of rounds divided by the number of rounds that are achieved using the MaxWill algorithm to select the MPRs.

In Figure 17 we assess the performance of the MaxWill algorithm when the nodes are assigned different power levels. We consider three cases: Power level 1 where $p_v = 1$ for all nodes; Power level 2 where, for each v, p_v is randomly selected from the set $\{1, 2\}$; Power level 3 where, for each v, p_v is randomly selected from the set $\{1, 2, 3, 4\}$. Once chosen, the power levels are fixed for all rounds. For each simulation, we randomly generate a graph with 10 nodes and with probability $p = 0.5$ of having an edge between two nodes. We randomly assign battery levels and power levels as described above.

We see that for the Power level 3, the performance ratio is statistically close to 1. For Power level 1 and Power level 2 MaxWill performs worse. In the case

of Power level 1 MaxWill performs slightly worse than in the case of Power level 2.

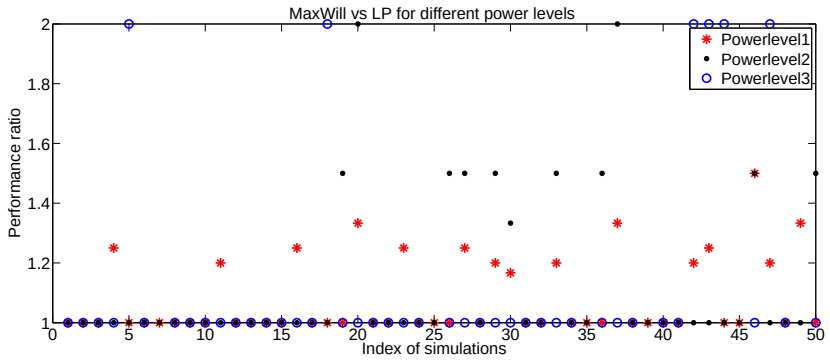


Figure 17: Performance ratios: The optimality of the MaxWill algorithm was tested for 50 different random graphs, with $n = 10$ nodes for different power levels. The x-axes correspond to the simulation number. The y-axes correspond to the performance ratios $R_{\text{IP}}/R_{\text{MW}}$ of the optimal number of rounds divided by the number of rounds that are achieved using the MaxWill algorithm to select the MPRs.

In Figure 18 we study the performance of MaxWill for graphs that have been generated with different probabilities. The different probabilities measure the sparseness of the graph. We took $p = 0.1, 0.3, 0.5, 0.7$ and performed 50 simulations for each case. The plot suggests that the performance indeed depends on the sparseness of the graph. The sparser the graph, the better the performance of MaxWill.

Runtimes. We compare the runtime of the MaxWill algorithm with the runtime needed to find the optimal number of rounds R_{IP} (which was found by solving the binary integer program (IP) for $\mathcal{R} = R_{\text{MW}}$ to $R_{\text{IP}} + 1$). See Table 2, which shows the average runtimes in seconds for the simulations given in Figure 16. Note that the runtimes depend partly on our implementation of the algorithms and no attempt was made to optimize the code. For interest we also present the runtimes of solving the linear programming problem (LP) that is obtained by relaxing the condition $x_{r,s,v} \in \{0, 1\}$ of (IP) to allow $x_{r,s,v}$ to take any value in the interval $[0, 1]$.

As expected, since (IP) is NP-complete, for small n the runtimes of the three algorithms are similar but for large n the binary integer program (IP) can take far longer. Note the big difference between the mean and the median runtime for (IP) when $n = 15$.

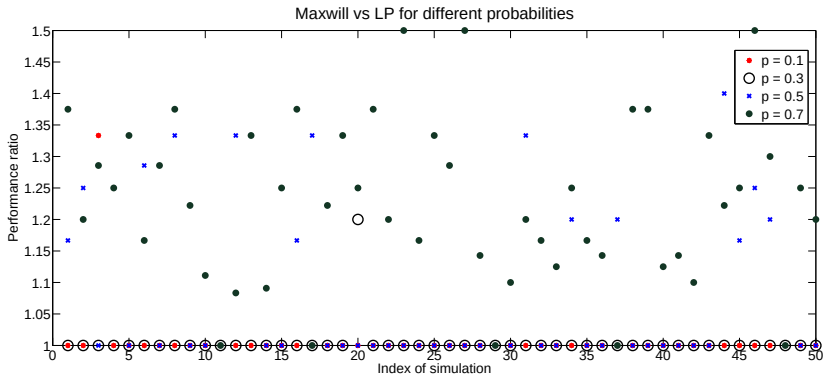


Figure 18: Performance ratios: The optimality of the MaxWill algorithm was tested for 50 different random graphs with $n = 10$ nodes, for different probabilities used for generating the graphs. The x-axes correspond to the simulation number. The y-axes correspond to the performance ratios $R_{\text{IP}}/R_{\text{MW}}$ of the optimal number of rounds divided by the number of rounds that are achieved using the MaxWill algorithm to select the MPRs.

n	MaxWill		IP		LP	
	mean	median	mean	median	mean	median
5	0.06	0.05	0.14	0.12	0.09	0.08
10	0.13	0.13	7.02	0.54	0.27	0.25
15	0.21	0.21	651.66	7.81	0.84	0.71

Table 2: Runtimes: Average runtimes in seconds of the simulations given in Figure 16, plus also the runtimes of the (LP) relaxation of the binary integer program (IP).

This shows that in practice, except for networks with few nodes (around 10 or less), the main role of the binary integer programming formulation (IP) is to assess the performance of the MaxWill algorithm and other heuristics. If the network is small and if the number of desired rounds to be broadcast is known in advance, then (IP) could be solved to choose the MPRs optimally. (Note that (IP) can only be used if you know a priori that the nodes will broadcast in rounds for a given, fixed number of rounds.)

It is interesting to note that, if the number of nodes is small, then the (LP) relaxation of (IP) tends to give the same number of feasible rounds, i.e., $R_{\text{IP}} = R_{\text{LP}}$ when n is small: For $n = 5$ we found that $R_{\text{IP}} = R_{\text{LP}}$ for every simulation, for $n = 10$ we found that $R_{\text{IP}} = R_{\text{LP}}$ for 98 % of the simulations, and for $n = 15$ we found that $R_{\text{IP}} = R_{\text{LP}}$ for 75 % of the simulations.

5 Extensions and Concluding Remarks

In this section we recall the questions posed by Thales Nederland and based on these questions we make some conclusions and recommendations for Thales and discuss possible extensions.

Direction 1: What would be the optimal MPR selection algorithm? With a linear battery decrease model it should be possible to formulate this as a linear program. How much does the optimal solution differ from the known heuristics? Can we define easily a better heuristic than the Maximum Willingness heuristic algorithm used by Thales?

Outside the OLSR framework. In the first part of this paper, Section 3, we relaxed the constraint of Thales that messages have to be sent in layers. We showed in Section 3.1 that the problem of selecting the relay nodes optimally to maximize the lifetime of the network is NP-complete (this is also the case with the layer constraint). In Section 3.3 we introduced a polynomial-time algorithm for choosing the relay nodes. We demonstrated numerically (Section 3.4) that this algorithm out-performs the MaxWill algorithm, in some cases significantly. Therefore if it were possible for Thales to work outside the OLSR framework, Thales could benefit greatly from the use of this new algorithm instead of the MaxWill algorithm.

Inside the OLSR framework. In Section 4 we introduced the constraint of Thales that messages should be broadcast in layers (the OLSR framework). In Section 4.1 we gave a (non-generic) example to show that the MaxWill algorithm can perform arbitrarily badly. To see how it performs in general, two binary linear programs were derived to choose the MPRs optimally. See Sections 4.2 and 4.3. In Section 4.4 the second linear program was implemented to show quantitatively how far the MaxWill algorithm is from being optimal. Thales could use this to decide whether to continue using the MaxWill algorithm or whether to search for a better heuristic. Our linear programming formulations are mainly intended for comparison purposes (to test the optimality of heuristics) but could also be used as algorithms for choosing the relay nodes when the network is small.

Finally we tie together Sections 3 and 4 by commenting on how the performance of MaxWill depends on whether or not the layer constraint is included, for the case of sparse and dense graphs.

Figure 18 shows that, within the OLSR framework, the MaxWill algorithm performs nearly optimally for sparse graphs. On the other hand, from Figure 8 we see that the Path-Based algorithm, which lies outside the OLSR framework, far out-performs the MaxWill algorithm for sparse (but not too sparse) graphs. This

suggests that for sparse graphs a significant improvement in the lifetime of the network could be achieved by working outside the OLSR framework.

Conversely, in the case of dense graphs, Figure 8 shows that the Path-Based algorithm does not perform better than MaxWill (even though the layer constraint is relaxed). From Figure 18 we see that also within the OLSR framework the MaxWill algorithm is far from being optimal. This suggests that for dense graphs it is worth looking for a better polynomial-time algorithm even within the OLSR framework.

Direction 2: Assume additionally that a node can choose between different power levels. For a higher power level a node will have larger set of neighbors to choose its MPR-set from. Can we formulate the optimization problem and find some good heuristic to solve it (a solution being an assignment of transmit powers and MPR-sets)? What would be the impact on the network lifetime?

A further direction of investigation proposed by Thales Nederland consists in allowing the nodes the freedom of choosing the power levels at which they transmit a message (the transmission, however, will still proceed layer by layer). This additional degree of freedom means that the topology of the network changes with every transmission according to the power level each node selects. Hence the set of edges connecting the nodes is essentially an unknown of the problem as well.

We notice that, although the transmission has still to respect the layered structure of the graph, since we can change the topology of the graph before sending the message, examples of the type in Figure 2 showing the weakness of the MaxWill algorithm can be ruled out by properly tuning the power levels. In fact, in the graph in Figure 2, it is possible to turn node 3 into a leaf by suitably choosing its power level. Hence a possible strategy for optimizing the lifetime of the network is to tune the power levels of the nodes so that, in the corresponding graph, the transmission path generated by the Path-Based algorithm becomes admissible within the OLSR framework.

For example, in the case of only two power levels, we could choose the power level of each node using a threshold argument: we could define a threshold battery level such that all nodes with battery level higher than the threshold will broadcast with the highest power level, while nodes with battery levels lower than the threshold will select the lowest power level. This heuristic prevents all “low battery" nodes from transmitting with the high power level, and hence might prolong the lifetime of the network.

Here is another strategy. Assume again that all nodes can use two trans-

mission powers ($p_{\min}$ and $p_{\max}$). First assume that the source uses power $p_{\min}$, provided that it can transmit to at least one node. All 1-hop neighbors of the source will be ordered according to their battery levels in a descending order. Furthermore, consider the two extreme cases where all the nodes transmit with the minimum power $p_{\min}$ or all the nodes transmit with the maximum power $p_{\max}$. Let $L_{\min}$ and $L_{\max}$ denote the corresponding number of relays used according to the MaxWill algorithm. If $L_{\min} \leq L_{\max}$ then all nodes are assigned $p_{\min}$. Otherwise, define the fraction of nodes transmitting with power $p_{\max}$ as follows:

$$R = \begin{cases} (L_{\max} * p_{\max} - L_{\min} * p_{\min})/(L_{\max} * p_{\max}), & \text{if } L_{\max} * p_{\max} - L_{\min} * p_{\min} \geq 0, \\ 1, & \text{otherwise.} \end{cases}$$

We choose which nodes transmit with power $p_{\max}$ according to their battery level, starting from the highest battery level. Having assigned the transmission powers we can now use the MaxWill algorithm to choose the relays in the 1-hop neighborhood of the source. We continue in this way until we cover the entire graph. Finally, we repeat this algorithm for the case where the source uses $p_{\max}$ and decide between the two options.

Of course, these approaches require further investigation and are not complete, but could be the start of a sequel paper.

Direction 3: What is the effect on the network lifetime problem when using a battery model with a recovery effect?

Although Direction 3 is very interesting, due to time limitations we were not able to extend our analysis in this direction. A very frivolous approach to the recovery effect could be to assume that the battery is depleted by a fixed amount for each message transmission and replenished by a fixed amount for each round that it remains idle. Of course this is simply a first thought on tackling the problem, but we would be very much interested to look into it in the future.

Acknowledgements

We would like to thank Dr. Maurits de Graaf for introducing the sensor lifetime maximization problem to us. Moreover, we would like to express our gratitude to Thales Nederland for sharing their problem with us and to the SWI 2012 committee for the wonderful opportunity to work on such an interesting and exciting problem. We are grateful to Prof. Remco van der Hofstad for many useful suggestions and improvements regarding this manuscript.

References

[1] N. Alon and J. H. Spencer. *The Probabilistic Method*. New York: John Wiley, 2000.

[2] S.-H. Cheong, K.-I. Lee, Y.-W. Si, and L. H. U. Lifeline emergency ad hoc network. *Computational Intelligence and Security (CIS), 2011 Seventh International Conference on*, pages 283 –289, 2011.

[3] T. Clausen and P. Jacquet. RFC 3626: Optimized link state routing protocol (OLSR). Internet draft, `http://www.ietf.org/rfc/rfc3626.txt`, 2003.

[4] T. Coenen, J.-K. van Ommeren, and M. de Graaf. Routing versus energy optimization in a linear network. In *Architecture of Computing Systems (ARCS), 2010 23rd International Conference on*, pages 1–6. VDE, 2010.

[5] U. Feige, M. M. Halldórsson, G. Kortsarz, and A. Srinivasan. Approximating the domatic number. *SIAM J. Comput.*, 32(1):172–195, 2002.

[6] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms And Combinatorial Optimization*. Springer, 1993.

[7] T. Moscibroda and R. Wattenhofer. Maximizing the lifetime of dominating sets. In *Parallel and Distributed Processing Symposium, 2005. Proceedings. 19th IEEE International*, 2005.

[8] V. V. Vazirani. *Approximation Algorithms*. Springer, 2010.

[9] J.-M. Verbree, M. de Graaf, and J. Hurink. An analysis of the lifetime of OLSR networks. *Ad Hoc Networks*, 8(4):391–399, 2010.

[10] L. Viennot. Complexity results on election of multipoint relays in wireless networks. Report RR-3584, INRIA, 1998.

[11] D. P. Williamson and D. B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, 2011.

Non-imaging Optics for LED-Lighting

Jan Bouwe van den Berg (VU University), Rui Castro (Eindhoven University of Technology), Jan Draisma (Eindhoven University of Technology), Joep Evers (Eindhoven University of Technology), Maxim Hendriks (Eindhoven University of Technology), Oleh Krehel (Friedrich-Alexander-Universität Erlangen-Nürnberg), Ivan Kryven (University of Amsterdam), Karin Mora (University of Bath), Botond Szabó (Eindhoven University of Technology), Piotr Zwiernik (Eindhoven University of Technology)

Abstract

In this report, several methods are investigated to rapidly compute the light intensity function, either in the far field or on a finite-distance screen, of light emanating from a light fixture with a given shape. Different shapes are considered, namely polygonal and (piecewise) smooth. In the first case, analytic methods are sought to circumvent the use of Monte Carlo methods and ray-tracing with large sample size. In the second case, refinements of the Monte Carlo method (notably using a bootstrap procedure) are devised to minimize the number of samples needed for a good approximation of the intensity function.

KEYWORDS: optics, light, geometry, reflection, light ray, phase space, far field, polygon, ray-tracing, root approximation, Monte Carlo, bootstrap

1 Introduction

Philips Lighting is interested in assessing and improving the design of optical systems. These systems consist e.g. of lenses and mirrors and transfer light from a source to a target. To obtain characteristics of the light exiting the device (e.g. luminous intensity, color point and brightness), an efficient way of relating the input from the light source to the output at the target is needed. Currently, Monte Carlo methods are used which sample from the space of all possible rays that are emitted from the source. Using geometrical optics, it is possible to trace the path of each ray through the system. If a sufficiently large sample of rays is taken the corresponding distribution of rays over the target is representative for the output created by the device in reality. The words 'sufficiently large' are crucial here since in practice up to a million paths need to be traced to obtain reasonable results. In view of the fact that one wants to improve the design and redo the calculations step-by-step a less-involved procedure would be beneficial.

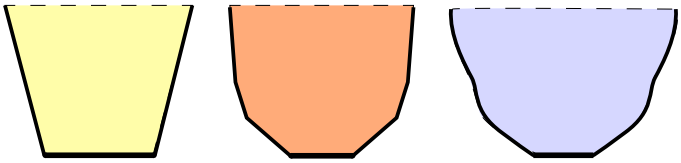


Figure 1: Different cup shapes in order of increasing difficulty: faceted (2 or more) and piecewise smooth.

The request to the Philips SWI 2012 group (here, formulated in its widest form) was to come up with a better (less time-consuming) approach. In view of the limited amount of time available, the problem owners indicated that they would not be disappointed if only the 2D case was considered. However, a short overview of further ideas on the 3D case would be highly appreciated.

During the week the team considered only the following:

- Devices consisting of mirrors. Lenses are not taken into consideration.
- Luminous intensity and no other photometric quantities (like color and brightness).

Moreover, the shapes of the light fixtures, also referred to as cups, can be subdivided in three classes, cf. Figure 1. Firstly, we consider two-faceted cups: devices with one flat wall at either side of the light source (Figure 1, left). Secondly, there are cups with piecewise linear walls, here called multi-faceted (Figure 1, center). Finally, the most general devices we consider are those whose sides are only required to be piecewise smooth (Figure 1, right). In order to assess the quality of the light bundle emitted from a lighting device, a screen is placed at some specific distance from, and parallel to the source. Each ray emitted from the source can be characterized by its position and angle. Once it arrives at the target again, it does so at a certain position and under a certain angle. For both the source and target, the position and angle constitute a two-dimensional phase space. Hence, the effect of the lighting device can be represented by a mapping from the source phase space, $\text{Phase}_{\text{source}}$, to the target phase space, $\text{Phase}_{\text{target}}$. Assume that X and Θ are the spaces of position and angle of a ray, respectively, when emitted from the source. Analogously, the position and angle at the target are in $\tilde{X}$ and $\tilde{\Theta}$, respectively. Under these assumptions, we have

$$\text{Phase}_{\text{source}} = X \times \Theta$$
$$\text{Phase}_{\text{target}} = \tilde{X} \times \tilde{\Theta}.$$

In practice ,i.e. in case of real, manufactured devices, either the target screen is at a relatively large distance from the device or one does not know before-

hand where the target ,e.g. a wall in the customer's home, will be. In such situations it makes sense to consider the so-called *far field* **F**, only. This far field can be considered as a target screen 'at infinity' where one only regards the angle-component of the target phase space and disregards the position-component, then

$$\mathbf{F} = \left(-\frac{\pi}{2}, \frac{\pi}{2}\right).$$

The further away a screen is from the source the better the far field approximation becomes.

The source emits light but it need not necessarily do so equally strong in all directions from all points. We model this by the intensity function $I_{\text{source}} : \text{Phase}_{\text{source}} \to \mathbf{R}_{\geq 0}$. Philips assumed

$$I_{\text{source}}(x, \theta) = \mathcal{I}_0 \cos\theta, \tag{1}$$

where $\mathcal{I}_0$ is a given constant. A source obeying this law is called *Lambertian*. Note that, in general, a suitable composition of I_{source} and the mapping $\text{Phase}_{\text{source}} \to \text{Phase}_{\text{target}}$ yields a corresponding function $I_{\text{target}} = I_{\text{target}}(\tilde{x}, \tilde{\theta})$ on the target phase space which contains the information on the luminous quantities we are interested in. Our goal is to compute this function efficiently. By integrating I_{target} over $\tilde{x}$, a quantity $I_{\mathbf{F}}$ called *intensity* is obtained, whereas the *illumination* follows from integrating over $\tilde{\theta}$.

2 Two-dimensional faceted cups

2.1 The 2-facet cup

One of the simplest shapes of a light fixture from the viewpoint of mathematical analysis is the 2-facet cup comprised only of straight lines. This axis-symmetric fixture consists of the base (light source) and two inclined edges which are 100% reflective and equally long.

Let us fix some notation as illustrated in Figure 2. A Cartesian coordinate system is set up by letting the origin O be the intersection of the lines through the reflective edges. The x-axis is parallel to the fixture's base and the y-axis divides the cup into two equal, symmetrical parts. Further, let a denote the distance of the base to the origin, let γ be the angle between the y-axis and each fixture edge, and let h be the height of the cup.

We assume an emitted light ray to be a straight line with initial position (x_0, a) where $x_0 \in (-a\tan\gamma, a\tan\gamma)$ and initial angle θ which is measured anti-clockwise to the y-axis. A light ray normal to the source, i.e. $\theta = 0$, corresponds to the direction $(0, 1)$.

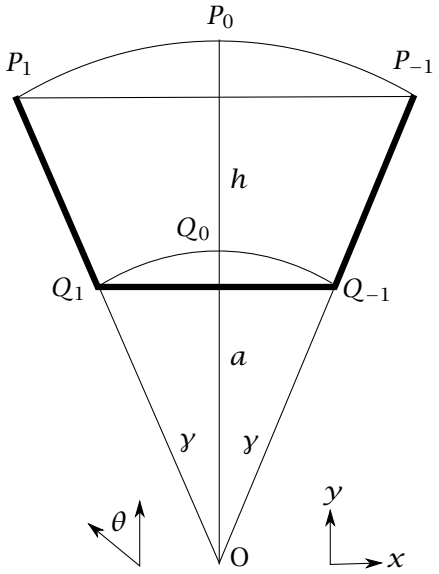


Figure 2: The 2-facet cup (bold lines) in our Cartesian reference frame.

Furthermore, we define the point $P_0 := (0, (a+h)/\cos\gamma)$ to lie on the y-axis and as far away from the origin as the highest point of the cup's edge. Similarly, $Q_0 := (0, a/\cos\gamma)$ has the same distance to the origin as the bottom corners of the cup.

In this particular case, we have:

$$\text{Phase}_{\text{source}} = X \times \Theta = [-a\tan\gamma, a\tan\gamma] \times \left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$$
$$\text{Phase}_{\text{target}} = \tilde{X} \times \tilde{\Theta} = [-(a+h)\tan\gamma, (a+h)\tan\gamma] \times \left(-\frac{\pi}{2}, \frac{\pi}{2}\right).$$

In the following, we address how to analytically determine the number of reflections the light ray undergoes before leaving the fixture. When the light ray meets the cup's wall we reflect the cup at that edge instead of the ray itself. Then the light ray, now a straight line, can be easily traced. This is illustrated in Figure 3 where the red and blue rays are reflected once and twice, respectively. This is an example cup that Philips suggested for study purposes.

For the general case we define the lines $P_{2k+1}Q_{2k+1}$ and $P_{2k-1}Q_{2k-1}$ to be the cup's edges after k reflections where

$$P_k := \begin{pmatrix} \cos\gamma & -\sin\gamma \\ \sin\gamma & \cos\gamma \end{pmatrix}^k P_0, \qquad \text{for } k \in \mathbf{Z}$$

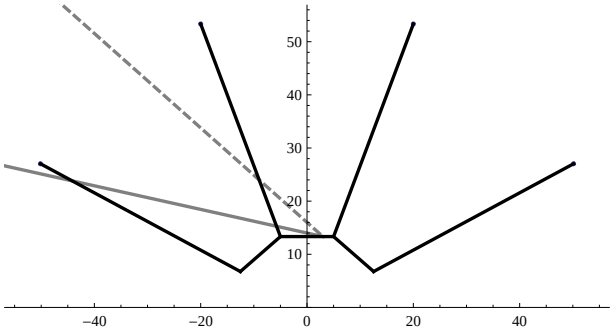


Figure 3: The 2-facet cup with reflections into its sides. Two emitted light rays are shown; the dashed ray with initial conditions (IC) $(x,\theta) = (3, 0.844)$ undergoing one reflection and the other with IC $(x,\theta) = (3, 1.352)$ undergoing two reflections. Cup parameters are $a = 40/3$, $\gamma = \arctan(3/8)$ and $h = 40$ (the Philips example).

are clockwise, $k < 0$, and anti-clockwise, $k > 0$, rotations of P_0 by an angle γ. A similar expression can be obtained for Q_k. Also, $P_{k,1}$ and $P_{k,2}$ are the x and y coordinates of P_k, respectively. The same notation is adopted for any other point, e.g. Q_k.

Then, it becomes obvious that for a fixed cup geometry there is a maximum number of reflections a ray can undergo before escaping the cup. This is the first number $j \in \mathbf{N}$ such that the y-coordinate of the point P_{2j+1} satisfies

$$P_{2j+1,2} \leq a$$

for $\theta \in (0, \pi/2)$. By symmetry we obtain a similar condition for initial ray angle $\theta \in (-\pi/2, 0)$ and y-coordinate of the top edge point $P_{2j-1,2}$ with $j < 0$.

We thus have a finite set $K \subset \mathbf{Z}$ that indexes regions in $\text{Phase}_{\text{source}}$ for which the rays show the same reflective behavior. Define $M_k \subseteq \text{Phase}_{\text{source}}$ $(k \in K)$ to be the set of rays that will undergo $|k|$ reflections before hitting the target where we define the first reflection with the left edge, P_1Q_1 to be positive, i.e. $k > 0$, and negative otherwise, i.e. $k < 0$. If $k = 0$, the ray will not hit any edge at all before reaching the target.

The set $K \subset \mathbf{Z}$ can be determined by solving $P_{k,2} > a$, resulting in

$$K = \left\{ -\left\lfloor \frac{1}{2\gamma} \arccos\left(\frac{a\cos\gamma}{a+h}\right) + \frac{1}{2} \right\rfloor, \ldots, \left\lfloor \frac{1}{2\gamma} \arccos\left(\frac{a\cos\gamma}{a+h}\right) + \frac{1}{2} \right\rfloor \right\}$$

where $\lfloor \cdot \rfloor$ denotes the floor function, giving the nearest smaller integer. Motivated by Figure 3, we calculate equations for the boundaries of M_k since points on such a boundary correspond to planar rays from (x, a) to P_{2k+1}. Considering a suitable triangle in the figure yields a linear equation for those boundaries in

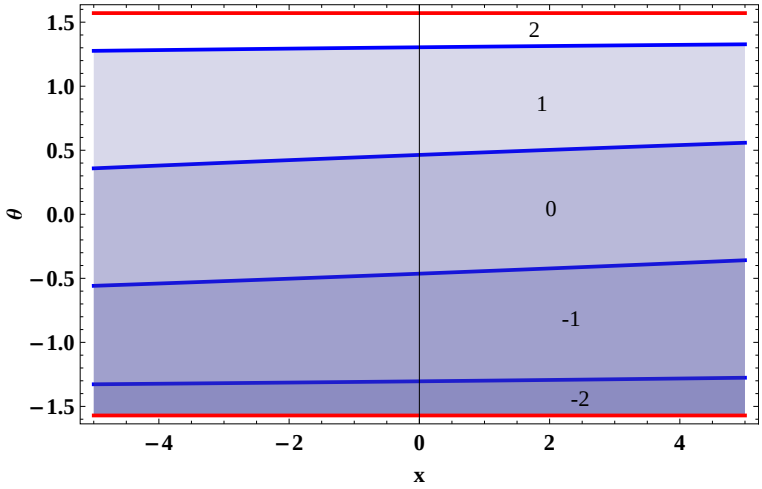


Figure 4: The division of $\text{Phase}_{\text{source}}$ into the regions M_k for $a = 40/3$, $\gamma = \arctan(3/8)$, $h = 40$ (the Philips example).

the coordinates $(x, \tan\theta)$

$$\tan(\theta) = -\frac{P_{2k+1,1} - x}{P_{2k+1,2} - a}.$$

Thus, the boundaries are simply straight lines in the $(x, \tan\theta)$-plane. The subdivision of the phase space $\text{Phase}_{\text{source}}$ into these regions is shown in Figure 4; note that the figure shows the (x, θ)-plane. In this specific case, the boundaries appear straight in these coordinates as well.

To compute the intensities I_{target} or $I_{\mathbf{F}}$ we have to re-trace a ray $(\tilde{x}, \tilde{\theta}) \in \text{Phase}_{\text{target}}$, respectively $\tilde{\theta} \in \mathbf{F}$, back to a set of rays (x, θ) in $\text{Phase}_{\text{source}}$. To do this, we first follow an emitted ray (x, θ). The reflection process is represented by a transformation $T : \text{Phase}_{\text{source}} \to \text{Phase}_{\text{target}}$ which we can describe in several parts, using the restrictions $T|_{M_k}$.

From Figure 5 we conclude that reflection in the positive, respectively negative edge result in the transformations

$$\text{Ref}_+ : \theta \mapsto 2\gamma - \theta$$
$$\text{Ref}_- : \theta \mapsto -2\gamma - \theta$$

Since rays are alternatively reflected on both sides of the cup, if they are reflected at all, we derive, for a general ray in M_k,

$$\text{pr}_2 \circ T|_{M_k}(x, \theta) = (-1)^{k+1}(2k\gamma - \theta). \tag{2}$$

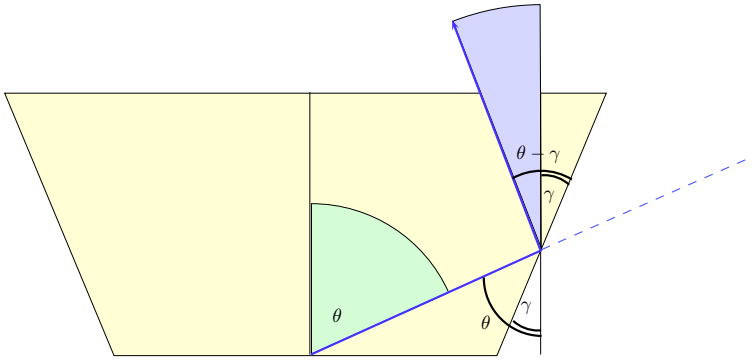
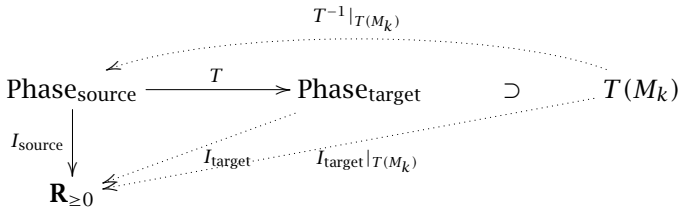


Figure 5: How the angle θ is affected by reflection.

Here, pr_2 denotes the function that projects a vector on its second component, in this case the $\tilde{\theta}$-component of $\mathrm{Phase}_{\mathrm{target}}$. This is a generic function that can be used on any space, and depends on the basis chosen for that space.
Note that it follows from (2) that parallel rays in the same subset M_k of $\mathrm{Phase}_{\mathrm{source}}$ will result in the same final $\tilde{\theta}$.
Now, we can formulate the problem of computing the intensity I_{target} in more detail. For a point $(\tilde{x}, \tilde{\theta}) \in \mathrm{Phase}_{\mathrm{target}}$ we can compose the inverse map T^{-1} with the source intensity. We say that we *push forward* the function I_{source} using T or that we *pull it back* using T^{-1}. To get explicit formulas we have to split up the computation over the different regions $T^{-1}|_{T(M_k)}$ for $k \in K$, since on each region T^{-1} has a different analytic description. This is visualized by the diagram



For simplicity, we concentrate for the moment only on the far field intensity $I_{\mathbf{F}}$.[1] Since, for any $\tilde{\theta}$ in the far field, multiple values of x could in principle have resulted in this final direction we have to integrate over these values on $\mathrm{Phase}_{\mathrm{target}}$ to find the intensity on $\mathbf{F}$, i.e.

$$I_{\mathbf{F}}(\tilde{\theta}) = \sum_{k \in K} \int_{x \in X} I_{\mathrm{source}}(x, (\mathrm{pr}_2 \circ T|_{M_k})^{-1}(\tilde{\theta})) \cdot \updownarrow_{(x, (\mathrm{pr}_2 \circ T|_{M_k})^{-1}(\tilde{\theta})) \in M_k} dx. \qquad (3)$$

[1] Later, we will see, however, that this might actually complicate the computations.

In the case of the 2-facet cup, the regions M_k yield (for each x) intervals with boundaries depending on k and θ for the pull-back integration regions

$$\{x \in X : (x, (\mathrm{pr}_2 \circ T|_{M_k})^{-1}(\tilde{\theta})) \in M_k\},$$

and the previous expression reduces to

$$I_{\mathbf{F}}(\tilde{\theta}) = \sum_{k \in K} \int_{x_{\min}(k,\tilde{\theta})}^{x_{\max}(k,\tilde{\theta})} I_{\mathrm{source}}(x, (\mathrm{pr}_2 \circ T|_{M_k})^{-1}(\tilde{\theta}))dx,$$

where $x_{\min}(k, \tilde{\theta})$ and $x_{\max}(k, \tilde{\theta})$ are the lower and upper bounds of the aforementioned intervals. Since the suggested Lambertian source intensity function (1) is in fact independent of x the above equation simplifies further

$$I_{\mathbf{F}}(\tilde{\theta}) = \mathcal{I}_0 \sum_{k \in K} \cos T_k^{-1}(\tilde{\theta}) \max\{0, x_{\min}(k, \tilde{\theta}) - x_{\max}(k, \tilde{\theta})\}$$

We can compute an analytic expression for this function given parameters (a, γ, h). Utilizing the formulae obtained above for the boundaries of M_k, we derive that

$$\begin{aligned} x_{\max}(\tilde{\theta}) &= \max\{-a\tan\gamma, (P_{2k+1,1} - (P_{2k+1,2} - a)\tan(-T^{-1}(\tilde{\theta}))\} \\ x_{\min}(\tilde{\theta}) &= \min\{a\tan\gamma, (P_{2k-1,1} - (P_{2k-1,2} - a)\tan(-T^{-1}(\tilde{\theta}))\} \end{aligned}$$

Figure 6 shows the intensity profile $I_{\mathbf{F}}$ computed for the Philips example.

To check for correctness of computations: the area under this curve is identical to the integral of the original intensity, corresponding to the total amount of emitted energy which should be conserved. For the given Lambertian source, this energy equals twice the width of the bottom of the cup:

$$E = \int_{\theta=-\frac{\pi}{2}}^{\theta=\frac{\pi}{2}} \int_{x=-a\tan\gamma}^{x=a\tan\gamma} I_0 \cos\theta dx d\theta = 2a\tan(\gamma)\mathcal{I}_0 \cdot \int_{\theta=-\frac{\pi}{2}}^{\theta=\frac{\pi}{2}} \cos\theta d\theta = 4a\tan(\gamma)\mathcal{I}_0.$$

Philips apparently chose their cup shape carefully. Not all cup shapes yield an intensity profile that looks as simple as the one in Figure 6. Specifically, it need not be unimodal even for the 2-facet case, for example see Figure 7 with parameters $a = 3$, $\gamma = \arctan(5/3)$ and $h = 10$ in the far field. We see a bright core appear straight in front of the fixture. Just left and right from the center of the graph there is a dip. This means that on a distant screen a relatively dark ring is present around the center of the illuminated zone which in turn is surrounded by a bright ring. We could call the presence of the dark ring the 'flashlight' or 'torch' effect; this phenomenon is often observed when using these devices even though the reflective sides of a flashlight are mostly not flat.

Instead of considering the intensity on the far field, we could also try to compute it on a nearby target phase space. The intensity on the far field can be obtained from it by integrating over x.

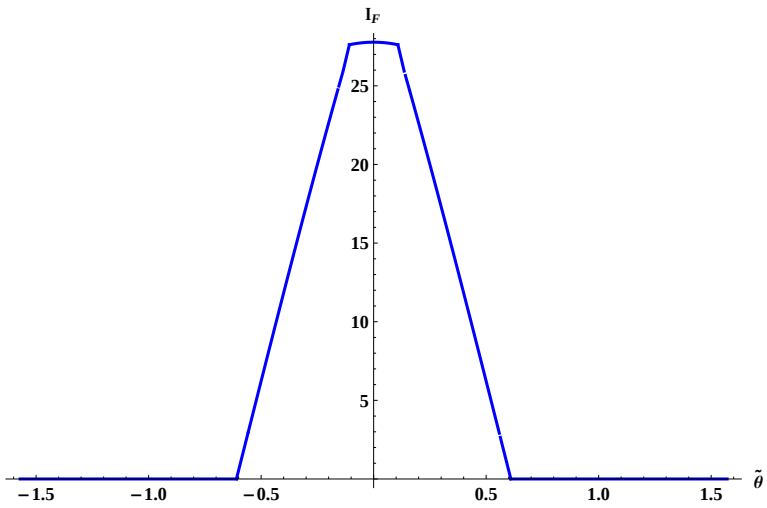


Figure 6: The intensity on **F** for $a = 40/3$, $\gamma = \arctan(3/8)$, $h = 40$ (the Philips example).

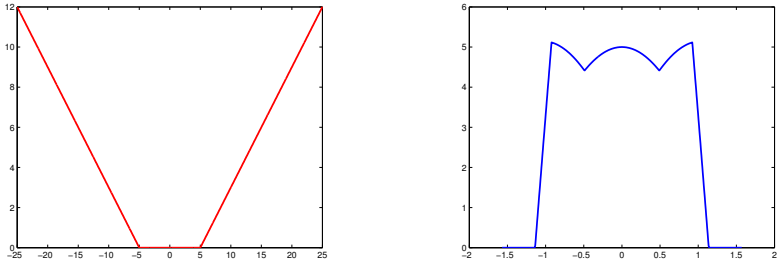


Figure 7: The cup and the intensity function $I_{\mathbf{F}}$ for $a = 3$, $\gamma = \arctan(5/3)$, $h = 10$.

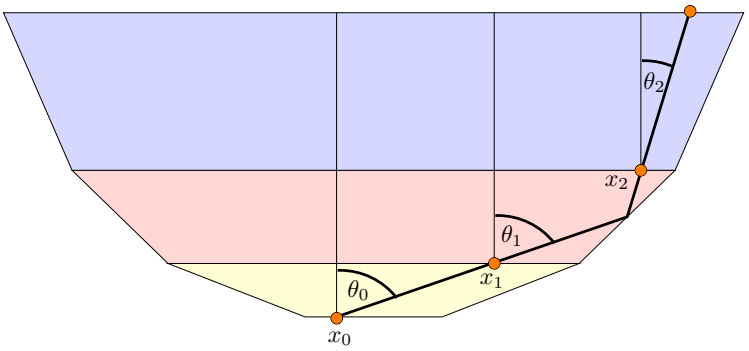
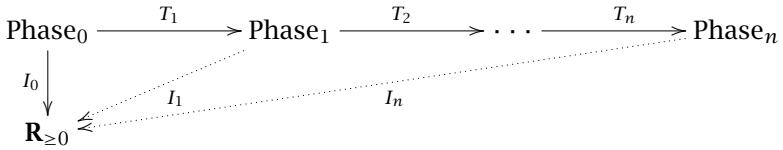


Figure 8: A multi-facet cup can be viewed as a stack of 2-facet cups.

This method allows us to study the symmetric (to the y-axis) multifaceted cup by subdividing the cup into identical subproblems, i.e. by viewing it as a stack of 2-facet cups. The light rays travel from the emitter to the top of the first cup, which is at the same time the bottom of the second. We can consider the continuation after that as a newly emitted ray from the bottom of this second level, see Figure 8. In total, this gives rise to a set of phase spaces

$$\text{Phase}_{\text{source}} =: \text{Phase}_0, \text{Phase}_1, \ldots, \text{Phase}_n := \text{Phase}_{\text{target}}\,.$$

Furthermore, we have a map $I_0 := I_{\text{source}} : \text{Phase}_0 \to \mathbf{R}_{\geq 0}$. Now, if we know the maps $T_i : \text{Phase}_{i-1} \to \text{Phase}_i$ then we also know the ras's path from the bottom of a 2-facet cup to the top and, more importantly, their inverses which means we can compute the intensity on $\text{Phase}_{\text{target}}$. Schematically, we want to push I_0 forward iteratively to the higher phase spaces:



The resulting intensity is

$$I_{\text{target}} = I_0 \circ T_1^{-1} \circ T_2^{-1} \circ \cdots \circ T_n^{-1}.$$

Several problems arise when one tries to implement this technique. Firstly, the maps T_i are not surjective: there are rays at the top of a 2-facet cup that could not have been emitted from the source irrespective of the number of reflections. A simple example one can think of is a ray exiting the top of the cup at the far left, with direction perpendicular to the left boundary. This example shows that the inverse map T_i^{-1} is not defined everywhere and we must keep track of its

domain. Secondly, the definition of T_i is piecewise, depending on the number of reflections, just as we saw above.

The description of T_i^{-1} on each piece where it is analytic is not very difficult. In order to deduce the formula for this map take a ray exiting the cup at $(\tilde{x}, a+h)$ in direction $\tilde{\theta}$. We draw a straight line downwards in direction $\tilde{\theta}+\pi$, retracing the light ray, see Figure 8. Assume that this straight line intersects the polygon $\ldots Q_{-1}Q_0Q_1\ldots$ in (x', y'), that the line segment $(\tilde{x}, a+h)(x', y')$ crosses k reflected sides of the cup and lies to the side of Q_1. The segments correspond to a ray emitted at (x, a) in direction θ that was reflected k times. We quickly deduce that

$$\begin{pmatrix} \cos\gamma & -\sin\gamma \\ \sin\gamma & \cos\gamma \end{pmatrix}^{2k} \begin{pmatrix} x \\ a \end{pmatrix} = \begin{pmatrix} x' \\ y' \end{pmatrix}$$
$$\tilde{x} - x' = -\tan\tilde{\theta}(a+h-y')$$

from which we compute

$$x = \frac{\tilde{x} + (a+h)\tan\tilde{\theta} + a\sin 2k\gamma - a\cos 2k\gamma\tan\tilde{\theta}}{\cos 2k\gamma + \sin 2k\gamma\tan\tilde{\theta}}.$$

Earlier, we already calculated $\mathrm{pr}_2 \circ T|_{M_k}$, and we see that

$$\theta = 2k\gamma + (-1)^k\tilde{\theta}.$$

This gives a complete description of $T^{-1}|_{T(M_k)}$. Due to time constraints we stop short of implementing this and producing a graph of I_{target} for the 2-facet cup.

2.2 Multifacet cups, following the beam backwards

Now, we describe an alternative, rather direct way to implement the computation for $I_{\mathbf{F}}$ of the multi-faceted cup by viewing such a cup as a stack of two-faceted cups.

Instead of starting at the source let us consider the part of the light emanating from the cup that travels in the direction of arbitrary angle $\tilde{\theta}$. Denote a whole interval of parallel rays a *beam*. Its width is the size of a perpendicular cross section of it. The goal is to determine the intensity of the light emitted in direction $\tilde{\theta}$, which corresponds to the width of the beam emanated in this direction. So we follow this beam back in time. Geometrically, we follow it down its path from top to bottom through the stack of cups. It enters the top cup on the line segment $[-(a+h)\tan\gamma, (a+h)\tan\gamma] \times \{a+h\}$. Inside the top cup the beam splits into several pieces (sub-beams), each characterized by the number of reflections in the top cup, where one also has to distinguish between the first reflection being a left or right reflection, as discussed before. These reflected

sub-beams reach the bottom $[-a\tan\gamma, a\tan\gamma]\times\{a\}$ of the top cup in intervals $[x_k,\bar{x}_k]\times\{a\}$ with angle $\theta_k = (-1)^{k+1}(2k\gamma-\tilde{\theta})$. Here k is an index running from $-\lfloor(\pi/2-\tilde{\theta})/(2\gamma)\rfloor$ to $\lfloor(\pi/2+\tilde{\theta})/(2\gamma)\rfloor$. Each interval $[x_k,\bar{x}_k]$ is determined by rotating the bottom of the cup over an angle $2k\gamma$, finding the intersection of this rotated line segment with the light beam, and rotating back. One also needs to take into account the number of reflections (odd or even).

Since the bottom of the upper cup is the top of the cup under it, we now repeat the above process in this next cup (with appropriate a, h and γ) for each of the sub-beams. We then continue inductively until we reach the bottom of the final (lowest) cup. This is where the light source is located, hence we can determine the contribution of the intensity in each sub-beam to the total intensity in the direction $\tilde{\theta}$.

This algorithm has been implemented recursively in Matlab. An example is shown in Figure 9.

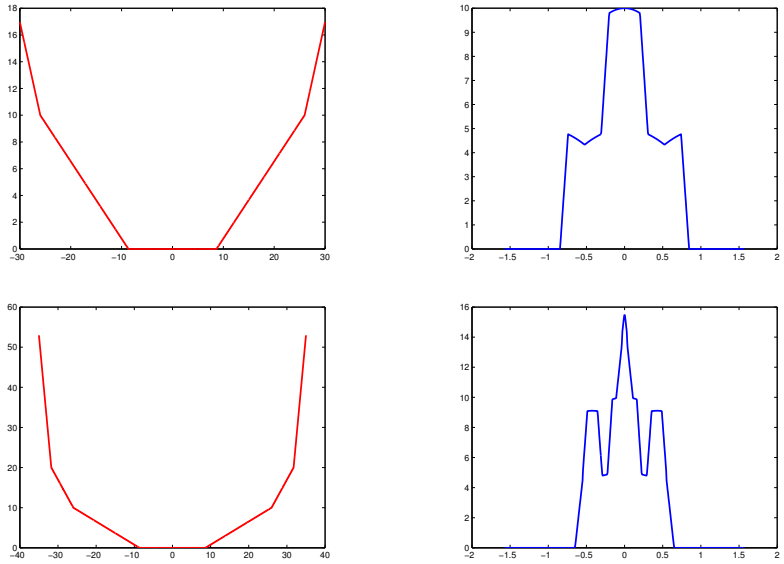


Figure 9: Top row A 4-facet cup with parameters $a_1 = 5$, $\gamma_1 = \pi/3$, $h_1 = 10$, $a_2 = 22.32$, $\gamma_2 = \pi/6$, $h_2 = 20$, and the resullting intensity $I_{\mathbf{F}}$. Bottom row A 6-facet cup with parameters $a_1 = 5$, $\gamma_1 = \pi/3$, $h_1 = 10$, $a_2 = 22.32$, $\gamma_2 = \pi/6$, $h_2 = 20$, $a_3 = 28.09$, $\gamma_3 = \pi/32$, $h_3 = 30$, and the resulting intensity $I_{\mathbf{F}}$. These graphs were obtained using a recursive computation by following a beam of light downwards.

2.3 Polygonal cups

In this section we generalize some of the previous results for cup-shaped fixtures even more, namely to (still two-dimensional) polygonal fixtures. To this end, let $Q \subseteq \mathbf{R}^2$ be a two-dimensional convex polygon. To each edge e of Q we associate a phase space Phase_e whose points parameterize rays leaving e into the interior of Q. Formally, Phase_e is the Cartesian product $e \times \mathbf{R}$, where the first component records the point where the ray leaves e and the second component records the tangent of the angle that the ray makes with the inward pointing normal vector to e.

The rationale for choosing the tangent rather than some other function of that angle is the following beautiful fact: *straight line(segment)s in* Phase_e *correspond bijectively to pencils of rays going through a common point.* Let us elaborate a bit on this. For $x \in e$ the vertical straight line $\{x\} \times \mathbf{R} \subseteq \text{Phase}_e$ corresponds to all rays emanating from x into Q, and for $t \in \mathbf{R}$ the horizontal straight line segment $e \times \{t\}$ corresponds to all lines emanating from e in the direction given by t (and hence intersecting in a common point at infinity in the projective plane). Let a, b be the endpoints of e, listed in clockwise order on vertices of Q. Fix distinct $t_a, t_b \in \mathbf{R}$ and draw the lines l_a, l_b (not just half-lines) from a, b with directions given by t_a, t_b, and let p be their common intersection. If $t_a < t_b$, then p lies on the same side of e as Q, while if $t_a > t_b$, then p lies "behind" e. In either case, a straightforward computation shows that the straight line segment in Phase_e between (a, t_a) and (b, t_b) parameterizes precisely the lines going through p.

Given two distinct edges e and f we define the set $K_{ef} \subseteq \text{Phase}_e$ as the set of all rays emitted from e that hit f next. This is, in fact, a convex polyhedron in Phase_e. To see this, we distinguish two cases. First assume that e and f are not adjacent in Q. Let a, b be the vertices of e in clockwise order, and let c, d be the vertices of f in clockwise order. Then the boundary of K_{ef} consists of all rays from vertex a to edge f, from vertex b to edge f, from edge e to vertex c, or from edge e to vertex d. Let $t_{ac}, t_{ad}, t_{bc}, t_{bd}$ denote the tangents of the angles that the lines ac, ad, bc, bd make with the inward normal of e. Then the observation above shows that K_{ef} is the convex quadrangle in Phase_e with corners $(a, t_{ac}), (b, t_{bc}), (b, t_{bd}), (a, t_{ad})$ (in clockwise order), see Figure 10.

Next, assume that e and f are adjacent. We first assume that f follows e in clockwise order; the opposite ordering is treated similarly. Let a, b be the vertices of e and let b, d be the vertices of f. Now the boundary of K_{ef} consists of the rays from a to f, forming the vertical half-line from (a, t_{ad}) downwards, the rays from b to f, forming the half-line from (b, t_{bd}) downwards [2], and the rays from e to f, forming the segment between (a, t_{ad}) to (b, t_{bd}). Hence, by abuse of notation, K_{ef} is the convex hull of $(a, -\infty), (b, -\infty), (b, t_{bd}), (a, t_{ad})$. See Figure 11, also for the case where e follows f in clockwise order and where

[2] to see that this is the right half-line, use a limiting argument: consider lines from a point $x \in e$ close to b to f

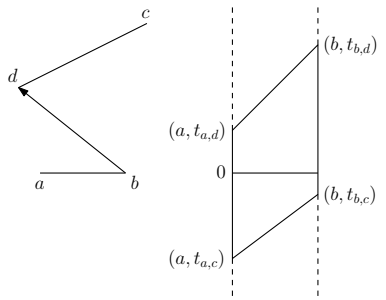


Figure 10: The polyhedron K_{ef} for e, f not adjacent.

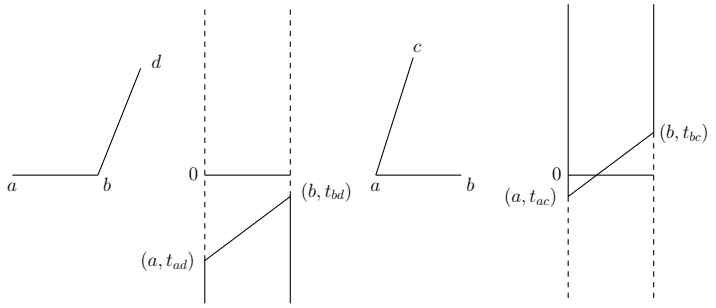


Figure 11: The polyhedron K_{ef} for e, f adjacent.

K_{ef} is the convex hull of $(a, t_{ac}), (b, t_{bc}), (b, +\infty), (a, +\infty)$.

Still for distinct edges e, f of Q we define the map $T_{ef} : K_{ef} \to \text{Phase}_f$ that takes a ray in Phase_e traveling from $x \in e$ to $y \in f$ and returns the ray in Phase_f traveling back from y to x (we do *not* yet reflect here). This map is *not* affine-linear. Indeed, as before, label the endpoints of e by a, b and those of f by c, d. Let u_e, u_f be inward pointing normals of e, f of the same lengths as e, f, respectively. Then the second component of $T_{ef}(x, t)$ equals the rational expression

$$\frac{[u_e + t(a-b)] \cdot [c-d]}{[u_e + t(a-b)] \cdot u_f}$$

in t, where $\cdot$ is the dot product on $\mathbf{R}^2$. The first component is computed by intersecting with f the ray emanating from $x \in e$ in direction t with f.

Note that T_{ef} is a bijection between K_{ef} and K_{fe} with inverse T_{fe}. Strictly speaking, this is only true for the *interiors* of these polyhedra, but it will be convenient in our computations to use their boundaries in our computations. To this end, when $f = (b, d)$ follows $e = (a, b)$ in clockwise order of edges, it will by convenient to set $T_{ef}(a, -\infty), T_{ef}(b, -\infty) := (b, t_{ba})$, where t_{ba} is the tangent of the (b, a) makes with the inward normal to f. Similarly, we set $T_{ef}(b, t_{bd})$

equal to *both* (d,∞) and (b,∞) (these should really be thought of as one and the same point in the projective plane). When $e=(a,b)$ follows $f=(d,b)$, we set $T_{ef}(a,\infty), T_{ef}(b,\infty) := (a,t_{ab})$ and $T_{ef}(a,t_{ac})$ equal to *both* $(c,-\infty)$ and $(a,-\infty)$. Note that if we use, on any edge $e=(a,b)$, the coordinate t for the point $(1-t)a+tb$, then T_{ef} is orientation reversing.

In spite of the fact that T_{ef} is not affine-linear, it *does* have the property that the pre-image $T_{ef}^{-1}m$ of a straight line segment $m\in \mathrm{Phase}_f$ is a straight line segment in K_{ef}. Indeed, the line segment m corresponds to (part of) the pencil of lines from f going through some common point p (possibly behind f or even at infinity). But then then all lines in $T_{ef}^{-1}m$ go through p, as well, and hence lie on a line segment in Phase_e. This argument shows that the bijection $T_{ef}: K_{ef}\to K_{fe}$ maps polyhedra to polyhedra. If a polyhedron in K_{ef} is given by its clockwise list of corners, then mapping T_{ef} to that list gives the counterclockwise list of corners. When e and f are adjacent, some of the corners may be mapped to an ordered "pair" of points at infinity.

For tracing rays traveling through Q we will need the maps $\tau_e : \mathrm{Phase}_e \to \mathrm{Phase}_e,\ (x,t)\mapsto(x,-t)$; these reflect rays in the normal vector u_e.

Now, consider a sequence $s=(e_1,\dots,e_m)$ of edges of Q, where $e_i\neq e_{i+1}$ for all i. We claim that the set K_s of rays in Phase_{e_1} that travel to e_2, and then after reflection to e_3, etc, and finally to e_m, form a (possibly empty) convex polyhedron. We have already seen this for $m=2$, where K_{e_1,e_2} is as above. For $m=1$ it is also true if we agree that $K_{(e_1)}$ is just all of Phase_{e_1}. For $m>2$ let $s'=(e_2,\dots,e_m)$ be the tail of s. Then $K_s = T_{e_1e_2}^{-1}\tau_{e_2}K_{s'}$, and the claim follows from the observation that τ_{e_2} is affine-linear and that $T_{e_1e_2}$ pulls back straight line segments to straight line segments. This also gives an inductive *algorithm* for computing K_s: intersect $\tau_{e_2}K_{s'}\subseteq\mathrm{Phase}_{e_2}$ with K_{e_2,e_1}. This gives a polyhedron. Map its corners into Phase_e by means of $T_{e_2e_1}$. The polyhedron with the corners thus obtained is K_s.

Now that we have an algorithm for computing K_s for any sequence of edges, we single out two edges source, target of Q representing the source and the target, respectively. We want to enumerate all sequences s of edges, starting with source and ending with target, for which K_s is non-empty (or even better, an honest, two-dimensional polyhedron), and then we want to draw the corresponding polyhedral subdivision of $\mathrm{Phase}_{\mathrm{source}}$. One issue with this is that for some triples $(Q,\mathrm{source},\mathrm{target})$ this subdivision is not finite. We will assume, however, that it is. Then it can be computed recursively as follows: given a string $s'=(e_2,\dots,e_m)$ with $m\geq 1$ for which $K_{s'}\subseteq\mathrm{Phase}_{e_2}$ is non-empty (respectively, two-dimensional), check whether $e_2=\mathrm{source}$. If so, then return s' and the corresponding polyhedron. If not, let e_1 run through all edges of Q distinct from e_2, set $s:=(e_1,\dots,e_m)$, and compute K_s from $K_{s'}$ as above. If it is non-empty (respectively, two-dimensional), call the procedure just described with the sequence s.

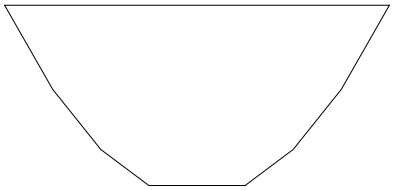


Figure 12: The polygon Q.

An example

Consider the polygon Q with vertices $(\pm i, (i^2-1)/4)$ for $i = 1, 2, 3, 4$, a reasonable approximation of a parabolic collimator, see Figure 12. We label the edges $1, \ldots, 8$ in clockwise order; edge 1 is source, edge 5 is target.

It turns out that there are 15 strings s for which K_s is non-empty. They are:

$$(1,5),(1,2,5),(1,3,5),(1,2,3,5),(1,4,5),$$
$$(1,2,4,5),(1,3,4,5),(1,2,3,4,5),(1,6,5),(1,7,6,5),$$
$$(1,8,7,6,5),(1,8,6,5),(1,7,5),(1,8,7,5),(1,8,5).$$

The corresponding polyhedra in $\text{Phase}_{\text{source}}$ and $\text{Phase}_{\text{target}}$ are depicted in Figure 13. Another example, now with 4 replaced by 7 to resemble a parabola even closer, gives rise to Figure 14.

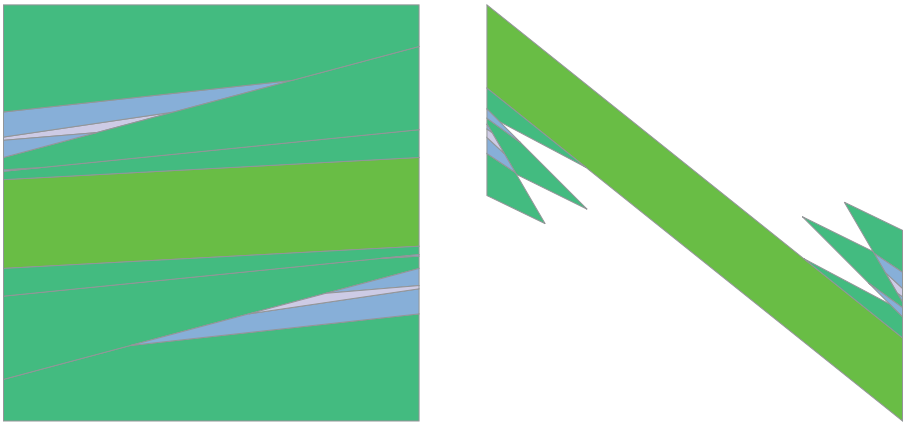


Figure 13: Left The polyhedra K_s—lighter colors indicate more reflections. The line segments separate K_s for different s. Two of the regions are too small to discern. Right The images in $\text{Phase}_{\text{target}}$ of the polyhedra K_s.

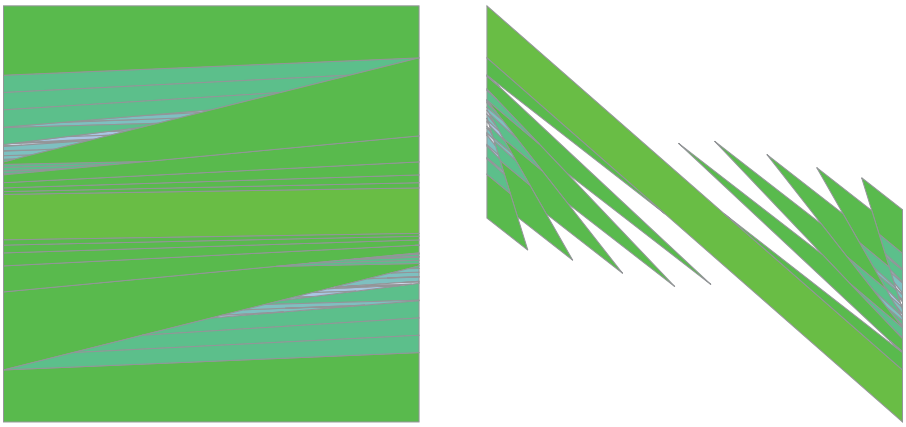


Figure 14: Left The polyhedra K_s—lighter colors indicate more reflections. The line segments separate K_s for different s. Two of the regions are too small to discern. Right The images in $\text{Phase}_{\text{target}}$ of the polyhedra K_s.

3 Two-dimensional smooth cups

In this section we will study a 2-dimensional cup with smooth edges. The aim is to find an efficient numerical algorithm to reconstruct the partition of $\text{Phase}_{\text{source}}$. This was also suggested by the problem owners. For now, this method still has a somewhat limiting restriction on the cup shape, the fourth assumption below. Possibly, it can be removed with a little further research and small adaptations of the method. We assume

1. $p(x) \in C^1((-1,1))$, where $p(x)$ is function that defines the shape of the cup;

2. $X = (h_{-1}, h_1) \times \{p(h_1)\}$, defined so that the light source occupies the whole space between the sides of the cup at height $p(h_1) \in [0, p(1))$;

3. $\tilde{X} = (-1,1) \times \{p(1)\}$;

4. the tangent to the cup does not tend to a vertical when approaching the left or right cup edge

Each point of the source emits light with angle $\theta \in (-\pi/2, \pi/2)$ and we deal with the space $\text{Phase}_{\text{source}}$ as before. We present results for a cup described by $p(x) = 10x^6$. The space $\tilde{X}$ is delimited by end points $P_{-1} = (-1, p(-1))$ and $P_1 = (1, p(1))$. Our example light source is the horizontal strip $(-0.5, 0.5) \times \{0.3\}$, see Figure 15.

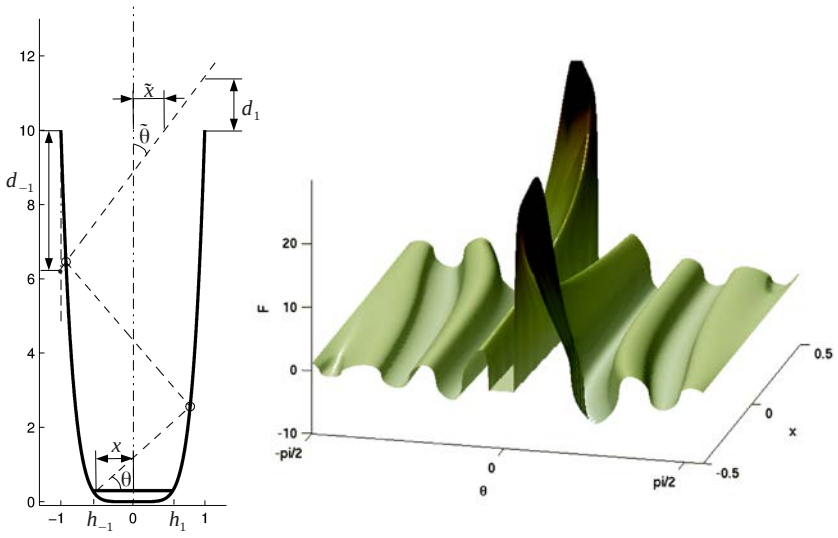


Figure 15: Left A smooth cup described by $p(x) = 10x^6$. The light source is the segment $(-0.5, 0.5) \times \{0.3\}$. The dashed line is the trajectory of a light ray with two reflections. Right The continuous function F is defined so that it vanishes on the boundaries of the sets M_k.

3.1 Reconstruction of the partition for $\mathrm{Phase}_{\mathrm{source}}$

We deal with the numerical reconstruction of the partition M_k of $\mathrm{Phase}_{\mathrm{source}}$ where

$$\bigcup_k \overline{M_k} = \mathrm{Phase}_{\mathrm{source}}, \qquad \bigcap_k M_k = \emptyset. \tag{4}$$

Since each set in the partition is a region (connected, open) in $\mathrm{Phase}_{\mathrm{source}}$ it is enough to efficiently reproduce the union $S = \cup_k \partial M_k$ of the boundaries separating the open sets M_k. It is easy to show that a ray corresponding to a point $s \in S$ passes exactly through one of the points P_{-1} or P_1. Note that the points $P_{-1,1}$ are in $\partial\tilde{X}$ and not in $\tilde{X}$. These facts play a key role in the method of the current section.

We denote $T(x,\theta)$ to be a ray-tracing map – a function that determines the angle and position of the exiting ray for the angle and position of the emitted ray

$$\begin{aligned} T : \mathrm{Phase}_{\mathrm{source}} &\longrightarrow \mathrm{Phase}_{\mathrm{target}} \\ (x,\theta) &\mapsto (\tilde{x},\tilde{\theta}). \end{aligned}$$

The map $T(x,\theta)$ is continuous when $(x,\theta) \in M_k$ and has a jump discontinuity when $(x,\theta) \in \partial M_k$. This fact dramatically affects the convergence speed of common numerical methods. Better results can be achieved by introducing the continuous function

$$F : \mathrm{Phase}_{\mathrm{source}} \longrightarrow \mathbf{R}$$

that we will now describe. Moreover, we define $d_{-1,1} : \mathrm{Phase}_{\mathrm{target}} - \tilde{X} \times \{0\} \to \mathbf{R}$ by

$$\begin{aligned} d_{-1}(\tilde{x},\tilde{\theta}) &:= \cot\tilde{\theta}(1+\tilde{x}) \\ d_1(\tilde{x},\tilde{\theta}) &:= -\cot\tilde{\theta}(1-\tilde{x}). \end{aligned}$$

The geometrical interpretation is that d_{-1} is the signed vertical distance from P_{-1} to the intersection of the line $x = -1$ with the line on which the exiting ray lies. It is positive if the intersection point lies above P_{-1}. Similarly, d_1 is the signed vertical distance from P_1 to the intersection of the line $x = 1$ with the line on which the exiting ray lies. It is positive if the intersection point lies above P_1. This is illustrated in Figure 15. Note that for $\theta = 0$, also $\tilde{\theta} = 0$, the light ray goes straight up, and d_{-1}, d_1 are undefined. Moreover, in the vicinity of the line $\theta = 0$ in $\mathrm{Phase}_{\mathrm{source}}$, d_{-1} and d_1 are unbounded. Our function F will handle this phenomenon, which is unpleasant for root finding. More importantly, if the light ray grazes an upper edge of the cup, either $d_{-1} \circ T$ or $d_1 \circ T$ has a jump discontinuity across zero and the other is zero, see Figure 16. The points for which this is the case form S. This discontinuity is bad for an approximation algorithm to localize the zeros. The function F, that we now define,will remedy

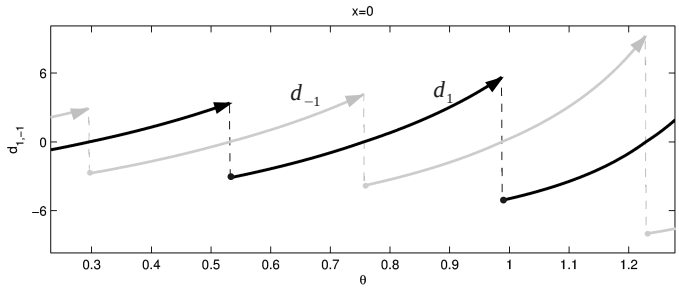


Figure 16: The functions $d_{-1} \circ T$ and $d_1 \circ T$ alternate their signs around any $s \in S$ when fixing x and varying θ. However, only one of the two is continuous at s.

this problem and allow us to find the zeros of both functions simultaneously. We set

$$F(x,\theta) := \begin{cases} (d_{-1} \circ T)(x,\theta) \cdot (d_1 \circ T)(x,\theta) \cdot \operatorname{sign}(\theta)\operatorname{sign}(\mathrm{pr}_2 \circ T(x,\theta)), & (x,\theta) \notin M_0 \\ F_0(x,\theta), & (x,\theta) \in M_0 \end{cases}$$

$$= \begin{cases} d_{-1}(\tilde{x},\tilde{\theta}) \cdot d_1(\tilde{x},\tilde{\theta}) \cdot \operatorname{sign}(\theta \cdot \tilde{\theta}), & d_{-1}(x,\theta) \cdot d_1(x,\theta) < 0 \\ F_0(x,\theta), & d_{-1}(x,\theta) \cdot d_1(x,\theta) \geq 0 \end{cases}$$

outside of S. Here F_0 is a simple continuous function of our choosing that is strictly positive on M_0 and zero on ∂M_0. It is introduced to overcome the discontinuity problem around $\theta = 0$. An example of a suitable F_0 is

$$F_0(x,\theta) = -\left(\theta - \arctan\left(\frac{p(1) - p(h_1)}{x-1}\right)\right)\left(\theta - \arctan\left(\frac{p(-1) - p(h_{-1})}{x+1}\right)\right).$$

It turns out that F is a continuous function on the whole $\mathrm{Phase}_{\mathrm{source}}$ and equal to zero precisely on S. The reason this is true is that when one of the compositions $d_{-1} \circ T$ and $d_1 \circ T$ continuously approaches zero, when we vary the ray such that it passes one of the edge points P_{-1}, P_1, the other stays bounded.

The other important property of F is that when we cross one of its zeros, the signs of both d_{-1} and d_1 switch. But also, the sign of $\tilde{\theta}$ switches. For completeness, the signs of d_{-1}, d_1 and $\tilde{\theta}$ are $(-,+,-)$, respectively, if the last reflection is on the cup's left side and $(+,-,+)$, respectively, if it is on the right. Also, the factor $\operatorname{sign}(\theta)$ ensures that sign switching occurs at ∂M_0. As a consequence, the function F always changes its sign at zeros. This is crucial to root approximation algorithms.

In summary, the problem of reconstructing (4) has now been expressed in terms of F

$$s \in \bigcup_{k \in K} \partial M_k \iff F(s) = 0. \tag{5}$$

Faced with problem (5) one has a vast choice of numerical root-finding methods. For example, both the secant method and inverse interpolation have been implemented in many mathematical packages, like GNU Octave or Matlab.

Usually, an implementation of a root-finding algorithm expects only a one-dimensional function as an input. Therefore, the following strategy might be useful. Fix a sequence of equally spaced points $(x_i)_{i=0}^n$, $x_i \in X$, so that $x_{i+1} - x_i = d$ is a small number, called a discretization step. We now iteratively solve the one-dimensional sub-problems $F(x_i, \theta) = 0$ for θ. To do this in a smart way, when computing the solution set $\{x_i\} \times \Theta_i$, we use the previously computed solution sets $\{x_j\} \times \Theta_j$ for $j = 1, 2, \ldots, i-1$ to determine the starting points for a one-dimensional root-finding algorithm.

More generally speaking, one may use a whole ensemble of numerical tricks introduced for ODE, e.g. self adaptive grids, multi-step methods. For example, while reconstructing trajectory in $\text{Phase}_{\text{source}}$, an algorithm may automatically vary the discretization step d so the overall result meets some error estimate, using as few nodes as possible. This is illustrated in Figure 17 on the left. The nodes are more densely distributed around critical places.

Despite the case with ODE systems, root-finding is a much "better behaved" process since no accumulative error is introduced. A root-finding algorithm will always converge to an exact value within margins of the predefined tolerance.

4 Monte-Carlo methods for intensity computation

A simple and practical, albeit computationally intensive way to compute the intensity profile for a given reflector is to use a Monte-Carlo method to perform essentially numerical integration. Such an approach is part of a larger class of generically known as Monte-Carlo (MC) methods. These rely mostly on simulation of the processes of interest (either random or not), and use the outcomes of the simulation to compute important quantities (in this case, the intensity profile). These methods are simple enough, can be endowed with performance guarantees, and rely on the fact that it is generally easy and quick to simulate the desired processes. The latter point is critical, and it is often the bottleneck of such approaches. For the problem under consideration simulation of the process involves ray-tracing which, despite its simplicity, can be too computationally expensive if one is required to compute several millions of rays to ensure the desired performance. Therefore a naïve and straightforward application of such methods can still be prohibitive. In this section we discuss two possible approaches to *bootstrap* the basic MC approach, which will use a small number of ray-tracing experiments to reconstruct the intensity profile to a high accuracy, by relying on the analytical considerations of the previous sections, as well as by using clever sampling strategies to choose only "important" rays to simulate.

The overall goal of this section is to compute the intensity function at the target

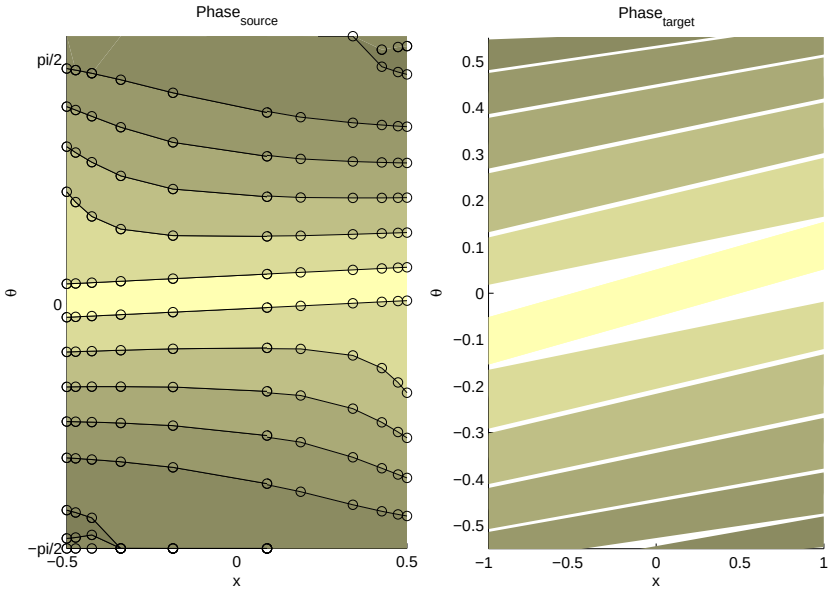


Figure 17: Left The partition of $\text{Phase}_{\text{source}}$ into M_k for the cup described in Figure 15 was obtained numerically. The $\circ$-signs mark grid points computed with a mix of the secant method and inverse interpolation root-finding algorithms. Right The partition of $\text{Phase}_{\text{target}}$ into regions with same number of reflections k. In both pictures, darker colors correspond to a higher number of reflections.

for different designs of optical systems. As was seen before, this requires both knowledge of the transformation T as well as a slightly complicated pull-back integration step. As should be clear by now, even for very simple two-dimensional faceted reflector designs deriving an analytical solution is already rather involved. Therefore extension to non-faceted cups and the three-dimensional setting is likely to be a daunting task. Another possibility is the use of Monte-Carlo integration to approximate the intensity function I_{target}. For a single ray, parameterized by a source position x and source angle θ we can evaluate the transformation function T and therefore the target angle $\tilde{\theta}$ and target position $\tilde{x}$. We restrict ourselves to the far field problem explained in the introduction, hence we are not interested in the target position. By collecting many such samples the integral (3) can be computed numerically to a desired precision level. Unfortunately, to have any reasonable accuracy this method requires a huge number of samples and it is not efficient. However, leveraging on the information for the previous section a different approach can be taken.

The analytical results of the previous sections inform us of the nature of the source phase space which for faceted cups can be decomposed into several polygonal (and furthermore convex) regions in which rays display the same reflective behavior. Since for all rays in each region the reflective behavior is qualitatively the same (i.e. these rays "hit" exactly the same reflective surfaces) this means that two "close-by" rays within the same region (i.e. rays starting at nearby points x_1 and x_2, with similar starting angles θ_1 and θ_2) will display nearly the same behavior. In fact for faceted cups given a few rays in each region it is possible to quickly infer the behavior of *any other ray* in that same region. This means that a Monte-Carlo approach only needs to trace a very small number of rays in each region. The previous statement is obviously conditional on the knowledge of the way the input phase-space is partitioned. In this section we describe two ways of using MC methods based on this rationale:

1. The partition of $\text{Phase}_{\text{source}}$ is given – this can be accomplished either using analytical methods (see Section 2), or through clever use of ray-tracing (see Section 3).

2. The regions partitioning $\text{Phase}_{\text{source}}$ into equivalence classes of reflective behavior can be assumed to be convex (or nearly convex). In this case we propose an algorithm that carefully chooses rays to be traced, and effectively estimates the desired partition of the input phase space, while also computing the intensity profile.

4.1 Method 1: Known input phase space partition

Assume we know how the input phase space is partitioned into regions with the same reflective behavior. In particular, we know for each ray $(x, \theta) \in X \times \Theta$ the number of reflections it experiences. For multifaceted cups this information can

be provided analytically. For smooth-shaped reflectors the method in Section 3 can be used to approximately get this information. The method described is similar in spirit to the original naïve MC method, and relies on the characterization of the behavior of a large number of rays. However, for most of these rays this characterization does not require invoking a ray-tracing routine, therefore dramatically decreasing the computational demands.

The proposed method begins by generating a small sample of n_1 points uniformly distributed over the input phase space $\text{Phase}_{\text{source}}$. Let this sample be denoted as the *preliminary sample*

$$\{(x_i, \theta_i)\}_{i=1}^{n_1} .$$

For each one of these points we run a ray-tracing routine. This means that for each point (x_i, θ_i) in the preliminary sample we know both the target angle $\tilde{\theta}(x_i, \theta_i)$ and the number of reflections $r(x_i, \theta_i)$. This step forms the first stage of the method.

In the second stage we generate a much larger uniform sample of n_2 points from the source phase-space, given by the set

$$\{(x_j, \theta_j)\}_{j=n_1+1}^{n_1+n_2}$$

which we denote by the *estimation sample*. Now, instead of running the ray-tracing routine for each of these points we are going to make use of the assumed knowledge about the phase-space. In particular, with that knowledge we can compute the number of reflections $r(x_j, \theta_j)$ for any point in our sample, without resorting to ray-tracing. So all that is left to be calculated is the exit angle $\tilde{\theta}(x_j, \theta_j)$. Let $j \in \{n_1 + 1, \ldots, n_1 + n_2\}$ and consider the point (x_j, θ_j). Now simply take the three "nearest" rays with the same number of reflections in the preliminary sample.[3] Define

$$d(i, j) \equiv \sqrt{(x_i - x_j)^2 + (\theta_i - \theta_j)^2} , \tag{6}$$

and let $G_j = \{i \in \{1, \ldots, n_1\} : r(x_i, \theta_i) = r(x_j, \theta_j)\}$. Finally, let $i_1, i_2, i_3 \in G_j$ be three distinct points so that for any $i \in G_j$ (unequal to i_1, i_2, i_3) we have $d(i, j) \geq \max\{d(i_1, j), d(i_2, j), d(i_3, j)\}$. We determine $\tilde{\theta}(x_j, \theta_j)$ by linear interpolation using the three nearest neighbours. In other words, determine $\tilde{\theta}(x_j, \theta_j)$ so that the points

$$(x_{i_1}, \theta_{i_1}, \tilde{\theta}(x_{i_1}, \theta_{i_1})),\ (x_{i_2}, \theta_{i_2}, \tilde{\theta}(x_{i_2}, \theta_{i_2})),\ (x_{i_3}, \theta_{i_3}, \tilde{\theta}(x_{i_3}, \theta_{i_3})),\ \text{and}\ (x_j, \theta_j, \tilde{\theta}(x_j, \theta_j))$$

are all co-planar. This simple computation is done for all the points in the estimation sample, and can be significantly faster than performing ray-tracing. For

[3] Note that n_1 needs to be sufficiently large, such that each region contains at least three rays from the preliminary sample.

faceted reflectors (in two and three dimensions) this interpolation is actually exact, so in the end its outcome is essentially equivalent to ray-tracing, but less computationally demanding. Finally, with all these in hand, one can just proceed as in regular Monte-Carlo simulation to compute the intensity function, using the $n_1 + n_2$ points.

Remarks: At this point it is important to issue a number of remarks.

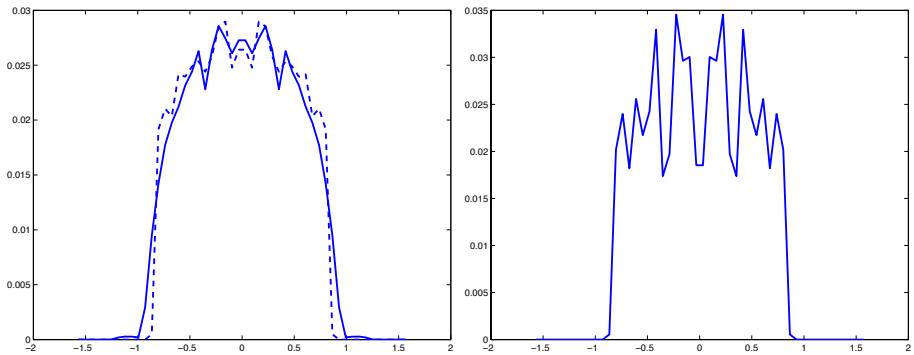
- For faceted reflectors the interpolation step is not approximate, as the exit angle does not depend on the ray position within each of the regions of the source phase-space, and the source and target angles satisfy linear relations.

- The number of reflections does not uniquely identify each of the regions in the phase space, but for our purposes (in two dimensions) it does suffice. This method can easily be extended to resolve this problem, by using instead a label for each reflective path.

- This method will work extremely well *provided* the partition of the phase space is accurate, otherwise it will introduce some systematic errors that can affect the intensity profile estimation. This can be rather undesirable and motivates the second method proposed (see also the simulation results below).

- The notion of distance in (6) is obviously open to discussion as it does not reflect the physics of the problem. The above choice was taken only for simplicity and there are certainly other distances that can be considered.

To assess the potential of this method we used it to compute the intensity profile of the smooth reflector similar to the one used in Section 3. The method described in that section gave rise to the estimated partition of the input phase space, which resembles the one depicted in Figure 17. Using this partition as input to the method described above we constructed an estimate of the intensity profile. In Figure 18a we took $n_1 = 512$ preliminary rays, and ran this method using $n_2 = 2^{18}$. Also in this figure is the result of applying the naïve Monte-Carlo approach with $n_1 = 2^{18}$ rays (this is essentially the true intensity profile). Note that qualitatively the profile obtained by bootstrapping only 512 rays is similar to the "true" profile, however, there are striking differences that cannot be ignored. This systematic error is due mostly to misspecification of the input phase space partition, in which curves were approximated by interpolation of several grid points (see Section 3 for details). To mitigate this drawback we propose a second method that does not rely on such prior knowledge about the input phase space.



(a) (b)

Figure 18: Intensity profile of a smooth reflector cup obtained using bootstrapped MC method 1 using 512 rays (solid line in panel a), and using regular MC with 2^{18} rays (dashed line in panel a). In panel b the result obtained using regular MC with only 512 rays is also depicted.

4.2 Method 2: Unknown partition

In this subsection we give a modified version of the MC method, this time not relying on the complete specification of the input phase space partition. The problem with the application of Method 1, proposed in the preceding paragraphs, is that one needs to have a very accurate description of the input phase space partition and small errors in such description can give rise to significant errors in the intensity profile. Whenever one has an analytical description of the partitioning this is not a problem, but when numerical methods are used to construct such a partition then it becomes difficult to provide any performance guarantees.

The procedure we propose in this section makes very few assumptions about the true partition of $\text{Phase}_{\text{source}}$ (where each region corresponds to a certain reflectivity path). In particular these regions are assumed to be connected (true for any convex reflector). Furthermore we assume these regions are convex when $\text{Phase}_{\text{source}}$ is parameterized by position x and tangent of the source angle $\tan\theta$. The second assumption is true for two-dimensional faceted cups (as the partition regions are convex polygons in such parametrization), but in general false for smooth cups. Nevertheless, if the curvature of these boundaries is modest this assumption is approximately true locally which is all that is required by the method (see more remarks on this later on).

The first step of this method is the same as before: generate a uniformly distributed *preliminary sample* of n_1 points in $\text{Phase}_{\text{source}}$.

$$\{(x_i, \theta_i)\}_{i=1}^{n_1} .$$

As before, for each of these points we run the ray-tracing routine. This means

that for each point (x_i, θ_i) in the preliminary sample, we know both the target angle $\tilde{\theta}(x_i, \theta_i)$ and the number of reflections $r(x_i, \theta_i)$. The second step of the method begins in the same fashion as before, by generating a much larger uniform sample of n_2 points from the source phase space, given by the set

$$\{(x_j, \theta_j)\}_{j=n_1+1}^{n_1+n_2} ,$$

which we denote by the *estimation sample*.

Now, take $j \in \{n_1 + 1, \ldots, n_1 + n_2\}$. Begin by identifying the $K \geq 3$ closest neighbours in the preliminary sample (using again the distance in (6)). For our purposes $K = 12$ was a reasonable option but the proposed method is not terribly sensitive to this choice. For concreteness let $i_1, \ldots, i_K$ denote the K neighbours. The next step is to divide the set of K neighbours into groups of points with the same number of reflections (alternatively the same reflective path). More concretely, for a certain number of reflections $c \in \mathbf{N}_0$ define

$$G_c = \{k \in \{1, \ldots, K\} : r(x_{i_k}, \theta_{i_k}) = c\} .$$

For each $c \in \mathbf{N}_0$ we check if the point $(x_j, \tan\theta_j)$ is *inside* the convex hull of the points $(x_{i_k}, \tan\theta_{i_k})$, with $k \in G_c$. If our assumptions hold then this will happen for at most one group and ensures that the number of reflections for ray (x_j, θ_j) is exactly given by the number of reflections of that group. Suppose this condition holds for c_j^* reflections. Then we are guaranteed that $r(x_j, \theta_j) = c_j^*$ and so we can use all points in $G_{c_j^*}$ to estimate the target angle which can be easily done by linear regression. If the convex hull test is negative for all values c then we cannot identify the point membership and need to run the ray-tracing routine, after which we can add this ray to the preliminary sample rays. Finally, once all the points in the estimation sample have been processed one can proceed with the regular MC integration, as before.

Remarks: Before discussing some experimental results it is important to issue a number of remarks:

1. Note that this method simultaneously identifies the partition of $\text{Phase}_{\text{source}}$ and estimates the intensity profile. Under the faceted cup assumption the partition is convex and the target angle is a linear function of the input angle. This means that the method is exact, in the sense that its outcome is the same as if all the rays were ray-traced. However, the number of ray-traced points is much smaller, mostly consisting on rays near the partition boundaries.

2. The convex partition assumption is required to ensure that the partition membership test is accurate. When this assumption is not met a non-vanishing bias in the estimation of the intensity profile will most likely be present. Nevertheless this bias should typically be quite small: the approximation quality depends on the curvature of the boundaries between

the various regions of the $\text{Phase}_{\text{source}}$ space, and on the average distance between neighboring ray-traced points. This means that in practice the convexity assumption is essentially valid for most smooth cups one might want to consider. Therefore, the primary source of error is the linear regression step. A possible way to ensure there is no asymptotic bias (due to non-convexity) is to modify the proposed procedure in the following way: whenever the convex hull test is positive one "flips" a coin such that with probability $p > 0$ (but small) the corresponding ray-trace is collected.

3. Higher-order regression models can also be considered, and will likely reduce the bias created by using simple linear regression.

4. With the current approach one always ends up at a uniform sample of rays from $\text{Phase}_{\text{source}}$. This is not necessarily the best way to proceed, although it is convenient for the simple MC integration method used. One could use the proposed sampling ideas to construct non-uniform samples that might increase the efficiency of the numerical integration. This is a possibility for future research and was not explored in the current work.

5. It is possible to endow the MC integration step with performance guarantees, which means that, for two-dimensional faceted cups, the performance of this method can be fully characterized. We conjecture that, for three-dimensional faceted reflectors with planar polygonal sources, similar convexity assumptions hold. This means that this method can also be endowed with performance guarantees in that case.

We illustrate this method for the two-faceted cup and the smooth cup described in Section 3. First, we deal with the two-faceted cup. We choose a preliminary sample size $n_1 = 2^9$ and the estimation size $n_2 = 2^{19}$. In Figure 19a we plot the obtained intensity profile. Compare this with the analytical result in Figure 6. Note that in this case the number of calls to the ray-tracing subroutine was only 31906. In Figure 19b we applied the naïve MC method with $n_1 = 31000$ and in Figure 19c we used the same naïve method with $n_1 = 2^{19}$ rays. Note that panels a and c are nearly identical but the latter required a large number of ray-tracing operations. It is important to notice that we invoke the ray-tracing routine mostly for points near the partition boundaries. Figure 20a illustrates this, where we plot the number of reflections vs. source position and angle. In green are the points in the preliminary sample, in red the points of the estimation sample that were ray-traced, and in blue are the points for which we used linear regression. In Figure 20b we zoom in on a particular region so that this phenomena is more visible.

Next we consider a smooth reflector cup, similar to the one of Section 3. For this reflector cup ray-tracing can be rather time consuming, as the typical number of reflections is relatively high. Furthermore, one must use numerical methods in the ray-tracing routine. To illustrate our method we ran it using

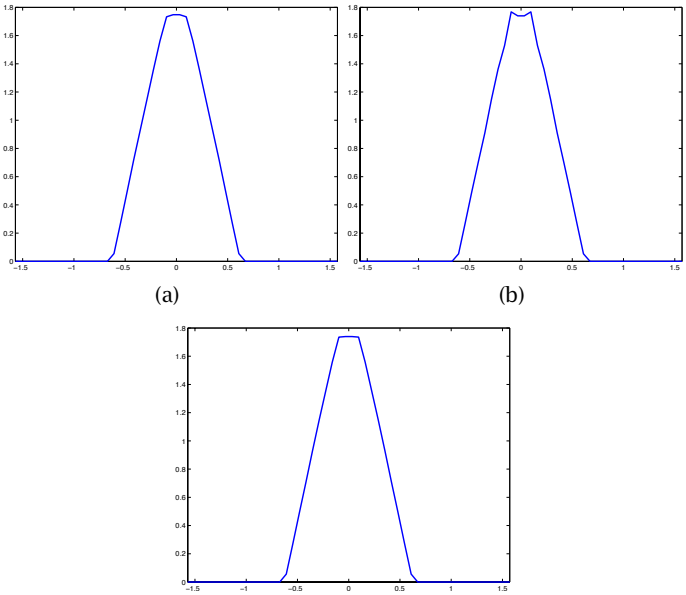
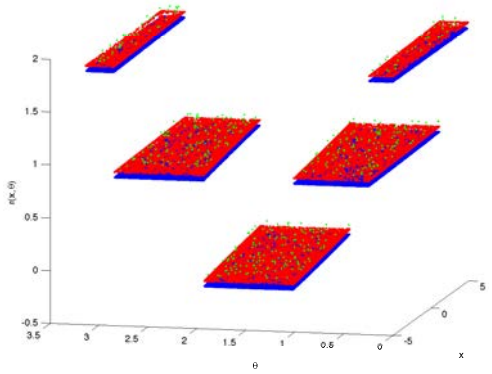
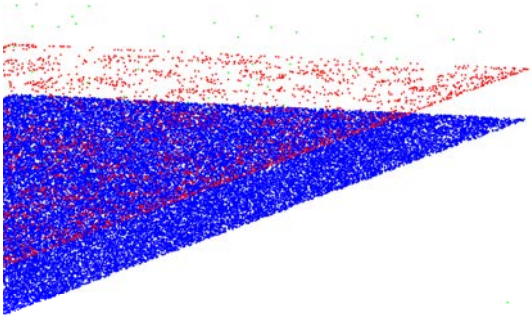


Figure 19: Intensity profile for the faceted reflector cup of Figure 2. In panel a we see the result of our method (bootstrapped MC). In panel b we see the result of a naïve MC method using 31906 rays, and in panel c we display the result for a naïve MC that uses 2^{19} rays.



(a)



(b)

Figure 20: Number of reflections for each sampled point in the phase space. In green are the points in the preliminary sample, in red the points that were later sampled using ray-tracing, and in blue all the points for which we used linear regression. Panel b zooms in a particular part of the plot.

preliminary sample of size $n_1 = 2^9$ and an estimation sample of size $n_2 = 2^{19}$. In the end the method called the ray-tracing subroutine only 38006 times. The resulting intensity plot is depicted in Figure 21a, and computing it took only a few minutes using a rather crude implementation in Matlab. This can be compared with the resulting intensity profile when using a naïve MC method with $n = 2^{18}$ samples which took several hours to compute on the same machine, and is depicted in Figure 21b. As one can see, the two profiles are nearly identical, demonstrating the efficiency of the proposed method. Furthermore, the differences between the two figures are within the accuracy of the Monte-Carlo integration methods when $n_2 = 2^{18}$ rays.

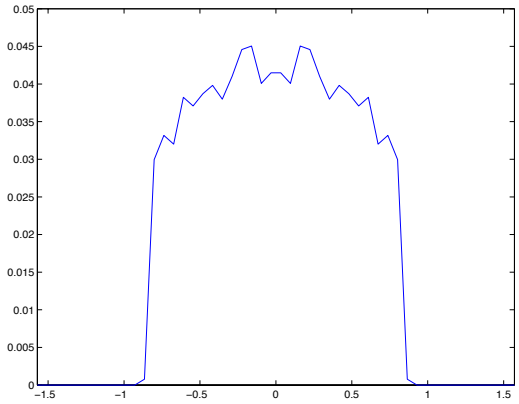
Finally, in Figure 22 we plot the number of reflections as a function of a ray's position and initial angle. Unlike in the faceted cup, the input phase space partition has many more regions, some with a large number of reflections. Nevertheless calls to the ray-tracing routine were done mostly for rays in the boundary of each region.

5 Conclusions and Outlook

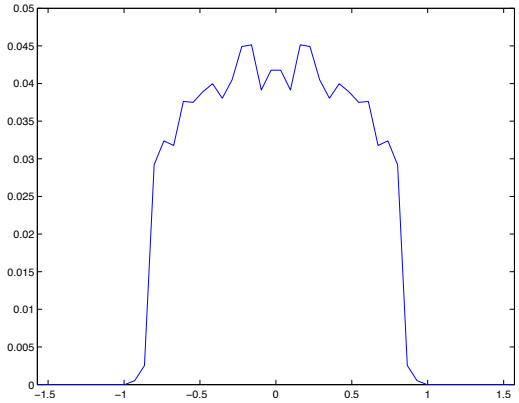
This report shows that, for faceted cups, the analytic approach already yields a lot of information about the target phase space (see Section 2). We started from the relatively simple two-faceted cup (Subsection 2.1) and the philosophy is now that a (symmetric) multi-faceted cup can be regarded as a stack of two-faceted cups where the luminous input in a cup is given by the output of the underlying one. The corresponding results are given in Subsection 2.2. A different class of faceted cups was treated in Subsection 2.3 in which we considered general (convex) polygonal devices.

Section 3 is in two ways different from the preceding ones. First of all, we switched from faceted cups to smooth ones. Secondly, the exact analytic approach was no longer applicable. We proposed a method to approximate the partition of the phase space (both at the source and at the target) numerically. This does not involve random sampling and is based on a completely deterministic method. Moreover, it makes computations quite inexpensive and gives an insight into the solution of the 3-dimensional problem.

We used the information obtained about the partition of the phase space to develop a smarter Monte-Carlo sampling method in Section 4. The first method proposed uses specific information about the subdivision of the source phase space. This is no problem when one indeed has this exact information. However, if only approximate information is available (e.g. it was obtained by the method of Section 3), the corresponding errors in the input might be transferred into the output. Therefore, a second method is introduced that relies on less *a priori* knowledge about the partition.



(a)



(b)

Figure 21: Intensity profile for the a smooth reflector cup. In panel a we see the result of Method 2 (bootstrapped MC), which made use of 38006 calls to the ray-tracing routine. In panel b we see the result of a naïve MC method using 2^{18} calls to the ray-tracing routine. As one can see, differences between the two panels are very minute.

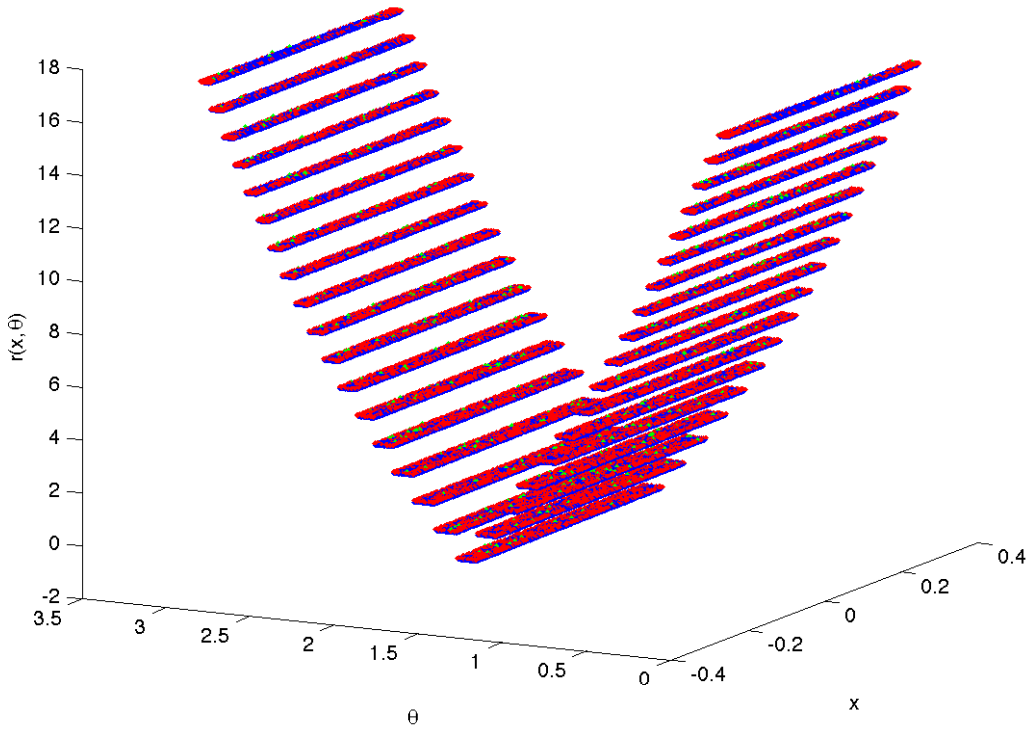


Figure 22: Number of reflections for each sampled point in the phase space, for the smooth cup. In green are the points in the preliminary sample, in red the points that were later sampled using ray-tracing, and in blue all the points for which we used linear regression.

To conclude this report we give some suggestions and directions for future work interesting from either a mathematical or a practical viewpoint. These originate from thoughts and questions that arose during and after the week but did not evolve into concrete results yet.

1. What is the area of the regions in phase space for the 2-facet cup?
2. For what cup shapes are the regions in $\text{Phase}_{\text{source}}$ and $\text{Phase}_{\text{target}}$ convex?
3. What are the θ for which the intensity graph of the far field is non-smooth?
4. Can we express the intensity function on the far field piecewise by elementary (trigonometric) functions when the cup is multifaceted?
5. How to model light rays that graze the surface and follow it, instead of reflecting? Wilbert IJzerman mentioned during the week that these rays arise and can form caustics.

Acknowledgments

We would like to thank Teus Tukker, Ferry Zijp, and Wilbert IJzerman (Philips Lighting) for posing their problem, and for their help during the week. Furthermore, we would like to thank the organizers of SWI 2012 for the opportunity to work together for a week and letting us use the facilities of Eindhoven University of Technology. Special thanks go to Mark Peletier, not only for thinking along about our problem, but also for inviting us all over to his house for a wonderful evening full of tasty desserts.

References

During the Study Group, we did of course not have occasion to study existing literature on the topic in detail. Points of entry suggested by the Philips team are the following.

[1] D. Rausch and A. M. Herkommer. Phase space approach to the use of integrator rods and optical arrays in illumination systems. *Adv. Opt. Techn.*, 1, 2012.

[2] H. Ries and A. Rabl. Edge-ray principle of nonimaging optics. *JOSA A*, 11:pp. 2627–2632, 1994.

[3] R. Winston, J. C. Miñano, and P. Benítez. *Nonimaging Optics*. Academic Press, 2005.

Up and Beyond - Building a Mountain in the Netherlands

Paulo J. De Andrade Serra (Eindhoven University of Technology), Tasnim Fatima (Eindhoven University of Technology), Andrea Fernandez (University of Bath), Tim Hulshof (Eindhoven University of Technology), Tagi Khaniyev (Middle East Technical University), Patrick J.P. van Meurs (Eindhoven University of Technology), Jan-Jaap Oosterwijk (Eindhoven University of Technology), Stefanie Postma (Leiden University), Vivi Rottschäfer (Leiden University), Lotte Sewalt (Leiden University), Frits Veerman (Leiden University)

Abstract

We discuss the idea of building a 2 km high mountain in the Netherlands. In this paper, we give suggestions on three important areas for the completion of this project. Issues like location, structure and sustainability are investigated and discussed in detail.

KEYWORDS: building a mountain, high structure, the Netherlands

1 Introduction

The Netherlands does not have any tall mountains. Indeed, its name even derives from the fact that it is essentially flat. According to Thijs Zonneveld, a journalist and former professional cyclist, this is a serious shortcoming of his country. As a possible remedy, he proposed *building* a 2 kilometer high mountain in the Netherlands. The response was immense. Immediately, there was a lot of excitement at the prospect of building a mountain, but also a fair amount of skepticism about whether it can actually be done (see [11]). In this report we aim to address some of the obstacles and opportunities that may arise in the construction of such a mountain.

The idea of building a massive structure is not new. In the past, numerous plans have been proposed for extremely tall buildings and structures. However, what all these plans have in common is that they never left the drawing board. The Dutch, however, are renowned for their large-scale engineering works such as the dikes, polders, and the Delta Works. Still, it is not hard to see that building a mountain would dwarf these accomplishments by comparison. Consider

We thank M.P. Chaudhary, M.A. Peletier and M.A.A. Boon for their contribution in the problem analysis and the presentation of the subsequent results during SWI 2012.

that currently, at a height of 828 meters, the Burj Khalifa in Dubai is the tallest building in the world – truly a marvel of modern engineering. Imagine then the extremely special care and consideration, the vast amount of work and the incredible ingenuity that is required to achieve a structure that is more than double that height.

After Thijs Zonneveld proposed building a mountain, a group of companies joined forces in the organization 'Die Berg komt er!' [9]. The aim of this organization is to bring Zonneveld's vision into reality and build a mountain. Right now, they concentrate on studying the feasibility of building a 2 kilometer high mountain in the Netherlands. One of the companies involved, Bartels Consulting Engineers, brought this problem to the Study Group Mathematics with Industry (SWI) held in Eindhoven from 30 January to 3 February 2012 to aid in this investigation.

The main questions Bartels Consulting Engineers posed at the SWI were:

1. Where should the mountain be built?
2. What shape, size and structure should the mountain have?
3. Which materials can be used to build a mountain?
4. How will the mountain impact the environment, soil levels and (local) weather?
5. Can the mountain be made sustainable?
6. How could one set up the necessary infrastructure?
7. How can the mountain be used? (Both during construction and after completion.)

Our aim during the SWI was to answer these questions as best we could. In Section 2, we discuss possible locations for the mountain and in Section 3, we discuss the impact a mountain would have on the ground it is built on. In Section 4 we make some general remarks about possible ways of constructing the mountain, in Section 5 we take a more in-depth look at possible materials that may be used, and in Section 6 we discuss possibilities for making the mountain sustainable. In Section 7 we conclude with a summary of our results, we present our conclusions and make suggestions for further research.

One final remark: we will assume throughout this report that the man-made mountain will have a height of 2 kilometers and a width at the base of roughly 14 kilometers.

2 Location

At the start of the Study Group, Bartels Consulting Engineers handed us the following selection of eight possible locations to build the mountain, which are

listed in the table below. The numbers in the table corresponding to the locations are positioned on the map of the Netherlands in Figure 1.

Number of location	Description	In sea/on land
1	near Bergen aan Zee	in sea
2	near The Hague	in sea
3	off the coast of Zeeland	in sea
4	off the coast of Texel	in sea
5	in the IJsselmeer, near the Afsluitdijk	in sea
6	in the IJsselmeer, close to Flevoland	in sea
7	in the Markermeer	in sea
8	in the province Flevoland	on land

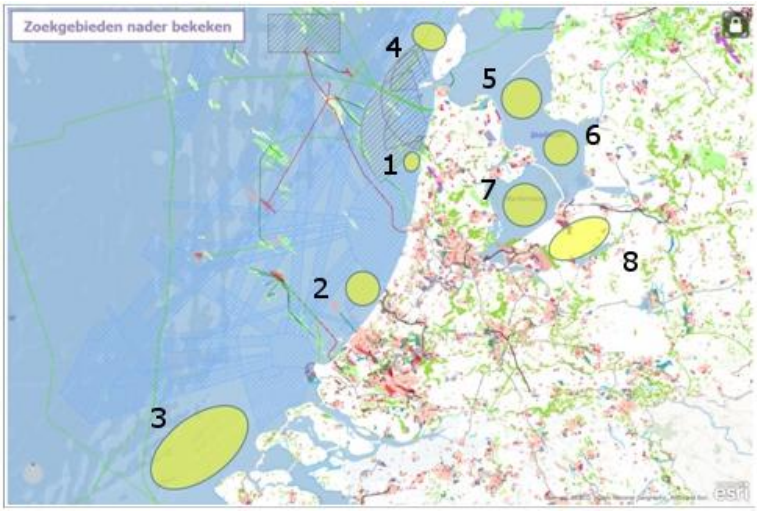


Figure 1: Eight possible locations to build the mountain. The numbers correspond to the numbers in the table of locations.

To select the most suitable location, we formulated several criteria:

- *Flight routes.* Schiphol is one of the largest airports in Europe. The air traffic should not be hindered in any way by the mountain. Hence, the mountain should not be placed on a flight route of Schiphol. These flight routes are depicted in Figure 2. Locations 2 and 7 are on busy flight routes, making them less preferable as a building location.

- *Shipping routes.* The seaport of Rotterdam is one of the largest in the world and it is crucial for the Dutch economy. Therefore, it is unwise to hinder

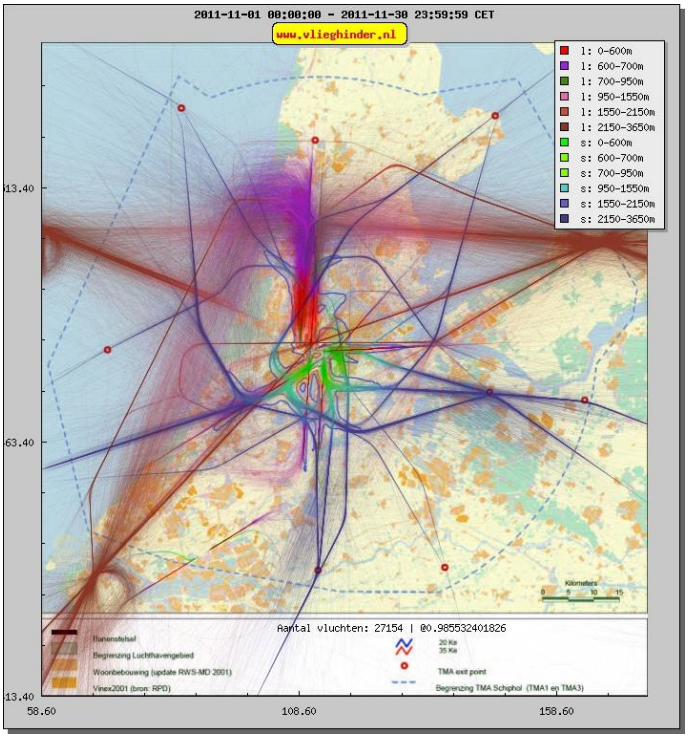


Figure 2: Routes of flights from and to Schiphol.

ship traffic to and from this seaport. Moreover, ships coming from the IJssel crossing the IJsselmeer should not be hindered either. This effectively rules out locations 3 and 6.

- *Sea currents.* The sea currents are quite strong near the North Sea coast. Building a mountain there would have a severe influence on the flow of these currents. This would cause changes in the location and shape of sand banks and would cause coastal erosion. Also, the currents will put stress on the building foundations. It would need to be verified through modeling and simulation, but it is likely that building a mountain at location 2 would have just such an amplifying effect on the local currents. This does, however, raise the question whether the power of these currents could be harnessed for energy production, for instance, by large turbines. Again, (computer) modeling would likely provide some insights.

- *Protected environment/nature.* Sustainable development and renewable energy will be a key issue in this project. Naturally, the construction and placement should have a minimal impact on the existing ecosystem, flora and fauna. Proximity of protected environments is therefore an important

limiting aspect. Location 4 is close to the island of Texel, which has a rich and rather unique ecosystem, and for this reason we recommend against building the mountain there.

- *Accessibility.* Construction resources (e.g. people, material, machines) need to efficiently reach the construction site. This is hard to assess for a given location as it depends on many different factors, and needs to be looked at in more detail in future studies.
- *Impact on society.* Existing societal structures (e.g. cities, infrastructure) need to experience as little interference as possible from the project. Location 5, in particular, does not meet this criterium, as it is near important infrastructure (i.e. 'de Afsluitdijk').

Under these criteria, only locations 1 and 8 do not raise any immediate objections: one is in sea and the other is on land, see Figure 3. As we will explain below in Section 3, we prefer location 1, which is in the sea near Bergen aan Zee because we estimate that the soil will respond rather extremely to the pressure that the mountain would exert, and this would have less severe consequences if this happened off-shore.

Figure 3: The two locations that we believe are most favorable. One location is in the North Sea, about 15 km offshore near Bergen aan Zee. The other location is on land, situated in the province Flevoland.

3 Soil mechanics

In this section we focus on the effect of the mountain on soil, by estimating how far the mountain will sink into the ground. We also estimate how much soil will be displaced, and whether this will cause hills or depressions to form nearby. Since these effects would likely need to be prevented, we propose some methods to do so.

3.1 Model

Let us first look at a model for soil mechanics that is popular in the field of high-rise buildings (see, for example, [19]). Figure 4 shows a schematic picture of the soil and the load exerted on it by the building. The lines in the ground illustrate the curves along which the soil would slide if the load of the building is too high.

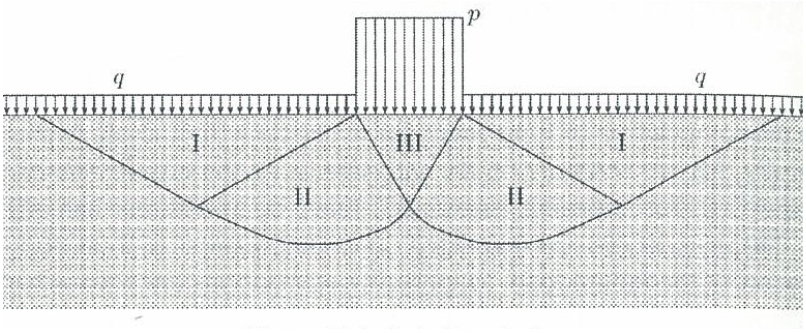


Figure 4: Popular model for the failing of soil caused by skyscrapers, [19].

The model states that if the load is higher than a certain value, the soil will slide. More specifically, zone III will move downwards, sliding along both zones II. As a result, zones II will slide sideways along the curved line at the bottom. This causes zones I to slide sideways and to rotate a bit. The result is that the ground will rise in the shape of a funnel.

These are exactly the effects that we want to avoid during and after the construction phase. However, it is not clear whether the above model can still be applied in our case. The expected area of the base of the mountain is several orders of magnitude greater than the area of the base of a large building. Furthermore, the load of the mountain on the soil (being orders of magnitude greater than the load caused by skyscrapers) causes zones II to be several kilometers deep. The soil is too heterogeneous at these depths (also containing rock type areas) for these failure curves to make sense. A more realistic model is thus needed to comprehensively understand the effects of the load of a mountain.

The shape of the mountain has not yet been decided upon, so we allow ourselves to make major simplifications in modeling the mountain and the soil. A schematic picture is given by Figure 5. We model the mountain as a solid cone. The assumptions on the size, weight and other relevant quantities are given in Table 1. For the expected weight of the mountain, we assume that it will be constructed mainly from concrete. The value of the resulting weight is estimated in Section 4. Note that these estimates only serve the purpose of supplying rough estimates on the weight and base area of the mountain.

Soil consists of layers, and our model needs to take this into account. We only use common knowledge about these layers, which can be found for example in

Quantity	Value	Description
A	$1.5 \cdot 10^8$ m^2	base area of the mountain
d	$1.4 \cdot 10^4$ m	diameter of the mountain
E	$2.0 \cdot 10^7$ N m^{-2}	static stress-strain modulus for the soil
g	9.8 m s^{-2}	Earth gravitational acceleration
h	$2 \cdot 10^3$ m	height of the mountain
ℓ	$5.0 \cdot 10^2$ m	thickness of the soft soil layer
m	$6.9 \cdot 10^{12}$ kg	mass of the mountain

Table 1: Quantities and their units.

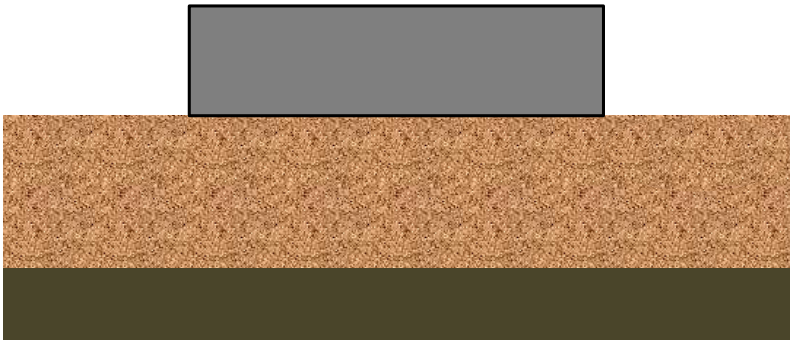


Figure 5: Schematic picture of our model of the mountain and the two soil layers. The first layer (light brown) is the soft layer. The bottom layer (dark brown) is modeled as a hard layer (not deformable), and consists of a second and third layer as mentioned below. The thickness of the layers as depicted here does not correspond to the actual ratio between them.

[10].

The first layer from the top, formed during the Quaternary period (this period started 2.6 million years ago and is still ongoing) is very thin in comparison to the other layers below, so we will neglect this layer in our model.

The second layer, formed during the Neogene period (between 23 and 2.6 million years ago) consists mainly of sublayers of either clay or sand. Its thickness depends heavily on the location. In Flevoland, it is about 500 meters thick (see [7]).

The third layer, formed during the Paleocene period (between 66 and 23 million years ago) also consists mainly of clay and sand, but because of the pressure induced by the second layer, it is more compressed. In Flevoland, this layer is also about 500 meters thick (see [7]).

The fourth layer, formed during the Cretaceous period (between 146 and 66 million years ago) consists mainly of limestone.

Based on these data, we propose a model for the soil in Flevoland with some

major simplifications. We propose a two-layer system, with a soft (mobile) layer to model the Neogene layer, and a rigid (immobile) layer to model the Paleocene and Cretaceous layer. Depending on the phenomena we want to study, we model the soft layer either by an incompressible viscous fluid or by an elastic medium with a linear stress-strain relation given by $\sigma = E\epsilon$ (see Table 1), where σ denotes the stress and ϵ the strain. The value of E that we take is based upon the values of this modulus for soft clay and loose sand (see [12], Table 2-7 on page 99).

Since the values of E for the Paleocene and the Cretaceous layer are at least an order of magnitude greater, we model these layers as being rigid. Hence our model of the soil consists of two layers; the soft layer on top, and a rigid layer below (see Figure 5).

3.2 Results

First we calculate how far the mountain will sink into the soil in our model. This distance is indicated by $\Delta\ell$. We will model the part of the soft soil layer under the mountain as an elastic medium for which we have the linear stress-strain relation

$$\sigma = E\epsilon. \tag{1}$$

The stress on the soil is given by the force that the mountain exerts on the soil divided by the area: $\sigma = F/A$ (for the sake of simplicity, we assume here that the force is equally distributed over the area). F is the gravitational force of the mountain, which is given by the mass of the mountain times the gravitational acceleration: $F = mg$. The strain is given by the ratio between the thickness of the soil and how far it is compressed: $\epsilon = \Delta\ell/\ell$. Substitution of these quantities into (1) yields

$$\frac{mg}{A} = E\frac{\Delta\ell}{\ell}.$$

Hence, we obtain

$$\Delta\ell = \frac{mg\ell}{EA} \approx 11 \text{ m}, \tag{2}$$

where we have used the values in Table 1.

To predict how the soil around the mountain will react, we model the soft soil layer as an incompressible fluid, while using the result given by (2). A schematic view of this situation is given by Figure 6. If the mountain sinks a distance of $\Delta\ell = 11$ m, then this means that the following volume of soil needs to be displaced around the mountain:

$$V = \Delta\ell A \approx 1.7 \cdot 10^9 \text{ m}^3.$$

If this volume of soil would be distributed over a ring-shaped area around the mountain up to 3 kilometers away from the mountain, this would mean that on average, in this area the ground would rise 11 meters upwards.

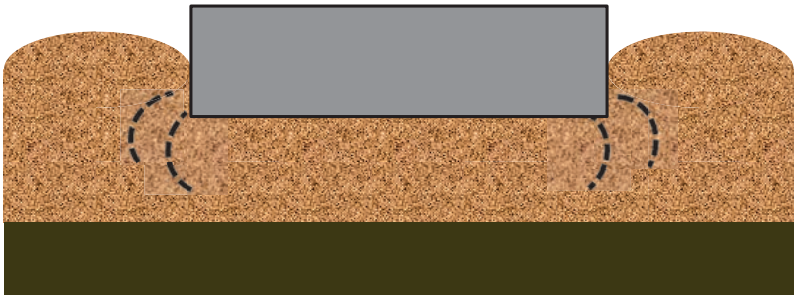


Figure 6: Schematic picture of the situation after the mountain has sunk into the soil. The shape of the surface of the soil in the surroundings is artificial.

3.3 Discussion

Since our model is a huge simplification of reality, our estimate that the mountain will sink 11 meters into the soil may be far from accurate. It is important to enhance the model to get a more accurate estimate. The estimates can be improved significantly by running computer simulations. This would allow one to take into account a number of aspects that our estimates ignore, such as the fact that the load is likely not uniformly distributed, and that the soil is in reality a complex and highly heterogeneous medium.

However, suppose that our estimate of 11 meters is of the right order of magnitude (or too small), then major problems can be expected while building the mountain. Even if one can come up with a solution that keeps the structure from collapsing while the mountain sinks into the soil, the problem remains that a huge amount of soil will be displaced to the surrounding area. If the mountain is going to be built on land, this could cause serious problems. If the mountain is going to be built in sea, it is still necessary to investigate whether this excess of soil can cause problems, but it seems less likely.

It may be better to prevent the mountain from sinking into the soil. A naive approach would be to use a foundation with long concrete pillars (approx. 100 m); see Figure 7. This has several advantages: first, the soil gets compressed by the driving force of the pillars, thus becoming more resistant to the load. Second, these concrete pillars will experience a shear force by the surrounding soil, which will carry a significant part of the weight of the mountain.

The downside of this approach is that one needs to cover at least 30% of the base area of the mountain by pillars to prevent the concrete from failing under the bulk pressure induced by the weight of the mountain.

A more innovative approach would make the mountain less dense than the soil that it rests on (see Figure 7). Since the soil behaves like a liquid on the length scale of the mountain, this idea is based on the same principle that makes a boat float on water. However, this is likely not possible with currently available materials and construction methods, and so would call for a major innovation.

Figure 7: Two different types of foundation: (concrete) pillars or a very light structure.

To conclude, our model indicates that one cannot neglect the depth that the mountain will sink into the soil. This will not only make it more difficult to construct the mountain, but will also raise the ground surrounding the mountain. Therefore, we advise to investigate how one can prevent the mountain from sinking into the soil. We propose to look either at a deep foundation with pillars, or to make the mountain much less dense than the soil it rests on.

4 Structure

It should be noted first off that there are no physical objections to building a mountain. However, the practicality of such an endeavor can be called into question.

Due to the broad nature of the project it is difficult to make precise statements about the sort of structure that this construction should conform to. We do know that it should have the appropriate shape and height, and it should be stable enough to serve as a platform for the construction of other buildings and facilitate activities that involve a large number of people. Based on this, some considerations can readily be made.

Comparing our man-made mountain with those in nature might seem like an obvious starting point; however, such comparison does not yield any useful insights. When it comes to natural mountains there is no intention to them, nor, for that matter, a design that one could copy effectively.

An obvious approach to building a mountain would be to simply pile on sand and rocks until a mountain is created. A number of artificial archipelagos have been created this way off the coast of Dubai. For instance, consider the 'The World' archipelago. This is a group of islands that have been shaped to resemble a map of the earth when viewed from above. It has a surface area of roughly 5.6 km^2, and an average elevation of 13 m when measured from the sea floor. Approximately 0.3 km^3 of sand was deposited over the course of 5 years by an approximate work force of 30,000 men and women. At the start of construction the cost was estimated at 14 billion dollars. A report by Fugro NPA Satellite

Mapping suggested that the islands were both eroding and sinking; Nakheel, the company in charge of development of 'The World', denied those claims (see [6]).

It would seem then that scaling this construction method to the size of a mountain, where approximately 100 km^3 of sand would have to be deposited, is not feasible for reasons of cost and stability.

Let us then consider the scenario in which the mountain is merely a structure that from the outside looks like a mountain, so that a comparison with a high-rise building is more appropriate.

In general, tall buildings are constructed because of their ability to provide large usable spaces while taking up a small area at the ground level. This is quite an attractive feature for buildings to have in large cities, where space for construction is scarce. It is important to understand however that the motivation for building exceedingly tall high-rise buildings, such as the current record holder, Burj Khalifa in Dubai, is not economical but mainly comes from the prestige that goes with owning (and demonstrating the ability to construct) such a tall structure. It is commonly believed that construction costs grow exponentially with the height of the building. This, put together with the extra structural precautions required to make the building withstand extreme winds and earthquakes, not to mention the vast stresses and strains generated by the weight of the building itself, calls into question whether it is sensible or cost-effective to build so tall a building. Their main commonality with a man-made mountain would therefore be that both these constructions are built mostly for their impressive height, rather than for their practicality. We thus have to ask ourselves whether the stated goals of having a structure that can both function as a mountain on the outside and provide practical and cost-effective spaces on the interior are at all compatible.

There are two major differences between an artificial mountain and a high-rise building. First, with an artificial mountain it is more important to have a functional exterior than a functional interior, while with high-rise buildings it is the other way around. Whereas the facade of ordinary buildings typically will not contribute significantly to the total weight, with an artificial mountain one would expect its facade – e.g. the soil and secondary buildings on top of it – to have considerable weight. The second difference is that the width-to-height aspect ratio in high-rise buildings is roughly 3:7 while for a mountain these values would most likely be inverted. Also, high-rise buildings in general have vertical facades, whereas we want the facade of our artificial mountain to be slanted, in order to use it.

The general rule with high-rise buildings, when it comes to structure, is that about 30% of the volume of the building is made up of structural elements such as walls and pillars. Although it is conceivable that a clever method of construction, such as a (geodesic) dome, could reduce this number in case of an artificial mountain, it seems unlikely that any dramatic improvements can be made. While the lower height-to-width aspect ratio would facilitate a more widely and

evenly distributed pressure at the base of the building, the central section of the base, where the mountain is tallest, would have to withstand large pressures and would therefore need to be of a higher density and be made of materials that could withstand extreme compressive forces. Furthermore, the surface of the structure would have to be heavily reinforced to facilitate secondary structures, such as other buildings, roads, or simply soil and flora, natural structures like rivers and lakes, and ice formations – factors which are usually not relevant in traditional buildings. In short, it may even be optimistic to assume that only 30% of the volume of the mountain would correspond to structural elements. This number is important since it can be used to estimate the minimal volume of materials needed in the construction based on an approximation of the volume of the mountain.

Finally, it had been suggested to us that the mountain could be constructed in stages. Here, for instance, one would start with a small hill and gradually expand it over time, so that during the construction the mountain is already functional in some way. The following question then arises: How does the expansion relate to the amount of materials added? That is, at which stage of the construction would one have a structure capable of satisfying at least some of the functional requirements? To tackle this, we can express the height of the mountain as a function of the volume: let h be the height of the mountain, V the volume of the mountain and ρ_m the ratio of material to air inside the mountain. Assume that the mountain has a conical shape with a fixed slope s, then

$$h = \left(\frac{3}{\pi} s^2 \rho_m V\right)^{\frac{1}{3}}. \tag{3}$$

Hence, setting $\rho_m = 30\%$ and $s = 2/7$, starting with 1 km^3 of material would result in a hill of roughly 450 meters tall. Adding another 1 km^3 (spreading it uniformly over the surface) would increase the height to 571 meters. Repeating this process, the gain in height per added unit of volume decreases steeply. Indeed, increasing the volume of the mountain from 99 km^3 to 100 km^3 would result in a height gain of only 8 meters (cf. Figure 8). Looking at it from another perspective, one could say that if one wanted to double the height, one would have to use 8 times the material already used. Also note that the slow-down occurs at the end of the construction in this scenario. Different ways of adding the material (e.g. building the mountain in layers) could move this phenomenon to a different stage in the construction. The fact remains, however, that this slow-down has to occur at some stage of the construction. This intrinsic slow-down is thus another serious issue to contend with, if the mountain is to be a (financial) success.

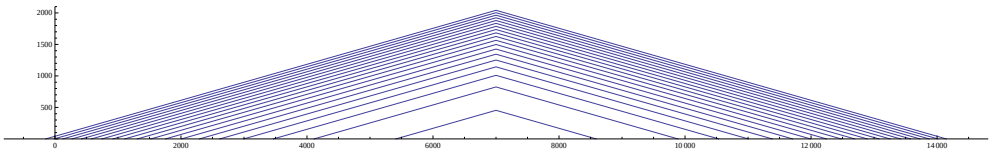


Figure 8: Schematic of the cross section of a conical mountain for 5 km^3 increments with a first layer of 1 km^3.

5 Materials

One of the largest limiting factors when building a mountain would be the amount of materials needed, the production costs of this material, its availability and the environmental impact of mining and/or manufacturing materials in such large quantities.

In this section we again approximate the total volume of the mountain by assuming its shape resembles a cone. For the sake of comparison we consider a cone with a diameter at the base of 14 km and a height of 2 km, resulting in a volume of approximately 100 km^3. Under the optimistic assumption that only 30% of this volume would be comprised of actual material we reach a total volume of 30 km^3 of material.

In Table 2 we consider for a few common construction materials with some rough estimates for the total amount of material needed, how many yearly world productions that represents, the total prices, and how many times the total yearly emissions of carbon dioxide of the Netherlands (= 176 mega tonnes of CO_2 per year [1]) the production of these quantities of materials would represent.

In all rows the values refer to 30 km^3 worth of material.

Materials	Required (10^{13} kg)	World Prod. (Years)	Total Price (10^{12} Euro)	NL CO_2 Emissions (Years)
Rock	7.50	–	1.95	–
Sand	7.95	–	2.10	–
Concrete	6.90	3.8	3.90	357
Plastic	2.91	29.1	29.10	1005
Steel	24.00	30.0	135.00	641
Glass	7.80	–	270.00	422

Table 2: Estimates on the required material, see [2], [5], [8], [4].

The table is quite clear: a man-made mountain cannot be built using traditional construction materials. The CO_2 emissions in particular stand out. An obvious requirement of any building materials used would therefore be that, besides

availability, they could be produced in a more-or-less CO_2 neutral way. This may very well be the biggest hurdle when building a mountain. Note that even if our estimated material density of 30% is off by, say, three orders of magnitude, that is, one could build it with a material density of 0.03% instead, then this would still result in the equivalent of 350 years worth of Dutch CO_2 production, in the most favorable case.

To these estimates one would have to add costs, both financial and environmental, for transporting these materials. Not to mention salaries for the workforce which would likely have to consist of tens of thousands of people, the material required, the costs of purchasing the land where the mountain would be built and many other costs which would add significantly to the already exorbitant numbers seen above.

Another obstacle is simply the amount of time required to finish the project; if we take the fastest material to produce from Table 2 it would take almost 40 years just to produce the material, provided that one could buy 10% of all the material made in the world during that period.

Taking all the above in consideration, it is clear that the current materials and techniques are simply not sufficient for this project. Innovative ideas are needed to see the mountain become reality. One step forward that comes to mind is to explore new construction techniques based on cellular or foam-like structures. Geodesic domes are a prime example of this. They use a minimal amount of material to cover a large area. The drawback is that they are expensive, difficult to build, and they cannot withstand the large pressures that for instance a classically constructed building can handle. As an illustration of how geodesic domes could be used to construct a mountain, consider the artist's impression of a multi-stage building consisting of dome-like elements, shown in Figure 9. Note that in this impression, the mountain 'grows' outward from a certain initial core structure.

6 Sustainability

In this section, we investigate how natural resources can be used to produce renewable energy at the site of the mountain, during and after construction, and how the mountain can be designed in such a way that the most is gained from the available resources. We will focus on means of generating energy that exploit the mountain's height, since there would be no point in generating energy on a mountain if it could be generated more efficiently elsewhere.

The aim in building a mountain would be to construct a so-called Zero-Energy Building. This is a popular term to describe a building with a zero net energy consumption and zero carbon emissions per year. Hence, the total amount of energy produced on-site should at least compensate the total amount of energy used in the building, but also, it should compensate for the energy spent during the construction phase. Considering that the estimates on the CO_2 emissions in

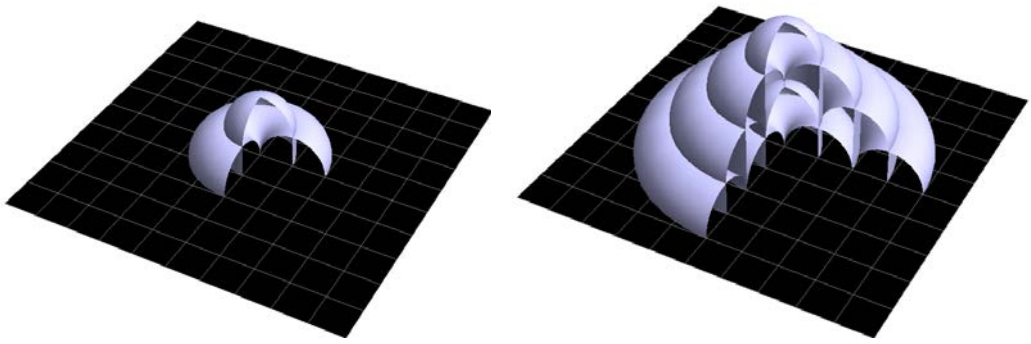


Figure 9: An artist's impression on building a mountain in stages.

the previous section were enormous, compensation for these emissions poses another serious problem.

We propose to use wind, sunlight and water as possible sources of energy.

6.1 Wind energy

Windmills have historically played a major role in the Netherlands by providing an alternative to water driven mills. More recently, wind power is being used as a renewable energy source. Today, around 2500 wind turbines are operational in the Netherlands, with a total wind energy production of 4000 GWh/year.

In the last decades, much research has been performed concerning wind energy production and the efficiency of wind turbines. This research varies from development of new types of wind turbines to determining the optimal spacing between turbines in a field of wind turbines, a so-called wind farm.

Commercial wind turbines, usually vertical, have a height varying from 80 to 150 meter producing around $2-5$ MW. Although many features of the turbine play a role in determining the capacity of the turbine, the major difference in capacity is due to the blade length.

The capacity of a wind turbine can be determined as follows (W: wind power)

$$W = \frac{1}{2} A_{\text{swept}} \rho_{\text{air}} V^3. \tag{4}$$

Here, A_{swept} is a number that is mainly determined by the blade length, ρ_{air} is the density of air and V is the wind speed. This formula shows that the amount of energy produced is very much influenced by the wind speed. Optimization

studies have shown that the ideal spacing between different turbines in a wind farm is about 12-20 times the blade length, since then the wind speed arriving at a turbine is not affected by the surrounding turbines. Thus, in order to achieve the most efficient wind energy production level, this has to be taken into account when placing wind turbines in a wind farm.

Since our aim is to build a 2 km high mountain, it is worthwhile to look at how high altitudes affect the wind speed and influence the energy production, when using wind turbines. As can be seen from equation (4), the power produced is proportional to the cube of the wind speed, i.e. $W \propto V^3$.

The effect of altitude on wind speed can be estimated as follows (see [15]):

$$V_x = V_y \left(\frac{h_x}{h_y} \right)^{\alpha}, \quad \alpha \equiv \frac{1}{7} \tag{5}$$

Here V_y is the wind speed at a given height h_y, take for example 10 meters, and V_x is the wind speed at altitude h_x. Formula (5) implies that at 2 km above sea level the wind speed is approximately twice the speed at sea level (10 meters). By using relation (4) we can see that this results in eight times more energy per turbine at 2 km altitude than at sea level.

As promising as it may seem, achieving this large gain may as of yet not be possible. Turbines constructed with today's technology are not built to cope with very high wind speeds. Nevertheless, recent developments raise hope for producing much more stable and strong wind turbines, see [13].

Another important aspect to consider is the stage-by-stage construction of the mountain. This provides the possibility to produce wind energy already after the construction of the first stage. Furthermore, since wind energy is as yet not often produced at high altitude, this way of construction will provide opportunity to develop and implement the cutting-edge technology in this field.

6.1.1 Tunnels in the mountain

One of the methods for wind energy production worth considering is wind tunneling. Making a long, relatively narrow tunnel through a building or mountain will cause air to be sucked in at a high velocity, similar to a chimney. This method is not widely used in traditional high-rise buildings due to stability problems, as having a high wind speed at the top introduces a large horizontal strain. However, stability in this sense is not an issue for the mountain, because of the height-to-width ratio is very small. Thus, tunneling appears to be a perfect way to use the construction to generate power from wind. Studies have shown that wind power can be increased by approximately 5-6 times with the tunneling effect, see [18]. Taking into account both the effects of altitude and wind tunneling, it seems that one could generate 30 to 40 times more power from a turbine in a wind tunnel at the top of the mountain than than one could generate from a traditional wind mill at sea level of the same size as the diameter of the tunnel.

6.2 Solar energy

A popular source of renewable energy is solar energy. Nowadays this is mostly collected using photovoltaic solar panels. Of course, an enormous area such as the surface of a mountain could easily serve as a subsoil for solar panels. Also, if the mountain would be built in layers, the panels could be installed on the first layer and reused in the next. This way there would be gain already in an early stage of construction. However, it should be noted that solar panels do not seem to be significantly more efficient at high altitude than at sea level. Still, they might be implemented in places that would otherwise go unused.

Heat storage could generate energy as well, and solar radiation could contribute to the heating. On sunny days, heat could be collected inside or outside the structure. We now discuss some possibilities to use heat in more detail.

6.2.1 Solar chimneys

A number of new techniques in renewable energy have been developed over the years, one of them is a so-called 'solar chimney', an installation which combines three simple techniques. It consists of three essential elements: a glass roof, a chimney and wind turbines. Basically, this construction works as follows. Solar radiation heats up air below the glass roof with open sides. Attached to this roof is a high chimney. Air at large altitudes is cooler, and the difference in temperature of the air below the glass roof and at the top of the chimney causes the hot air to rise, creating a draft within the chimney. This principle of air acceleration causes high wind speeds which can generate energy using wind turbines (see Figure 10).

To have an effective solar chimney, a large area at ground level should be covered by a transparent roof, so as to catch as much heat from the sun as possible. A single 1000 meter tall solar chimney can provide energy for 30,000 Dutch households, see [16] (see also [3], [14] for further literature on solar chimneys).

Using solar chimneys has several advantages compared to other energy sources. For instance, since it uses both direct and diffuse radiation it is more suited to the Dutch weather conditions, whereas traditional solar power plants can only use direct radiation, which means they only work on sunny days. Also, because the construction of a solar chimney is relatively simple and there are few moving parts, the structure is very reliable (and therefore it requires little maintenance). Moreover, the power plant needs no cooling water, which is commonly used in solar power plants today. The greatest advantage, however, is that all the necessary technologies are already widely available and relatively cheap.

In 1982, a prototype solar chimney was built in Manzanares, Spain. It is a 195 m high chimney with a diameter of 10 m, with a collector that is 244 m in diameter. It achieved a yield of 50 kW. Designs for chimneys with a yield of 100 MW exist. Until now these chimneys have not often been used. There are two reasons for this. First, the efficiency of a solar chimney is fairly low compared to

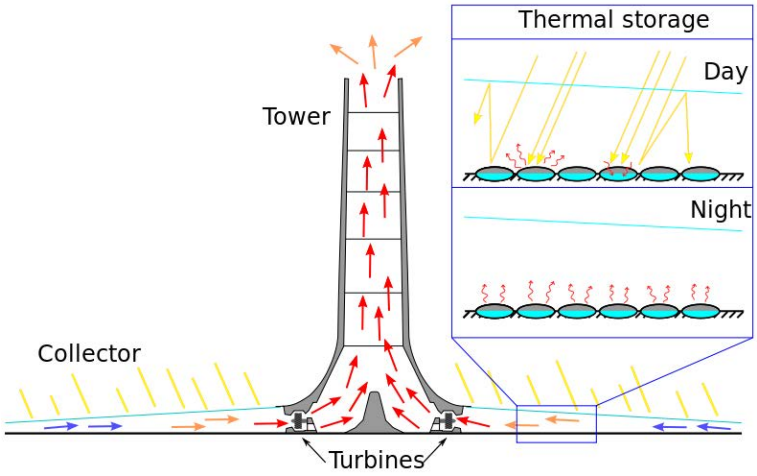


Figure 10: Principle of the solar chimney: a glass roof collector, chimney tube and wind turbines. The enlargement shows the use of water filled tubes.

that of traditional power plants. Second, their size often raises objections from the people who would have to live near them. Considering that we are trying to build a mountain that is much larger still, the latter issue seems to be minor by comparison.

6.2.2 The 'Tower of MegaPower'

A similar idea to a solar chimney that may be investigated further is that of the so-called 'Tower of MegaPower' (ToMP). The idea was proposed in [17] where a more detailed description and a first study can be found.

A ToMP is a tall tunnel-shaped device that is placed upright above a body of water and that generates energy through the following process: at ground-level, water evaporates and rises upward through the tunnel. Due to the colder air at high altitudes, the vapor will condensate and fall back down. By placing a turbine in the condensed water's path, energy can be generated. In other words, a ToMP imitates the rain cycle in a sustained way.

Like a solar chimney, a ToMP seems very well suited for integration into the mountain.

6.3 Hydroelectric energy

A simple way of generating energy that is easily integrated in the project is hydroelectricity. One can construct several lakes on the mountain that collect rain-

water. These lakes can be used to store and produce energy, but could also provide venues for recreation. The lakes can be constructed at different altitudes. This way, a surplus of generated energy can be stored by pumping up water from one lake to another, higher up the mountain. This energy can then later be reclaimed by using a hydroelectric power installation. This way, the system functions a lot like a battery. Apart from using the water to store and generate energy, this water can also be used in various ways to supply the needs of the mountain's facilities.

7 Summary, conclusions and recommendations

We have taken on the challenge posed by Bartels Consulting Engineers during the Study Group Mathematics with Industry in Eindhoven, the Netherlands, and investigated both the possibilities of and the difficulties in constructing a 2 kilometer tall artificial mountain in the Netherlands.

Where to build a mountain. The Netherlands is a relatively small but densely populated country. The mountain would have to be built in an uninhabited zone. This clearly creates a severe restriction. We considered eight possible locations suggested by the organization of 'Die Berg komt er!', and came to the conclusion that it is best to either build it on land in the province of Flevoland, or to build it in the North Sea near (the appropriately named) Bergen aan Zee. We chose these locations on the basis of several criteria imposed by the presence of settlements, industry, infrastructure and nature.

Another important factor to take into account when choosing a location is the effect such a massive structure would have on the underlying soil, and specifically on the areas around it.

Of the two locations that seem the most convenient, we believe that the mountain would best be placed in sea rather than on land, considering the effects the building of a mountain would have on the surrounding soil, as the soft Dutch soil would shift significantly under the load of a mountain. In fact, we estimated that without a proper foundation the mountain could sink as much as 11 meters into the ground, and that this displacement would result in raising the ground around the mountain on average by 11 meters as far as 3 kilometers away.

Furthermore, we conclude that to minimize the effects of the mountain on the surrounding soil, it would be best to come up with a method that would prevent the mountain from sinking at all. The traditional way would require many and long concrete pillars to be driven into the soil; alternatively, one could lower the average density of the mountain, for instance by building it on a cushion made of a very light material, so that the mountain floats on the soil, like a ship in the water.

We recommend that extensive measurements of the profile of the soil layers is performed, down to much greater depths than is standard practice. This data

can be used in a computer model to assess the ramifications on the surrounding soil (and possibly, sea currents) and the structural requirements in much greater detail than we were able to, in a couple of days.

How to build a mountain. Building a 2 kilometer high mountain would be an endeavor of unprecedented scale in human history. Comparison to existing man-made archipelagos tell us that a solid mountain made of sand and rocks would cost too much, erode too easily, and involve too much work to be completed in a feasible amount of time. This leads us to conclude that instead one would need to apply more refined construction techniques that allow for rapid progress and that require significantly less material than a solid mountain would.

In Section 4 we presented a table that states for a number of common building materials the total price, a comparison of the necessary amount with the current world production, and a comparison with the total annual emission of CO_2 in the Netherlands. These estimates give us a strong indication that not enough of these materials is available (in the world) to finish the project on a time scale of decades. Furthermore, even if availability was not a problem, the production of these materials would be far too damaging for the (global) environment to be justifiable. Thus, we have to conclude that with the materials and techniques that are currently common, a 2 kilometer high mountain cannot be built.

Hence, the problem of producing massive quantities of cheap building materials without producing large amounts of CO_2 or other pollutants must first be resolved before a mountain can be built. This is certainly the most important and difficult problem that comes with building a mountain, but also the one where the reward is the greatest.

How to make the mountain sustainable. The mountain can be designed in such a way that electrical energy can be generated in several ways. Some modern ideas like a 'Tower of MegaPower', wind tunnels through the mountain and solar chimneys seem good candidates, since the mountain is very susceptible to the integration of such devices. Indeed, the integration of devices that rely on the difference in temperature at the ground and at greater altitude, or on the higher wind speeds that come with greater altitude, seem to offer the greatest rewards.

Summary. With the current techniques, materials and knowledge it is not possible to build a 2 kilometer high mountain. It seems that vast leaps in thinking about structural design and material use have to be made: the mountain needs to be as light as possible, as cheap as possible, as 'clean' as possible, and it needs to be built in a relatively short time as well. This opens up grand new challenges in material development, design and logistics.

As for the sustainability of the project, there are certainly many ways in which the mountain can be used to generate clean energy.

The building of a mountain in the Netherlands turns out to be both challenging and inspiring; it certainly invites one to go up and beyond.

8 Acknowledgments

We would like to thank Bartels Consulting Engineers and in particular ir. Sigrid Mulders for participating in the Study Group Mathematics with Industry and proposing this wonderful challenge. During the week, we consulted experts in several fields. Thanks are due to prof.dr.ir. Bert Blocken, dr.ir. Hans Hoenderkamp, ir. Rijk Blok and ir. Juan Manuel Davila Delgado for their time and effort.

References

[1] http://statline.cbs.nl/statweb/publication/?vw=t&dm=slnl&pa=70946ned&d1=a&d2=0-1,15&d3=a&hd=121112-1143&hdr=t&stb=g1,g2.

[2] http://minerals.usgs.gov/minerals/pubs/commodity/cement/index.html.

[3] http://www.floatingsolarchimney.gr/.

[4] http://ec.europa.eu/enterprise/sectors/metals-minerals/non-metallic-mineral-products/glass/index_en.htm.

[5] http://en.wikipedia.org/wiki/plastic#production_of_plastics.

[6] http://www.telegraph.co.uk/news/worldnews/middleeast/dubai/8271643/the-world-is-sinking-dubai-islands-falling-into-the-sea.html.

[7] http://www.geologievannederland.nl/boorprofiel.

[8] http://en.wikipedia.org/wiki/list_of_countries_by_steel_production.

[9] http://www.diebergkomter.nl/.

[10] http://en.wikipedia.org/wiki/geologic_time_scale.

[11] http://www.nu.nl/column-vrijdag/2604966/berg-onzin.html.

[12] J. E. Bowles. *Foundation Analysis and Design*. S.l.: McGraw-Hill, 1977.

[13] G. G. H. D. GmbH. Market status of high altitude wind energy. *http://www.gl-garradhassan.com/en/highaltitudewind.php/*, 2011.

[14] C. D. Papageorgiou. Floating solar chimney technology for eu and mediterranean countries. *Unpublished*, 2009.

[15] E. Peterson and J. Hennessey. On the use of power laws for estimates of wind power potential. *Journal for Applied Meteorology*, 17:390–394, 1978.

[16] S. B. und Partner. The solar chimney. *Unpublished*, 2002.

[17] R. van Ginkel, F. Hoos, R. Krom, and P. van Summeren. Toren van 5 km in Noordzee voor opwekken energie. *De Ingenieur*, 20: http://www.lgwkater.nl/energie/index1.htm, 1995.

[18] T. Van Hooff, B. Blocken, L. Aanen, and B. Bronsema. A venturi-shaped roof for wind-induced natural ventilation of buildings: Wind tunnel and CFD evaluation of different design configurations. *Building and Environment*, 46:1797–1807, 2011.

[19] A. Verruijt and S. van Baars. *Soil Mechanics*. Delft : VSSD, 2007.

Identification of a Response Amplitude Operator for Ships

Giovanni A. Bonaschi (Eindhoven University of Technology), Olena Filatova (Eindhoven University of Technology), Carlo Mercuri (Eindhoven University of Technology), Adrian Muntean (Eindhoven University of Technology), Mark A. Peletier (Eindhoven University of Technology), Volha Shchetnikava (Eindhoven University of Technology), Eric Siero (Leiden University), Iason Zisis (Eindhoven University of Technology)

Abstract

At the European Study Group Mathematics with Industry 2012 in Eindhoven, the Maritime Research Institute Netherlands (MARIN) presented the problem of identifying the response amplitude operator (RAO) for a ship, given input information on the amplitudes of the sea waves and output information on the movement of the ship. We approach the problem from a threefold perspective: a direct least-squares approach, an approach based on truncated Fourier series, and an approach using low-dimensional measures of the RAO. We give a few recommendations for possible further investigations.

KEYWORDS: Parameter/structure identification, inverse problem, response amplitude operator, ship structure, fatigue estimation

1 Introduction

In the present paper we deal with a problem proposed by MARIN during the SWI 2012 workshop in Eindhoven. MARIN, the Maritime Research Institute Netherlands, is an independent service provider for the maritime industry. MARIN's customers include commercial ship builders, fleet owners, navies, naval architects, and offshore companies.

The problem we tackle here is the identification of the structure response amplitude operator (RAO) of a 230m long FPSO, given sets of input-output data, which will be explained in Section 2.

A floating production, storage and offloading (FPSO) unit is a floating vessel used by the offshore industry for the storage and processing of oil and gas, and it is typically moored at a fixed position at sea. The structure is exposed to a natural process of degradation related to the cyclic loading of the structure through time: *fatigue*. This is due to continuously incoming sea waves and wind.

This topic has been studied extensively in the literature according to different points of view; see for example [1, 3, 4] and the references mentioned therein.

The interest of MARIN in the identification of the RAO lies in its use to estimate the expected time until the formation of fatigue cracks. The methods that we discuss in this report might be used to improve the accuracy of numerically calculated RAOs, and lead to a better estimate of the fatigue lifetime. We keep this in mind when discussing the different possible working strategies.

2 The data

The data provided by MARIN are generated by two different detection devices.

- A buoy at some distance from the FPSO measures water surface height and angle, and converts these into a wave energy spectrum. For each 30-minute interval indexed by k this results in a discretely defined function $S_\zeta^{(k)}(\omega, \theta)$, which gives the energy contained in waves moving in direction θ with frequency ω.
- A number of strain gauges on the FPSO measures a local strain in the structure, and converts this into another energy spectrum. This results in a discretely defined function $S_R^{(k)}(\omega, d(k))$, measured at the same time k, which gives the energy contained in harmonic bending modes with frequency ω.
- The *draft* $d(k)$ is the vertical distance between the waterline and the bottom of the hull at the time of measurement k. This draft changes over time, since the structure accumulates oil and gas over time, and periodically offloads it to transport ships. According to MARIN, the draft has a significant effect on the behaviour of the structure, and this is why this draft is taken into account.

The measurement data is organized as follows.

(θ) The measurements of S_ζ are taken along discretized directions of 4 degrees each (91 in total; $\theta_1 = 0$ and $\theta_{91} = 360$ coincide).

(ω) The frequency range for ω is $0.025 - 0.580\,\text{Hz}$ for S_ζ and $0 - 0.995\,\text{Hz}$ for S_R. For S_ζ there are 64 different frequencies, 200 for S_R. Since the available data for S_R and S_ζ do not correspond to the same frequency, we convert the values of S_R to the 64-value discretization of S_ζ by interpolation. As a consequence we do not analyse values of S_R at frequencies greater than $0.58Hz$ and below $0.025Hz$.

(d) The draft of the vessel ranges from 9 to 15 (meters) with $\Delta d = 0.5$, so that there are 12 different drafts.

(k) Measurements are taken along a period of one year with different draft values as shown in Table 1.

Table 1: Number of measurements for different periods and drafts.

Draft	July 2007 April 2008	May 2008 Sept 2008	Total
9.0-9.5	0	91	91
9.5-10.0	0	45	45
10.0-10.5	221	207	428
10.5-11.0	710	555	1265
11.0-11.5	1482	422	1904
11.5-12.0	1408	464	1872
12.0-12.5	893	588	1481
12.5-13.0	1052	124	1176
13.0-13.5	902	426	1328
13.5-14.0	370	783	1153
14.0-14.5	109	202	311
14.5-15.0	0	92	92
Total	7147	3999	11146

3 The mathematical problem

We now describe the mathematical problem that we consider. The response of the FPSO is assumed to follow linear response theory, resulting in the (theoretical) equation (see [1, 2, 3])

$$\forall\, \omega, d: \quad S_R(\omega, d) = \int \Phi_R(\omega, \theta, d) S_\zeta(\omega, \theta)\, d\theta, \tag{1}$$

where, as we described above, S_R and S_ζ are respectively the total response of the structure and the profile of incoming waves at different angle θ, at a certain frequency ω. S_R, S_ζ and Φ_R are positive functions; Φ_R and S_R are assumed to depend also on the draft d. The unknown function Φ_R is, by definition, the response amplitude operator (RAO), and its identification is the aim of this work.

We first convert equation (1) into a discrete, experiment-dependent version:

$$\forall\, \omega, k: \quad S_R^{(k)}(\omega, d(k)) = \sum_\theta \Phi_R(\omega, \theta, d(k))\, S_\zeta^{(k)}(\omega, \theta). \tag{2}$$

We will also reduce to the case of a single draft, using only the 1176 data points corresponding to draft range 12.5–13.0. Therefore we can omit the explicit draft dependence in S_R and Φ_R, and then the equation becomes

$$\forall\, \omega, k: \quad S_R^{(k)}(\omega) = \sum_\theta \Phi_R(\omega, \theta)\, S_\zeta^{(k)}(\omega, \theta). \tag{3}$$

The central question of this paper is therefore:

Can we construct methods for the determination of Φ_R in (3), given data on S_R and S_ζ?

4 Inverse problems and least squares

This problem is a classical inverse problem: determining a physical law from experimental data (see e.g. [6]). For each ω we need to determine the 91 values of $\Phi_R(\omega, \cdot)$; since for each ω we have 1176 data points to do so, this is an *a priori* strongly overdetermined problem. The method of first choice in this situation is the least-squares solution.

Unconstrained least squares

The least-squares method can be interpreted as a method of fitting data. The best fit in the least-square sense is that instance of the model for which the sum of squared residuals has its lowest value, the residual being the difference between an observed value and the value given by the model.

Fix ω, and write $a_{k,j} := S_\zeta^{(k)}(\omega, \theta_j)$, $b_k := S_R^{(k)}$, and $x_j := \Phi_R(\omega, \theta_j)$. Writing A for the matrix of $a_{k,j}$, equation (3) becomes

$$Ax = b \qquad \Longleftrightarrow \qquad \forall k : \sum_j a_{k,j} x_j = b_k. \tag{4}$$

A least-squares solution of (4) is a vector x that minimizes the residual of (4), i.e.

$$x = \arg\min_x \|b - Ax\|_2^2, \tag{5}$$

where $\|\cdot\|_2$ is the standard Euclidean norm.
If A has maximal rank, then this x is given by

$$x = (A^T A)^{-1} A^T b.$$

The MATLAB backslash operator implements this solution concept.

Constrained least squares

A least-squares solution has no reason to be nonnegative, while the RAO Φ_R is necessarily nonnegative. The minimization problem (5) has a natural generalization

$$x = \arg\min_{x \geq 0} \|b - Ax\|_2^2,$$

in which $x \geq 0$ should be interpreted as component wise non-negativity. In MATLAB the routine `lsqnonneg` implements this constrained least-squares solution.

5 Organization of the report

During the Study Group three different approaches were investigated.

1. The first approach is to apply the constrained or unconstrained least-squares method directly. In an attempt to reduce the impact of noise, we first made a selection of the most relevant data. This approach is outlined in Section 6.
2. A second approach used a truncated Fourier series representation of Φ_R, and determined the RAO again by least-squares fitting (Section 7).
3. A final approach focused on low-dimensional properties of the RAO (see Section 8).

6 Ansatz-free solutions after data selection

In this approach the idea is to solve equation (3) for fixed frequency ω and then repeat for all 64 frequencies for which there are both respons and wave data available. For fixed ω the equation reads:

$$S_R = \Phi_R(\theta_1)S_\zeta(\theta_1) + \Phi_R(\theta_2)S_\zeta(\theta_2) + ... + \Phi_R(\theta_{90})S_\zeta(\theta_{90});$$

where $\Phi_R(\theta_1), \Phi_R(\theta_2), ..., \Phi_R(\theta_{90})$ are 90 unknowns.[1] Thus if one obtains 90 of these equations, then, generically, it should be possible to solve for the unknowns. From every simultaneous measurement of S_R and S_ζ it is possible to obtain such an equation.

6.1 Data selection

In real life some of the data are bad. For instance, when a ship passes the measuring buoy, this affects S_ζ, but does not change S_R. The relationship resulting from this measurement will be inherently false. To reduce the impact of erroneous data we make a selection, by taking at given frequency ω the data with the highest response S_R at that frequency. The idea is that, to obtain a good relation at a given frequency, the frequency should be represented in the measurement. This is guaranteed if the FPSO shows a response at this frequency.

Is, for fixed frequency, every angle represented in some of the chosen data? If $S_\zeta(\omega, \theta)$ is small in every measurement, then the response of the ship to components of waves coming from this angle is impossible to determine. As a consequence the RAO may have a peak at this angle, without any meaning. This corresponds to the RAO being (partly) underdetermined. This has not been checked during the Study Group.

[1] Since both 0 and 360 degrees are represented in the data, it has been decided to exclude the data for 360 degrees from the calculations in this approach.

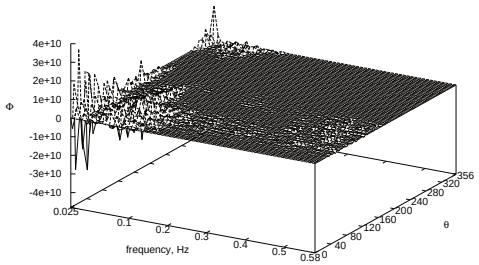


Figure 1: unconstrained RAO with negative components.

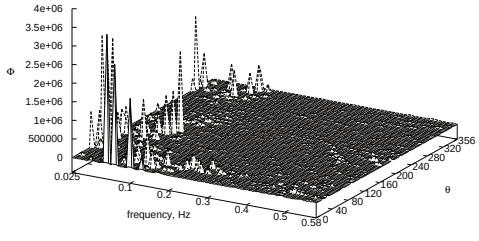


Figure 2: RAO calculated from data with 100 highest stress responses.

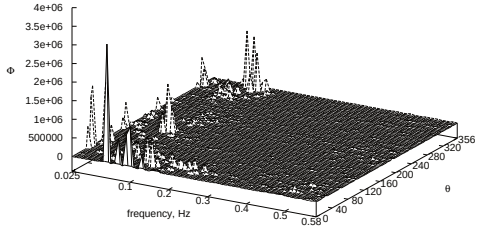


Figure 3: RAO calculated from data with 200 highest stress responses.

6.2 Least-squares solutions

Use of the unconstrained least squares solver (the MATLAB backslash operator) leads to a solution with negative components. This is illustrated by Figure 1, where the RAO computed from 100 data points with the highest stress response is plotted. Along the vertical axes the component of the RAO $\Phi_R(\omega, \theta)$ is drawn for each of the 90 angles θ and each of the 64 frequencies ω. Since the RAO should be non-negative, the unconstrained solver is not useful.

Thus we switched to the constrained least-squares `lsqnonneg` solver, since this solver finds a least squares solution under the constraint that every component must be non-negative. In figures 2 and 3, results are shown that are computed using respectively the 100 and 200 data with the highest stress response, for every frequency separately. As one can see, the solutions are very

spiky. Moreover, it has been observed that these spikes have the tendency to move to a neighbouring angle upon small changes in the input data.

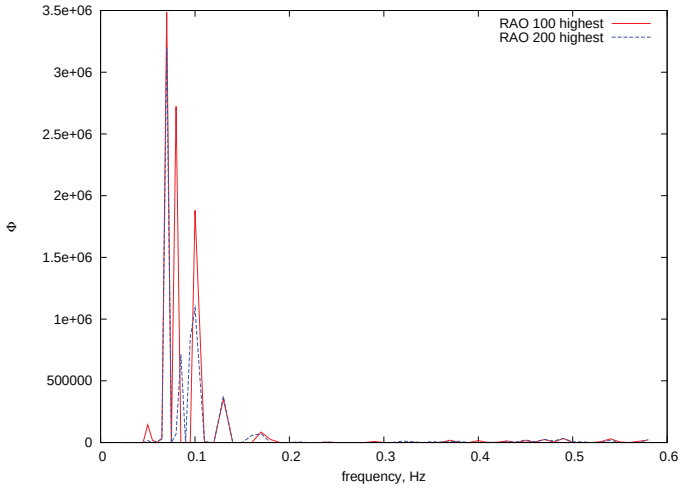


Figure 4: Graph of $\Phi_R(\omega)|_{\theta=0}$.

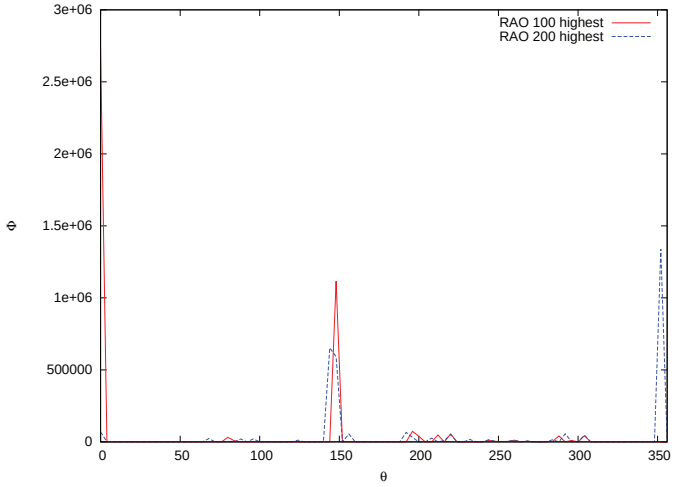


Figure 5: Graph of $\Phi_R(\theta)|_{\omega=0.8}$.

If we fix $\theta = 0$, then Figure 4 shows graphs of Φ_R as a function of ω, which corresponds to taking a slice from Figures 2 and 3. The peak of the solid red line at $\omega = 0.8$ is not present in the dashed blue line. If we fix $\omega = 0.8$, then Figure 5 shows graphs of Φ_R as a function of θ, which corresponds to taking a

slice in the other direction. From this graph we see that for the 'dashed blue' RAO based on 200 data, there is a peak for $\omega = 0.8$ near $\theta = 0$, at $\theta = 352$. This is illustrated further by the behaviour near $\theta = 150$. Although in Figure 5 the RAOs practically coincide near $\theta = 150$, this will not be reflected by taking slices for fixed $\theta = 144$ or $\theta = 148$.

6.3 Intermediate conclusion

The calculations shown in this section suggest that using the least-squares method one can calculate an approximate RAO, but the resulting RAO will be rather sensitive to differences in data point selection. Because the computed RAOs contain spikes instead of having a more smooth profile, it is not possible to reliably plot $\Phi_R(\omega)$ for a fixed θ.

7 Fourier expansions

7.1 Motivation

In the previous section we showed that, most likely, the straightforward least-squares approach leads to a sensitive dependence of the RAO on the choice of the data. This is a common occurrence when dealing with inverse problems, and is intimately related to the intrinsic ill-posedness of the problem (see again e.g. [6]). We now investigate whether this issue can be limited by restricting the set of RAO's to a smaller set.

We postulate a solution Φ_R of the form

$$\Phi_R(\omega, \theta) = K(\omega)\Psi(\theta). \tag{6}$$

Such an expression allows to consider dependency on ω and θ separately, and simplifies our calculations. For more precise approximation it is also worthwhile to replace (6) by

$$\Phi_R(\omega, \theta) = \sum_{\ell=0}^{n_\ell} K_\ell(\omega)\Psi_\ell(\theta). \tag{7}$$

Due to the anisotropy of the ship's geometry it was suggested by MARIN to choose $\Psi_\ell(\theta) = \cos(\ell\theta)$. Thus the final form of our approximation Ansatz is

$$\Phi_R(\omega, \theta) = \sum_{\ell=0}^{n_\ell} K_\ell(\omega)\cos(\ell\theta). \tag{8}$$

This Ansatz can be viewed as representing Φ_R by a truncated Fourier series in the terms of θ variable.

This approach again defines a linear least squares problem, which we refer to as the LLSP. As a result we expect to find $K_\ell(\omega)$ which determine the final

approximation of the solution. We will estimate residuals for different numbers of terms n_ℓ in (8). Moreover the relative error of the solution $\Phi_R(\omega,\theta)$ and $S_R(\omega)$ predicted by our model will be estimated in section 7.4 for different amounts of data used.

7.2 Implementation of the model

For each fixed ω we define

$$P := \begin{pmatrix} S_\zeta^{(1)}(\omega,\theta_1) & \dots & S_\zeta^{(N)}(\omega,\theta_1) \\ \dots & \dots & \dots \\ S_\zeta^{(1)}(\omega,\theta_{91}) & \dots & S_\zeta^{(N)}(\omega,\theta_{91}) \end{pmatrix}, \tag{9}$$

and

$$C := \begin{pmatrix} 1 & \cos\theta_1 & \dots & \cos n_\ell\theta_1 \\ 1 & \cos\theta_2 & \dots & \cos n_\ell\theta_2 \\ \vdots & & & \vdots \\ 1 & \cos\theta_{91} & \dots & \cos n_\ell\theta_{91} \end{pmatrix}. \tag{10}$$

The LLSP then consists of solving, in the least-squares manner, the equation $Ax = b$, with

$$A = P^T C, \qquad b = [S_R^{(1)}(\omega),\dots,S_R^{(N)}(\omega)]^T, \tag{11}$$
$$\text{and} \quad x = [K_1(\omega),\dots,K_{n_\ell}(\omega)]^T.$$

By repeating the procedure for each ω we obtain K_ℓ and thus $\Phi_R(\omega,\theta)$.

7.3 Analysis of the method

It is important to understand how well this model constructs Φ_R and predicts S_R. On the other hand, we wish to analyze to which extent the constructed Φ_R is data-dependent.

7.4 Approximation error

First we analyze the approximation error, which is the discrepancy between the exact values of S_R and their approximation by the LLSP.

We split the total available data for the draft 12.5–13.0 into disjoint groups of different sizes. We fix a number N of data points. Let D_κ, $\kappa = 1,2$ be two disjoint sets of data of size N. Also let $K^{[\kappa]}$ be the solution of the LLSP with data D_κ, i.e. the minimizer of the norm of the residuals for data set D_κ. In other words, we have

$$K^{[\kappa]}(\omega) := \arg\min_x ||S_R^{[\kappa]}(\omega) - Ax||_2,$$

where $S_R^{[\kappa]}(\omega)$ is the N-vector of response data corresponding to data set D_κ and A is defined in (11).

Then, for each frequency ω, we define the approximation error of LLSP-solution κ for the data D_λ as follows:

$$F(\omega, K^{[\kappa]}, D_\lambda) := \frac{\|S_R^{[\lambda]}(\omega) - AK^{[\kappa]}(\omega)\|_2}{\|S_R^{[\lambda]}(\omega)\|_2}, \qquad \kappa, \lambda = 1, 2, \tag{12}$$

where $S_R^{[\kappa]}(\omega)$ is the N-vector of response data corresponding to data set D_κ, and $K^{[\lambda]}$ is the solution of the LLSP for the data set D_λ.

First we study the influence of the number N of data points. We choose two sets D_1 and D_2 of size $N = 350$ corresponding to the data from February and August 2008. We compare $F(\omega, K^{[1]}, D_1)$ and $F(\omega, K^{[2]}, D_1)$ for the amount of terms in (8) $n_\ell = 3$. This can be interpreted as a measure of how well data D_2 predicts data D_1.

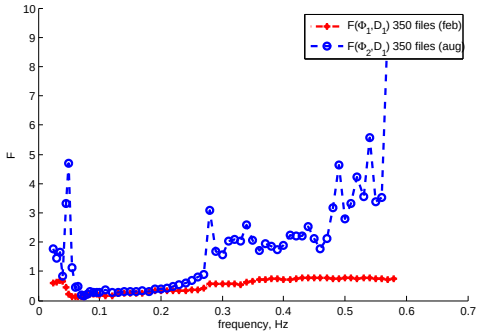


Figure 6: The approximation error for $F(\omega, K^{[1]}, D_1)$ and $F(\omega, K^{[2]}, D_1)$, $N = 350$.

The high value of the $F(\omega, K^{[2]}, D_1)$ in Figure 6 may well be explained by the fact that during different seasons the intensity of some frequencies differs. We next compare $F(\omega, K^{[2]}, D_2)$ and $F(\omega, K^{[1]}, D_2)$. In Figure 7 we see that the prediction of the August response by the February data is much better than vice versa. It can be useful to see how this fact changes with increasing the size of data sets used.

From Figures 8–11 it is clear that the bigger data sets we use, the closer to each other the approximation errors of the corresponding solutions become. But for several small frequencies the approximation error is still very high. We believe that this happens due to measurement errors of the experiments.

At this stage the conclusion is that it is best to use the biggest available amount of data for the further analysis of the approximation error on the number of terms in expansion (8). Now, in Figures 12–15, we vary the number of

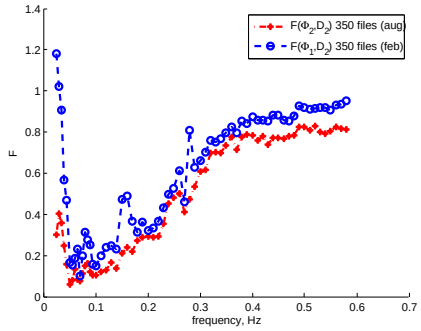


Figure 7: The approximation error for $F(\omega, K^{[2]}, D_2)$ and $F(\omega, K^{[1]}, D_2)$, $N = 350$.

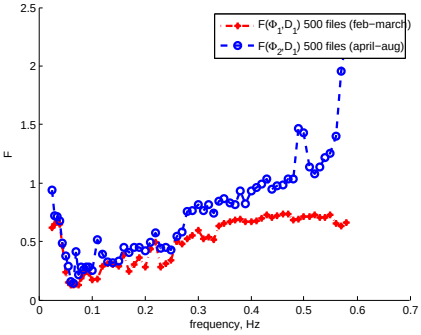


Figure 8: The approximation error for $F(\omega, K^{[1]}, D_1)$ and $F(\omega, K^{[2]}, D_1)$, $N = 500$.

terms n_ℓ and fix the size of the data sets $N = 715$, as this is the half of the available data for the chosen draft. The case $n_\ell = 0$ corresponds to the fact that Φ_R is approximated by a function that is constant in θ.

We study not only the approximation error but also two other values of interest ($\varepsilon(\omega)$ and $\varepsilon(\theta)$ defined in equations (13),(14)). The first of them is the relative error of $\Phi_R^{[\kappa]}$ with respect to $\Phi_R^{[\lambda]}$ for each frequency:

$$\varepsilon(\omega) = \frac{\|\Phi_R^{[\kappa]}(\omega, \cdot) - \Phi_R^{[\lambda]}(\omega, \cdot)\|_2}{\|\Phi_R^{[\kappa]}(\omega, \cdot)\|_2}, \tag{13}$$

where $\Phi_R^{[\kappa]}(\omega, \theta)$ and $\Phi_R^{[\lambda]}(\omega, \theta)$ are approximated values of Φ_R calculated via corresponding solutions $K^{[\kappa]}$ and $K^{[\lambda]}$ of LLSP using two distinct data sets D_κ and D_λ of the same size. The norms above are the L^2-norms over θ.

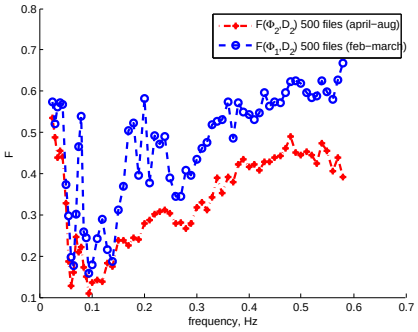


Figure 9: The approximation error for $F(\omega, K^{[1]}, D_2)$ and $F(\omega, K^{[2]}, D_2)$, $N = 500$.

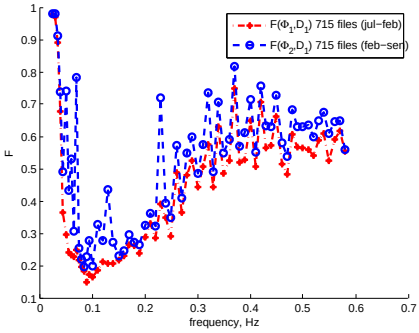


Figure 10: The approximation error for $F(\omega, K^{[1]}, D_1)$ and $F(\omega, K^{[2]}, D_1)$, $N = 715$.

A similar quantity can be calculated for each angle θ, where the norms are calculated by summing over ω:

$$\varepsilon(\theta) = \frac{\|\Phi_R^{[\kappa]}(\cdot,\theta) - \Phi_R^{[\lambda]}(\cdot,\theta)\|_2}{\|\Phi_R^{[\kappa]}(\cdot,\theta)\|_2}. \tag{14}$$

Increasing n_ℓ gives the system more freedom to adjust the parameters. From this point of view using more terms is a good idea. At the same time it leads to an increasing amount of oscillations, as can be seen in Figures 12-15. From the numerical experiments we suggest to use $n_\ell = 2$, because of two reasons:

- the peak of the approximation error for small frequencies is not high yet;
- the relative errors $\varepsilon(\omega)$ and $\varepsilon(\theta)$ are still reasonable (below 1).

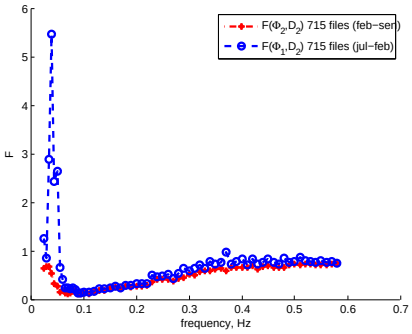


Figure 11: The approximation error for $F(\omega, K^{[1]}, D_2)$ and $F(\omega, K^{[2]}, D_2)$, $N = 715$.

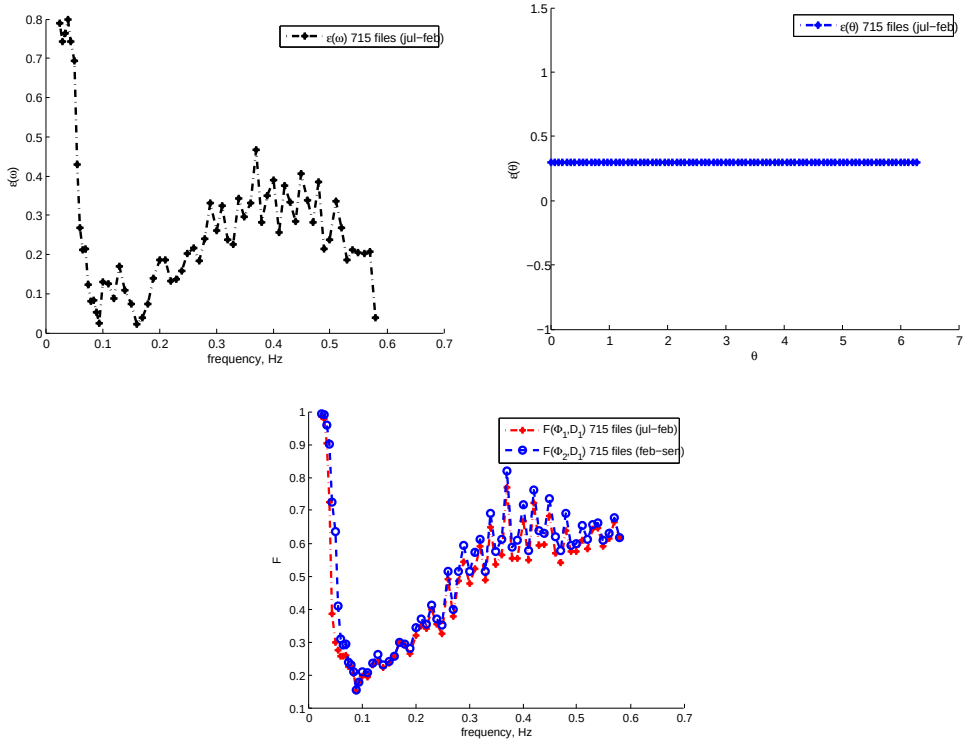


Figure 12: The approximation error for $F(\omega, K^{[1]}, D_1)$ and $F(\omega, K^{[2]}, D_1)$, $n_\ell = 0$. Note that $\varepsilon(\theta)$ is constant, since with $n_\ell = 0$, Φ_R is independent of θ.

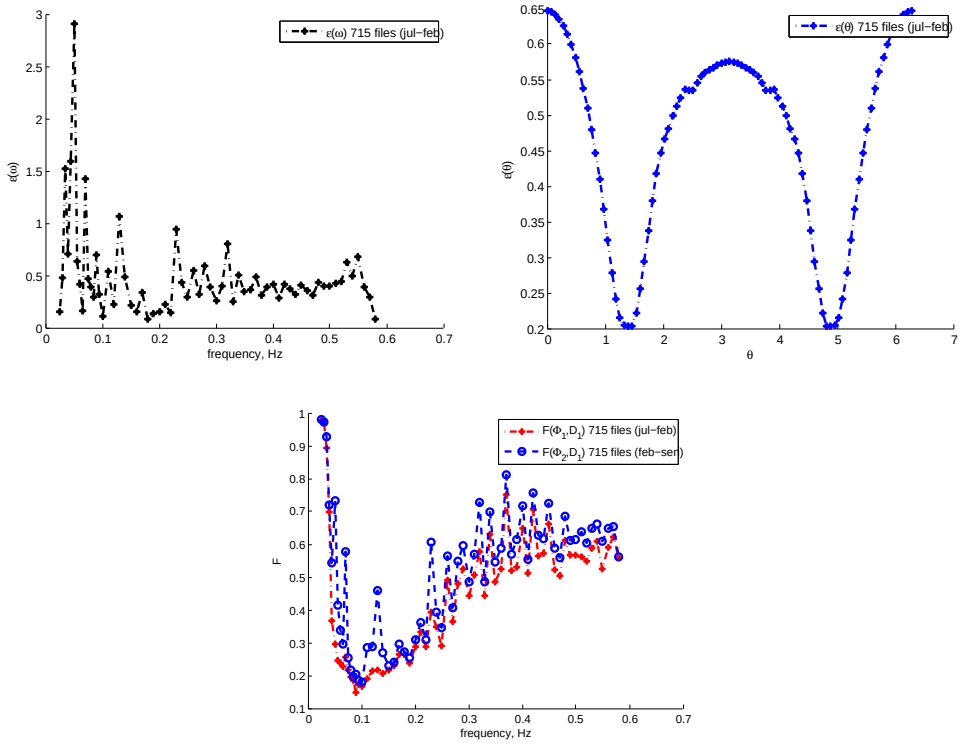


Figure 13: The approximation error for $F(\omega, K^{[1]}, D_1)$ and $F(\omega, K^{[2]}, D_1)$, $n_\ell = 2$.

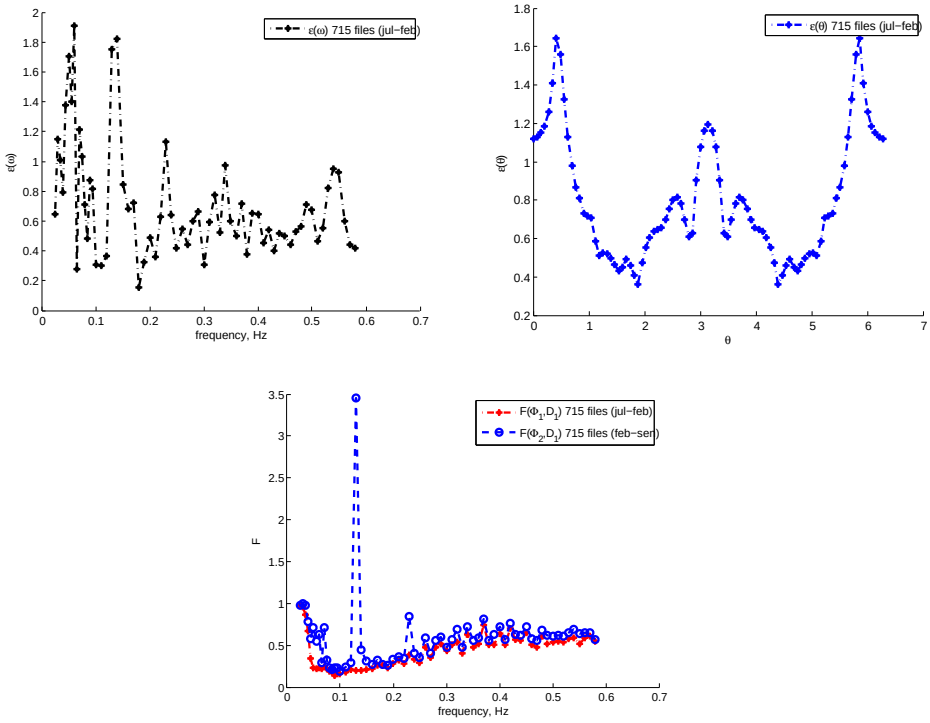


Figure 14: The approximation error for $F(\omega, K^{[1]}, D_1)$ and $F(\omega, K^{[2]}, D_1)$, $n_\ell = 6$.

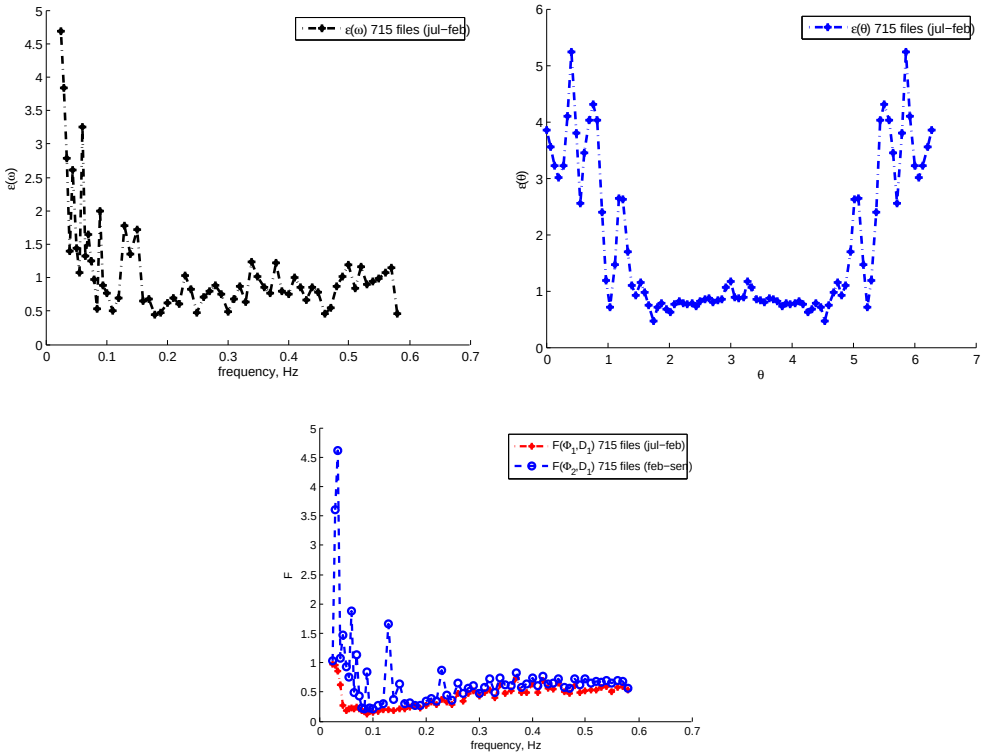


Figure 15: The approximation error for $F(\omega, K^{[1]}, D_1)$ and $F(\omega, K^{[2]}, D_1)$, $n_\ell = 10$.

7.5 Prediction of S_R

The comparison of predicted (dashed line) and experimental dependence (solid line) of Φ_R on ω is done for different data files. We again use the set of 715 files and predict the values of S_R for three specific data points which are not included in those 715 files. All calculations are done for $n_l = 2$. Also corresponding $\Phi_R(\omega)$ for several angles is presented in Figures 19-21.

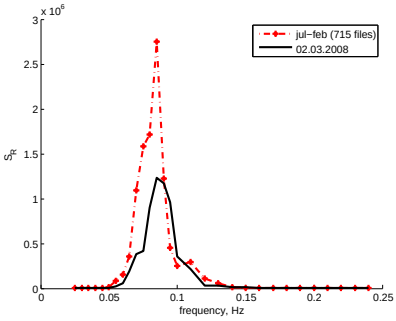


Figure 16: The predicted values have an overshoot.

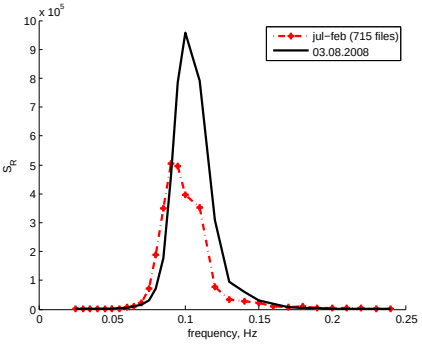


Figure 17: The predicted values are too small.

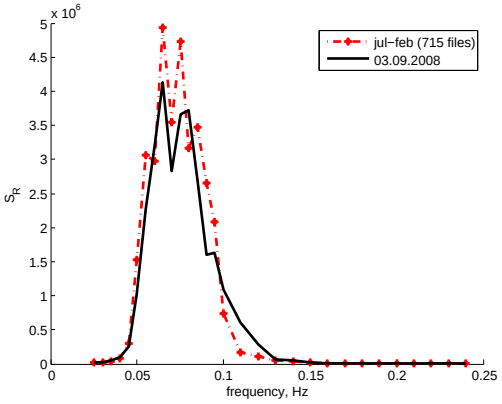


Figure 18: The predicted and experimental behavior fit well together.

Figures 16–18 show that, depending on the date which we pick for forecast, the result differs. One way of explaining this phenomenon is due to experimental errors. Results obtained for the 1st and 2nd of July 2008 on Figures 22 and

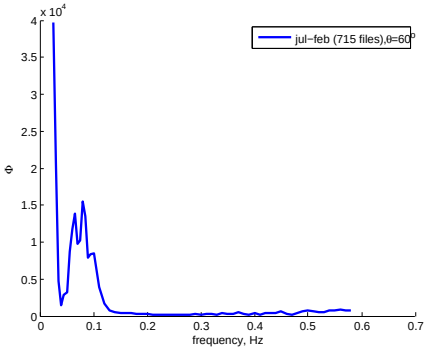


Figure 19: The prediction of $\Phi_R(\omega, 60)$.

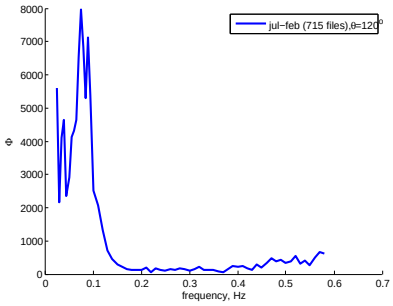


Figure 20: The prediction of $\Phi_R(\omega, 120)$.

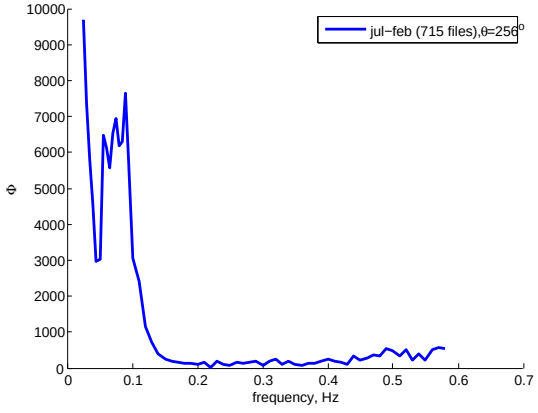


Figure 21: The prediction of $\Phi_R(\omega, 256)$.

23 are examples of this. Usually the response on consequent days changes continuously but for these dates it is not the case.

Another reason could be an insufficient period of measurements used in calculations. Therefore our recommendation is to use the observations of several years to predict Φ_R.

7.6 Intermediate conclusions

From these calculations we draw the following conclusions:

1. If the number of data points is large enough ($N \geq 500$ seems a reasonable

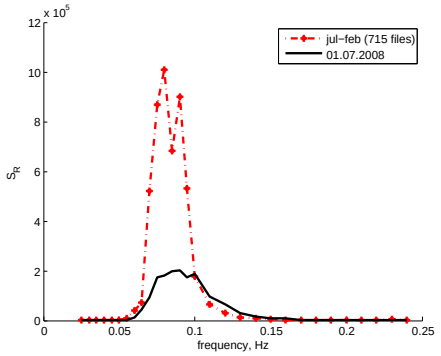


Figure 22: Results for the 1st of July.

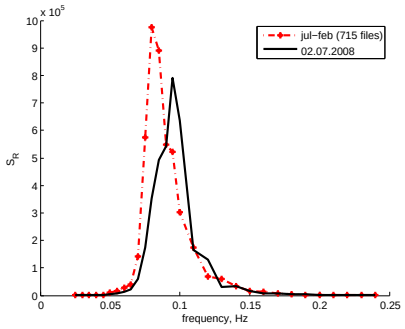


Figure 23: Results for the 2nd of July.

lower bound) then the cross-approximation error $F(\omega, K^{[2]}, D_1)$ often is practically as good as the self-approximation error $F(\omega, K^{[1]}, D_1)$. In words: the least-squares error for data 1, based on the parameters determined with data 2, is close to the error calculated with the optimal parameters for data 1. (See e.g. Figures 12–15).

2. The number n_ℓ of Fourier modes is a free parameter in this inverse problem. By definition, the self-approximation error decreases with increasing n_ℓ, since the minimization is performed over larger sets. But for larger n_ℓ, the Fourier coefficients K_ℓ becomes highly sensitive to the choice of data. The calculations done above suggest to keep n_ℓ low, e.g. $n_\ell \approx 2$.

8 Reduced measures

During the Study Group the question arose whether the identification of the RAO might be used to *detect* fatigue: can fatigue cracks determine a modified stress response, and therefore result in a modified RAO? By tracking changes in the RAO over time, we thought that these changes might be detected. This idea turned out to be incorrect. Indeed a fatigue crack cannot determine any significant change in the vertical bending moment, by which the global behavior of the structure is analyzed. This consideration has been communicated to us by MARIN.

In the spirit of the the previous section, we focus here on a particular Ansatz, investigating whether an RAO could be determined with sufficient accuracy and confidence. The ill-posed nature of the problem suggests to replace the aim of determining Φ_R by determining some low-dimensional properties of Φ_R that might (a) function as fatigue markers, and (b) be more stable.

Given that our data only spans 15 months, and that the fatigue time scale is expected to be longer than this, it was difficult to test any hypothesis concerning stability with respect to time. Instead, we investigate below a simple hypothesis concerning a special form of RAO with respect to the dependence on θ and ω.

Equation (3) can be written as

$$\forall\, \omega, k: \quad \sum_{\theta} \Phi_R(\omega,\theta) S^{(k)}(\omega,\theta) = 1, \tag{15}$$

where $S^{(k)}(\omega,\theta) := S_\zeta^{(k)}(\omega,\theta)/S_R^{(k)}(\omega)$. One Ansatz for Φ_R would be that Φ_R is of the form

$$\Phi_R(\omega,\theta) = c(\theta)(S^{(k)})^{-1}(\omega,\theta), \tag{16}$$

for some function $c(\theta)$ satisfying $\sum_\theta c(\theta) = 1$. Note that since Φ_R is k-independent, this requires $S^{(k)}$ also to be k-independent. This is a condition that we can test directly the available experimental data.

8.1 Data analysis

We analyse if the ratio $(S^{(k)})^{-1}$ is data- (k-) independent. This analysis is achieved by means of an estimator. We average the ratio among measurements, but they must belong to the same draft to avoid the draft dependence (that will be analysed in the following). So we define

$$f_d(\omega,\theta) := \frac{1}{\tilde{S}(\omega,\theta,d)} = \frac{\sum_{k\in I(d)} S_k^{-1}(\omega,\theta,d)}{\#I(d)}, \tag{17}$$

where $I(d)$ is the set of all the measurements obtained for a certain draft. From now on for brevity we write S_k^{-1} instead of $(S^{(k)})^{-1}$. If there is an independence; then we expect the standard deviation to be small. We analyse then the relative error:

$$g_d(\omega,\theta) = \sigma^2 = \frac{\sum_{k\in I}(S_k^{-1} - \tilde{S}^{-1})^2}{\#I}, \tag{18}$$

$$\text{relative error } := h_d(\omega,\theta) = \frac{\sqrt{g(\omega,\theta)}}{f(\omega,\theta)}. \tag{19}$$

A suitable way to analyse the relative error is to perform the average of the relative error over angles or frequencies:

$$a(\omega) := \frac{\sum_\theta h(\omega,\theta)}{\#\theta},\ b(\theta) := \frac{\sum_\omega h(\omega,\theta)}{\#\omega}. \tag{20}$$

The above formulas give us an estimator of the oscillations occurring in the data, depending only on one variable. We perform this procedure because we cannot represent all the values of $h(\omega,\theta)$ (it is a $90 * 64$ matrix). The problem that can appear when averaging is related to huge oscillations giving an irrelevant average. This will not be our case as it will be showed in the following.

8.2 Draft dependence

We need to make a reasonable choice of a single draft. To do this we analyze the total average of the relative error to check to which extent the independence (of the ratio with respect to data) is a reasonable assumption and to see which measurements present a strong correlation:

$$c(d) := \frac{\sum_\theta \sum_\omega h(\omega,\theta,d)}{\#\omega \times \#\theta} = \frac{\sum_\theta b(\theta,d)}{\#\theta} = \frac{\sum_\omega a(\omega,d)}{\#\omega}. \tag{21}$$

Table 2 shows the value of $c(d)$ for each d. Note that the values for the middle

$c(d)$	142%	12.1%	3.8%	2.2%	1.7%	2.7%
Draft	9.5	10	10.5	11	11.5	12
$c(d)$	2.2%	2%	2%	2.4%	7.1%	9.5%
Draft	12.5	13	13.5	14	14.5	15

Table 2: For each draft d the value of $c(d)$ estimates to which extent the function S_k^{-1} can be considered measurement- (k-) independent.

range of d are relatively small, giving support to the conjecture (16). In choosing a specific draft d for further analysis, it makes sense to avoid the extremal values for which $c(d)$ is larger.

8.3 Correlation

In Figure 8.3, we plot the functions $a(\omega)$, $b(\theta)$ defined in (20). They refer to the fixed draft 12–12.5 m, that presents a low total average ($c(d)$). We note that the values are relatively small and they show weak fluctuations. This gives more relevance to the choice of the estimator $c(d)$ and it is a way to quantify the independence of the ratio S_k^{-1} with respect to measurements.

8.4 Time evolution

The small values of $c(d)$ allow us to focus in the data analysis on a single draft. We now want to investigate whether the time scale of the data measurements could provide a reasonable RAO. A necessary condition is that the vessel does not experience excessive changes in its structure. In order to check this fact, we choose suitable estimators and we analyze their values on each month in a time range of 15 months. For fixed θ, we consider N_{exp} measurements in a certain time range. We use (a normalized) S^{-1} as a probability distribution, then, and we compute its ω-average for each θ:

$$\omega_\theta^{(k)} = \frac{\sum_\omega \omega S^{-1}(\omega,\theta)}{\sum_\omega S^{-1}(\omega,\theta)}, \qquad k = 1,\dots,N_{\text{exp}}. \tag{22}$$

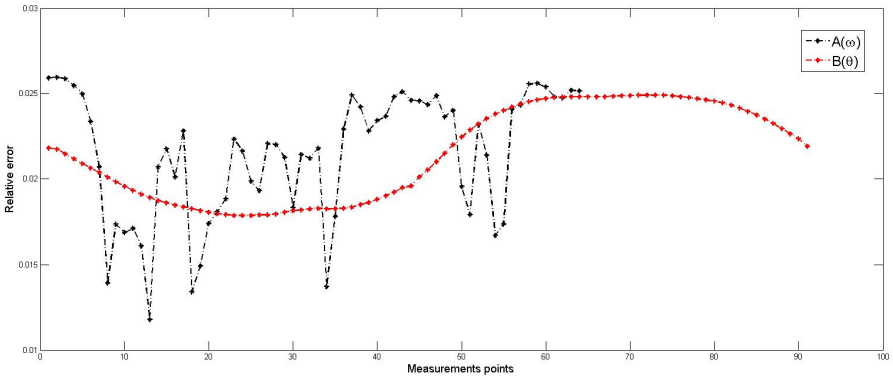


Figure 24: Relative standard deviations.

Now we average along measurements and determine the mean value and standard deviation at fixed angles:

$$\bar{\omega}_\theta := \frac{1}{N_{\text{exp}}} \sum_{k=1}^{N_{\text{exp}}} \omega_\theta^{(k)}, \tag{23}$$

$$\sigma_\theta^2 := \frac{1}{N_{\text{exp}}} \sum (\bar{\omega}_\theta - \omega_\theta^{(k)})^2, \tag{24}$$

where $\omega_\theta^{(k)}$ is given by (22). $\bar{\omega}_\theta$ can be interpreted as an average eigenfrequency of the structure.

In Table 3 are shown average and standard deviations related to five angles, in a time range of 15 months. We note that the values at 0 and 360 coincide,

θ	0	90	180	270	360
$\bar{\omega}_\theta$	0.1337	0.1359	0.1396	0.1704	0.1337
σ_θ	0.0255	0.0331	0.0426	0.0516	0.0264

Table 3: Angles, averages and standard deviations.

confirming the expected periodicity in θ of the data sets.

In Figure 25 we plot the average and standard deviations for each month, i.e. $\bar{\omega}_\theta$ and σ_θ^2 calculated for each month separately. We choose two angles, $\theta = 90$ and $\theta = 270$. The number of measurements per month is given by the following table:

1st	2nd	5th	8th	9th	10th	12th	13th	14th	15th
106	59	140	473	37	78	11	109	351	117

Table 4: Measurements analyzed per months, starting from July 2007, ending in September 2008.

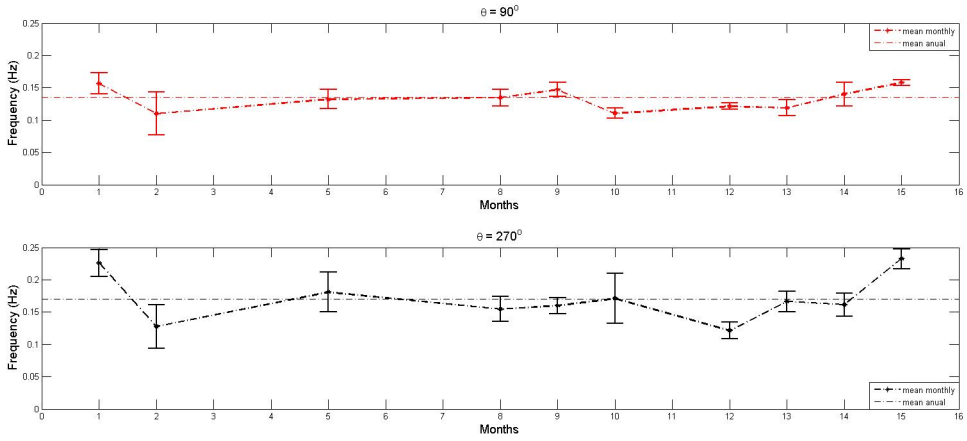


Figure 25: Monthly averages $\bar{\omega}_\theta$ and standard deviations σ_θ for two angles $\theta = 90, 270$.

8.5 Intermediate conclusions

Figure 25 shows no significant drift of ω_θ over the five months, and the fluctuations are of the magnitude that is to be expected. This observation can be interpreted as suggesting that the object that we calculate here (the expected frequency, according to the weighting given by the 'probability distribution' S^{-1}) is approximately constant over the 15 months of the data. This consideration together with Table 2 estimates to which extent (16) is a reasonable choice.

An interesting analysis would be to have a θ-dependent picture of the behavior of the structure measured by the fluctuations of the monthly averages and their relative standard deviation, which, during SWI 2012, we thought to be more considerable in those directions where the vessel is affected by more relevant damage and structural changes. This idea turned out to be wrong, after discussing with our collaborator from MARIN, as we already mentioned at the beginning of this section.

9 Summary and conclusions

We have seen that despite the large amount of data, determining the RAO to any accuracy is a hard problem. This is illustrated, for instance, in the strong data-dependence that we observed when doing direct (constrained) least-squares fitting in Section 6. This is a classical difficulty in inverse problems, and is related to the ill-posed nature of the problem.

The classical 'solution' to this difficulty is to restrict the class of admissible RAOs and perform the fitting in this smaller class. This is the core idea in Section 7 and Section 8 (see respectively equations (6) and (16)), where a special form for RAO has been postulated. This restriction brings in Section 7 the identification of an RAO.
A different form of RAO has been considered in Section 8, where initially we tried to address the question whether the data could provide information about existence of fatigue-induced drift. The performed data analysis is meant to verify to which extent (confidence) the Ansatz (16) is a reasonable guess, as a preliminary step in the identification of an RAO.

There are many possible avenues for further research and algorithm construction. Below we sum up those that we considered during the five-day SWI 2012 study group to be the most important.

- All methods should be set in a suitable stochastic framework in order to treat the unavoidable *noise* brought in by the measurements. We expect a faithful modelling of the characteristics of this noise will improve the quality of the fitting methods.

- Connected to the above consideration is the study of the *rank* of the available data files. Referring for simplicity to the Ansatz-free approach, it would worth analyzing how *independent* the data are, in order to yield the solvability of the linear system of equations. Intuitively, this is related to the non-vanishing determinants of the sub-matrices associated to the linear system.

Acknowledgments

We would like to thank Bas van der Linden, Sjoerd W. Rienstra from Eindhoven University of Technology for their help and useful remarks. We are grateful to Ingo Drummen from MARIN for his kind availability and Marin itself for driving our attention to a so interesting problem.

References

[1] J. Journée and W. Massie, *Offshore Hydromechanics*. Delft University of Technology, Course notes; http://www.shipmotions.nl/DUT/LectureNotes/OffshoreHydromechanics.pdf

[2] E. H. Cramer, R. Løseth and K. Olaisen Fatigue assessment of ship structures. *Marine Structures*,8:359–383, 1995.

[3] I. Drummen, M.K. Wu, M. L. Kaminski, and T. Moan, Numerical investigation into the application of response conditioned waves for long-term nonlinear fatigue analyses of rigid hulls. *Ocean Engineering*, 36:1208–1216, 2009.

[4] I. Drummen, G. Storhaug, and T. Moan, Experimental and numerical investigation of fatigue damage due to wave-induced vibrations in a containership in head seas. J. Mar. Sci. Tech., 13:428–445, 2008.

[5] B. Girod, R. Rabenstein, and A. Stenger, *Signals and Systems*. Wiley, NY, 2001.

[6] A. Kirsch, *An Introduction to the Mathematical Theory of Inverse Problems*, Springer Verlag, 2011.

[7] A. Powell, *An Introduction to Acoustic Fatigue*. in Acoustical Fatigue in Aerospace Structures, NY, 1965.

[8] A. Tarantola, *Inverse Problem Theory and Methods for Model Parameter Estimation*. SIAM, NY, 2005.

Acknowledgments

The generous financial support from NWO and STW together with the contributions by KWG, 3TU AMI and by the problem owners (Endinet, Philips Lighting, Thales, MARIN, Tata Steel, and Bartels Engineering) made the SWI 2012 possible. The success of the meeting is due to both the active involvement of the mathematicians and the transparence as well as close collaboration of the industrial partners. We thank you all.

The organizers of SWI 2012