Proceedings of the

Study Group Mathematics With Industry

held at the TU Delft

on January 27th - January 31st, 2014

ISBN: 978-94-6186-306-5

Foreword

How to bridge the gap between mathematics in academia and problem solving relevant to industry? In their pioneering answer, mathematicians at Oxford university organized the first Study Group Mathematics with Industry in 1971. Since then, their idea was copied numerous times worldwide. The hundredth edition of the European Study Group Mathematics with Industry will be held in Oxford in April 2014. The pioneering idea currently serves as a model for similar study groups in physics and computer science in the Netherlands.

As in recent previous editions of the Study Group Mathematics with Industry held in the Netherlands, six external entities brought a problem to work on to the 2014 edition. In the order in which they appear in these proceedings, these six entities are Statistics Netherlands, HZPC, Waterlaboratorium Noord BV, Witteveen-Bos, MARIN and INCAS3. The study group saw sixty participants from fifteen affiliations. Representatives from the external entities presented the problem on Monday morning. The study groups worked on a problem from Monday through Thursday and presented the results obtained on Friday morning. In the remainder of this foreword we briefly summarize the findings for the six individual problems.

Statistics Netherlands (www.cbs.nl) is responsible for collecting and processing data in order to publish statistics made available to policymakers and academics. It posed a problem on the counting of traffic on Dutch roads. The study group working on this problem employed graph techniques to measure this traffic independently of the amount of sensors and their locations.

HZPC (www.hzpc.com) is the largest seed potato supplier worldwide. It challenged the study group to quantify the market value of a batch of potatoes for the french fries production industry. To this end, the study group developed the finite fry method that allows to predict the amount of fries cut from a single potato as well as several parameters affecting the perceived quality of the french fry. The finite fry method can directly be incorporated in the HZPC production process.

Waterlaboratorium Noord BV (www.wln.nl) supplies drinking water to households in the north of the Netherlands. The need for better insight into the measurement techniques ensuring sufficient quality of the drinking water motivated WLN's participation to the SWI. The study group developed a hybrid stochastic-continuous model for the iron concentration in the different stages of the water purification plant. Numerical results show good qualitative agreement with measurements in the field. From this agreement the group was able to formulate recommendations on how to carry out future measurements.

Witteveen-Bos (www.witteveenbos.nl) is an engineering firm specialized in the design of infrastructure to contain water. Its concern for the flooding of the old city of Delft caused by heavy rain and its wish to improve existing numerical models for the water flow through the sewage system motivated their participation in the study week. In their contribution the study group develops a protocol for citizens to report the amount of rainfall allowing numerical models to be calibrated. Their work has resulted in valuable insights for Witteveen-Bos.

The Maritime Research Institute Netherlands (www.marin.nl) consults in various branches of the ship and off-shore industry. The institute proposed a problem on the estimation of parameters in system of ordinary differential equations modeling ship movement. Using polynomial interpolation on sparse grids the study group was able to find the optimal parameters for a large range of ship motion scenarios. Numerical results indicate that the results obtained carry over to more complex models of ship motion.

INCAS3 (www.incas3.nl) is an independent, non-profit and private consulting firm active in cognitive systems. The institute has been looking into the notoriously difficult problem of hearing aids for babies for a number of years. The study group formulated the problem as a inverse problem that can be solved via an adjoint formulation. This contribution offers INCAS3 new perspectives in developing effective hearing devices.

A promotional video on the 2014 edition of study group held in Delft will be made available on Youtube at the end of May 2014.

The organization of the 2014 study group gratefully acknowledges the generous financial support from NWO and STW. Additional funding was provided by the 3TU Applied Mathematics Institute and the Dutch Mathematical Society. The organizers sincerely thank Marco Puts and Erik van Bracht from Statistics Netherlands, Hans van Doorn, Jaqueline Verdijck, Rob Klooster and Pieter-Jelte Lindenbergh from HZPC, Peter van de Maas from WLN, Hans Korving and Rina Clemens from Witteveen-Bos, Ed van Daalen from MARIN and Peter van Herschel from INCAS3 from for their valuable input and contributions to making the week successful. The enthusiasm and efforts of all participants have been instrumental in making the study group successful. The organizers would like to especially acknowledge the six corresponding authors of the contributions to the proceedings. The week finally would not have been possible without the help of the support staff. Our warm gratitude in this respect goes to Deborah Dongor, Dorothee Engering and Evelyne Sharabi.

Johan Dubbeldam
Wolter Groenevelt
Arnold Heemink
Domenico Lahaye
Corine Meerman
Frank van der Meulen

Organizers of the 2014 edition of the Mathematics with Industry Study Group held in Delft

Table of Contents

Index of Authors

, , , , , , ,

Calculating Traffic based on Road Sensor Data

Rob Bisseling[1],
Fengnan Gao[2],
Patrick Hafkenscheid[3],
Reijer Idema[4], *
Tomasz Jetka[5],
Valia Guerra Ones[6],
Debanshu Ratha[6],
and Monika Sikora[7]

[1] Universiteit Utrecht
[2] Universiteit Leiden
[3] VU Amsterdam
[4] VORTECH
[5] IPPT, Polish Academy of Science
[6] TU Delft
[7] TU Wrocław, Poland

Abstract

Road sensors gather a lot of statistical data about traffic. In this paper, we discuss how a measure for the amount of traffic on the roads can be derived from this data, such that the measure is independent of the number and placement of sensors, and the calculations can be performed quickly for large amounts of data.

We discuss how a graph of the road sensors can be constructed, and how the number of cars and car-kilometers can be estimated on this graph. Further, methods for dealing with missing data are presented, and the benefits of principal component analysis are discussed.

Keywords: traffic index, road sensor, graph construction, statistical imputation, principal component analysis, CUR decomposition.

*Corresponding author, email: `reijer.idema@vortech.nl`

1 Introduction

This paper reports on findings regarding the *traffic index* problem, as posed by the CBS (Centraal Bureau voor de Statistiek [NL], Statistics Netherlands [EN]) at the Study Group Mathematics with Industry 2014, held at the Delft University of Technology.

The problem posed to us was to determine a traffic index comparing the average traffic on the highway system in the Netherlands (or a region such as South Limburg) from a particular year to a previous year, based on traffic measurements by road sensors (such as inductive loops, traffic cameras, etc.) taken every minute of the day, every day of the year. We call this problem TI. We decided to tackle an even more ambitious problem, which we call problem C, namely estimating how many vehicles (cars) there are on a particular road, at a given moment in time, based on the measurements of the past minute. Solving this instantaneous estimation problem by computing the number of cars C on the road, will also give a solution to the TI problem, by averaging over all the minutes of the year, and adding the results for all the roads of the network.

An advantage of tackling the more general problem C is that a solution can give more detailed information, such as changes in traffic patterns during the day, or differences between different days of the week, and it also enables zooming in on certain regions, roads, or even road segments. Another advantage of trying to achieve a precise estimate of an actual physically meaningful number is that such an approach is fault tolerant and adaptable to changes, e.g. new road sensors appearing, old sensors being removed, and some sensors malfunctioning temporarily.

The increase in our ambitions from computing mere statistical indicators to achieving a very precise estimate of the actual situation on the road is possible because of the wealth of detailed data that is now available. So to speak, Big Data is driving analysis from statistics towards detailed answers to specific questions on the systems and subsystems studied. For this approach to work, it is essential that computational algorithms become available that are efficient and preferably scale linearly in the number of data entries.

To formulate our problem, we define a number of variables, where we first consider a road segment between sensors A and B. Define d_{AB} as the distance between sensor A and sensor B. Let t be the current time and T the time interval of measurement (1 minute in our data). The measurements at time t are the *intensity* $I_A(t)$ of the traffic, i.e., the number of cars measured at sensor A in time interval $[t, t+T)$, and the average *velocity* $v_A(t)$ of the traffic measured at sensor A (for our data, the arithmetic average during the time interval). Furthermore, a road segment concerns one direction, see Figure 1.

In Section 2, we discuss methods to construct a graph of the road network

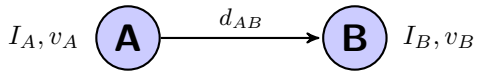


Figure 1: Road segment from sensor A to sensor B, with length d_{AB} and at each sensor the measured traffic intensity I and velocity v.

from the data, such that the edges are road segments as in Figure 1, and in Section 3 methods are presented to calculate traffic indices based on car count and on car-kilometer count, using the constructed graph of road sensors. In Section 4, ideas are presented that can help deal with missing data, and Section 5 discusses how Principal Component Analysis can be used to support traffic index calculations. Finally, in Section 6 conclusions are presented.

2 Graph construction

The idea to reconstruct the road network has been devised to utilize the maximum information provided by the data. Constructing a graph for the full road network would require more than just the road sensor data, e.g., OpenStreetMap data. Here, we only construct a separate directed graph for each road that has sensor data. This means that roads without sensors, including all connecting roads with exits from and entrances to roads with sensors, are ignored.

For the purpose of calculating a traffic index, we feel that this is the best way to deal with these 'dark roads'. First, there is no way to tell what exactly happened. How many cars exited and entered a measured road between sensors A and B? What dark road did they go on or come from, and did they drive that entire road or were they just visiting the closest house along that road? Second, assuming that the number of cars on dark roads is, in approximation, a constant fraction of the total number of cars, missing data of these roads should not significantly impact the relative traffic index.

We can roughly split the construction of the graph for a single road into two parts: a) determining most likely neighbours of sensors, and b) determining an order of the sensors that a car traveling on a road follows. We will need the following three ingredients:

1. coordinates of each sensor in latitude and longitude,
2. the name of the road each sensor is on,
3. the direction of the recorded traffic at each sensor.

The latitude and longitude coordinates of a sensor are used to locate the sensor on the map, and to compute the distance between two sensors. If the earth is assumed to be a perfect sphere, then the shortest distance between two points on the sphere is the smaller arc length of the great circle passing through these two points on the sphere. This is called the *geodesic* distance, and can be calculated from:

$$\text{haversin}\left(\frac{d_{AB}}{R}\right) = \text{haversin}(\phi_B - \phi_A) + \cos\phi_A \cos\phi_B \,\text{haversin}(\lambda_B - \lambda_A), \tag{1}$$

where d_{AB} is the geodesic distance between points $A(\phi_A, \lambda_A)$ and $B(\phi_B, \lambda_B)$ on a sphere of radius R, ϕ_A and ϕ_B are the latitudes, λ_A and λ_B the longitudes, for A and B, respectively, and

$$\text{haversin}(\theta) = \sin^2\frac{\theta}{2} = \frac{1-\cos\theta}{2}. \tag{2}$$

Note that MATLAB has a built-in geodesic distance function. The region of South Limburg has latitudes between 50.75° N and 51.05° N and longitudes between 5.7° E and 6.1° E.

Using the name of the road each sensor is on, the sensors can be assigned to their respective roads, and using the direction data the lanes in opposite directions can be separated. The main aim is to find the *successor* sensor for each sensor when travelling in a certain direction, as our car traffic calculation method requires that we know the order in which the sensors are traversed. For this we employ a modified nearest-neighbour algorithm, processing one direction of one road at a time.

When we have the order of the sensors for one direction of a road, we create a *pseudo-connection matrix* P of size $n \times n$, with a row and column for each of the n sensors (the name will be explained later on). For each sensor A in the data, we first find the sensor B that is closest to A. In the matrix P, there will then be a 1 in the (B, A)-entry.

Except for the first and last sensor on the road, each sensor has two neighbours. Rather than just looking for the sensor that is closest to A after B, we apply a second criterion to make the result more realistic, i.e., we only consider sensor C where the angle between AC and AB is at least 90 degrees. We illustrate this criterion with an example. In Figure 2 the closest neighbour to A is B, and the second closest neighbour is C. Still, because we assume that roads do not make turns sharper than 90 degrees, we consider D the more likely second neighbour of A, as in Figure 3.

We apply the rules for first and second neighbours to all of the sensors involved, and fill the matrix P accordingly. This matrix will only have 0s

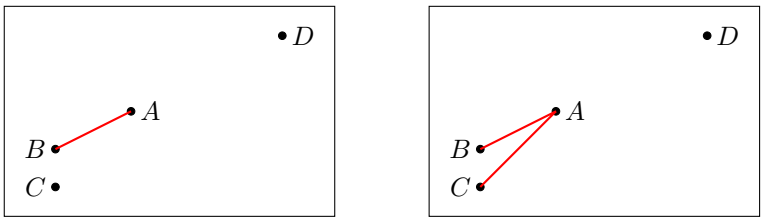


Figure 2: Neighbours of A when not using the 90 degrees criterion.

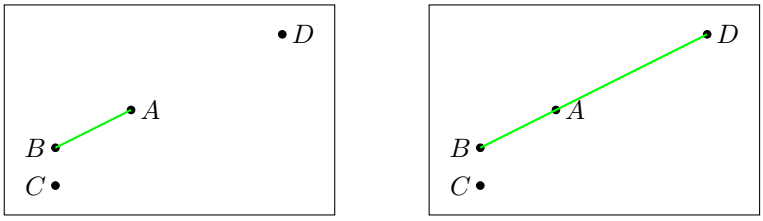


Figure 3: Neighbours of A when using the 90 degrees criterion.

and 1s, with at most two 1s per column. The reason we call this a pseudo-connection matrix is that it is not the standard connection (adjacency) matrix for a directed graph; it also need not be symmetric, as the connection matrix of an undirected graph would be.

The next step is to use the direction of the road to make sure we start at the beginning of the road and then find the successor sensors. To determine the starting point we simply look at the raw coordinates, in combination with the direction, e.g., if a road is eastbound we start at the westmost point. This first sensor, with index o_1, should only have a single connection to other nodes, i.e., the o_1-th column of P has a single entry at some row o_2. Thus, we have found the second sensor on the road. We then keep going by looking at the o_2-th column, which should have 2 nonzero entries, one at the o_1-th row and one at some row o_3. If the sensors are placed nicely on a straight road, we can continue this way until the end of the road (see Figure 4).

However, when the road is not straight this method may sometimes go awry, as illustrated in Figure 5. Starting at B we go to the successor A. There we have a problem, since A connects to both C and D. To fix problems as these we choose whichever point has the largest distance to A, skipping the other point. We may lose some sensors in the process but this will not invalidate the traffic estimation; it only reduces the accuracy of the approximation (see Section 3)

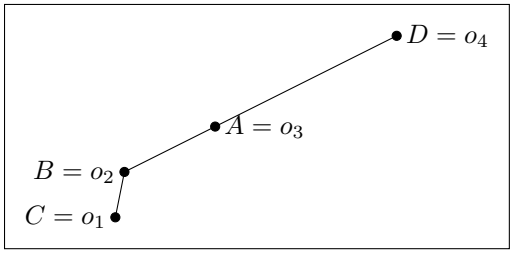


Figure 4: Eastbound road constructed from sensor coordinates.

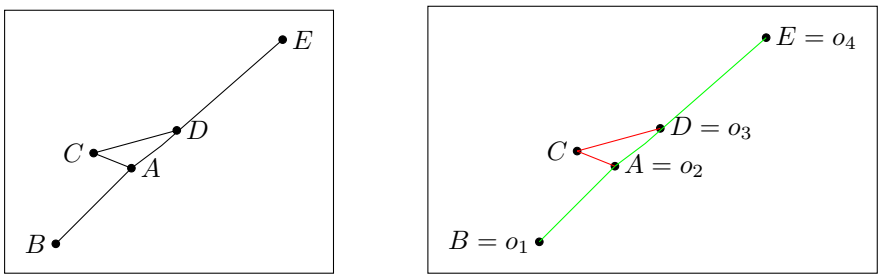


Figure 5: A problematic graph and the used solution.

Another method to construct the graph for a single road, is to assume that the shortest path that connects all sensors is the correct way to connect them. This graph can be calculated by adding a dummy node with distance 0 to all other nodes, and then solving the Traveling Salesman Problem (TSP).

Solving a TSP for each road is doable as long as the number of sensors on a single road is not too high, but using the knowledge that a road should be mostly straight can also inspire a number of heuristics that are quick to calculate. For instance, a linear regression line through the sensors can be calculated, and then sensors can be ordered by projecting them onto that line. If needed, simple local search heuristics can be used to improve on the initial solution.

3 Calculating traffic

Given a directed graph, where the nodes represent sensors and the edges represent the road segments between these sensors, the intensity and velocity measurements, in conjunction with the segment lengths, can be used to estimate traffic quantities. Here, we discuss estimating the number of cars on the road at a given time, and the number of kilometers driven by all cars on the road during a certain time period.

3.1 Car count

To estimate the number of cars on the road at a given time, we estimate the number of cars on each road segment. In the simplest case, we assume cars to be driving at constant speed for some time around the measuring sensor. The estimated number of cars on the road segment AB, based on the measurements at sensor A and sensor B, respectively, equals

$$C_A(t) = I_A(t)\frac{d_{AB}}{v_A(t)}, \tag{3}$$

$$C_B(t) = I_B(t)\frac{d_{AB}}{v_B(t)}. \tag{4}$$

Note that to be correct, the estimates $C_A(t)$ and $C_B(t)$ need the average velocities $v_A(t)$ and $v_B(t)$ to be the harmonic mean of the velocities of the passing cars. Unfortunately, only the arithmetic mean is available in the data. This issue is further discussed in Section 4.

The estimate $C_A(t)$ assumes all cars that enter the road segment at sensor A to drive along the entire segment. It does not take into account cars leaving the road somewhere along the segment AB. Similarly, $C_B(t)$ does not take into account that cars may have entered somewhere along the segment.

Assuming that leaving and entering occurs halfway along the road segment, these two effects can be incorporated in the car count by averaging $C_A(t)$ and $C_B(t)$:

$$C_{AB}(t) = \frac{C_A(t) + C_B(t)}{2} = \frac{d_{AB}}{2}\left(\frac{I_A(t)}{v_A(t)} + \frac{I_B(t)}{v_B(t)}\right). \tag{5}$$

Note that this formulation is independent of how many cars leave and enter the road on segment AB. All that matters is the difference between the number of cars that left and that entered.

For short road segments AB, specifically if $\frac{d_{AB}}{v_A(t)} < T$, equation (5) should give a good estimate for the number of cars on the segment. For longer road segments, the constant speed assumption may be too much of a simplication. This problem can be alleviated by using measurements of multiple time intervals. We still assume the cars to traverse the road segment with constant speed as measured at the sensor, but the different speed of cars that arrive within different time intervals will be taken into account.

In the time interval $[t + kT, t + (k + 1)T)$, cars are measured at a sensor with average speed $v(t + kT)$. Relative to the placement of that sensor, at time t these cars are expected to be at $(-(k + 1)Tv(t + kT), -kTv(t + kT)]$.

At sensor A only measurements before t are interesting, i.e., $k < 0$, because measurements after t correspond to cars that had not entered the segment AB yet at time t. For $k < 0$, if

$$-kTv_A(t + kT) \leq d_{AB} \Leftrightarrow \frac{d_{AB}}{Tv_A(t + kT)} + k + 1 \geq 1 \tag{6}$$

then all the measured cars are within the segment AB at time t, while if

$$-(k + 1)Tv_A(t + kT) \geq d_{AB} \Leftrightarrow \frac{d_{AB}}{Tv_A(t + kT)} + k + 1 \leq 0 \tag{7}$$

then all the cars already passed the segment at time t. In all other cases, assuming a uniform distribution of the cars within the measured time, the fraction of the measured cars that are in segment AB at time t is equal to

$$\frac{d_{AB} - [-(k + 1)Tv_A(t + kT)]}{Tv_A(t + kT)} = \frac{d_{AB}}{Tv_A(t + kT)} + k + 1. \tag{8}$$

The number of cars on the road segment AB can then be estimated from the measurements in sensor A by adding contributions for all $k < 0$,

$$C_A(t) = T \sum_{k=-N}^{-1} I_A(t + kT) \max\left\{0,\ \min\left\{\frac{d_{AB}}{Tv_A(t + kT)} + k + 1,\ 1\right\}\right\}. \tag{9}$$

Similarly, the number of cars on segment AB can be estimated from the measurements in sensor B by

$$C_B(t) = T \sum_{k=0}^{N-1} I_B(t+kT) \max\left\{0,\ \min\left\{\frac{d_{AB}}{Tv_B(t+kT)} - k,\ 1\right\}\right\}. \tag{10}$$

Here, the truncation value N should be such that no contributing measurements are neglected. A safe value for N can quickly be calculated from

$$N = \frac{d_{AB}}{T \min_{t,\xi\in\{A,B\}} v_\xi(t)}. \tag{11}$$

Again averaging to account for cars leaving and entering the road along the segment AB, we get

$$\begin{aligned} C_{AB}(t) = \frac{T}{2}\Bigg(&\sum_{k=-N}^{-1} I_A(t+kT) \max\left\{0,\ \min\left\{\frac{d_{AB}}{Tv_A(t+kT)} + k + 1,\ 1\right\}\right\} \\ &+ \sum_{k=0}^{N-1} I_B(t+kT)\ \max\left\{0,\ \min\left\{\frac{d_{AB}}{Tv_B(t+kT)} - k,\ 1\right\}\right\}\Bigg). \end{aligned} \tag{12}$$

Note that the treated methods for counting traffic are essentially independent of the number and placement of the sensors. That is, if the simplifications assumed to model the traffic on a road segment would be exact, then using measurements from any set of sensors would lead to the same traffic count, provided that the same part of the roads is covered. Evidently, using more sensors that are closer together does lead to more reliable estimates.

Many more extensions are possible to better estimate the number of cars on a road segment at a given time. For instance, using an interpolated function for intensity and velocity, or incorporating a typical distribution of velocities for a certain road. However, the intended use of these statistics is a traffic index, i.e., an aggregation over a time period and a geographical area. In this case, the provided estimates are expected to be accurate enough, as the local approximation effects should not significantly impact the aggregate.

Figures 6–8 show the estimated car count on two major roads in South Limburg, based on equation (5), using the minute data of the road sensors. Figure 6 shows the estimated traffic on the A76 on a Friday. The morning and afternoon rush hours are clearly visible. There is slightly more westbound traffic in the morning, and more eastbound traffic in the afternoon. Figure 7 shows the traffic on the A2 on a Friday. Again, the morning and afternoon rush hours are clearly visible. Further, it is clear that there is a lot more southbound traffic, towards the city of Maastricht, the entire day. Figure 8 shows the traffic on the A2 on a Saturday. There is still a lot more southbound traffic, but there is no morning or afternoon rush hour.

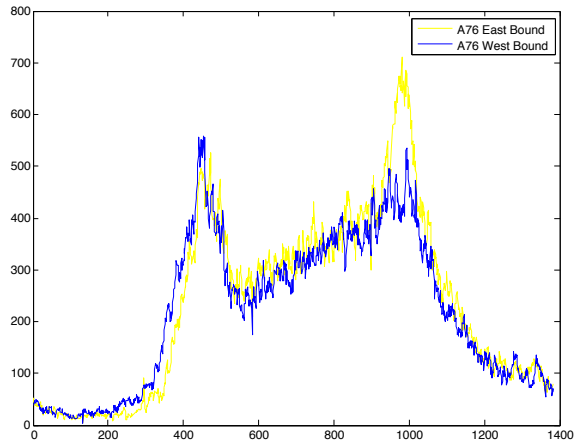


Figure 6: Car count on the A76 on a Friday (Feb. 1, 2013).
Horizontal: minutes past midnight. Vertical: number of cars on the road.

3.2 Car-kilometer count

We want to build a traffic index as an indicator of the traffic usage on the infrastructure. One way of measuring this is the average number of cars on the road, as described in the previous section. Another way would be the total usage of the roads by car-kilometers. In this case, one is not only interested in the current number of cars but also how far they travel. This is measured by the total number of car-kilometers, i.e., if there is a way of recording all cars and their whereabouts, the sum of kilometers of all car trips on the road network. The car-kilometer approach is studied in this section.

Imagine an abstract straight road $[0, A]$. The road sensors (observation points) are located at positions $x_1, x_2, \ldots, x_K$ (assume $0 = x_0 < x_1 < x_2 < \cdots < x_K < x_{K+1} = A$). Take $\rho(x, t)$ as the car density at location x at time t, such that the total number of car-kilometers on the road, in the time period $[0, \tau]$, is

$$K(\tau) = \int_0^\tau \int_0^A \rho(x, t) dx dt. \tag{13}$$

We do not know $\rho(x, t)$ at every x, but we have traffic records at the sensors x_i. Thus, we can approximate $\rho(x, t)$ with piecewise constant functions using our observations $\rho(x_i, t)$, and then come up with an approximation of K. From a numerical integral theory viewpoint, the best method is to divide the intervals to construct the piecewise function as follows. Given $\rho(x_i, t)$, we

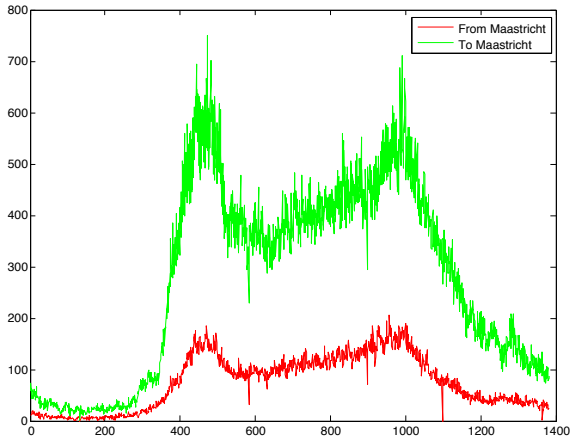


Figure 7: Car count on the A2 on a Friday (Feb. 1, 2013).
Horizontal: minutes past midnight. Vertical: number of cars on the road.

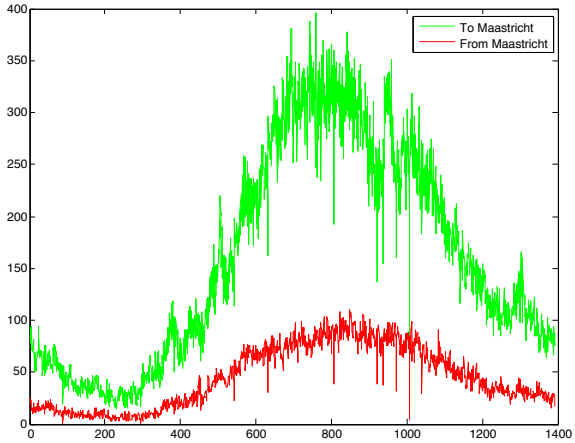


Figure 8: Car count on the A2 on a Saturday (Feb. 2, 2013).
Horizontal: minutes past midnight. Vertical: number of cars on the road.

approximate $\rho(x,t)$ by

$$\rho(x,t) \approx \sum_{i=1}^{K-1} \mathbb{1}_{\left((x_{i-1}+x_i)/2,(x_i+x_{i+1})/2\right]}(x)\rho(x_i,t). \tag{14}$$

The values $\rho(x_i,t)$ can be approximated by the observations at sensor x_i. We then arrive at the following approximation of K:

$$K(\tau) \approx \sum_i d_i N_i, \tag{15}$$

where N_i is the number of cars passing sensor x_i in the time interval $[0,\tau)$, and d_i is the length of the interval around x_i, i.e., for three consecutive sensors x_{i-1}, x_i, and x_{i+1} take $d_i = \frac{1}{2}(\|x_{i+1}-x_i\| + \|x_i - x_{i-1}\|)$. Thus, d_i is the estimated travel distance of a car captured by sensor i, covering half of the interval before sensor i and half the interval after.

In terms of the provided data, an approximation $\mathrm{CK}(t)$ of the car-kilometer count in the time interval $[t, t+T)$ can be obtained by

$$\mathrm{CK}(t) = \sum_i d_i I_i(t) T, \tag{16}$$

where $I_i(t)T$ is the number of cars passing sensor x_i in the time interval $[t, t+T)$, and d_i is a length that represents the part of the road around x_i, as in equation (15).

We developed the above method for an abstract straight road. However, the principle works for any topography. We just have to associate the correct length to sensors. Naturally these lengths are difficult to deal with, as they need detailed information on the geography of roads. However, this approach has some nice properties. First, it does not care whether a car is recorded by several sensors. If a car is recorded twice, it means the car travels more kilometers, which means more damage to the roads. (It could also mean more air pollution.) Second, this approach does not mind too many sensors, as it is cheap to compute, and the more sensors the more accurate the approximation will be.

The car-kilometer count is a fairer indicator of infrastructure usage than the car count. It represents how widely and extensively the infrastructure has been used for a certain time period. It is simple to calculate and easy to implement, and robust to using more or fewer sensors.

siteID	date	location	roadNo	lane	direction	flow	speed
...				...			

Table 1: Excerpt from the sensor data.

4 Data reconstruction

In the previous sections, a detailed method for the calculation of a traffic index has been proposed and formulated. Although theoretical considerations cover some of the practical problems, there are still several issues with the input data that need detailed analysis. In this section, the structure and characteristics of actual data are discussed, the main problems are identified, and solutions and corrections are proposed.

4.1 Data format and data problems

The data provided by CBS covers both 1 minute and day measurements. The former are restricted to the region of South Limburg for two consecutive days: Friday, 1 Feb. 2013 and Saturday, 2 Feb. 2013, and include 2 million observations from 424 road sensors. The latter include observations from all of the Netherlands for a period of 3 months, and include 900 000 observations from 15144 sensors. The available variables describe a wide range of different features, including flow and velocity measurements, directional information, and location coordinates.

Table 1 presents an excerpt of the data that covers the most important variables from the point of view of the method proposed in previous sections.

As the traffic system is constantly changing, sensitive to unpredictable phenomena, and vulnerable to different factors, one must also expect possible errors in the traffic data. Fortunately, the system is designed to return "-1" if there was an error. This allows to differentiate between no traffic (zero measurements) and malfunctioning equipment.

A preliminary analysis of the data shows several obstacles for the implementation of the proposed solution. First, we found that there is a problem with the availability of directional data of sensors. Second, we identified also missing and incorrect intensity and velocity data. Finally, the velocity averaging method needs some investigation.

Brief comments regarding all the variables are summarized in Table 2.

Variable	Comment	Problems
siteID	Unique number for each sensor	No problems
date	Time of measurements	No problems
location	Geographical coordinates	No problems
roadNo	Exact road number	No problems
lane	Number of the lane	No problems
direction	Direction of the sensor	Many missing entries
flow	Measured intensity	Both errors (-1) and missing data (unexpected 0)
speed	Measured velocity	Averaging: arithmetic mean Both errors (-1) and missing data (unexpected 0)

Table 2: Summary of variables and problems in the data.

4.2 Missing data

The first step of our solution is to create a graph using the data on the location and direction of the sensors. Therefore, it is important to be able to match every sensor to an exact geographical point. Figure 9 shows all sensors in the Netherlands and in South Limburg, drawn according to the provided longitude and latitude information.

As the presented figures illustrate, the sensors represent a network of the main roads in the Netherlands, which is compatible with our proposed approach. However, if we were to use only sensors for which directional data is available, we would have to omit a considerable part of the Netherlands. About 30% of the sensors lack information on the direction, see Figure 10. Hence, having a systematic procedure to reconstruct the missing data is vital, in order to provide a reliable calculation of the traffic index.

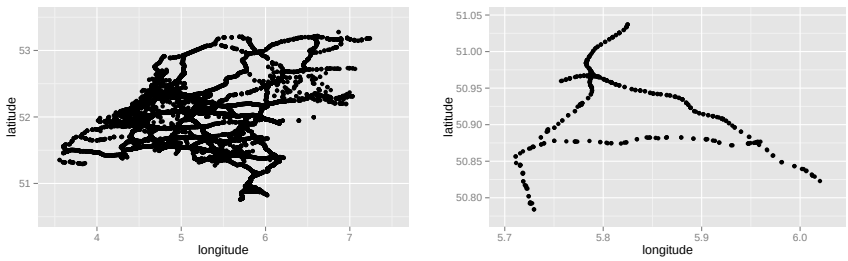


Figure 9: Sensors in the Netherlands (left) and South Limburg (right).

The second problem with the data concerns missing and incorrect mea-

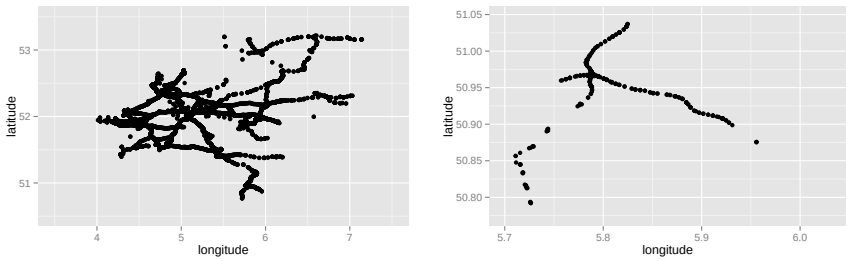


Figure 10: Sensor with available direction data in the Netherlands (left) and South Limburg (right).

surements of intensity and velocity of traffic. We distinguish between two possible errors: 'blind sensors' and 'blind time points'. The first indicate the situation where a sensor does not give any data for any period of time, whereas the second represents holes in the time series data for a given sensor. Table 3 shows the percentages of these two types of errors in the provided data.

	Velocity	Intensity
NL	6.5	4.5
SL	10.5	8.0

	Velocity	Intensity
NL	2.5	2.0
SL	15.5	0.0

Table 3: Percentage of blind sensors (left) and blind time points (right), for the Netherlands (NL) and South Limburg (SL)

The percentages of errors are tolerable, as our proposed solution does not need to be applied using all possible measurements. Problematic sensors can be excluded with a little loss of precision.

The third identified problem concerning the data lies in the method of averaging velocity measurements. For our purpose the correct way to compute an average of speed is to use the harmonic mean of the observations. Unfortunately, all the provided data give are the value of the arithmetic mean.

4.3 Direction reconstruction

The main problem with the data is the lack of directional information in a substantial number of sensors. We propose to retrieve this data on the basis of the behaviour of the traffic intensity. We hypothesized that, as far as different directions of one road are concerned, patterns of traffic should differ within a weekday. For example, in the morning rush hours the traffic should converge mostly to the biggest city in the neighbourhood, whereas during the afternoon the profile should be the opposite. The same is true in the case of a significant event in a specific place.

From the mathematical point of view, we would like to compare a set of time series data with each other and then cluster the sensors according to some distance function for a pair of sensors. We identified two approaches, which can be applied to this problem:

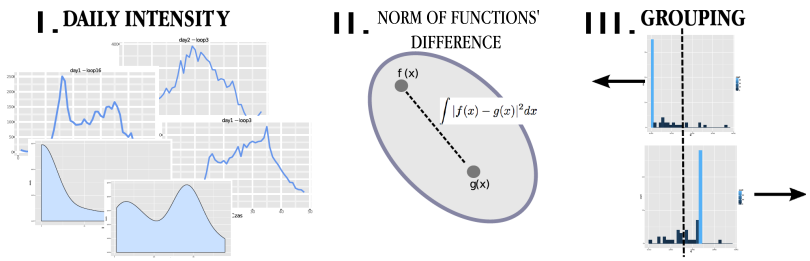
1. Compare a whole-day time series for each sensor. One simple method is just to use a mean square error as a distance function. This will not take into account the possible time dependence (time lag) between consecutive sensors. Hence, we propose to distinguish them using a cross-covariance (cross-correlation) function, which compares two time series and their lagged transforms[1].

2. Calculate the distribution function of the intensity for each sensor in one, chosen rush hour, i.e., either in the morning or in the evening. In order to estimate the distribution, one can use histograms or kernel density estimators. In the next step, the distance function would be the difference between two histograms/densities, which can be calculated using Kullback-Leibler divergence or just the mean square error.

The method which compares only distributions of intensities is much simpler to apply and cheaper to implement, but it aggregates and loses information included in the time series. Therefore, it has a potentially smaller range of applicability. Nonetheless, if the hypothesis stated above is true the distinction made in that way should be reliable.

[1] In the statistical software package R, this is implemented via the function *ccf*.

The presented idea can be summarized in the following picture:



The steps for method 1 are:

a. Identify continuous approximation of intensities.

b. Compute how the time series differ between sensors.

c. Divide sensors into two groups according to the distance.

For method 2, the steps are:

a. Identify distribution of intensities in the rush hour.

b. Compute distance between densities (Kullback-Leibler).

c. Divide sensors into two groups according to the distance.

We believe that such an approach should be useful, when there is strong evidence that the traffic is very different in two directions. Unfortunately, as the provided dataset for South Limburg shows, it is not a general rule that always holds. In particular, such an approach was not successful enough in differentiating the direction of sensors for roads A2 and A76, where we know most directions. We cannot assume then that the approach can be applied for the road A79 in South Limburg. In that case, we must in addition make use of information on the location of the sensors. Therefore we propose the following procedure, which is a modification of the initial proposition.

Let us assume we are given a set S of sensors within a single road without direction. Sets S_1 and S_2 will include sensors in different directions.

1. Choose an arbitrary sensor $s_0 \in S$, add it to S_1 and delete it from S.

2. Choose n (we propose $n = 4$) different sensors $s_1, \ldots, s_n \in S$ that are the nearest (in the sense of location) to s_0, but not farther than a fixed tolerance distance d (we propose $d = 3$km). If it is not possible, go to step 1 (with different s_0).

3. Calculate the cross covariance for the intensity time series data between s_0 and each sensor from the set $\{s_1, \ldots, s_n\}$. The number of calculated lags (mostly 1, 2 or 3) should be chosen according to the average velocities and distance between sensors; car-count formulas from section 3.1 can be used here.

4. Choose one sensor, s_h, with the highest covariance. Add s_h to S_1 and delete it from S.

5. Choose one sensor, s_l, with the lowest covariance. Add s_l to S_2 and delete it from S.

6. If S is empty, stop. Otherwise, go to step 2 and repeat on the current S with $s_0 := s_h$.

The method will be most efficient, if the first chosen sensor is approximately in the middle of the set S in the sense of location. Moreover, the algorithm will work better, if sensors are uniformly distributed in both directions, i.e., the number of sensors in each direction is similar. The more the number of sensors in each direction differs, the higher the choice of the variable n should be. The procedure needs to be carried out only once, but should be tested on a regular basis with newly available data.

Using this method, the identification of the directions for roads A2 and A76 in South Limburg was at a level of 80%. We apply the method to the data for A79, a reconstruction of which is presented in Figure 11.

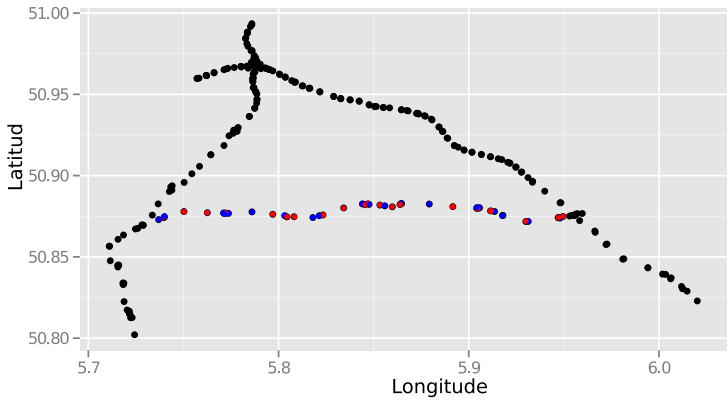


Figure 11: Reconstructed directions (red/blue) for the A79 in South Limburg.

4.4 Velocity averaging

Another issue that needs special attention, is that the velocity is given as an arithmetic mean instead of the harmonic mean. The best method to correct this problem is of course to get the harmonic mean from the source of the measurements. If this is impossible, however, the suggestions below can be used to improve the calculations.

An elementary inequality between means states that for positive $x_1, \ldots, x_n$

$$\frac{n}{\frac{1}{x_1} + \ldots + \frac{1}{x_n}} \leq \frac{x_1 + \ldots + x_n}{n}, \tag{17}$$

i.e., the harmonic mean is less than or equal to the arithmetic mean.

In the case of the calculation of a traffic index, this means that using an arithmetic mean of observations leads to an upper bound on the true value. Unfortunately, there is no method to quantify the error. The difference between the arithmetic and harmonic mean can attain any value depending on the dispersion of observations. The higher the difference between the observation values, the larger the error of using the arithmetic mean.

We see the following options to achieve more accurate calculations:

- aggregate different types of vehicles into one group using a harmonic mean for the velocity,
- aggregate minute measurements into longer periods using a harmonic mean for the velocity,
- aggregate observations with respect to location, using a harmonic mean for the velocity.

All these proposals give better approximations to the traffic index, but at the same time they sacrifice some details of the data, e.g., information on different types of vehicles. This approach to correcting the dataset will be effective especially in the case of low-traffic periods. Moreover, the proposed solution in previous sections implictly assumes harmonic averaging with respect to location, which already alleviates the problem somewhat.

A more involving method could be devised, if a typical distribution of vehicle speeds on the road is available. The arithmetic mean could then be used to select a likely set of velocities that realised that mean, and from this set the harmonic mean could be calculated. The accuracy of such a method would rely heavily on the predictability of the vehicle speeds.

5 Principal Component Analysis (PCA)

The procedure described in the previous sections gives an approximation of the number of cars on the roads in a specific time interval using the minute-by-minute information collected by the sensors. However, this information could not be available in certain scenarios where the collected dataset is reduced to the number of cars detected by the sensors in one hour or one day.

In this case, the data show a main feature: they contain much redundant information because one car can be detected by several sensors. In matrix terms, it means that if we assume that the information of each sensor is in the columns of a matrix A, then the dimension of the linear subspace spanned by the columns of A is much smaller than the number of sensors.

In this case, describing the data using an orthonormal basis of this 'smaller' subspace that contains a compressed representation of A, is a good strategy. The Singular Value Decomposition (SVD, Golub and Van Loan (2012)) of A permits to calculate a possible set of basis vectors formed by the singular vectors corresponding to the largest singular values of the matrix A.

This fundamental data analysis tool, known as Principal Components Analysis (PCA), is one of the methods investigated by the CBS specialists to know the variability of the data and calculate a traffic index.

Some recent approaches can improve the application of PCA on the traffic data:

- Robust PCA, Ke and Kanade (2005): Techniques to recover the low-rank matrix approximations from highly corrupted and/or missing measurements.

- CUR matrix decomposition, Mahoney and Drineas (2009): Here, the basis vectors are explicitly expressed in terms of a small number of actual columns and/or actual rows of the data matrix. In the traffic index context, the application of the CUR matrix decomposition could detect sets of sensors where the redundance in the information is minimized. In contrasts with the known limitation of the classic PCA, with CUR one can interpret the produced basis in terms of the original data. An additional advantage of PCA based on CUR decomposition is avoiding the calculation of the Singular Value Decomposition, giving the possibility of working with a much larger dataset.

A final observation: the calculation of a traffic index using a PCA version on the hourly or daily data does not use the information of the velocity of the cars on the road. This can be an advantage considering the high proportion of missing data for this parameter.

6 Conclusions

In this paper, the problem of calculating a measure of the amount of traffic on the roads, based on data from road sensors, has been treated.

A method was presented for constructing a graph, based on the sensor data, with the sensors as nodes and the road segments between sensors as edges. Using such a graph, we presented methods for estimating the number of cars, and the number of car-kilometers. Both estimates are more accurate when more sensors are used, but are otherwise independent of the number and placement of sensors used, provided that the same set of roads is covered by the sensors.

Further, procedures were proposed that deal with missing data, focussing on reconstructing the traffic direction for sensors that missed the direction data field, and on dealing with the arithmetic velocity mean being given, when our methods rely on the harmonic velocity mean.

Finally, the uses of principal component analysis for this particular problem were discussed.

References

Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. The Johns Hopkins University Press, Baltimore, MD, fourth edition, 2012.

Q. Ke and T. Kanade. Robust l^1-norm factorization in the presence of outliers and missing data. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2005.

M. W. Mahoney and P. Drineas. CUR matrix decompositions for improved data analysis. *PNAS*, 106(3):697–702, 2009.

, , , , , , , ,

The Mathematics of French Fries

Nicodemus Banagaaya[1],
Neil Budko[2] [*],
Hans van Doorn[3†],
Giorgi Khimshiashvili[4],
Rob Klooster[3],
Pieter-Jelte Lindenbergh[3],
Jacqueline Verdijck[3],
Fred Vermolen[2]
and Ian Zwaan[1]

[1] TU Eindhoven
[2] TU Delft
[3] HZPC
[4] Ilia State Univ., Georgia

Abstract

Although the act of cutting a single potato (*Solanum tuberosum*) into french fries may appear to be trivial, the questions concerning the efficiency of this process on an industrial scale are quite daunting. Therefore, many producers are looking for a rigorous method to evaluate the market potential of a given potato crop by predicting the number and parameters of the fries that can be cut from it. Applying the methods of geometry and numerical analysis our group was able to propose several algorithms that can be directly incorporated into the existing production process.

Keywords: French fries, geometry, cutting, Finite Fry Method, simulations, histograms

[*]For questions about mathematics: n.v.budko@tudelft.nl
[†]For questions about potatoes: hans.vandoorn@hzpc.nl

1 Introduction

The HZPC Holland B.V. company is a major international supplier of potatoes. The question posed by HZPC could be summarized as follows:

Question: *Given a set of potato tubers with approximate information about the size of each tuber and the desired cross-section of a fry, how many french fries of various lengths and textural quality can be obtained from this set, and what will be the volume fraction of waste?*

During the SWI week our group was able to solve the geometrical part of the problem. Whereas, the question concerning the textural quality remains partially open. Below we describe two methods to cut tubers into virtual fries and estimate their number, their length distribution, and the amount of waste. The analytical approach provides with the formula for the volume of each tuber based on the best elliptic approximation of the surface and with a quick and dirty way to compute the fry length histogram from experimental data. The second approach, which we call the Finite Fry Method (FFM), has been implemented as two separate numerical algorithms: one that is tuned to work with the currently available experimental data, and the other that allows generating tubers of arbitrary shape and perform Monte-Carlo studies on the obtained ensemble. Finally, we have developed an analytical procedure to interpolate the DMC quality trait distribution sampled in a small number of points onto the whole volume of the tuber.

2 Geometrical modeling

The available information about the shape of each tuber consists of its maximal extent (*length*) and several measurements in other directions. Namely, in nine equidistant planes orthogonal to the length direction the measurements of the *width* and *height* are available, two in each plane. However, although it is known that the directions of the width and height are mutually orthogonal and consistent over all nine planes, the measurements do not provide with the actual coordinates of the boundary points.

This limited information allows constructing only a simplified geometrical model of a tuber. We consider the intersection of the tuber boundary surface with each of the nine cross-sectional measurement planes to be a concentric but not necessarily a confocal ellipse with known semi-axes.

Computing the volume of a tuber requires interpolating its boundary surface between the aforementioned cross-sectional ellipses. Depending on the chosen interpolation model the surface between the planes may or may not have an elliptic cross-section. Without losing much precision we assume that *all* cross-sections of the tuber in the direction orthogonal to its length are indeed elliptic. The advantage of this assumption is the ability to compute the

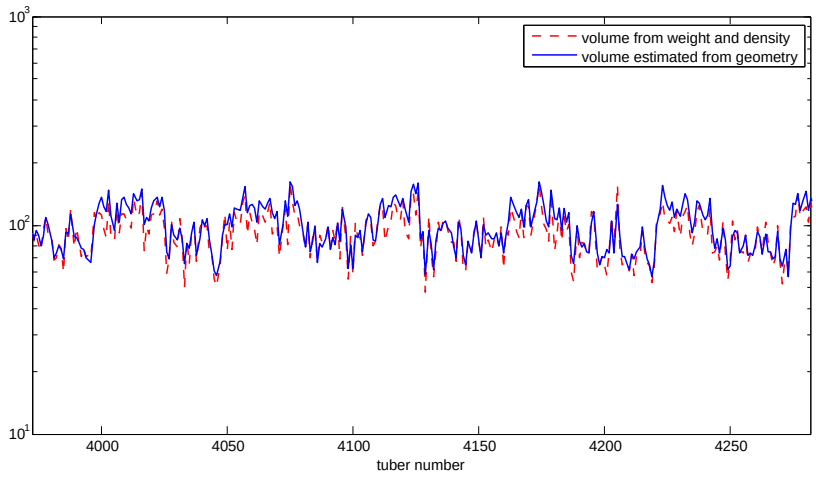


Figure 1: Calculated volume of the geometric model compared with the volume estimated from the weight and density of tubers.

volume of a tuber explicitly as a sum of the volumes of eight twisted elliptic cylinders defined by the measurement planes plus the volume of two caps that can be modeled as elliptic cones. The result of computing the volume via this approach is shown in Figure 1 where it is compared with the volume estimate based on the weight and density of a tuber.

Further improvement of this geometric model may consist in a better description of the tuber caps, for example, as elliptic paraboloids instead of elliptic cones.

3 Cutting potatoes into fries

With the model of the tuber boundary surface at hand one can proceed 'cutting' the potato into fries and estimating the waste – fries that do not conform to the industry standards.

Let the z-axis be along the length direction. Introduce a uniform two-dimensional Cartesian grid in the horizontal (x, y)-plane with the grid step h – the chosen width of a fry ($h = 6, 7, 8, 9, 10, 12, 14$, or 16 mm). Cut the tuber into fries numerically by constructing finite elements that extend along the z-axis with their vertical edges coinciding with the grid points of the previously defined horizontal grid. The surface of the tuber is approximated

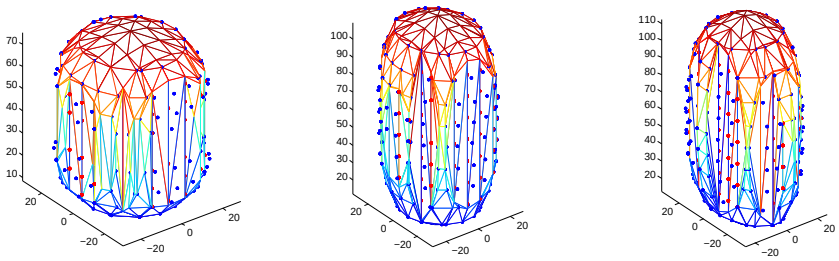


Figure 2: Three tubers from the $40 - 50$ mm class (experimental data) cut into fries of width $h = 6$ mm. Only acceptable fries are indicated.

in a piecewise linear fashion at this stage. Discard all fries that do not conform to the industry standard and compute the waste.

The result of the application of this numerical algorithm, which we call the Finite Fry Method (FFM), to the first three tubers in the $40 - 50$ mm width class is shown in Figure 2.

Although, we were able to run this algorithm on as many as 11333 tubers, the discretization of a large number of tubers into fries may take significant time (about an hour on a laptop). Therefore, we have also developed an approximate but simple method to estimate the number of fries that can be cut from each tuber. The main idea is to count only the fries that have a square cross-section discarding all wedge-shaped fries that are cut from the sides of a tuber. This method detects the smallest of the cross-sectional ellipses of each tuber, and is extremely fast (seconds) and surprisingly accurate (see Figure 3).

4 Modelling variations in tuber geometry

Another realization of the finite-fry method (FFM) was developed to relax the current experimentally imposed limitations on the tuber geometry. Namely, as was mentioned above, the measurements did not provide with the actual coordinates of the boundary points and forced us to assume the concentric elliptical geometry when working with the data. In reality, however, tubers can have very asymmetric shapes. In this second realization of the FFM we have implemented a more systematic finite-element approach and have simulated a large statistical ensemble of asymmetric tubers.

As before, the potato shape is determined by a number of slices with elliptic contours orthogonal to its maximal (length, vertical) dimension. However,

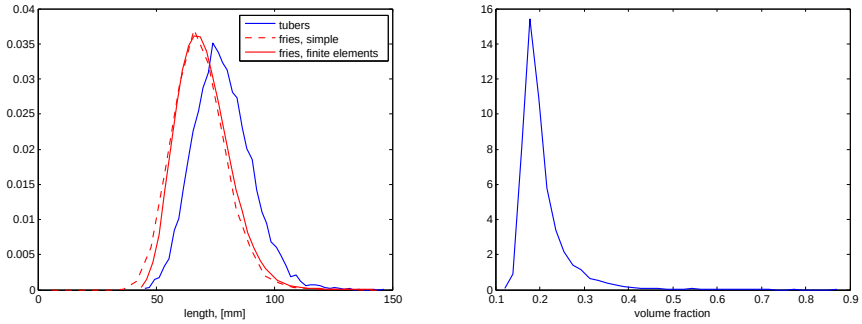


Figure 3: Left: Distribution of lengths of all acceptable 6 mm wide fries that can be cut out of 11333 tubers from the $40 - 50$ mm class calculated using a simple geometrical estimate and a more exact numerical procedure. Also shown is the distribution of the tuber length within the class. Right: Distribution of the volume fraction of waste.

this time not only the dimensions and orientation of the ellipses are allowed to vary but also their centres are allowed to shift off the length axis. In Figure 4 we show a schematic representation of tubers defined by a set of algorithmically generated ellipses.

To get the number of fries, we use a background (horizontal) square domain with the width and height chosen in such a way that all tubers of the simulated ensemble can be projected onto this domain. This domain is referred to as the *reference surface*. This area is divided into squares of equal size, which we call elements. Each elliptic slice is projected onto the reference area so that each square fits either completely, or partially, or not at all in the projected slice.

To make the whole procedure more systematic we construct an ordered list containing the indexes of all points in groups of four – the vertices of each element. Such lists are usually referred to as the *topology* of the grid. This list makes it easy to determine whether a fry belongs to a projected ellipse by testing for each vertex and each element whether the coordinates (x, y) of the vertex satisfy

$$\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 < 1.$$

If that is the case, then the vertex point and thus the edge of the fry is inside the tuber. If an element, which represents the cross-section of a fry, fits either

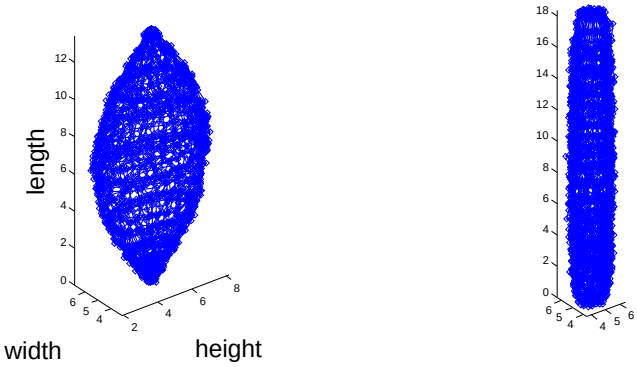


Figure 4: Two examples (Case I and Case II) of computer-generated tubers defined by ellipses with variable length, width, and center offset. Here, the length and width are deterministic functions, whereas the coordinates of the centres are taken from a normal distribution with zero mean and the standard deviation of 10 mm

partially or entirely within the projection of the slice onto the reference area, then this element is interpreted as a 'fry'.

Among all the fries we determine the list of *accepted fries* by accepting a fry if the intersection of the corresponding element with the projected ellipse has an area of at least $0.7\,A$, where A denotes the element area. A schematic of accepted and rejected fries over a single slice is given in Figure 5. This procedure is repeated for all the elliptic slices of the tuber.

The length of a fry corresponding to a certain element is determined by adding all the distances between adjacent slices that have yielded an acceptable intersection area with this element during the previous step. Due to the allowed variations in the tuber geometry some fries may have to be cut into pieces. In such cases each piece is considered to be a separate fry. As before, fries are only accepted if their length exceeds 40 mm. All the acceptance criteria for the fries can be changed in the model whenever necessary.

In the Monte-Carlo simulations, the tuber length, the centres, and the orientation and length of the semi-axes of the elliptic cross-sections are all assumed to be random normally distributed variables with adjustable means and variances. Furthermore, the mean width and height of each ellipse change according to a prescribed function in the length direction. We note that this function can be adjusted to model any desired shape.

Similarly to Figure 3 we present the results by comparing the distribution of the length of tubers (which is an input parameter) to the distribution of the

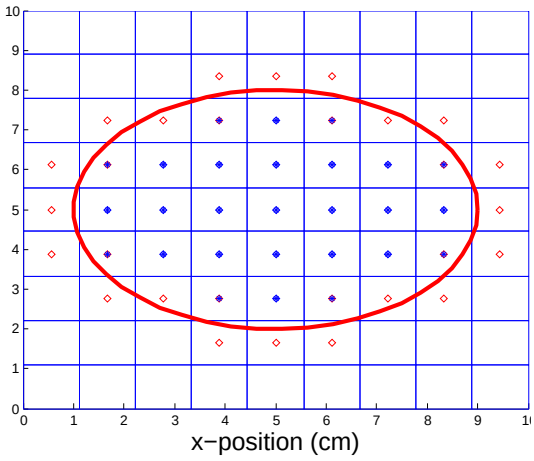


Figure 5: A schematic representation of accepted fries over an elliptic slice of a tuber. The red and blue dots, respectively, indicate the centres of the fries that are rejected and accepted

length of accepted fries (output parameter). As a tuber shape, we consider a mean tuber length of $L = 150$ mm, with the standard deviation of 35 mm, that is $L \sim \mathcal{N}(150, 35^2)$. The elliptic projections were simulated according to the formula:

$$\left(\frac{x - x_c}{a(z)}\right)^2 + \left(\frac{y - y_c}{b(z)}\right)^2 = 1,$$

where z is the vertical coordinate and $x_c, y_c \sim \mathcal{N}(0, 10^2)$ are the center coordinates taken from a normal distribution. We present the results for the following two shape functions:

$$\mathbb{E}\{a(z)\} = \frac{4}{L} z(L - z) = \mathbb{E}\{b(z)\}, \quad \text{Case I;} \tag{1}$$

$$\mathbb{E}\{a(z)\} = \left(\frac{4}{L} z(L - z)\right)^{\frac{1}{8}} = \mathbb{E}\{b(z)\}, \quad \text{Case II;} \tag{2}$$

where $\mathbb{E}\{\dots\}$ denotes the expectation used in the Monte-Carlo simulations. Note that we also set $a = b$ in the current simulations but we have the flexibility to use any value.

The results are presented in Figure 6. It can be seen that, in agreement with the experimental results of Figure 3, the average length of a fry is lower

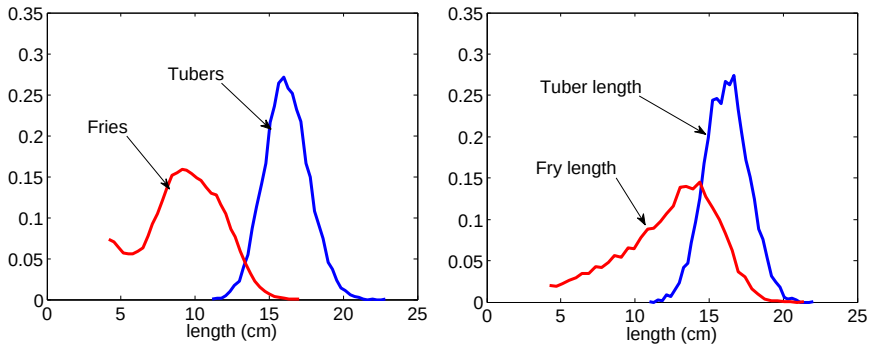


Figure 6: Statistical distribution of the tuber and fry lengths for two classes of tuber shapes, see (1)–(2).

than the average tuber length. This is a direct consequence of the large number of fries that is cut from the part of the tuber that is further away from the main axis. Along the main axis the length of fries is (approximately) equal to the tuber length, whereas at positions away from the main axis the fries are shorter and more numerous. This causes the relatively large number of fries with a length of 40 mm in the first case.

While the tuber shape is more ellipsoidal in the first case, in the second case the shape is more cylindrical (see Figure 4). Therefore, in the second case, the number of small fries is smaller. Thus, the proposed mathematical models can be used to quantify the observations. In particular, it is interesting to observe that the tuber geometry determines the shape of the probability density function.

5 Fry texture and the DMC quality trait

Another question posed by the HZPC company was to investigate the texture of fries after frying and the so-called DMC quality trait of raw fries. The DMC trait is currently measured in just a few points over the volume of some of the tubers. To interpolate the very sparse measurements of the DMC so that they can be used to determine the DMC of each individual fry we have introduced and computed the distance between an inner point of an elliptic slice and its bounding ellipse. This distance is then inserted into a parametric expression approximating the more recent fine-scale measurements of the DMC:

$$\phi(x, y) = \phi_c - \alpha(\mathrm{dist}((x, y); \Gamma))^2, \tag{3}$$

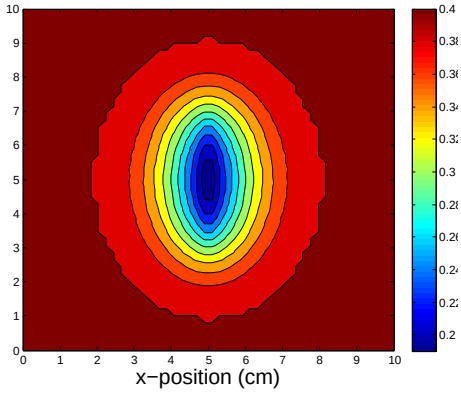


Figure 7: The distribution of the DMC quality trait over an elliptic slice.

where (x, y), ϕ_c, α and Γ, respectively, represent any point within the ellipse, the DMC content near the tuber surface (the maximum), and two tunable constants. The spatial distribution of DMC is denoted by ϕ. The proposed function is shown in Figure 7. We note that all tunable parameters can be easily determined by an optimization algorithm. In the current simulation, $\phi_0 = 0.4$, and $\alpha = 0.025$. We did not move further into this direction during the SWI-week.

6 Conclusions

The problem posed by the HZPC company was to find a mathematical technique that could help evaluate the market potential of a given set of potato tubers. For the particular French fries market the main quality factor is the amount and type of fries that can be cut from the tubers. As tubers vary in shape and size even after having been sorted into several size-related classes, the best approach appears to be the analysis of a histogram depicting the distribution of length among the fries. In order to obtain such a histogram we have developed an approximate and fast analytical procedure and a more exact numerical technique, which both cut tubers into virtual fries with a chosen cross-section. The advantage of the numerical approach stems from the fact that it allows investigating the influence of rather arbitrary shape perturbations on the histogram by running a Monte-Carlo simulation with a computer generated set of tubers. Both methods were tested on the provided experimental data and showed similar results. The analytical expression for

the volume of each tuber based on the available spatial measurements is in good agreement with the volume computed from the weight and density data.

The second part of the question concerned the distribution of the DMC quality trait inside the tubers and its relation to the texture of French fries. In this respect we have proposed a parametric expression that can be fit to the available sparse experimental data in order to interpolate the DMC density over the fry volume. Armed with this DMC trait distribution one could extend the histogram approach by showing the statistics of the DMC for each fry length.

, , , ,

Modelling a water purification process for quality monitoring

Frank van der Meulen[1], Stijn Luca[2]*, Gosse Overal[3], Johan Dubbeldam[1], Alessandro Di Bucchianico[4] and Geurt Jongbloed[1]

[1] TU Delft
[2] KU Leuven, Belgium
[3] VU Amsterdam
[4] TU Eindhoven

Abstract

This paper deals with a quality engineering problem introduced by 'Waterlaboratorium Noord' (WLN) situated at the Netherlands. Interest lies in determining an optimal sampling frequency that provides sufficient information on the water quality in a drinking water purification plant. The water purification plant that is studied consists of two aeration and filtration processes and a clear water reservoir where water is saved until distribution to households. One of the main processes during these filtration processes is iron removal.

A stochastic model is proposed that describes the decreasing effects on iron concentration after the filtration processes by multiplicative effects. This model is combined with an ordinary differential equation to model the amount of iron in the clear water reservoir that fluctuates due to the quality of the incoming filtrated water and the varying water demand.

In this way the iron concentration levels in the different compartments of a water purification plant can be simulated. Range and fluctuations approximate those of the observed data. Hence a realistic benchmark for detecting anomalies is obtained.

Keywords: quality engineering, sampling frequency, stochastic model, differential equation, water quality

*Corresponding author

1 Introduction

1.1 WLN

The quality of water has an important impact on public health. Whether it is used for drinking, for food production or just for recreational purposes, contaminated water can lead to severe health problems (Karanis et al., 2007). Therefore monitoring and keeping the quality of drinking water at a safe level is of crucial importance for our society.

The company 'Waterlaboratorium Noord' (WLN) takes care for the quality control of drinking water in Groningen and Drenthe. It monitors and improves the quality of drinking water provided by the water supply companies 'Waterbedrijf Groningen' and 'Waterleidingsmaatschappij Overijssel'.

Generally water is retrieved as surface water or groundwater. This water needs to be treated in a purification plant for several reasons, to remove harmful substances, to ensure it looks clear and to remove pathogenic microorganisms, to name a few. This purified clear water is then pumped through a distribution network and finally flows clearly and steadily into the households. Figure 1 depicts a typical treatment scheme for groundwater purification.

1.2 Outline of the problem

The main question posed by WLN is the following:

> Which monitoring frequency at the various stages in the water purification process is required for obtaining sufficient information on the water quality to prevent contaminated water to be delivered to households?

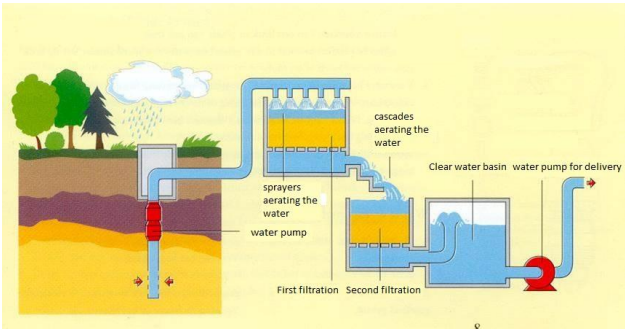


Figure 1: Illustration of a water purification process when groundwater is used as source.

One of the reasons this question is posed can be found in the Dutch legislation concerning the water quality. This legislation prescribes very strictly the quality control in the drinking water sources, in the clear water reservoirs and in the distribution networks. However, for the water inside the water purification plant there is only a rough guideline saying that the water company has to monitor the purification process for an 'adequate process control'. Hence the only demand is that it is being frequently monitored on several locations spread along the plant such that a sufficient overview of quality can be obtained. At this moment only expert judgement is used to determine this monitoring frequency.

1.3 Approach

In this paper we restrict attention to a single water treatment plant with groundwater as a source and two filtration steps. One of the main goals is to detect malfunctions of the purification process which can result in unacceptable water quality. In an attempt to detect such anomalies a model is introduced for predicting iron concentrations at each place in the purification process given the iron concentrations in the groundwater. A next step would be to monitor the differences between these predictions and the observations to reassure the quality of Dutch water.

Figure 2 shows a schematic overview of the proposed model of the purification process. The four main locations of the purification process are interpreted as basins where measurements are taken from. The transitions between the first three basins are modeled using a stochastic model with a multiplicative effect while the fluctuations of the iron in the clear water before distribution is modeled using a differential equation. The implementation is performed using the statistical software package R (R Core Team, 2012).

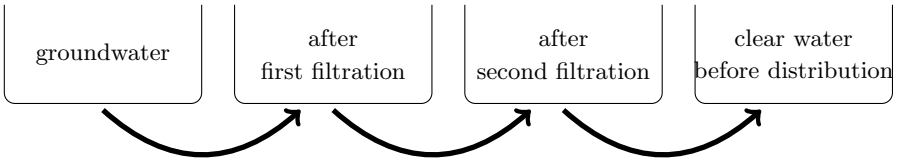


Figure 2: Schematic illustration of the water purification process

1.4 Outline of this article

In the next section we detail the data made available by WLN. In section 3 we present a model that relates the concentration of iron in groundwater

to the concentration in the clear water basin. This model uses both ideas from stochastic processes and differential equations for modellng conservation laws. We then move on to the results obtained. The final section contains conclusions, discussion and recommendations.

2 Available data

The data concern a water treatment plant that uses groundwater and basically removes substances through aeration and filtration. There are no pathogenic microbes in the water, because it is pumped up from deep beneath Earth's surface. The dataset consists of measurements of concentrations of several ions like e.g. Fe, Mg, NH_4. These measurements have been taken at four places during the purification process:

1. in the groundwater,
2. after the first filtration,
3. after a second filtration,
4. before the distribution (clear water).

The data include measurements over a period of 5 years starting from January 2009. The measurement frequency is about 1-2 times a week. Figure 3(a) shows the observed iron concentrations on each of the four locations mentioned above. As one can see the iron concentration decreases after each filtration.

Flushes of the filters at regular times cause fluctuations in the iron concentration after each filtration as can be seen in figure 3(b). Immediately after a flush the remaining fraction of iron in the water reaches a peak and subsequently decreases rapidly back to its original value. The peaks after first and second filtration in figure 3(b) seem to occur periodically, however an exact flush period is not made available. Figure 4 shows some typical flush curves describing the remaining fraction of iron in the water as a function of time since the filter has been cleaned. These samples are taken every 10 minutes during a period of roughly 12 hours. The curve can be modeled using non-linear regression analysis as will be discussed in the next section.

Fluctuations in the clear water are expected to be lower as the large volume of the clear water reservoir averages out most of the fluctuations. Fluctuations in this basin can, among other things, be caused by variations in the water demand.

In what follows t denotes the inspection time of the measurements, expressed in hours, since the first available measurement. Note that the time scale t is different than the one used to describe the flush curves. There, the

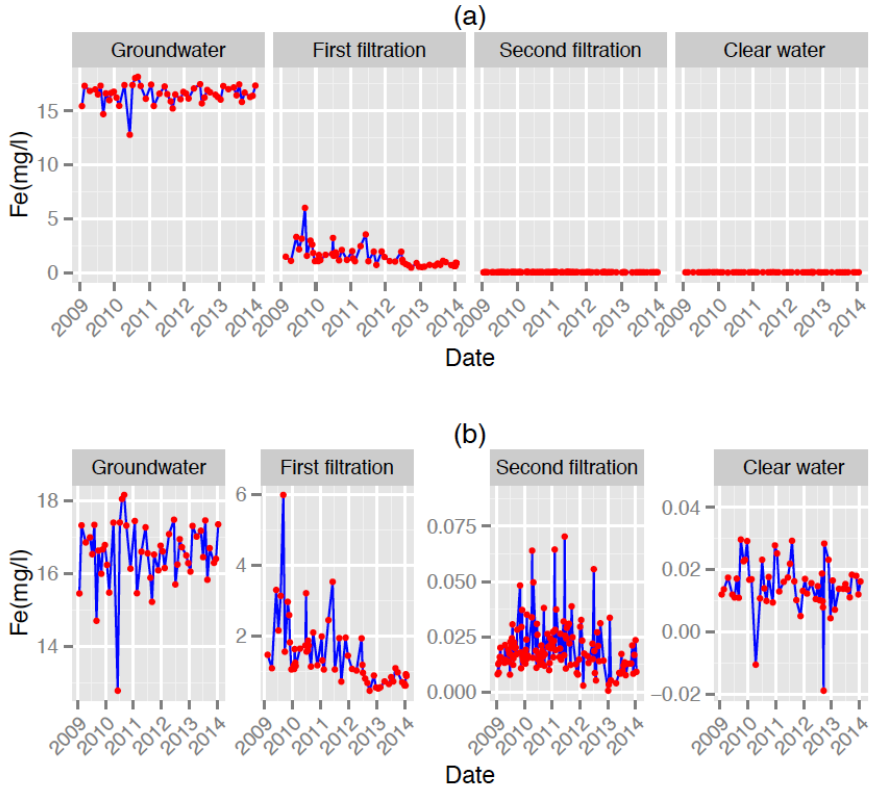


Figure 3: (a) Plots of observed iron concentrations at each location. (b) Plots with free axes scales to illustrate the fluctuations.

inspection time is expressed in hours after the last flushing. No link between these time scales is available, in the sense that for measurements y_t and z_t taken after first and second filtration respectively the time that is passed since the last flushing is not known.

3 Modelling the water purification process

3.1 A stochastic model for the first filtration process

The iron concentration in the water just before it enters the first filter is denoted by X_t and can be modeled using a time series model. However the observations in the dataset are obtained with 2-4 days in between. Therefore

the 'incoming iron concentrations' can assumed to be independent realizations of a random variable X having density function f_X. As a function of time s after flushing the filter is assumed to have a multiplicative effect on the iron concentration in the water. This means that the iron concentration in the water measured at time s after flushing is given by

$$Y_s = X \cdot \alpha(s) \tag{1}$$

where $s \in [0, 12]$ and the cleaning of the filter is assumed to happen every twelve hours. As noted before, the function $\alpha(s)$ indicates the temporal dependence of suspension in the water after the first filtration and is measured as the remaining fraction of iron in the water as a function of time s since the last filter cleaning. Based on the analysis of this *flush curve* $s \mapsto \alpha(s)$ we assume the following model:

$$\alpha(s) = a\left(1 + \frac{s}{b}\right)^{-n} + c, \;\; s \geq 0. \tag{2}$$

The function $s \mapsto \alpha(s)$ has a horizontal asymptote at $y = c$. The parameter n determines the shape (and steepness) of the curve whereas the parameter a determines the intercept at $(0, a+c)$. The parameter b on the other hand can be viewed as a scale parameter. Non-linear regression analyzes were performed to show the appropriateness of the chosen model in 2, see figure 4.

An optimal fit of the parameters $\theta = (a, b, c, n)$ is found using maximum likelihood estimation based on the measurements of the iron concentrations in the groundwater and after first filtration. This estimation is based on the following assumptions:

1. The 'incoming iron concentrations' are assumed to be independent and identically distributed according to a normal distribution $N(\mu_X, \sigma_X)$. Figure 5 indicates that the normality assumption is indeed not violated severely.

2. As noted before, the inspection time s after filtration at which the concentration Y_s is measured, is not known. For now this time is modelled by an uniformly distributed random variable on $[0, 12]$.

Assuming that the inspection times s for Y_s are drawn from a uniform distribution implies

$$Y = X \cdot \alpha(12U), \tag{3}$$

where $X \sim f_X$ and $U \sim$Unif$(0, 1)$ are independent. Based on the normality assumption it is clear that the distribution of Y will depend on the parameters $(\mu_X, \sigma_X, \theta)$.

Note that no paired observations are available, in the sense that for a particular input with concentration X, the corresponding concentration Y

is measured. Such information could improve the model and would be recommended in the future. For the moment it is assumed that a number of independent realizations of Y is observed, originating from (3). We now derive the density of Y. First recall the formula for the probability density of the product Y of two independent continuous non-negative random variables X and Z:

$$f_Y(y) = \int_{x=0}^{\infty} \frac{1}{x} f_Z\left(\frac{y}{x}\right) f_X(x)\, dx. \tag{4}$$

In order to use this relation to obtain the density of Y in (3), we need the density of the random variable

$$Z = \alpha(12U) = a\left(1 + \frac{12U}{b}\right)^{-n} + c,$$

which is given by:

$$f_Z(z) = \frac{b}{12na}\left(\frac{a}{z-c}\right)^{\frac{1}{n}+1}, \; z \in \left[c + \frac{a}{(1+12/b)^n}, c+a\right].$$

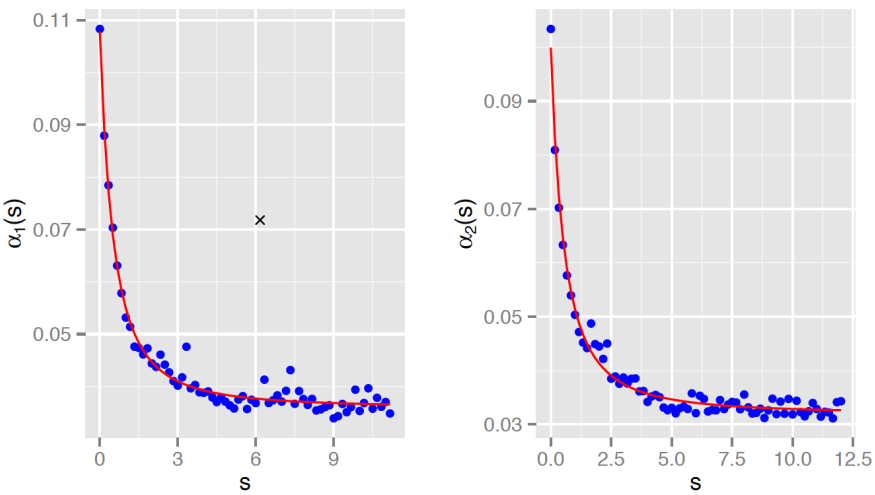


Figure 4: Flush curves of the first filtration process obtained after flushing the filter. The time s is expressed in hours after the moment of flushing. The fitted curves are respectively given by $\alpha_1(s) = 0.04 + \frac{0.07}{(1+s)^{1.92}}$ and $\alpha_2(s) = 0.03 + \frac{0.07}{(1+s)^{1.80}}$, where $b = 1$. The cross in the left plot is considered to be an outlier and is not incorporated in the regression analysis.

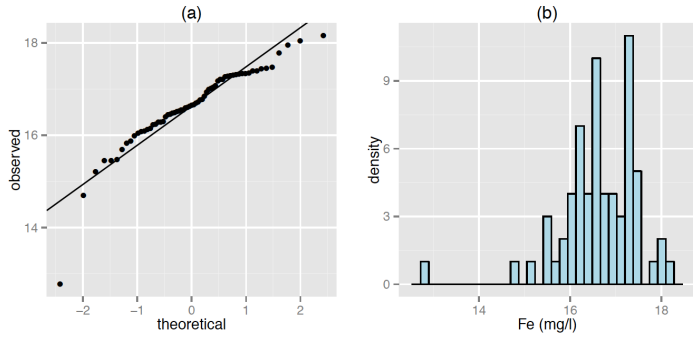


Figure 5: A quantile-quantile plot of the iron concentrations in groundwater with respect to the normal distribution (left) and a histogram of the iron concentrations (right).

Combined with (4), this leads to the following two-parameter model for the observed concentrations $Y_1, \ldots, Y_N$ of iron after the first filter:

$$f_Y(y; \mu_X, \sigma_X, \theta) = \frac{ba^{1/n}}{12n} \int_{x_\ell}^{x_r} x^{1/n} f_X(x, \mu_X, \sigma_X) \, (y - cx)^{-1-1/n} \, dx$$

where we make the dependence on the parameters $(\mu_X, \sigma_X, \theta)$ explicit and where the integration bounds are given by

$$x_\ell = x_\ell(y, \theta) = \frac{y}{c + a} \quad \text{and} \quad x_r = x_r(y, \theta) = \frac{y}{c + a(1 + 12/b)^{-n}}.$$

Given the observed values $y_1, \ldots, y_N$ of Y, the log likelihood function is given by

$$\ell(\mu_X, \sigma_X, \theta) = \sum_{i=1}^{N} \log f_Y(y_i; \mu_X, \sigma_X, \theta). \tag{5}$$

This function can be maximised numerically using the so-called limited memory BFGS algorithm (Byrd et al., 1995). This optimization algorithm allows constraints on the parameters and is implemented in R under the 'stats' package.

3.2 A stochastic model for the second filtration process

We feel the second filtration step can be dealt with in a similar fashion as the first filtration. Due to time limitations, this step has not been worked out yet. During the simulation, we simply divide the measurements after the second

filtration step by a fixed number (chosen to match the data on average). Of course, this is a big simplification that needs to be handled more accurately in future work.

3.3 Modeling the concentration of iron in the clear water basin

To examine the propagation of the compounds in the water when the water is transferred between different basins, we developed an ordinary differential equation model. This model describes how the amount of a contaminant in the water, in the case under consideration this is iron, changes in the different basins. To illustrate the method, we derive the equations for the amount of iron in the clear water basin, but a similar procedure could be followed for the other basins as well.

We assume that the volume of the clear water basin $V(t)$ changes in time according to the following equation:

$$\frac{d}{dt}V(t) = c_0(t) - f(t), \tag{6}$$

where $c_0(t)$ denotes the volume influx to the clear water basin and $f(t)$ is the water outward flux that typically consists of a constant part and a fluctuating part, as the water demand is a fluctuating quantity. For example, during day time more water is used than in the night:

$$f(t) = 60(83 + 5\sin(2\pi t/24)).$$

The numbers in this expression designate that 83 l/s is the typical rate at which water leaves the clear water reservoir. Furthermore the value of the amplitude of the sine function is chosen to correspond to the size of the fluctuations of iron concentrations, which we will see in the next section.

Next, we model the amount of iron, denoted by $u(t)$ in the clear water reservoir. The governing equation for $u(t)$ is

$$\frac{d}{dt}u(t) = c_0(t)g_{in}(t) - u(t)f(t)/V(t). \tag{7}$$

Equation (7) describes the influx of iron with rate $c_0(t)g_{in}(t)$ and outward flux with rate $u(t)f(t)/V(t)$. The function $g_{in}(t)$ which models the concentration of iron after the second filter is not known as a function of time. However from measurements that were performed at random times a typical size of the fluctuations can be inferred. Equations (6) and (7) can be solved in time using a simple Euler forward method which works as the system considered consists of linear equations (Xie, 2010).

3.4 Assessing the fit of the model

To assess the fit of the model a simulation was performed. Results of this simulation can then be compared with the given observations of the clear water.

We propose to model the iron concentration in the groundwater by a stationary autoregressive time series model of order 1:

$$X_t = \mu_X + r(X_{t-1} - \mu_X) + \varepsilon_t, \tag{8}$$

where $\{\varepsilon_t\}_t$ is a sequence of independent $N(0, \sigma_\varepsilon^2)$ distributed random variable. Assuming $|r| < 1$ ensures that the (causal) stationary distribution of (8) exists uniquely and is normally distributed. As no data at this time are available to fit this model, we somewhat arbitrarily chose $r = 0.8$. This choice implies positive dependence of iron concentrations in the groundwater over different lags. We simulated 10^6 points with a time step of $dt = 0.5$. This corresponds to a simulation of 5.7 years of data which approximates the period over which the observed data of WLN is taken. Using our model one is able to predict the outcome when this data is considered to be the iron concentration of the groundwater.

The corresponding concentrations after first filtration are obtained by applying formula (1) using ratios $\alpha(s)$ evaluated at 10^6 equidistant times with a time leap of dt starting from $s = 0$. The concentrations after the second filtration were obtained in a similar way applying the same multiplicative procedure on the iron concentration after the first filtration. This resulted in a discrete time series consisting of 10^6 points that can be used as input for $g_{in}(t)$ of the differential equation in (7). The initial conditions to solve the differential equations were chosen to correspond to the stationary state values of $u(t)$ and $V(t)$:

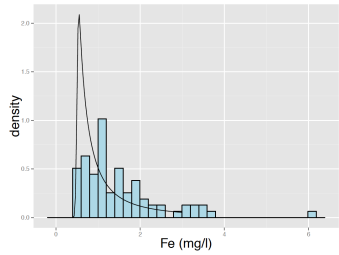
$$\begin{cases} u(0) &= 0.015\, V(0) \text{ mg} \\ V(0) &= 7500000 \text{ l} \end{cases} \tag{9}$$

Finally a number of 70 points are randomly chosen from the simulated 10^6 points and compared with the observations of WLN.

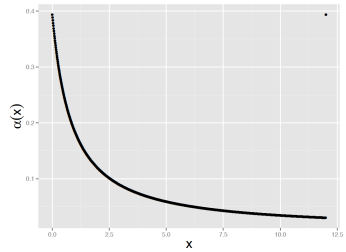
4 Results

In the non-linear regression analysis of the flush curves, the parameter b in (2) could be set to 1 while keeping a satisfactory curve fitting. The parameters $(\mu_X, \sigma_X, a, c, n)$ were estimated by maximizing the log likelihood function (5):

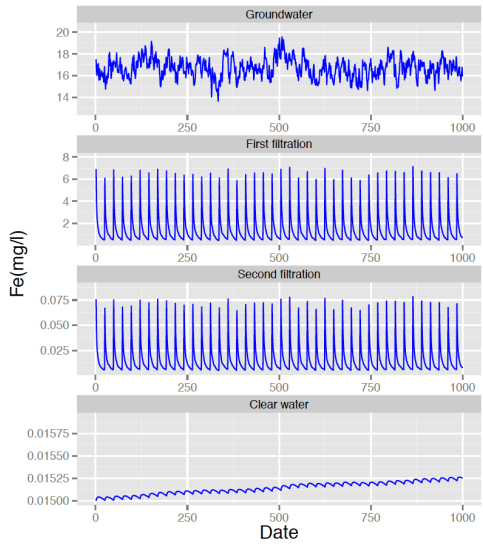
$$\hat{\mu}_X = 16.62, \quad \hat{\sigma}_X = 0.88, \quad \hat{a} = 0.38, \quad \hat{c} = 0.01, \quad \hat{n} = 1.1496307$$



(a) The fitted distribution f_Y on the observed data of Y.



(b) The flush curve after first filtration obtained using our MLE approach.



(c) Plots of simulated time series data in each basin for the first 1000 points simulated.

Figure 6(a) shows the fit of the density distribution f_Y to the observed iron concentrations after the first filtration. A slight underestimation of the variance is noted. Figure 6(b) shows the flush curve that is obtained using the estimated parameters $(\hat{a}, \hat{c}, \hat{n})$.

Figure 6(c) shows the simulated concentrations of iron in each basin using the time series data generated from (8). The x-axis shows the indices of the simulated data corresponding to some moment in time where the origin $t = 0$ has to be interpreted as the first inspection time available in the dataset of WLN.

As one can see the ranges of the simulated data in the first three basins of our model approximate those of the observed data in figure 3. Furthermore the decrease of the iron concentrations after first and second filtration is clear. The clear water contains an iron concentration of 0.015 at $t = 0$ as induced by (9). The figure shows an initial increase of the iron concentrations.

Figure 6 shows the complete simulated time series of iron concentrations in the clear water basin. After some time period the iron concentrations fluctuate around a stationary value of approximately 0.02. This corresponds with the observed data in the clear water basin shown in figure 3(b). This is, of course, in accordance with expectations as the large volume of the reservoir

averages out most of the flucatuations.

The observations in figure 3 are sampled at irregular moments in time and with an irregular frequency, typically given by 1-2 times a week at different weekdays. Because of the flush activities the sampled observations reach a peak when taken at a moment short after flushing. This happens typically when a sample is taken within an hour after flushing, see figure 6(b). In a later time period, when the last flush occured more than 3 hours ago, more steady concentrations are observed. Hence the observed fluctuations depicted in figure 3 are inherent to the sampling method.

This sampling method can be mimicked by randomly choosing a number of N observations from the time series obtained for the clear water basin (figure 7). To make a consistent comparison with the observed data, N is chosen to equal the number of observations available from the clear water basin, i.e. $N = 70$. In this way we were able to simulate fluctuations in the iron concentrations that are approximately of the same magnitude as the measured fluctuations.

5 Discussion

In this section we first summarize the conclusions that we may draw from our results. We then present recommendations to WLN for further actions to address the problem at hand. Finally, we suggest some further lines of research.

5.1 Conclusions

At this stage of research, the model as presented in section 3 may be too simple to realistically model the water purification process. This is partly due to

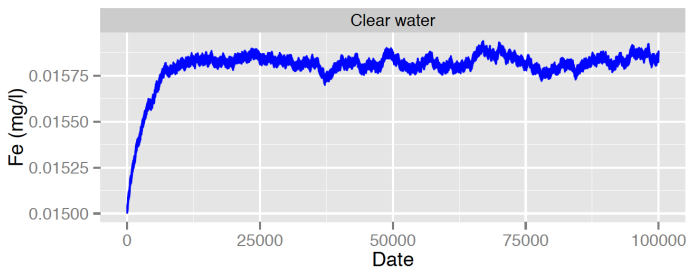


Figure 6: (a) Complete simulated time series of iron concentrations in the clear water basin.

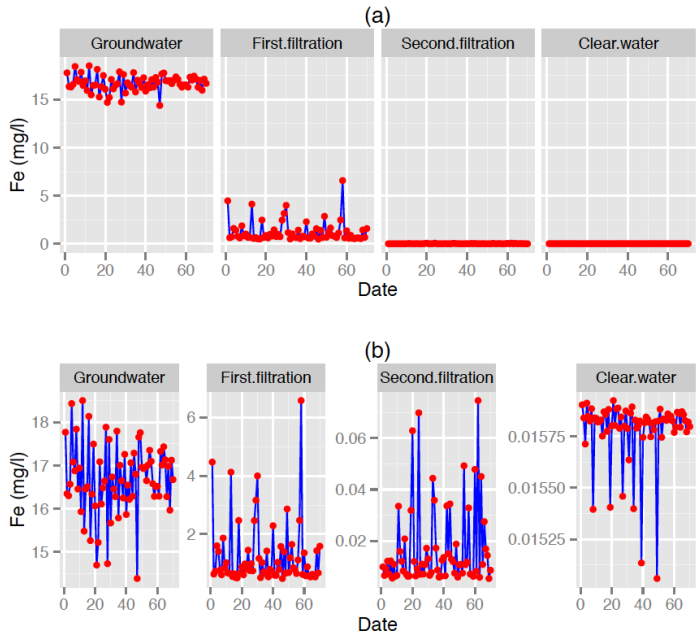


Figure 7: (a) Plots of simulated observed data in each basin. (b) Plots with free axes scales to illustrate the fluctuations.

time constraints for building the model, but also due to the nature of the available data right now. Presently, the data provided by WLN only allows for a rough calibration of the model parameters. More accurate estimation may be accomplished in the future by setting up a specific measurement scheme with the purpose of model calibration in mind. Once such measurements have been obtained and the model is sufficiently refined (i.e. realistically modeling the characteristics of the water purification process), process inherent variation can be estimated. The latter directly gives information on the required monitoring frequency: wildly fluctuating iron concentrations will require a relative high monitoring frequency, whereas a process with low variability may allow for less measurements taken over time.

To develop an effective and efficient monitoring procedure it is necessarily to properly define the process to be monitored and the desired performance of the monitoring procedure. These descriptions come from operational and legal constraints, but must be translated into statistical descriptions in order to be able to provide a sound basis for the assessment of a monitoring procedure.

A proper definition of the monitored process includes a description of

the available data, a list of variables to be monitored, a description of an acceptable or "in-control" situation and the sampling strategy (Hawkins and Olwell, 1998; Kenett and Zacks, 1998). In this application the variables that are monitored are the concentrations of ions (e.g. iron). In this case an "in-control" situation for each variable is defined by so-called control limits. These control limits are driven by the natural variability of the concentration in the drinking water. For instance, in the case at hand one would desire that the iron concentration in the clear water remains at a constant (probably low) level while fluctuations are limited.

However it seems that the current approach performed by WLN is based on the monitoring of univariate variables evaluated relative to specification limits. These specification limits describe the maximal allowable deviation from a desired value of the variable, called the target value. In contrast to the control limits these specifications are determined externally and are not related to the natural variability of the variables. Therefore these specifications mostly allow a variability that is larger than the variability naturally induced by an in-control system. For instance a malfunctioning filter or a slow but persistent increasing value of a certain concentration does not have to lead to exceedances of the specification limits. Hence information about inherent changes of the water quality and the filtration processes can be lost using this approach.

The importantace of the distinction between specification and control limits is widely known in the literature of statistical process control (Montgomery, 2013). To illustrate this difference further we can say that a purification process performs within specifications when the water quality is acceptable from a public health point of view. However at the same time the water company may suffer from excessive costs due to an out-of-control process caused by some malfunctioning. When one is aware of such malfunctioning preventive maintenance actions or increasing surveillance could be performed without interrupting purification. In this way unnecessarily costs can be avoided.

Furthermore, the performance of the monitoring procedure is mainly determined by the time that is needed to detect a malfunctioning or contamination. Using proper inherent control limits of a process that is in a steady state one can objectively determine appropriate sampling frequencies that assure the desired performance (Montgomery, 2013). However, due to flushing in the normal state after the first filtration, the process is not in a steady state. On the contrary it rather shows cyclic behavour. Therefore, the effect of flushing for a single location was modeled with a simple stochastic model obtaining a reasonable agreement with actual data from iron concentrations. Monitoring using this model is then in fact monitoring a profile (Noorossana et al., 2012). In order to connect flows from one filter to another, we set up a system of coupled differential equations obtained from simple conservation laws.

Finally, remark that one should also take into account that the lab needs time to perform analyses. Thus demanding detection of abnormal levels of certain ion concentrations within e.g. 2 hours is not feasible when the water quality analysis in the lab takes 1 day. Another important aspect is to determine optimal measurement locations in the purification plant. It may seem optimal to monitor as much as possible upstream in the purification plant to assure on-time detection. However, in this way one may fail to detect anomalies downstream. Thus, it is sensible to monitor at several locations.

To summarize, the main conclusions are:

(i) monitoring is now performed by checking individual specification limits, and thus fails to take into account deviations from normal process deviations

(ii) the cyclic concentration levels due to flushing may be modelled adequately by a simple model so that we have a realistic benchmark for detecting anomalies

(iii) the flows from one filter to another may be modeled by a system of coupled differential equations obtained from simple conservation laws

5.2 Recommendations

We start with some recommendations on monitoring in general.

1. Change the monitoring procedure from checking specification limits to checking inherent concentration fluctuations in order to obtain a more responsive monitoring system (and thus have a well-grounded justification for the associated sampling frequency). These statistical limits will be more strict than the chemical/health limits, thus there is no risk for exceeding chemical/health limits.

2. Specify detection performance for all concentrations, taking into account processing time of analytical analyses in the lab and out-of-control scenarios like trends (see e.g., Frisén (2003) for a discussion of different performance metrics).

3. Study correlations between concentrations of different ions so that one may use the available data more efficiently.

4. Synchronize timing of measurements between compartments in a purification plant in order to track the flow water drops and thus improve detection performance.

5. Reduce variance in the beginning by blending the right wells.

6. Perform a pilot study with in-line measurements on all measurement locations in one purification plant in order to improve the models of this paper.

The following recommendations concern modeling the effects of flushing.

1. Measure the incoming iron concentration together with corresponding (time-aligned) iron concentration after flushing ; this allows for much more accurate modelling.

2. Keep track of the actual mixture of incoming water sources since this has an impact on the distribution of the incoming iron concentrations.

5.3 Future Research

The study in this paper was restricted to the monitoring of iron concentrations due to time limitations. In future research other parameters could be studied as well. Furthermore, the model has to be completed to describe the second filtration process. As a next step simulations could be performed where anomalies are artificially implemented. Such simulations would be useful in finding an optimal monitoring frequency that enables us to detect a small (to be determined) shift (with a certain probability). Finally, it would be interesting to adapt our model to include correlations between different ions. In this way an alarm system can be build to detect dangers of combined high levels.

Acknowledgements

The authors would like to express their appreciation to Peter van der Maas from WLN who has introduced the problem treated in this paper that was fascinating and challenging to work on. Also a warm gratitude goes out to the organising committee of the Study Group Mathematics with Industry held at TU Delft to give us the possibility to work on this problem in a collaborative atmosphere.

References

Richard H. Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995.

M. Frisén. Statistical surveillance. Optimality and methods. *International Statistical Review*, 71(2):403–434, 2003.

D.M. Hawkins and D.H. Olwell. *Cumulative Sum Charts and Charting for Quality Improvement*. Springer, New York, 1998.

P. Karanis, C. Kourenti, and H. Smith. Waterborne transmission of protozoan parasites: A worldwide review of outbreaks and lessons learnt. *Journal of Water and Health*, 5(1):1–38, 2007.

R.S. Kenett and S. Zacks. *Modern Industrial Statistics*. Duxbury Press, 1998.

D.C. Montgomery. *Statistical Quality Control*. John Wiley & Sons, 2013.

R. Noorossana, A. Saghaei, and A. Amiri. *Statistical Analysis of Profile Monitoring*, volume 865. Wiley, New York, 2012.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2012. ISBN 3-900051-07-0.

Wei-Chau Xie. *Differential Equations for Engineers*. Cambridge University Press, 2010.

, , , ,

Monitoring the sewer system

Arnold Heemink[1],
Corine Meerman[2],
Sanjay Ramawadh[3],
Vivi Rottschäfer[2],
and Willem van Zuijlen[2]*

[1] TU Delft
[2] Leiden University
[3] Utrecht University

Abstract

The old city centre of Delft is sensitive to flooding caused by rainfall. We discuss the possibility and methods of combining data from observations, measurements and a theoretical model to determine the frequency and locations of these flooding events. We describe a protocol how the observations should be collected and present the determination of the measurement sensor positions as an optimisation problem. We also show some ideas about improving the presently used theoretical model.

Keywords: Sewer system, modeling, monitoring, optimisation, knapsack problem, de st. Venant equations

Acknowledgements: The authors are grateful to Hans Korving (Witteveen en Bos) and Hai Xiang Lin (TU Delft) for discussions.

*Corresponding author: willem.van.zuijlen@math.leidenuniv.nl

1 Introduction

At the 98th Study Group Mathematics with Industry (SWI), held at Delft University of Technology from 27 – 31 January 2014 one of the questions was formulated by the company 'Witteveen en Bos'. Witteveen en Bos (from now on W+B) is a consultancy and engineering firm for water, infrastructure and environment. The problem they posed considers the sewer system of the city center of Delft. The old city centre is sensitive to flooding caused by rainfall. These events can cause sewage to flow from the sewer system back onto the streets and as a result the water on the streets will contain diluted (faecal) sewage [Man et al. (2014)]. Hence, the flooding does not only cause damage and leads to dangerous situations, with streets and buildings flooded, but also poses health risks due to exposure to contaminated water.

Inside the sewer, data is collected by using sensors that measure the water level. W+B already gathers data by using such sensors in the sewers in a district of the city Utrecht: Tuindorp. Their intention is to do the same in the city center of Delft. However, sensors are only able to measure the water level inside the sewer up to 30 cm below the surface level. When the water exceeds that level, there is no further information available and it is possible that water flows out of the sewer. In that case the sensors do not provide a accurate picture. Apart from obtaining data with sensors, W+B currently use a theoretical model to simulate the water levels inside the sewer. As in the case of the sensors, the theoretical model cannot predict the water level above surface. For these reasons, W+B desires to combine the known data below surface with data gathered above surface. W+B is interested in how this data above the surface should be collected and how it can be combined and compared with the data collected below the surface in the sewers. For this second question, our main objective is to detect at which locations and how often flooding occurs since this causes health threads. This will result in a complete picture of the sewer system. Another aim is to locate possible obstructions in the sewer system.

An obstruction in the sewer, such as a root intrusion or deposits, affects the sewer performance. These obstructions can be found using a camera survey. However, these surveys cannot be done regularly, since they are time consuming and expensive. The obstructions are not present in the theoretical model but they can be put in once their presence and position is known. This should be done on a regular basis since the occurrence of the obstructions changes over time. The data from the sensors gives more insight into the position of the obstructions and combining the sensor-data with data from above the surface will be very valuable.

Data that could be collected above the surface is for example photos of water (puddles) on the streets. At the moment, flooding incidents that are

reported by citizens are already stored in a data-base. Increasing this number of reported incidents would improve the reliability of this data. In addition, we consider the possibility of making and collecting photos. One option that W+B is looking into is to use school children for this. Via these children, their parents and other citizens can also be motivated to gather data.

In Section 2, we present ideas and a strategy for collecting data above the surface. Considering where and when data has to be collected, we provide a protocol for the schools to follow in order to obtain sensible data. In Section 3 we describe a strategy for placing the sensors in such a way that the most important information is collected. Moreover, we optimise this placement by combining data from below and above the surface. In Section 4 we describe the theoretical model that is currently used to simulate water levels and give ideas for improving this model. In Section 5, our recommendations are summarized.

2 Strategies for obtaining data above surface

In collecting data above the surface, we have developed several methods which can be combined to get an accurate picture. Where possible, we would like the citizens to collect data. One route to increase the public awareness is via children and their schools. Previously, W+B had another program running at primary schools of Delft. Driven by this positive experience they are cooperating with primary schools to set up a program that is not only educative, but can also provide useful data about the sewer. We will discuss a way to implement such a program. Since primary schools have a regular schedule, this program might be missing the most important events, namely the heavy rain events. Therefore, we suggest to combine this with another program that can be used for just these events.

For both approaches we take into account a few aspects:

• Amount of time: How much time is available to gather and to evaluate the data?

• Place: At which places to gather data and on how many places?

• Information of a photo: What kind of information do we want a photo to contain?

After discussing these schemes we will give some suggestions of an app that aims to increase the amount of data collected voluntarily by citizens.

2.1 A program at primary schools.

Primary schools have a weekly timetable. For that reason, it is most likely that the sewer-program will included at a fixed time in the week. The idea

is that the children spread over the city and take photos of puddles of water. However, it does not make sense to let them take photos if it has not rained lately. Therefore, we suggest to develop an alternative program as part of the sewer-program in case there is no water on the streets.

In case it has been raining lately, the photos should be taken at places randomly distributed over the city center. Since time and children are limited we do not expect to cover all of the city in one day. Therefore, there are multiple ways of choosing places, or, more specifically streets. Practical details, such as number and location of the schools, will point out whether it is possible to distribute the streets of interest uniformly or to have a circulating system over the schools such that different parts of the city at different moments are covered during the week.

Once the children have arrived at the (predetermined) streets they should take pictures of puddles of water on the street, and assign a measure of size to the puddle. In order to reduce the amount of data generated in this program – someone has to evaluate the pictures – a photo should only then be taken if there is water on the street. Even when puddles are not close to a manhole, the photos still contain useful information for the local government. To be able to link the photo to a particular manhole (or position), GPS information must also be included in the data. This program is summarised in the flow chart shown in Figure 1.

2.2 The approach in case of heavy rain events.

The method described above provides useful information on a rather regular base. However, the rain events that are very important, those of heavy rain or storms, might not be covered. These storm events could give the most valuable information, since flooding is most likely to occur during such an event. Therefore, the above school program should be complemented by another method for collecting information from these events. For this, one or more persons should be readily available.

Weather predictions can be used to determine the moments at which data should be collected. Data has to be gathered from the moment it starts raining until the moment that the water has disappeared from the streets.

We select streets where data should be collected. We select the streets with the use of the sensor data, for example, by choosing streets where the sensors indicated that the water level gets to a high level. Furthermore the places where an incident was reported will be visited. Then if there is a puddle, the person will take a photo and/or assigns a measure of size to the puddle and repeatedly come back until the water has disappeared.

Similar to the previous approach, GPS coordinates must be included. Moreover, in case an estimate of the amount of water on the street can be

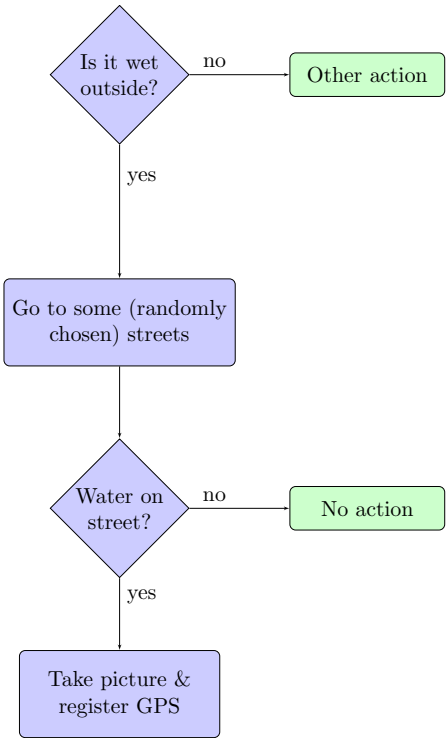


Figure 1: Flowchart of the program for school children.

given, this information should be added. The outline for this program is summarised in the flow-chart shown in Figure 2.

2.3 Suggestions for an app/website

At the moment, people can contact the local government by phone for incidents. We suggest that, apart from this, people should be able to report incidents via other media like internet. Incidents reported by phone-calls are already useful in order to detect possible defects or obstructions in the sewer system. However, information of phone-calls is rather subjective. More objective information can be obtained via a website or app that enables the persons to add extra information to the street name where the incident is mentioned, like pictures with GPS data and other observations. In such a way, one can also provide a guideline of how the person should include the data. By presenting such a guideline and an easy way to report incidents (without having to make a phone-call), we expect the number of reports to increase and to

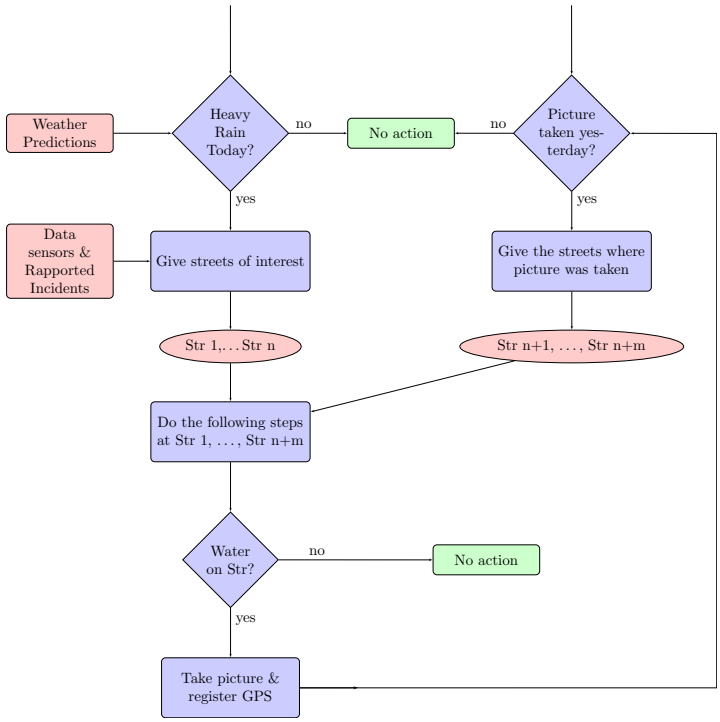


Figure 2: Flowchart of the program for heavy rain events.

be of better use. For this approach, the citizens should be made aware that their reports are very valuable and that they can help to reduce flooding in the future.

3 Placing of the sensors

In order to monitor the behavior of the sewer system below the surface efficiently, we want to place sensors at those manholes which are most likely to overflow. These sensors measure the level of the water in the sewer once every minute. Moreover, every sensor has a threshold up to 30 below the surface level. This threshold is known by W+B. The sensor data is not only useful for showing when and where a manhole has flooded, it can also be used to find obstructions in the sewer system. The method for finding these obstructions based on sensor data is also known by W+B [Bijnen et al. (2012)].

Next to the measurements with sensors, data is also available from reported incidents and photographs by citizens. We want to combine these

data with the sensor data to determine those manholes which are most likely to be flooded; the problem spots.

A maximization problem is set up to find the optimal placement of the sensors. The sewer network can be seen as a graph $G = (V, E)$, where the set of vertices V corresponds to the manholes and the set of the edges E corresponds to the pipelines. Let us define $n := |V|$. The main assumption will be that the sewer network contains no obstructions. We start with a clean network which, besides containing no obstructions, also has no sensors placed yet. W+B has a theoretical model of the sewer network, which can accurately simulate the level of the water below the surface in a clean network. Using this theoretical model, a well-educated guess is made for the initial placement of the sensors. Several simulations are done with the model, using different values of rainfall as an input. A sensor will be placed at those vertices where flooding is most likely to happen. If the well-educated guess requires placing less sensors than we have, we can place the remaining sensors at random locations.

3.1 Setting up the problem

The idea is to determine a placing of the sensors such that only the measurements of the sensors can give us an accurate view about the places of the incidents. We expect that the placement at the start is not suitable enough for this, so we want to replace certain sensors. We assume that the sensors are working properly, since malfunctioning sensors can be detected through data validation. If a placed sensor indicates that the threshold has been reached and at the same time there is observed data about an incident, then the sensor should not be removed. If the observed data shows no incident, then it does not matter whether the sensor is removed or not. If a vertex does not have a sensor and observed data shows an incident, a sensor should be placed there. If the observed data does not show any incidents, it does not matter whether the sensor is placed or not.

Each vertex will be given a value, which depends on whether or not the corresponding manhole has a sensor and also on the data available about this manhole. The value of a vertex indicates how likely it is that the corresponding manhole overflows; the higher the value, the more likely it is to happen. The value of a vertex will be denoted by $\alpha_{t,l}$, where t stands for the day of the measurement and l represents the corresponding manhole. The value is determined as follows. First, assume that the corresponding manhole l has a sensor. Then, for each day t, the value $\alpha_{t,l}$ equals 1 if the sensor reaches the threshold on day t as well as there is an incident reported on day t for this manhole. Otherwise, the value is 0. Now, assume that the corresponding manhole l has no sensor. Then, for each day t, the value $\alpha_{t,l}$ equals 1 if

either there is an incident reported on day t for this manhole, or if there is an incident reported for some other manhole nearby in the network and at the same time a simulation of the rainfall in the theoretical model indicates that this manhole has flooded as well.

We assign, to each vertex l, a variable x_l which can only attain the values 0 and 1. The expression $x_l = 1$ means that we want to place a sensor at the manhole corresponding to vertex l. The expression $x_l = 0$ means that we do not want to place a sensor there. The variables x_l are subject to some constraints which arise from practical considerations. For example, sensors should be maintained regularly, which comes with a cost. Moreover, these maintenance costs might be different depending on the location of the sensor. To include this constraint, we define, for each vertex l, the variable w_l which is the average maintenance cost per day (or any other timespan) for a sensor at the manhole corresponding to vertex l. The sum $\sum_l w_l x_l$ then equals the average maintenance cost per day to place or remove sensors, which is likely required to be smaller than some predetermined constant W, the daily budget. Another constraint is given by the limited number of sensors. Since $\sum_l x_l$ equals the number of sensors we want to place, we require this sum to be smaller than some other predetermined constant C, the maximal amount of sensors we can place.

In order to determine the problem spots, that is, the vertices corresponding to those manholes which are most likely to overflow, we use the data gathered over a certain time frame, for example a year. For each vertex l, the sum $\sum_t \alpha_{t,l}$ is the total value of the vertex l in this timeframe. The quantity $\sum_{l \in V} \sum_t \alpha_{t,l} x_l$ now represents the total value of the vertices where a sensor should be placed. Since we want to place the sensors as efficient as possible, this total value needs to be as high as possible. The maximisation problem can now be written as:

$$\max \left\{ \sum_{l \in V} \sum_t \alpha_{t,l} x_l \;\middle|\; \begin{array}{l} x_l \in \{0,1\},\ l = 1, \cdots, n \\ \sum_{l \in V} w_l x_l \leq W \\ \sum_{l \in V} x_l \leq C \end{array} \right\}. \tag{1}$$

3.2 Solving the problem

The optimisation problem stated in expression (1) is equivalent to the 0/1 knapsack problem, which is a known problem in combinatorial optimisation. The 0/1 knapsack problem with one constraint is formulated as follows. Suppose we have n objects, each with a value v_l and a weight w_l. The objective is to choose objects in such a way that the total weight of the chosen objects does not exceed some predetermined constant W and, moreover, that there is no other selection of objects satisfying the same weight constraint which has a higher total value. The difference between the general knapsack problem and

the 0/1 knapsack problem is that, in contrast to the general knapsack problem, each object can be chosen either once or not at all. The 0/1 knapsack problem with one constraint is mathematically formulated as follows:

$$\max \left\{ \sum_{l=1}^{n} v_l x_l \;\middle|\; \begin{array}{c} x_l \in \{0,1\},\ l = 1, \cdots, n \\ \sum_{l=1}^{n} w_l x_l \leq W \end{array} \right\}. \tag{2}$$

The general 0/1 knapsack problem, that is, the 0/1 knapsack problem with any number of constraints, is an optimisation problem that is known to be NP-complete [Garey and Johnson]. All NP-complete problems are characterised by the fact that there is no known algorithm that solves these problems quickly. However, there are several heuristic methods known which find a solution close enough to the optimal solution. In the case of the general 0/1 knapsack problem, the most commonly used method is dynamic programming [Salkin and Kluyver (1975)]. In the specific case where the 0/1 knapsack problem is of the form in equation (2), with all weights equal and, without loss of generality, all equal to 1, the problem belongs to class P and can be solved exactly. The solution is $x_l = 1$ for the $\lfloor W \rfloor$ objects with the highest values, with $\lfloor W \rfloor$ the largest integer smaller than W. So the maximisation problem in (1) coincides with this specific case if the average maintenance costs per day for all sensors are equal, as the two constraints can be reduced to one.

4 Mathematical modelling of sewer networks

In this section we give a brief summary of a model that describes the dynamic behaviour of the water level in a sewer system. For a detailed description the reader is referred to [Cunge et al.]. We also discuss the possibility to use the data that will be collected above the surface and by the sensors in the sewer system to improve this model. Furthermore we will suggest a method to identify locations where obstructions might be located.

A model for open channel flow (developed originally for river systems) is based on the so-called "de st. Venant equations", also referred to as the "1D shallow water equations". This model can also be used for sewer systems. The equations are respectively derived from conservation of momentum and conservation of mass:

$$\begin{aligned} &\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + g\frac{\partial h}{\partial x} + gS_l - \mu S_f = 0, \\ &\frac{\partial h}{\partial t} + \frac{1}{b}\frac{\partial (uA)}{\partial x} = S, \end{aligned}$$

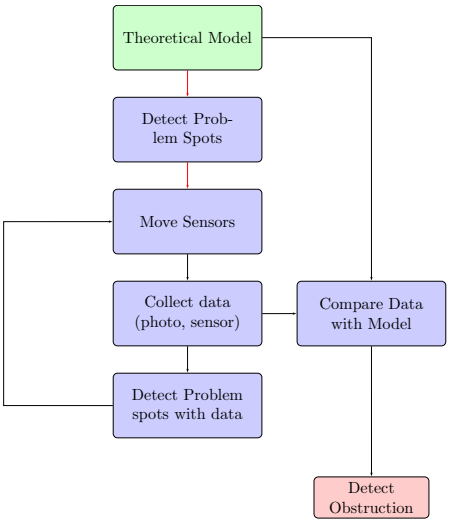


Figure 3: Flowchart summarising the recommendations about moving the sensors to a more optimal position and also about finding possible obstructions in the sewer network.

where x is the spatial coordinate (in one dimension along the sewer pipe), t is the time and

$$\begin{aligned} g &= \text{acceleration of gravity}, \\ u(x,t) &= \text{water velocity}, \\ h(x,t) &= \text{water level}, \\ S_l(x) &= \text{slope of the channel}, \\ S_f(x,t) &= \text{friction term}, \\ \mu(x) &= \text{friction coefficient}, \\ b(x,h) &= \text{width of the channel}, \\ A(x,h) &= \text{cross sectional area}, \\ S(x,t) &= \text{water source or sink}. \end{aligned}$$

The sewer system consists of pipes that are connected at junctions. The velocity and level of the water in each of these pipes are described by the above model. At the network junctions the amount of water that flows in has to be equal to the amount of water that flows out. In other words the total inflow is equal to the total outflow at a junction. This leads to one boundary condition for the partial differential equations described above. Since we still need one other boundary condition at the junction we could in addition assume

that the water level is continuous at the junction. This boundary condition does however not take into account the complicated flow pattern that may occur at a junction in case of a sharp curve. Therefore one often assumes continuity of the quantity

$$h + \alpha \frac{u^2}{2g}.$$

Here the second term is introduced to account for local effects at the junction due to the bulk motion of the fluid. This effect creates small differences of the water levels in the various pipes at the junction: The higher the velocity at the entrance of the pipe, the lower the water level at this location will be. The empirical factor $0 \leq \alpha \leq 1$ is a tuning parameter. In order to obtain a numerical model for the sewer network, the well-known Preissmann scheme [Cunge et al.] can be used. This scheme is very attractive for networks since it can deal easily with non-equidistant grids and is also capable to include the boundary conditions just described.

In the model S is the water source or sink and it represents the amount of water that enters or leaves the sewer network at the pipe (when S is negative, water flows out of the sewer network). If S is 0, then the total amount of water in the system is preserved. One of the difficulties in using this model is that there is hardly any information available about the water source or sink S. In the sewer system the water level can exceed the level of the street and then water will flow out of the system. But generally the amount of water that flows out of the sewer system is not known. Hence a good estimate of S is desirable. We suggest to explore the possibility to use the information that is gathered on street level (like pictures) together with the model simulations to estimate the parameter $S(x, t)$ in various sections of the sewer network model. This calibration procedure can be formulated as an optimisation problem by defining a cost function that measures for a given parameter the difference between the model results and the available measurements. Here usually the least squares criterion is taken as cost function. One value of this cost function for a given parameter (for S) can be computed by running the model with this parameter value and by comparing the results of the model at the sensor locations with the measurements. Using an optimisation scheme the parameter can be improved step by step and finally the best estimate of the parameter is the parameter for which the cost function is smallest and therefore for which the model simulation is as close as possible to the measurements available.

As we indicated before, water level measurements with sensors combined with information from photos can also indicate possible locations for obstructions. The available data can be combined with the model simulations in order to estimate the friction coefficient $\mu(x)$ at various locations in the network by means of the calibration procedure just described. If the estimated

value for the friction coefficient is significantly larger than the original value in the model at a certain location in the network, this is an indication that there might be an obstruction.

5 Recommendations

Here, we give a summary of the recommendations regarding collecting data, moving sensors and adjusting the theoretical model.

To collect data above street level we suggest to implement a program for the primary schools in Delft to collect photos of puddles of water in the streets (see also Figure 1). Apart from that, we suggest to use a complementary program to collect data in case of heavy rain events, described by the flowchart in Figure 2. Additionally to the two programs, we suggest to develop an easy app and/or website on which citizen can report their observations of a flooding incident.

We suggest to use the theoretical model as the basis for the initial placement of the sensors. This is done by simulating many different rainfall events and by choosing problem spots which are most likely to overflow. Once the sensors are placed, data can be collected from above the surface as well as from below the surface. After a certain, not necessarily predetermined, time frame, we use all the gathered data to optimize the positions of the sensors according to the reduced knapsack problem. This procedure can be repeated at any time. The flowchart in Figure 3 summarizes our recommendations.

For the theoretical model we suggest that the amount of escaping water could be estimated using the data from above the surface. Furthermore via an estimate of the friction coefficient that fits the measured data, one might be able to locate the obstructions by comparing this estimated friction coefficient to the normal friction coefficient.

References

M. van Bijnen, H. Korving, and F. Clemens. Impact of sewer condition on urban flooding: a comparison between simulated and measured system behaviour. *Proceedings of the Ninth International Conference on Urban Drainage Modelling*, 2012.

J. A. Cunge, F.M. Holly, and A. Verwey. *Practical aspects of computational river hydraulics.* Pitman Advanced Pub., Boston.

M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman and co., New York.

H. de Man, H.H.J.L. van den Berg, E.J.T.M. Leenen, J.F. Schijven, F.M. Schets, J.C. van der Vliet, Knapen F. van, and Roda Husman A.M. de. Quantitative assessment of infection risk from exposure to waterborne pathogens in urban floodwater. *Water Research*, 48:90–99, 2014.

H.M. Salkin and C.A. de Kluyver. The knapsack problem: A survey. *Naval Research Logistics*, 22:127–144, 1975.

, , , , ,

Model calibration for ship simulations

Ed van Daalen[1], Joseph Fehribach[2], Tristan van Leeuwen[3], Christian Reinhardt[4], Nick Schenkels[5] and Ray Sheombarsing[4]

[1] MARIN, Wageningen
[2] Worcester Polytechnic Institute, USA
[3] Centrum Wiskunde & Informatica, Amsterdam
[4] Vrije Universiteit, Amsterdam
[5] Universiteit Antwerpen, Belgium

Abstract

Model calibration is an important aspect in ship simulation. Here, ship motion is described by an ODE which includes tuning parameters that capture complex physical processes such as friction of the hull. In order for the simulations to be realistic for a wide range of scenarios these tuning parameters need to be calibrated to scale experiments. In principle, the optimal tuning parameters can be computed for any given scenario, but this would require a corresponding scale experiment to be conducted. The aim is to minimize the number of scenarios that need to be pre-calibrated while still being able to realistically model ship motion for a wide range of scenarios. In this paper we investigate the use of polynomial (sparse grid) interpolation to compute the optimal tuning parameters for *any* scenario from a few pre-calibrated optimal values.

Perturbation analysis of a simple model for roll damping indicates that the optimal tuning parameter may indeed vary strongly with the chosen scenario. Numerical experiments with this model confirm that the optimal tuning parameters vary strongly (but smoothly!) with the scenario and can be well approximated with polynomial interpolants. Further numerical experiments with a more complex modelling code for ship maneuvring are very promising.

Keywords: Model calibration, Parameter estimation, Chebyshev Interpolation, Sparse Grid Interpolation

1 Introduction

Ship simulators, used to train pilots, are based on simplified models for ship motion in order to enable real-time integration of the system. In such simplified models a lot of underlying physics is not explicitly modeled but is parametrized using tuning parameters. In order for the simulator to behave realistically, these models need to be calibrated to real-life (scale) experiments of actual ship motion under a wide variety of scenario's. This calibration process is depicted in figure 1. Here, and throughout the paper, we use the

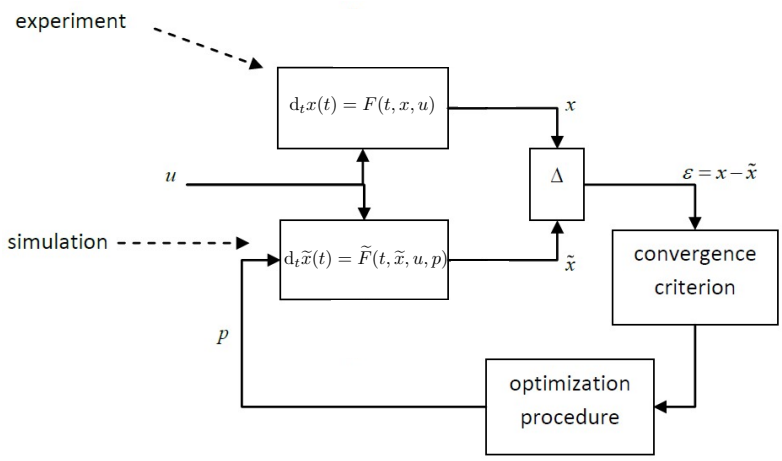


Figure 1: Schematic depiction of the calibration process.

following notation

- $\mathbf{u} = [u_1, u_2, \ldots, u_N]$ - vector with input scenario (rudder angle, propellor rpm, wave-height etc.);
- $\mathbf{p} = [p_1, p_2, \ldots, p_M]$ - vector with tuning parameters;
- $\mathbf{x} = [x_1, x_2, \ldots, x_K]$ - state vector describing *actual* ship movement – in the remainder of the paper we will treat this as the solution of an underlying *complex* model $\mathrm{d}_t \mathbf{x} = F(t, \mathbf{x}, \mathbf{u})$;
- $\widetilde{\mathbf{x}} = [\widetilde{x}_1, \widetilde{x}_2, \ldots, \widetilde{x}_K]$ - state vector describing *modeled* ship movement – this is the solution of the *simplified* model $\mathrm{d}_t \widetilde{\mathbf{x}} = \widetilde{F}(t, \widetilde{\mathbf{x}}, \mathbf{u}, \mathbf{p})$;

Unfortunately, there is no single setting of the tuning parameters for which the simple model will fit the complex model for all possible scenarios. For a given scenario, however, we can find the corresponding optimal calibration parameters as follows. First, we define a cost function $\mathcal{C}(\mathbf{u}, \mathbf{p})$ that measures the mismatch between $\mathbf{x}(t; \mathbf{u})$ and $\widetilde{\mathbf{x}}(t; \mathbf{u}, \mathbf{p})$ for a given $\mathbf{u}$ and $\mathbf{p}$. An example of such a cost function is the least-squares mismatch between the horizontal spatial coordinates (x_1, x_2)

$$\mathcal{C}(\mathbf{u}, \mathbf{p}) = \sum_{i=1}^{2} \int \mathrm{d}t \, \left(x_i(t; \mathbf{u}) - \widetilde{x}_i(t; \mathbf{u}, \mathbf{p})\right)^2 .$$

Then, the optimal $\mathbf{p}$ for a given scenario $\mathbf{u}$ is given by

$$\mathbf{p}^*(\mathbf{u}) = \operatorname*{argmin}_{\mathbf{p}} \mathcal{C}(\mathbf{u}, \mathbf{p}).$$

Given that the parameter space $\mathbf{p}$ is relatively small ($M \approx 20$), we can perform such a single calibration with a simple direct-search method (Kolda et al., 2003). Note, however, that each calibration requires not only multiple evaluations (by time integration) of the simple model, but also *one evaluation of the complex model (i.e. an experiment)*. Therefore, we would like to minimize the number of scenarios for which this calibration is performed. The question is: How can we efficiently calibrate the simple model for a range of scenarios while using only a limited number of evaluations of the complex model (i.e., experiments).

1.1 Approach

The main idea of our approach is to calibrate the simple model for a number of (cleverly chosen) scenarios $\{\mathbf{u}_k\}_{k=1}^{L}$, yielding optimal calibration parameters $\{\mathbf{p}_k^*\}_{k=1}^{L}$. For *any* given scenario $\mathbf{u}$, we then interpolate the optimal $\mathbf{p}^*$ elementwise based on these values

$$p_i^*(\mathbf{u}) = \sum_{k=1}^{L} w_{k,i} \psi_k(\mathbf{u}),$$

where ψ_k are basis functions specific to the type of interpolation used and $w_{k,i}$ are the corresponding weights chosen such that $p_i^*(\mathbf{u}_k) = p_{k,i}^*$. The main assumption here is that $\mathbf{p}^*$ varies smoothly with $\mathbf{u}$.

For a 1D scenario space (i.e., $N = 1$) we use Chebyshev interpolation in order to get high accuracy with only a few samples. In this case, the nodes (Chebyshev points) on $[-1, 1]$ are given by

$$u_k = \cos\left(\frac{2k-1}{2L}\pi\right).$$

For higher dimensional scenario spaces ($N > 1$), a simple cartesian product approach is not very attractive since the number of samples would grow exponentially with the dimension of the scenario space. In order to avoid this so-called *curse of dimensionality* we will consider *sparse grid* interpolation for $N > 1$ (Barthelmann et al., 2000). In sparse grids, the sampling points are clustered near the boundary of the domain and chosen more sparsely in the interior. Hereby the number of sampling points is considerably reduced when compared to a regular sampling. There are different choices of sparse grids that vary in number of grid points involved. A popular choice for the approximation of smooth functions is the so-called Clenshaw-Curtis grid. An example of such a grid at consecutive stages of refinement is shown in Figure 2. Table 1 shows how the number of sampling points grows with the stage of refinement. Note that L approximately doubles for each stage, whereas we would expect a quadrupling for a regular sampling in 2D. For more details on the accuracy and efficiency of sparse grid interpolation we refer to Barthelmann et al. (2000).

Stage	1	2	3	4	5	6	7
L	5	13	29	65	145	321	705

Table 1: Number of points in the sparse 2D grid in dependence of the stage depth. Note that L approximately doubles for each stage, whereas we would expect a quadrupling for a regular sampling for $N = 2$.

1.2 Outline

The remainder of the paper is organized as follows. First, we consider a model for *roll damping*. In this case, the models predict the roll motion (i.e. oscillations around the longitudinal axis) of the ship for given initial angle and forcing terms (which serve as the scenario parameters). The complex model F contains a non-linear damping term, while the simple model $\widetilde{F}$ contains only an equivalent linear damping term (which serves as the tuning parameter). For this model problem we perform a perturbation analysis and present a closed form solution for the optimal tuning parameter. We also perform a range of numerical experiments with both a 1D (with Chebyshev interpolation) and a 2D (with sparse grid interpolation) scenario space. Next, we present numerical experiments using a 6 degree-of-freedom model for rigid ship motion using the FREDYN modeling code. Here, the complex model is based on a frigate while the simple model is based on a lifeboat. We use a 1D scenario space (rudder angle) and we incorporate 6 tuning parameters governing rotational and drift forces. We compare two different mismatch criteria; based on horizontal spatial coordinates and based on the turning circle radius.

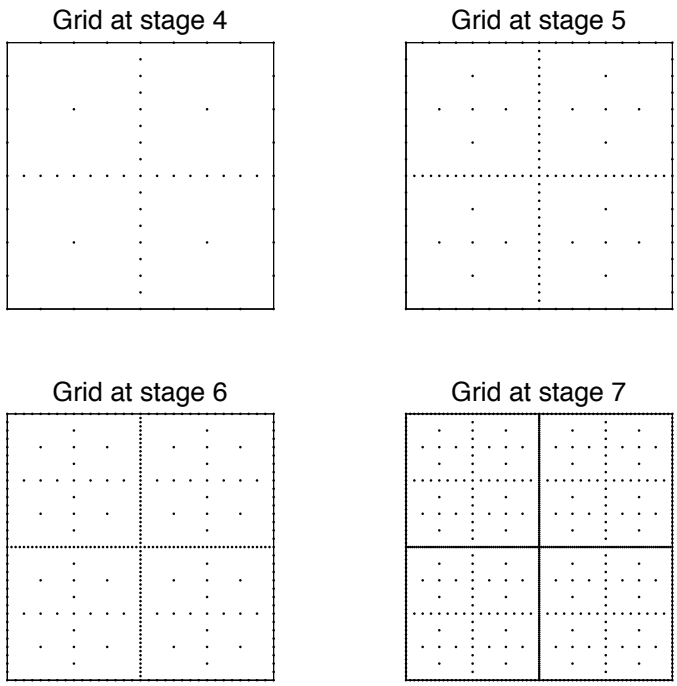


Figure 2: Grids used in various stages of Sparse-Grid interpolation procedure for $N = 2$. Note the clustering of the points at the boundary and sparsity in the interior of the domain.

Finally, we present conclusions and recommendations for further research.

2 A toy calibration problem: equivalent linear damping

The roll motion of a ship is modelled by the following ODE

$$(I + A)\ddot{\phi}(t) + B(\dot{\phi})\dot{\phi}(t) + C\phi(t) = M(t), \tag{1}$$

where I, A and C are constants, $B(\cdot) = b_1 + b_2|\cdot|$ is a damping term and $M(t)$ is a forcing term. In the remainder of the section we will consider this equation with *non-linear damping* (i.e., $b_2 \neq 0$) as the complex model, while the simple model only includes *linear* damping ($b_2 = 0$) and b_1 will serve as the tuning parameter. The initial angle $\dot{\phi}(0)$ and the amplitude and frequency of a periodic damping term $M(t) = a\sin(\omega t)$ will serve as the scenario parameters.

2.1 Linear damping

For $b_2 = 0$, equation (1) can be written in the form of a standard damped oscillator

$$\ddot{\phi} + 2\zeta\omega_o\dot{\phi} + \omega_o^2\phi = m \tag{2}$$

where

$$\omega_o := \sqrt{\frac{C}{I + A}}, \qquad m := \frac{M}{I + A} \qquad \text{and} \qquad \zeta := \frac{b_1}{2\sqrt{C(I + A)}}.$$

In the above equation, ω_o is the undamped oscillation frequency and ζ is the nondimensional damping coefficient. If $0 < \zeta < 1$, the system is sub-critically damped, and the general solution is

$$\phi(t) = \alpha e^{-\zeta\omega_o t}\cos(\omega_d t - \beta) + \phi_{\mathrm{p}}(t)$$

where α and β are free parameters determined by the initial conditions (α is the amplitude and β is the phase angle). The damped frequency is $\omega_d := \omega_o\sqrt{1 - \zeta^2}$, and $\phi_{\mathrm{p}}(t)$ is any particular solution which satisfies the nondimensional equation (2). For example, if $M(t) = M$ is constant, then $\phi_{\mathrm{p}}(t) = m/\omega_o^2 = M/C$ is the simplest particular solution.

2.2 Perturbation analysis

Now consider the homogeneous perturbed nondimensional equation

$$\ddot{\phi} + 2\zeta\omega_o(1 + \epsilon|\dot{\phi}|)\dot{\phi} + \omega_o^2\phi = 0. \tag{3}$$

For convenience, we drop the absolute values and allow the sign of ϵ to change when the sign of $\dot{\phi}$ changes [1]. Assuming that $|\epsilon| << 1$, one can make a regular perturbation expansion using the ansatz $\phi(t) = \phi_0(t) + \epsilon\phi_1(t) + O(\epsilon^2)$. Substituting this ansatz into (3), one finds that as before $\phi_0(t) = \alpha e^{-\zeta\omega_o t}\cos(\omega_d t - \beta_0)$ and that ϕ_1 must satisfy

$$\ddot{\phi}_1 + 2\zeta\omega_o\dot{\phi}_1 + \omega_o^2\phi_1 = -2\zeta\omega_o(\dot{\phi}_0)^2 \tag{4}$$

The form of the homogeneous solution for (4) is the same as before, but unless the initial position or velocity depend on ϵ (which would be unusual), this homogeneous solution is identically zero. Because of the form of the right-hand side of (4), the particular solution must be the linear combination of three terms:

$$\phi_1^{\mathrm{p}}(t) = A^{\mathrm{p}}e^{-2\zeta\omega_o t}\cos^2(\omega_d t - \beta_1) +$$
$$B^{\mathrm{p}}e^{-2\zeta\omega_o t}\cos(\omega_d t - \beta_1)\sin(\omega_d t - \beta_1) + C^{\mathrm{p}}e^{-2\zeta\omega_o t}\sin^2(\omega_d t - \beta_1)$$

where the coefficients A^{p}, B^{p} and C^{p} are determined by substituting this linear combination into (4), and β_1 is a shifted phase angle due to the presence of $\dot{\phi}_0$ in the right-hand side of (4), rather than just having ϕ_0.

2.3 Calibration

Now let us take the perturbed solution (for some ζ and ϵ) as the solution of the *complex system* and let us try to match this to the unperturbed solution using ζ as a tuning parameter. Specifically, let $\phi(t;\epsilon_1,\zeta_1) := \phi_0(t;\zeta_1) + \epsilon_1\phi_1^{\mathrm{p}}(t;\zeta_1)$ be the solution of the complex system and let $\widetilde{\phi}(t;p) = \phi_0(t;p)$ with p being the single tuning parameter. The question then is how should one set p so that the simple solution matches the complex solution?

Consider the absolute difference

$$\begin{aligned} |\phi(t;\epsilon_1,\zeta_1) - \widetilde{\phi}(t;p)| &= |\alpha(e^{-\zeta_1\omega_o t} - e^{-p\omega_o t})\cos(\omega_{\mathrm{d}} t - \beta_0) + \epsilon_1\phi_1^{\mathrm{p}}(t,\zeta_1)| \\ &= \alpha e^{-\zeta_1\omega_o t}|(1 - e^{(\zeta_1 - p)\omega_o t}\cos(\omega_{\mathrm{d}} t - \beta_0) \\ &\qquad + \epsilon_1 e^{-\zeta_1\omega_o t}(A^{\mathrm{p}}\cos^2 + B^{\mathrm{p}}\cos\sin + C^{\mathrm{p}}\sin^2)| \end{aligned}$$

Each of the last four trigonometric functions must be evaluated at $(\omega_{\mathrm{d}} t - \beta_1)$. Again the coefficients A^{p}, B^{p} and C^{p} are functions of ζ_1; they are determined by requiring that ϕ_1^{p} is actually a particular solution of (4).

Because of the decaying exponentials, the above absolute difference will decrease in time. But it is also possible to make this difference zero at a

[1] Because the sign of ϵ changes with each half oscillation, one must stop and restart the solutions to follow the motion through multiple oscillations.

specified time, for example, if $t = \beta_0/\omega_d$, then the absolute difference is zero when

$$p = \zeta_1 - \frac{\sqrt{1-\zeta_1^2}}{\beta_0} \ln(1 - \epsilon_1 e^{-\zeta_1\beta_0/\sqrt{1-\zeta_1^2}} D^{\mathrm{p}})$$

where $D^{\mathrm{p}} := A^{\mathrm{p}} \cos^2(\beta_0 - \beta_1) + B^{\mathrm{p}} \cos(\beta_0 - \beta_1)\sin(\beta_0 - \beta_1) + C^{\mathrm{p}} \sin^2(\beta_0 - \beta_1))$ So this is a relatively simple formula for determining the tuning parameter p in terms of ζ_1 and ϵ_1, the given parameters in the complex solution. That is, there is an explicit expression for selecting the tuning parameter in terms of the given parameters to minimize the absolute difference at least at one specific time. Of course, this approach only has the two solutions matching at one specified time, and then again for large time.

The tuning parameter of course could be chosen to minimize the absolute difference in other ways, for example, by selecting a different time, or by making a least squares fit across some interval of time. But since the absolute difference already decays in time, one would likely wish to set the difference to zero at some early time. So one could minimize

$$\int_0^T \Big(\phi(t;\epsilon_1,\zeta_1) - \widetilde{\phi}(t;p)\Big)^2 \, dt$$

by finding a stationary point p for which

$$\frac{\mathrm{d}}{\mathrm{d}p} \int_0^T \Big(\phi(t;\epsilon_1,\zeta_1) - \widetilde{\phi}(t;p)\Big)^2 \, dt = 0.$$

We expect that a closed-form expression for the optimal p can be derived in a simular manner as above but this investigation is outside the scope of the current report.

2.4 Numerical experiments

For the numerical experiments we numerically integrate the roll damping equation (1) with $I = 6.4$, $A = 0$, $C = 1$, $M(t) = a\sin(\omega t)$ and initial condition $\phi(0) = \phi_0$ and $\dot{\phi}(0) = 0$ for $T = 80$ seconds. We will use $\mathbf{u} = [\phi_0, a, \omega]$ as scenario parameters. We denote the solution of the *complex* system (with $b_1 = 0$ and $b_2 = 15$) by $\phi(t;\phi_0,a,\omega)$ while $\widetilde{\phi}(t;\phi_0,a,\omega;p)$ denotes the solution of the simple system (with $b_1 = p$ and $b_2 = 0$).

In these experiments we find the optimal p by minimizing the least-squares cost function

$$\begin{aligned} \mathcal{C}(p,\phi_0,a,\omega) \quad = \quad & \sum_i \Big(\phi(t_i;\phi_0,a,\omega) - \widetilde{\phi}(t_i;\phi_0,a,\omega;p)\Big)^2 \\ & + \Big(\dot{\phi}(t_i;\phi_0,a,\omega) - \dot{\widetilde{\phi}}(t_i;\phi_0,a,\omega;p)\Big)^2 \end{aligned}$$

using Matlabs `fminsearch`. For the 1D interpolation, we use the `chebfun` package (Trefethen, 2013). For 2D interpolation we use the `Sparse Grid Interpolation Toolbox` package (Klimke, 2007).

2.4.1 Case 1: roll decay with varying initial roll angle

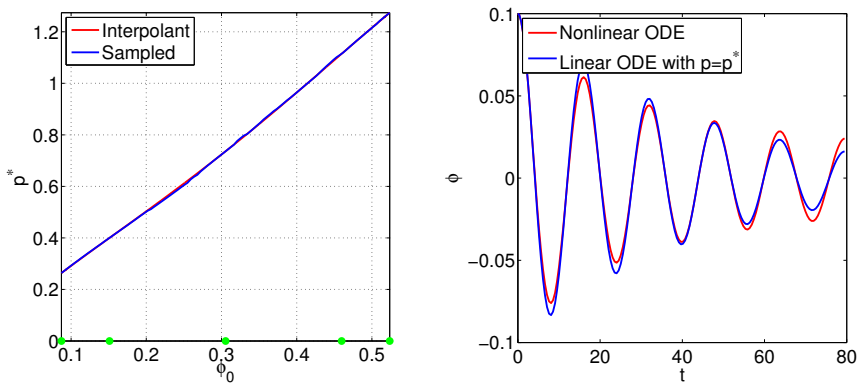
We set $a = 0, \omega = 0$ and vary only the initial condition $\phi_0 \in [\pi/36, \pi/6]$. The optimal values of p as a function of ϕ_0, obtained through Chebyshev interpolation with 5 points and brute-force sampling is shown in Figure 3 (a). The solutions for the complex and simple system (using the optimal p) for $\phi_0 = 0.1$ is shown in Figure 3 (b).

2.4.2 Case 2: regular forcing with varying amplitude

In this experiment, we set $\phi_0 = 0, \omega = 0.395$ and vary only the amplitude $a \in [0, 2]$. The optimal values of p as a function of a, obtained through Chebyshev interpolation with 5 points and brute-force sampling is shown in Figure 4 (a). The solutions for the complex and simple system (using the optimal p) for $a = 1.3$ is shown in Figure 4 (b).

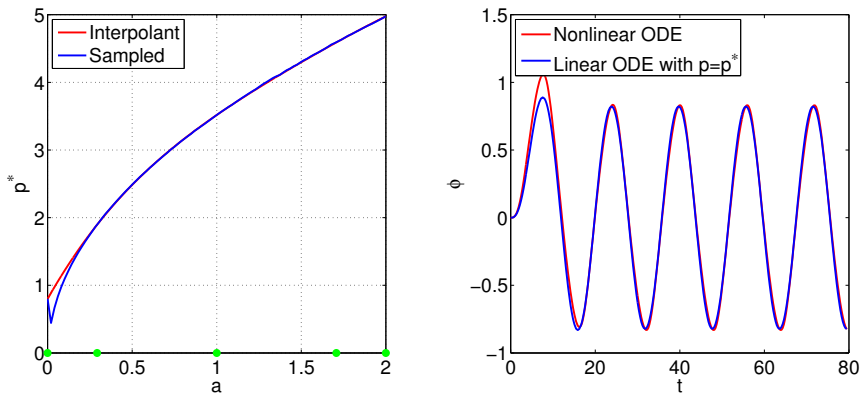
2.4.3 Case 3: regular forcing with varying amplitude and frequency

In this experiment we set $\phi_0 = 0$ and vary both $a \in [0, 2]$ and $\omega \in [0, 2]$. Figure 5 (a) shows a sparse interpolant on the Clenshaw-Curtis grid of stage 5 of the optimal p. The solutions for the complex and simple system (using the optimal p) for a particular choice of (a, ω) is shown in Figure 5 (b).



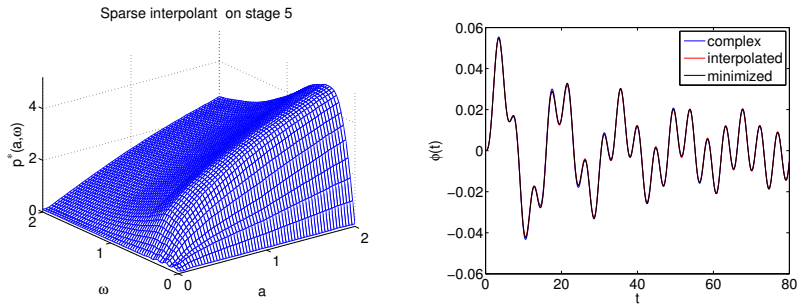
(a) (b)

Figure 3: Case 1: (a) p^* as a function of ϕ_0 obtained through Chebyshev interpolation with 5 points (blue) and brute-force sampling with 100 points (red). (b) Solutions of the complex (red) and simple systems (blue) for the optimal p obtained through interpolation, both for $\phi_0 = 0.1$.



(a) (b)

Figure 4: Case 2: (a) p^* as a function of a obtained through Chebyshev interpolation with 5 points (blue) and brute-force sampling with 100 points (red). (b) Solutions of the complex (red) and simple systems (blue) for the optimal p obtained through interpolation, both for $a = 1.3$.



(a) (b)

Figure 5: Case 3: (a) p^* as a function of (a, ω) obtained through sparse grid interpolation with 145 samples. (b) Solutions of the complex (blue) and simple systems for the optimal p obtained through interpolation (red) and by direct optimization (black), all for a randomly chosen (a, ω).

3 FREDYN code

In this section we present experiments with the `FREDYN` program, (Ypma, 2014). The complex model is based on a frigate and replaces the real life experiment. The simple model is based on a lifeboat. In these experiments $\mathbf{u}$ represents the rudder angle (i.e. $\mathbf{u}$ is 1-dimensional) which lies within the range $[5, 30]$ deg. As tuning parameters we use the following parameters that govern the drift and rotational forces:

$$\mathbf{p} = [X_{vv}, X_{rr}, X_{vr}, Y_{uv}, Y_{vv}, Y_{ur}].$$

The default values of these parameters are given by

$$\mathbf{p} = [18508, 0, 4117877, -82292, -201134, 3075682]. \tag{5}$$

To compare the simulations, we plot both the horizontal coordinates $(x_1(t), x_2(t))$ and the turning radius which is defined as

$$R(t) = \frac{\sqrt{v_1(t)^2 + v_2(t)^2}}{v_r(t)},$$

where v_1, v_2 are the horizontal velocities (surge, sway) in m/s and v_r is the angular velocity (yaw) in rad/s. We stop the simulations after the ship has completed a full turn. This means that simulations with a smaller rudder angle will run longer.

Figure 6 shows the behaviour of the complex and simple model for a rudder angle of 5 deg using the default $\mathbf{p}$. This clearly illustrates the need to fit the tuning parameters.

3.1 Finding an optimal p

We consider two different cost functions. The first cost function measures the misfit between the horizontal coordinates (x_1, x_2):

$$\mathcal{C}_1(\mathbf{u}, \mathbf{p}) = \int \mathrm{d}t \; (x_1(t) - \widetilde{x}_1(t))^2 + (x_2(t) - \widetilde{x}_2(t))^2 \,. \tag{Method 1}$$

The second cost function measures the misfit between the turning radii (cf. equation (3)),

$$\mathcal{C}_2(\mathbf{u}, \mathbf{p}) = \int \mathrm{d}t \; \left(R(t) - \widetilde{R}(t)\right)^2 . \tag{Method 2}$$

We use a direct-search method (Matlab's `fminsearch`) to find the optimal $\mathbf{p}$.

Figure 7 shows the simulations for the optimal $\mathbf{p}$ as obtained via Method 1 and Figure 8 shows the simulations for the optimal $\mathbf{p}$ as obtained via Method

2, both for a rudder angle of 5 deg. Comparing these to Figure 6 we see a dramatic improvement in the fit.

The 6th component of the optimal $\mathbf{p}$ (Y_{ur}) as a function of the rudder angle using 5 and 10 Chebyshev points is shown in Figure 9. We observe that the optimal $\mathbf{p}$ does not vary as smoothly with $\mathbf{u}$ as in the case of roll damping. In particular, we see a *staircase* effect that we do not fully understand. Still, the Chebyshev interpolation is able to capture the general trend. Figures 10 and 11 show how the interpolated optimal $\mathbf{p}$ for a rudder angle of 15 deg is able to produce a very good match between the simple and complex models.

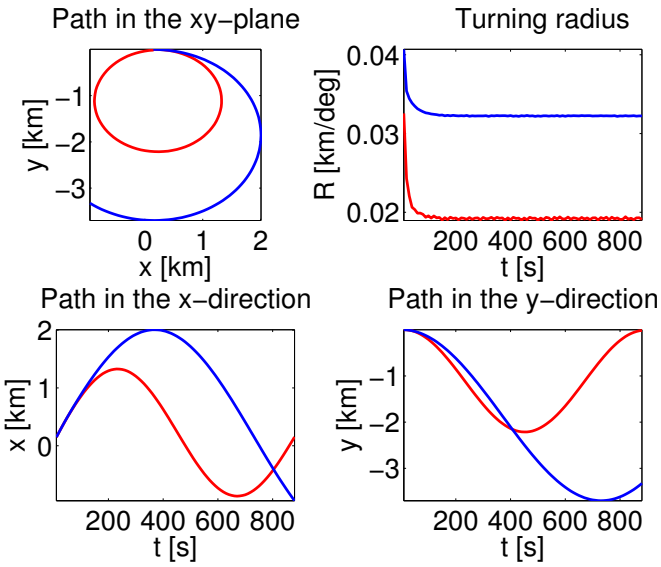


Figure 6: Simulation for the simple and complex model with a rudder angle of 5 deg using the default values for **p**. The red and blue lines represents the complex and simple model respectively.

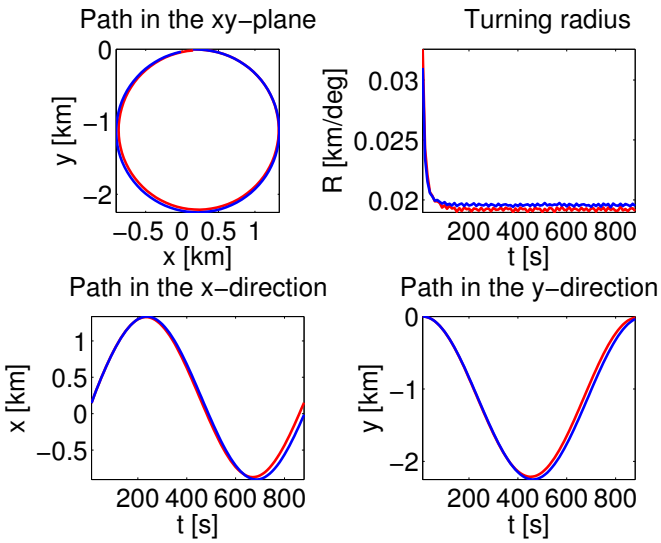


Figure 7: Simulation for the simple (blue) and complex (red) model with a rudder angle of 5 deg and the optimal $\mathbf{p}$ found by Method 1.

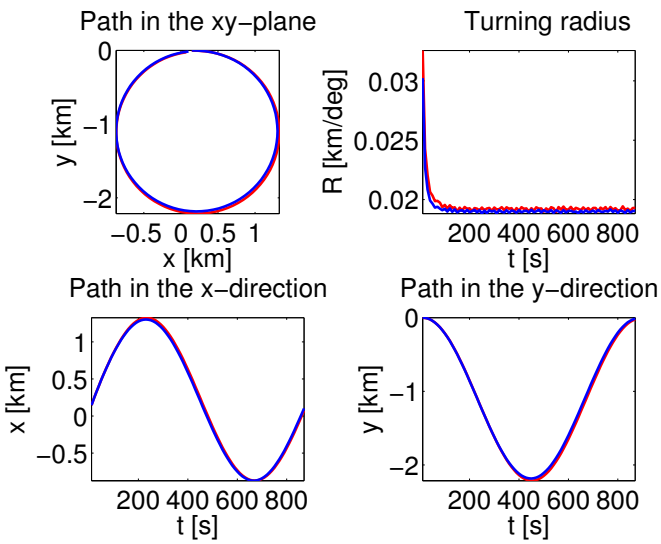


Figure 8: Simulation for the simple (blue) and complex (red) model with a rudder angle of 5 degrees and the optimal $\mathbf{p}$ found by Method 2.

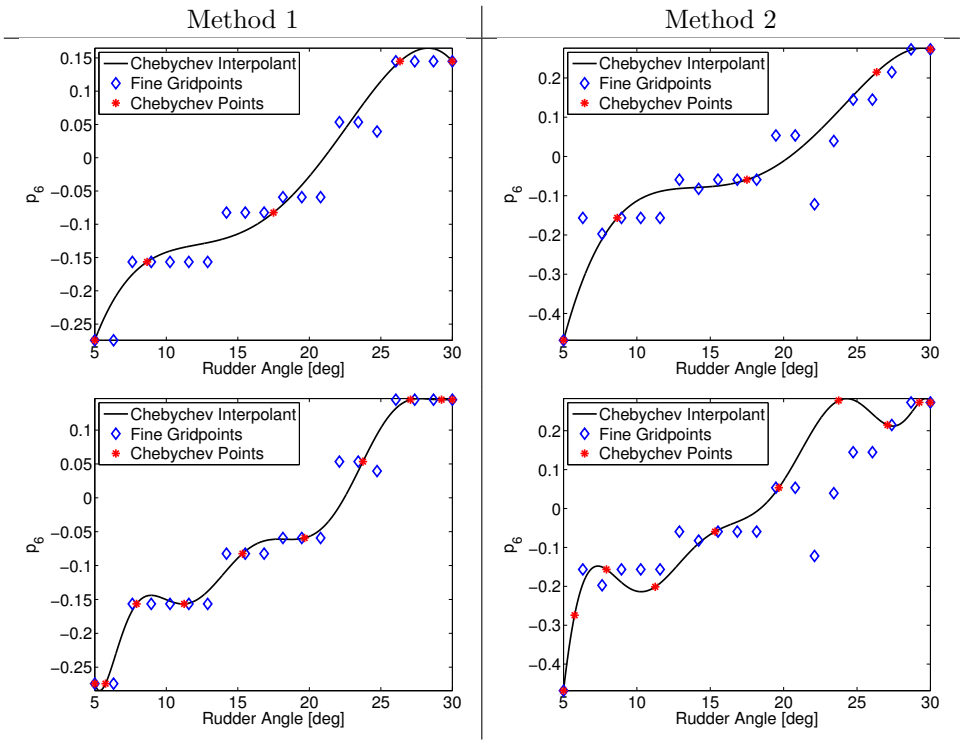


Figure 9: Chebyshev interpolation of the 6th component of $\mathbf{p}^*$ for the two methods. The first and second row use 5th and 10th order Chebyshev interpolation respectively. The values of p_6^* are scaled with the default value p_6 (see (5)).

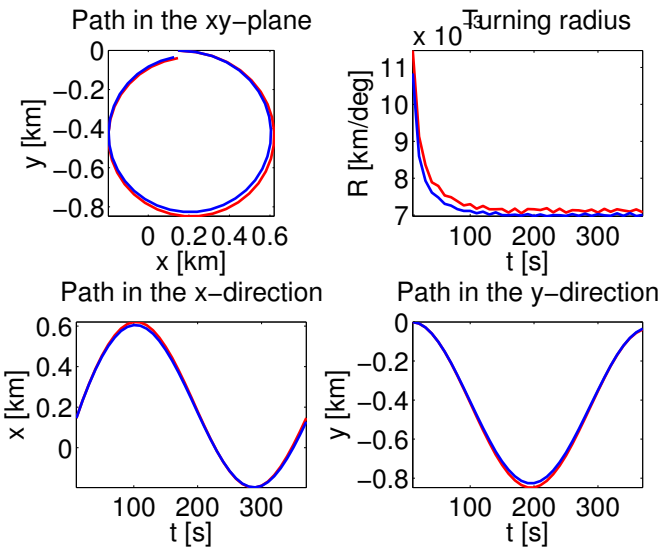


Figure 10: Simulation for the simple and complex model with a rudder angle of 15 degrees. The red and blue lines represents the complex and simple model respectively. Here the first method was used to determine the optimal value for p with Chebyshev interpolation of order 5.

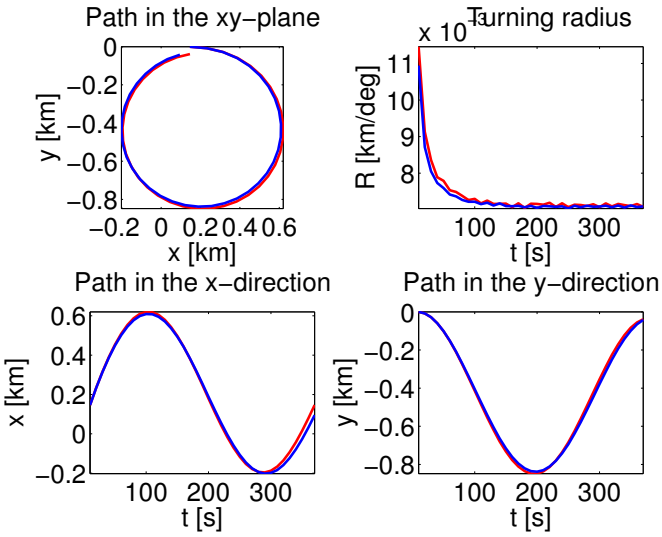


Figure 11: Simulation for the simple and complex model with a rudder angle of 15 degrees. The red and blue lines represents the complex and simple model respectively. Here the second method was used to determine the optimal value for p with Chebyshev interpolation of order 5.

4 Conclusions

We have presented a method for model calibration with application to ship simulations. The goal is to find the optimal tuning parameters $\mathbf{p}$ such that the solution of a *simple* system, $\widetilde{\mathbf{x}}(t; \mathbf{u}, \mathbf{p})$, matches the solution of a more complex system, $\mathbf{x}(t; \mathbf{u})$, for a range of scenarios $\mathbf{u}$. For a single scenario, this calibration can by done by minimizing a cost function that measures the difference between $\widetilde{\mathbf{x}}$ and $\mathbf{x}$. There are only a few (≈ 20) tuning parameters so that this minimization can be done with so-called direct-search methods. Such methods are very suitable for black-box optimization problems since they do not require gradient calculations of the cost function w.r.t. the tuning parameters.

However, each calibration requires an evaluation of the complex system (i.e., a scale experiment). In order to minimize the number of scale experiments that need to be done, we calibrate the simple model only for a small number of well-chosen scenarios $\{\mathbf{u}_k\}$, giving us the corresponding tuning parameters $\{\mathbf{p}_k\}$. We assume that the optimal tuning parameters vary smoothly with $\mathbf{u}$ and use polynomial interpolation to compute the optimal $\mathbf{p}$ for any given $\mathbf{u}$ from these points. When there is only one scenario parameter we use Chebyshev interpolation. This approach does not generalize well to higher dimensional scenario space as the required samples would grow exponentially with the dimension. To avoid this *curse of dimensionality* we resort to sparse-grid interpolation techniques.

Perturbation analysis of a model-problem (roll-damping) indicates that it is possible to obtain closed-form solutions for the optimal tuning parameter (the equivalent linear damping) in some specific cases. Numerical experiments indicate that the optimal $\mathbf{p}$ varies smoothly with $\mathbf{u}$. Both Chebyshev and sparse-grid interpolation perform well in this setting, as is confirmed by numerical experiments. Numerical experiments with a more complex system of ODEs that models full ship motion (using the FREDYN code) show promising results.

5 Recommendations

- The perturbation analysis and numerical experiments on the roll damping equation give some insight in how to choose the optimal equivalent linear damping term. It would be very insightful to verify the findings from the perturbation analysis numerically. The analysis might also be extended by considering other mismatch criteria and include a driving term. Such analysis should be able to predict the observed smooth dependency of the optimal $\mathbf{p}$ w.r.t. $\mathbf{u}$ and may even tell us how smooth the function is, allowing us to compute a-priori error estimates for the

interpolation.

- Further numerical testing with the FREDYN code with time-varying rudder setting, using the optimal tuning parameters appropriate for each rudder setting. This requires the ability to vary the tuning parameters with time.

- An alternative avenue – not tested in this report – is to try to find a $\mathbf{p}$ that is optimal over a range of scenarios $\mathbf{u}$

$$\mathbf{p}^* = \underset{\mathbf{p}}{\operatorname{argmin}} \int \mathrm{d}\mathbf{u}\, \mathcal{C}(\mathbf{u}, \mathbf{p}) \pi(\mathbf{u}),$$

  where $\pi(\mathbf{u})$ is a weighting function used to emphasize scenarios that are deemed more important [2]. A first step would be to use a brute force approach to compute the integral (i.e., by dense sampling of the scenario space). If the results are satisfactory, a generalized Polynomial Chaos expansion (gPC) can be employed to efficiently estimate the expectation (Xiu, 2010). The basic ideas are very similar to the ones discussed above; we need to sample a number of scenarios according to a quadrature rule (which will depend on π) and sparse grid techniques can be used to generalize to higher dimensions. Note that we cannot employ Monte-Carlo sampling methods to estimate the integral, as this would require evaluation of the misfit for many scenarios, which in turn would mean doing many experiments to evaluate the complex model for those scenarios.

- The optimization problem for a single scenario is very likely to have multiple minima (local *and* global) and suffer from ill-conditioning. This means that the variability of the optimal $\mathbf{p}$ as a function of $\mathbf{u}$ we observed may to some extent be artificial, in particular for the FREDYN examples. A sensitivity analysis of the problem (for example through the Jacobian of $\mathbf{p}^*$) may give some insight.

[2] Alternatively, we can interpret $\pi(\mathbf{u})$ as a probability, in which case we are aiming to find a $\mathbf{p}^*$ that minimizes the *expected* misfit over all possible scenarios.

A Matlab code

We give a short overview of the scripts and functions related to the generation of the above described results. We restrict to describe the dependencies and short explanations of the functionalities. Additional comments can be found in the files themselves.

A.1 1D Interpolation for damped oscillator

- `compareP`: computes two numerical approximations of $\theta_0 \mapsto p^*(\theta_0)$, where θ_0 is the parameter to be varied in an interval specified by the user. One approximation is determined by brute-force sampling and the other by Chebyshev interpolation. Moreover, $\theta_0 = \phi_0$ in the absence of an external force, and $\theta_0 = a$ in the presence of an external force. This subroutine calls `interpolateP` and `optimizeP`.

- `interpolateP`: computes the Chebyshev interpolant of the function $\theta_0 \mapsto p^*(\theta_0)$ by using the Chebfun-package developed in [reference to homepage of Chebfun]. This subroutine calls `optimizeP`.

- `optimizeP`: computes an optimal value for the damping coefficient p, such that the error between the solutions of the nonlinear and linear model is minimized, by applying the Matlab-subroutine `fminsearch` to the function-handle `difference`.

- `difference`: computes the error in the Euclidian norm between the time series of the nonlinear and linear model for a given value of p. This subroutine calls `integrateF`.

- `integrateF`: numerically integrates the ODE

$$(I + A)\frac{\mathrm{d}^2\theta}{\mathrm{d}t^2} + \left(b_1 + b_2\left|\frac{\mathrm{d}\theta}{\mathrm{d}t}\right|\right)\frac{\mathrm{d}\theta}{\mathrm{d}t} + \theta = M(t) \tag{6}$$

  by using the Matlab-subroutine `ode45`. This subroutine calls `F`.

- `F`: implements the first-order vector field associated to (6).

A.2 2D Interpolation for damped oscillator

- `interpolant_script`: main script implementing above described comparison. Used to produce Figure 5 (Note the randomized input a and ω hence output will not be the same). Calls `optimal_p` in the construction of the sparse interpolant

- `optimal_p`: implements optimization procedure, calls `difference` in the optimization using `fminsearch`
- `difference`: implements the least-squares misfit, calls `compute_timeseries`.
- `compute_timeseries` performs the simulation of the complex and simple system by writing the as first order systems, calls `nonlinear_osc_rhs` within `ode45`
- `nonlinear_osc_rhs`: implements the right hand side of the first order systems. Notation used: $I\ddot{\theta} + \left(b_1 + b_2|\dot{\theta}|\right)\dot{\theta} + \theta = M(t)$

A.3 1D Interpolation with `FREDYN` code

The m-files should be places in a folder which must contain two copies of the `./examples/manoeuvring/leander` folder, one called *leander_simpel* and the other `leander_complex`[3]. This is used to run both the complex fregat model and the simpel lifeboat model at the same time while keeping the results seperated.

In both folders the `leander_ship.xmf` file should be changed. The `RandomRudder` script must be removed or commented out and be replaced by the line

```
scripting::Scripting "ConstantRudder" {};
```

Some remarks:

- In all the m-files `dir` is short for directory and normaly this will be the string `leander_simpel` or `leander_complex`.
- The m-file `ReadData` loads the content of the `leander.dat` file into Matlab. This is usually refered to as `OutputC` for the complex model and *OutputS* for the simpel model one.
- Some of the m-files use the argument `Mode`. This is either 1 or 2 and refers to the way the variable p is optimized. For more details we refer to section 3.

[3] e.g. we placed them in `./examples/manoeuvring/`

M-Files

These first few m-files are functions that are used to change and run the simulation of the complex and simple model in various ways, as well as to determine the optimal value for the parameter p for a certain rudderangle.

- `WriteTweaking:` Used to change the value of the variable p in the `mo_leander_hull.xmf` file in the directory denoted by `dir`, which is either `leander_simpel` or `leander_complex`. All 26 components can be changed, but in the other m-files only the first six in the `leander_simpel` folder have been modified.

- `WriteParameters:` In order to easily change the rudderangle from within Matlab this m-file creates a python script `ConstantRudder.py` in the directory pointed to by dir (again, either `leander_simpel` or `leander_complex`). This is done by adding a line to the `ConstantRudderTemplate.py` defining the rudderangle and copying this to the chosen directory.

- `RunSimulation:` Runs the simpulation of the simple (`dir` = `leander_simpel`) or complex (`dir` = `leander_complex`) model and deletes the created `*_leander.out` files.

- `ReadData:` Reads the `leander.dat` file in the directory denoted by `dir` and loads it into Matlab.

- `PlotTraces:` Makes a plot of the data in `OutputC` and `OutputS`. Top left: the path in the xy-plane of both models. Top right: the radial distance R of both models. Botom left: timetrace of x of both models. Botom right: timetrace of y for both models.

- `OptimalP:` Given a `Mode` and a rudderangle (argument u) this runs the complex model and then searches for an optimal value for p (only the first six components are changed) by using the Matlab function `fminsearch` (only 20 iterations are used). Using this optimal p the simple model is computed. The function returns the output of both simulations (`OutputC` and `OutputS`) and the optimal value for p `OptP` (corresponding to this rudderangle u!).

- `NormDiff:` This function is used to determine how closely the simple model matches the complex one, e.g. it is used by `OptimalP`

The following m-files where used to create a.o. the results shown in the presentation. The can be seen as an example how the previous m-files can be used.

- `Example0:` Runs the complex and the simple model with the default value of p for three different rudderangles $(5, 15$ and $30)$ to show that this is not always a good choice of parameters. The results are saved.
- `Example1:` Determines the optimal value of p for an oversampling of the range $[5, 30]$ which is used as a reference value for the Chebychev interpolation. The results are saved.
- `Example2_5:` Determines the 5 Chebychev points in $[5, 30]$ and calculates the optimal value of p for them. The results are saved.
- `Example2_10:` Idem for 10 Chebychev points.
- `MakePlots:` Using the data calculated by the `Example` m-files, this m-files makes plots of the results and calculate the Chebychev interpolation. The plots are all saved in a subfolder `./Plots`

References

Volker Barthelmann, Erich Novak, and Klaus Ritter. High dimensional polynomial interpolation on sparse grids. *Advances in Computational Mathematics*, 12:273–288, 2000. URL `http://link.springer.com/article/10.1023/A:1018977404843`.

Andreas Klimke. Sparse Grid Interpolation Toolbox – user's guide. Technical Report IANS report 2007/017, University of Stuttgart, 2007.

Tamara G. Kolda, Robert Michael Lewis, and Virginia Torczon. Optimization by Direct Search: New Perspectives on Some Classical and Modern Methods. *SIAM Review*, 45(3):385–482, January 2003. ISSN 0036-1445. doi: 10.1137/S003614450242889. URL `http://epubs.siam.org/doi/abs/10.1137/S003614450242889`.

L. N. Trefethen. Chebfun version 4.3, 2013. `http://www.chebfun.org`.

Dongbin Xiu. *Numerical Methods for Stochastic Computations: A Spectral Method Approach*. Princeton University Press, 2010. ISBN 9781400835348.

E. Ypma. Generic forward-aft manoeuvring model. *MARIN Technical Report*, 2014.

, , , , , , ,

Nonlinear Cochlear Dynamics

Maarten Bosmans[1], Sarah Gaaf[2], Chris Groothede[3], Rohit Gupta[4], Marta Regis[2], Michael Tsardakas[5]*, Arthur Vromans[6] and Kees Vuik[4]

[1] VORTECH BV
[2] TU Eindhoven
[3] VU Amsterdam
[4] TU Eindhoven
[5] Heriot Watt University, UK
[6] Leiden University

Abstract

In this report we examine a model for human hearing. The unknown parameters in the model are estimated using experimental data and standard optimisation methods as described in the text. Additionally, we suggest possible improvements to the model as well as proposing a method to use the current model in locating which frequencies are affected in a damaged ear.

Keywords: cochlear model, delay differential equation, parameter estimation, hearing loss

*Corresponding author: mtsardakas@gmail.com

1 Introduction

INCAS[3] posed the problem of modeling the human hearing system in the Study Group of Mathematics with Industry held at TUDelft. More specifically, the company brought to our attention a model describing the part of the ear called cochlea as a series of coupled oscillators. Each oscillator is modelled by a second order linear ordinary differential equation with a delay term. The questions that mainly concerned us were the following:

1. Is it possible to improve this model?
2. Is it possible to estimate the parameters of this model using experimental data?

In the course of four days, we attempted to answer these questions as accurately as possible. In addition, the company was interested in a mathematical description of how a damaged ear works in comparison to a healthy one. While this proved to be an impossible task, we were able to suggest a method for locating the frequencies that are affected in a damaged ear.

This article is organised as follows: in section 2, we present the model used and the theoretical background it is based upon. In section 3, we analyse the model and attempt a physical interpretation of it. Section 4 deals with the mathematics of hearing loss diagnosis, while in section 5 we propose methods to estimate the parameters of the model. Finally, we present conclusions of our investigations as well as future directions of research.

2 Theoretical background

The model presented in this section is described in more detail in a paper of Zweig (1991).

Before analyzing the model that describes how the human ear works, let us first consider the anatomy of the human ear and, more specifically, the cochlea. It has been known for quite some time that the cochlea is a nonlinear, active system that converts sound into neural stimuli. In addition, the cochlea not only responds to the sound it receives, but emits sound as well. These OtoAcoustic Emissions (OAEs) can be accurately measured however, due to the nonlinearity of the cochlea, their use in revealing how the cochlea responds to certain (controlled or not) stimuli is non-trivial.

In his paper, Zweig considers a simplified model of the human ear by 'uncoiling' the cochlea, a model already existent in the literature. He notes that discrepancies appear between this theory and some experiments, which might be caused by deficiencies of the model, namely the possibility of oversimplifying what actually happens in the cochlea. However, relaxing the assumptions

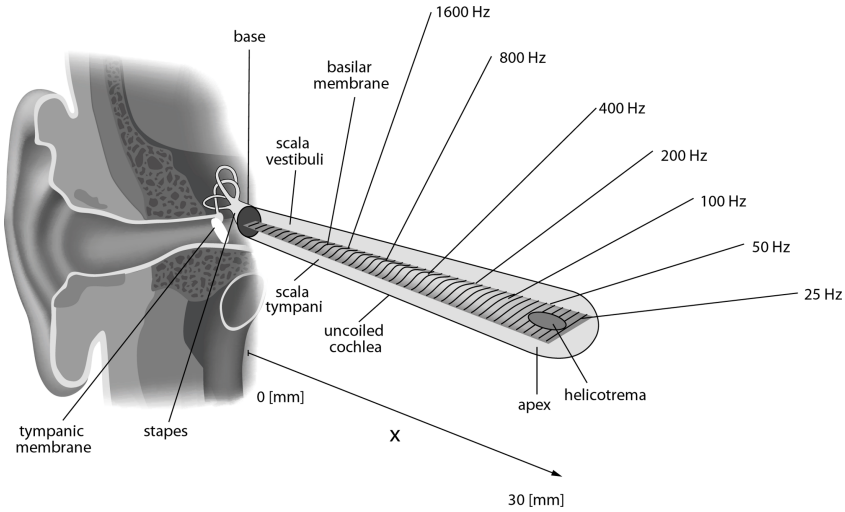


Figure 1: *Simplified model of the inner ear with uncoiled cochlea. Reproduced from* `wikimedia.com`

the model uses does not lead to any improvements. One of the assumptions, for instance, is that the geometry of the cochlear model is excessively simplified by the uncoiling of the cochlea. In addition, the fluids in the scalae are considered incompressible and inviscid. Relaxing these assumptions, i.e. assuming the cochlea is coiled and allowing the fluid to be compressible and viscous, only slightly changes the output of the model, which still does not fit the experimental data. Furthermore, the author questions the assumption that the scala media should be interpreted as an array of oscillators, coupled only through the fluid inside the ear. However, adding additional coupling between adjacent oscillators also fails to improve the output of the model.

Instead of relaxing the initial assumptions, the author uses a different approach. He considers the initial model as correct in a number of cases but acknowledges it is too simple to fully capture the correct behavior. Then, he proceeds to replace the harmonic oscillators of the model with more complex oscillators. Using the data available it is possible to approximate the form of the refined transport function and use it to obtain the more complex oscillator equation. We will now describe this approach in more detail, as there are parts of this procedure that could be altered, possibly resulting in a further improvement.

To get an oscillator equation from the transport equation, let us consider the latter as an integral equation for λ and suppose that T can be obtained

from experimental data:

$$T(s) \simeq C \frac{s}{\lambda^{3/2}(s)} \exp\left(-\int_{s_0}^{s} \frac{ds'}{\lambda(s')}\right) \tag{1}$$

We can now get the oscillator equation by Fourier transforming $\lambda^2 V \propto sP$ using the form of λ from the simple harmonic oscillators model:

$$\lambda^2 = \frac{s^2 + \delta s + 1}{(4N)^2},$$

where N is approximately equal to the number of wavelengths of the wave on the membrane. This will give us an inhomogeneous oscillator equation for the velocity v of a point on the basilar membrane, where $v = \mathcal{F}(V)$ and $\mathcal{F}$ denotes the Fourier transform, namely

$$\ddot{v}(\theta) + \delta \dot{v}(\theta) + v(\theta) = \frac{\dot{p}(\theta)}{\omega_{c0} M_0},$$

where $\theta(x) = \omega_c(x)t$ and $(\dot{\ }) = \partial/\partial\theta(x)$. We can now differentiate the transfer equation to solve for λ:

$$\lambda = -\left(1 + \frac{3}{2}\frac{d\lambda}{ds}\right)\left(\frac{d\ln(T/s)}{ds}\right)^{-1}. \tag{2}$$

Because the derivative of λ is small, we can get an approximation for λ as

$$\lambda \simeq -\frac{ds}{d\ln(T/s)}. \tag{3}$$

As λ is heavily influenced by the derivative of T, it is important to get an approximation for T as smooth as possible. To this end, a data fitting is performed by maximizing a modified likelihood function of the form $\chi^2 + \xi k^2$ where

$$k^2 = \int \left|\frac{T(s(\Omega))}{d\Omega^2}\right| d\Omega$$

and ξ plays the role of a Lagrange multiplier. This form of likelihood function is preferred over the usual χ^2 as s depends on a number of variables and this will cause T to be non uniformly distributed with respect to χ^2.

Let us now assume that the linear equation for the shunt impedance Z is correct but incomplete because it fails to capture all the mechanical properties of the cochlea. To correct it, we add an extra term that will account for these mechanical properties. The equation now is

$$Z = \frac{\omega_{c0} M_0}{s}\left(s^2 + \delta s + 1 + m(s)\right)$$

which can also be expressed as an equation for λ:

$$\lambda^2 = \frac{\left(s^2 + \delta s + 1 + m(s)\right)}{(4N)^2}.$$

To gain some insight into the form of the function $m(s)$, we fix N and δ and iteratively calculate λ from equations (2) and (3). We then plot the imaginary versus the real part of $m(s)$ for varying s. The author approximates the resulting points by a circle of the form $m(s) = \rho e^{-2\pi\mu s}$, ρ and μ being real constants. Substituting into the equation for the shunt impedance gives

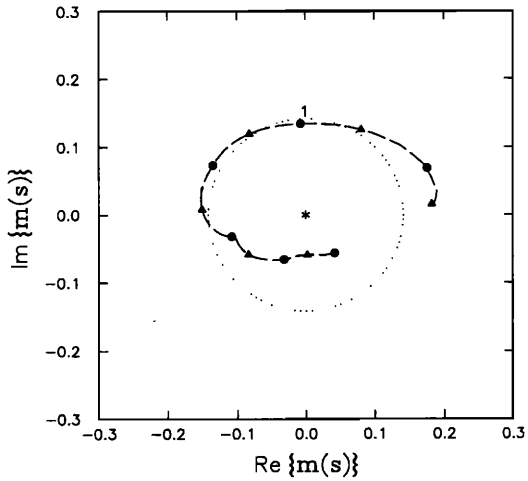


Figure 2: *The dashed curve is the measured values of $m(s)$ obtained from the transfer function T whereas the doted line is a circular approximation. The dots represent an equally spaced partition of the frequency domain.*

$$Z = \frac{\omega_{c0} M_0}{s}\left(s^2 + \delta s + 1 + \rho e^{-2\pi\mu s}\right).$$

Now using the Fourier transform as above yields the new oscillator equation for the velocity v of a point:

$$\ddot{v}(\theta) + \delta\dot{v}(\theta) + v(\theta) = \frac{\dot{p}(\theta)}{\omega_{c0} M_0} - \rho v(\theta - \psi),$$

where $\psi = 2\pi\mu$. Therefore, a section of the organ of Corti at position x behaves like a harmonic oscillator with angular frequency 1 and damping δ. It is also driven by two forces: one proportional to the derivative of the

pressure difference and one proportional to that section's velocity at the earlier time $\theta - \psi$. This delayed force is necessary to stabilize an otherwise unstable oscillator (recall that the damping is negative) and can be considered the active influence of the cochlea. An estimate of the time delay is given by

$$\frac{\psi}{\omega_c(x)} = 220\mu s,$$

where an approximation for μ (and therefore ψ as well) was obtained by fitting the model to experimental data. Zweig's model undoubtedly has a number of advantages. It describes the physical phenomenon much better than the simple harmonic oscillator model while at the same time remaining relatively simple for analysis and solution. However, an obvious point of improvement lies in the approximation of the form of the unknown function m by a circle. It is clear from the plot that an ellipsoid or a spiral would better fit the experimental data and result in a more accurate oscillator equation, without changing the assumptions of the original model or excessively complicating the calculations involved. We strongly believe that this point needs to be examined and reassessed in the future.

3 Physical interpretation

The processes involved in the perception of sound from a pressure wave are numerous. The most important process occurs in the cochlea, which makes the crucial transformation of a pressure wave into an electric signal, which can then be interpreted by the brain as sound. This transformation is a two-step process. First a small membrane inside the cochlea is made to oscillate by a propagating pressure wave. Then, the oscillation is registered by hair cells that activate the firing of an electric signal.

This oscillatory behavior of the cochlea was modeled by considering the cochlea as a tubular resonance cavity which encloses a membrane of oscillators that lies on the central horizontal plane. This plane divides the resonance cavity in two cavities, which are only connected at the far end of the cochlea. Research has shown that the oscillators responding to a pressure wave of a certain frequency have a position on the membrane which increases with decreasing frequency. In other words, higher frequencies are processed near the outer part of the cochlea and lower frequencies near the inner part. As we saw above, this frequency dependency of position was modeled with a delay differential equation. The model assumed a one-dimensional position x with domain $[0, 1]$, a total transversal pressure $p(x, t)$ and a transversal displacement $\xi(x, t)$ of the oscillators. The pressure p is actually the difference in pressure between the two cavities in the cochlea. The delay differential

equation in the paper is written in a slightly more general form as

$$p = m\ddot{\psi} + d\dot{\psi} + s\psi + s'\psi_{t-\tau}, \tag{4}$$

where $\psi_{t-\tau} = \psi(t-\tau)$ for some specific time $\tau > 0$. The differential equation without delay is that of a driven harmonic oscillator. There is extensive theory about this type of differential equation that leads us to expect certain parameter dependencies. INCAS[3] provided the following parameter dependencies

$$d = c_0\sqrt{sm}, \quad s' = c_1 s, \quad \tau = c_2\sqrt{\frac{m}{s}}, \tag{5}$$

where c_0, c_1 and c_2 are dimensionless constants.

Using these parameter dependencies we can transform equation (4) into a dimensional differential delay equation

$$\mathcal{P}(x,t) = \ddot{\xi}(x,t) + \frac{c_0 c_2}{\tau}\dot{\xi}(x,t) + \left(\frac{c_2}{\tau}\right)^2 \xi(x,t) + c_1\left(\frac{c_2}{\tau}\right)^2 \xi(x,t-\tau) \tag{6}$$

where a dot denotes partial differentiation with respect to time and $\mathcal{P}$ is defined as p/m, which has the same dimensions as acceleration.

Equation (6) fails to fully describe the ongoing process. All the oscillators are behaving independently and the pressure is known only at $x = 0$, since we only know the sound that enters the ear canal. The missing link are the cavities, which allow the existence of standing waves. However the Navier-Stokes equation for an incompressible, inviscid cochlear fluid with pressure differences only implies that an oscillator can influence these standing pressure waves. Therefore the Laplacian equation of standing waves becomes a Poisson equation

$$\frac{\partial^2 \mathcal{P}(x,t)}{\partial x^2} = \gamma\,\ddot{\xi}(x,t) \quad \text{with} \quad \gamma = \frac{\rho\, b_{BM}}{A/2} \tag{7}$$

and with constants ρ denoting the density of the cochlear fluid, b_{BM} the width of the membrane and A the diameter of the cochlea. The initial and boundary conditions for these differential equations are

$$\begin{cases} \xi(x,0) = 0\,, & \dot{\xi}(x,0) = 0 \\ \dfrac{\partial \mathcal{P}}{\partial x}(0,t) = P(t)\,, & \mathcal{P}(1,t) = 0 \end{cases} \tag{8}$$

for a known bounded function $P(t)$ with $P(t) = 0$ for $t < 0$. Together, equations (6) and (7) form a system of coupled ODEs describing how the cochlea responds to sound input. A comparison between the simulations generated using the model and the actual OAEs can be seen in figure 3.

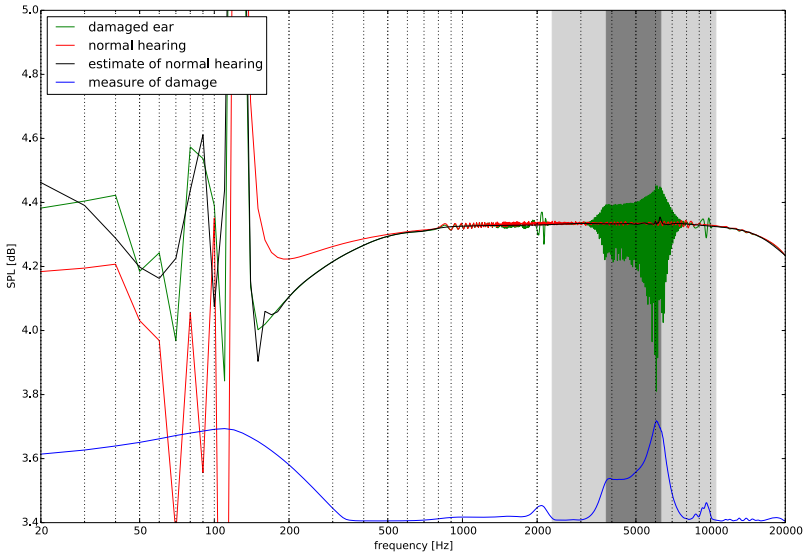


Figure 3: *Comparison of OAEs of normal hearing, damaged cochlea and estimated damage.*

3.1 The Fourier transform of the ODE

The incoming pressure wave is a representation of sound. It is therefore natural to use the frequency domain by means of the Fourier transformation in space. Let us use $\tilde{G}(x,\omega)$ to denote the Fourier transform of a function $G(x,t)$ with angular frequency ω. Then equations (6) and (7) are transformed into the system

$$\begin{cases} \tilde{\mathcal{P}}(x,\omega) = \left[-\omega^2 + i\omega\frac{c_0c_2}{\tau} + \left(\frac{c_2}{\tau}\right)^2\left(1 + c_1e^{-i\omega\tau}\right)\right]\tilde{\xi}(x,\omega) \\ \dfrac{\partial^2\tilde{\mathcal{P}}(x,\omega)}{\partial x^2} = -\omega^2\gamma\,\tilde{\xi}(x,\omega). \end{cases} \tag{9}$$

This system can be restated as

$$\begin{cases} \gamma\tilde{\mathcal{P}}(x,\omega) = \left[1 - i\frac{c_0c_2}{\omega\tau} - \left(\frac{c_2}{\omega\tau}\right)^2\left(1 + c_1e^{-i\omega\tau}\right)\right]\dfrac{\partial^2\tilde{\mathcal{P}}(x,\omega)}{\partial x^2} \\ \dfrac{\partial^2\tilde{\mathcal{P}}(x,\omega)}{\partial x^2} = -\omega^2\gamma\,\tilde{\xi}(x,\omega). \end{cases} \tag{10}$$

The first identity of this system has sufficient boundary conditions from the Fourier transformed boundary conditions of $\mathcal{P}$ in (8). The second identity

is satisfied by the first without imposing the boundary conditions of ξ in (8). Hence the unused boundary conditions must be satisfied automatically from the differential equation for $\tilde{\mathcal{P}}$. We can investigate this by solving the characteristic equation of the differential equation, which is

$$\gamma = \left[1 - i\frac{c_0 c_2}{\omega\tau} - \left(\frac{c_2}{\omega\tau}\right)^2 \left(1 + c_1 e^{-i\omega\tau}\right)\right] \lambda(\omega)^2. \tag{11}$$

The solution for $\tilde{\mathcal{P}}(x,\omega)$ and the boundary conditions $\xi(x,0)$ and $\dot{\xi}(x,0)$ are then equal to

$$\tilde{\mathcal{P}}(x,\omega) = -\frac{\tilde{P}(\omega)}{\lambda(\omega)} \frac{\sinh[(1-x)\lambda(\omega)]}{\cosh[\lambda(\omega)]} \tag{12}$$

$$\xi(x,t) = \frac{1}{\gamma\sqrt{2\pi}} \int_{-\infty}^{\infty} \lambda(\omega) \frac{\tilde{P}(\omega)}{\omega^2} \frac{\sinh[(1-x)\lambda(\omega)]}{\cosh[\lambda(\omega)]} e^{i\omega t} d\omega \tag{13}$$

$$\dot{\xi}(x,t) = \frac{i}{\gamma\sqrt{2\pi}} \int_{-\infty}^{\infty} \lambda(\omega) \frac{\tilde{P}(\omega)}{\omega} \frac{\sinh[(1-x)\lambda(\omega)]}{\cosh[\lambda(\omega)]} e^{i\omega t} d\omega \tag{14}$$

These identities imply that the cochlea has no spontaneous excitation modes without a forcing pressure $\tilde{P}(\omega) \neq 0$. Hence the condition $P(t) = 0$ for $t < 0$ guarantees the remaining boundary conditions due to causality, which is the property used in the Zweig paper to justify the delay term in equation (4) from data.

An advantage of the introduction of $\lambda(\omega)$ in equation (11) is the possibility to extend it to the form $\lambda(x,\omega)$ in a more general model where the constants c_0, c_1 or c_2 become functions of x. The function ξ is then still given by equation (13) by substituting $\lambda(x,\omega)$ instead of $\lambda(\omega)$. A second advantage of equation (11) is the existence of real and imaginary parts of λ, which allow not only oscillations, but decay as well. This decay will depend on the frequency, which reflects the frequency-position relationship of the oscillator response in the cochlea.

4 Input Functions

The physical model given in the introduction, derived in the theoretical background and explained in the physical interpretation is that of a resonance cavity, which resembles the cochlea and resonates due to an input pressure function. This input pressure is a representation of a sound wave. The reason INCAS^3 is interested in this model is to improve the current methods of diagnosing hearing loss. In these methods, a certain sound pulse is created and used with a certain procedure to determine the hearing loss. The quality

of the diagnosis is therefore highly dependent on the input sound pulse, measurement accuracy and measurement precision. Therefore, we tried to tackle the following problem. Is it possible to improve the input pulse to obtain a result faster without decreasing the accuracy or precision of the diagnosis?

Due to finite time measurements, the input function must be a pulse, which implies that it must be a function with compact support. Furthermore the Fourier transform of the input pulse may not have any zeros. This condition is necessary since hearing loss could occur at any frequency. Furthermore, hearing loss is determined as a spectral response loss, which implies that accuracy requirements need non-zero spectrum. Finally, we can observe only an interval of the frequency domain. It is therefore desirable that the Fourier transform of the input pulse is rapidly decaying in a known way outside the measurable interval of the frequency domain. A second desirable property would be the ability to modify the pulse such that a certain interval in the frequency domain can be examined. Hence, an input pulse must satisfy the following:

1. Compact support in the time domain.
2. Fourier transform without zeros.
3. Rapidly decaying Fourier transform.
4. Adjustable for having values above a given threshold for a given frequency interval.

A simple family of functions $G(t)$ which satisfy the requirements above are sums of a finite, symmetric interval part of the $\mathrm{sech}(t)$. Is is obvious that these functions have compact support. Additionally, their Fourier transform does not have zeroes due to the invariance of the sech under Fourier transform, which smoothes out all the zeroes due to the window. It is easy to see that they are rapidly decaying due to the properties of the sech by using the Riemann-Lebesgue lemma. Finally, we can adjust them at will since the sech is symmetric in the frequency domain and therefore sums of a symmetric finite interval part can cut away small frequency intervals for a given threshold.

Let us define by $\mathrm{rect}(a)(t)$ the unit function on the interval $[-a, a]$ and zero elsewhere. Then the function G and its Fourier transform $\tilde{G}$ are given by

$$\begin{aligned} G(a)(t) &= \sqrt{\frac{\pi}{2}}\mathrm{sech}(t)\mathrm{rect}(a)(t), \\ \tilde{G}(a)(\omega) &= \arctan\left(e^{a/(2\pi)-\omega}\right) - \arctan\left(e^{-a/(2\pi)-\omega}\right). \end{aligned}$$

Using the function G as a building block one obtains a new function S which has the property of selecting an interval in the frequency for which $\tilde{S}$ is a

threshold.

$$\begin{aligned} S(a,b)(t) &= G(a)(t) - \frac{1}{|b|}G(a)(t/b), \\ \tilde{S}(a,b)(\omega) &= \tilde{G}(a)(\omega) - \tilde{G}(a)(b\,\omega). \end{aligned}$$

If the threshold is equal to ϵ, then the interval endpoints are the positive

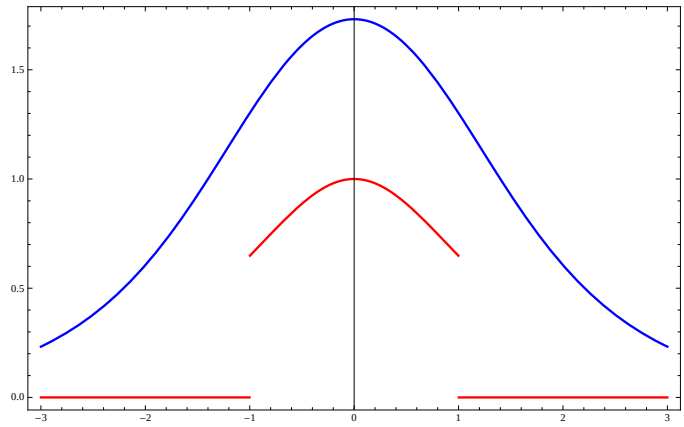


Figure 4: *Red line: plot of sech(t) multiplied with rect$(1)(t)$. Notice the compact support, the discontinuity at ± 1 and the absense of roots inside the window. Blue line: Fourier transform of the function above. This function also does not have any zeroes and is rapidly decaying.*

zeros of

$$\tan(\epsilon) = \frac{\sinh(a/\pi)\cosh(\omega) - \sinh(a/\pi)\cosh(b\,\omega)}{\sinh^2(a/\pi) + \cosh(\omega)\cosh(b\,\omega)}.$$

We can always find two positive zeroes for small enough ϵ. Hence S satisfies the properties of a desirable input pulse.

The existence of such a simple function that satisfies the properties needed is very important, as it will allow a relatively simple design and execution of experiments to locate where the damage lies in the frequency spectrum of the ear. Further research would help in refining the definition and use of the input function, thus drastically improving the way hearing damage diagnosis is performed.

5 Parameter estimation

5.1 Toy problem

Determining the value of a set of parameters using some experimental data about the solution of a problem is usually called the *inverse problem*. When experimental data are not available, numerical simulations may also be used. We will describe two different methods that are commonly used to attack inverse problems, namely the finite differences method and the adjoint method. Both share a common foundation, as in both cases we try to minimize an objective function that usually calculates the difference between the current set of parameters and the 'perfect' one. To illustrate these methods let us consider a simple example:

$$D(x)\frac{d^2T(x)}{dx^2} = 1, \quad x \in [0,1] \tag{15}$$

with boundary conditions $T(0) = 1$, $T(1) = 0$.

Suppose we have a given $\tilde{T}$ which satisfies (15), for some unknown $D(x)$. The inverse problem is to calculate the function $D(x)$ that corresponds to $\tilde{T}$. We define an objective function, F, which measures the difference between the exact solution $\tilde{T}$ and our approximation T; minimizing this function is now our goal, and the value of the parameter D corresponding to the minimum of the error function will be the best estimation, at least locally. The inverse problem is therefore tackled using an optimization procedure.

We will make use of gradient-based optimization algorithms, a subset of the class of line search methods, an optimization strategy based on two steps:

1. Find a direction along which F decreases rapidly.
2. Compute a step size which determines how far we should move along that direction.

It is obvious that successful use of a line search method requires the determination of both the direction and the step length.

Both strategies illustrated here, the finite differences method and the adjoint method, assume the gradient direction as the decreasing line, namely

$$D^{(k+1)} = D^{(k)} - \underbrace{\gamma}_{\text{step}} \underbrace{d_D F(T,D)}_{\text{direction}},$$

with d_D denoting the derivative with respect to D. However, these methods differ in the way they calculate the gradient. A first approach is to approximate it by finite differences over D. However, this includes the integration of n differential equations at each step, where n is the dimension of D. A more

sophisticated approach is to calculate the gradient using the adjoint method, which is significantly cheaper computationally, since at most two differential equations have to be integrated.

Using the procedure discovered above, starting from an initial guess D_0, we obtain a sequence $F(D_n)$ that satisfies

$$F(D_0) \geq F(D_1) \geq \cdots \geq F(D_k) \geq F(D_{k+1}) \geq \ldots$$

and converges to a minimum. A significant disadvantage of this method, as in every hill-climbing method, is the risk of getting stuck in a local minimum. Every minimum found is guaranteed to be a global minimum only if F is convex, which is usually either not true or difficult to prove.

5.2 Finite differences

The idea underlying this approach is the discretization of the problem with respect to the spatial variable. A second order approximation of the derivative of the objective function is calculated and used in the gradient descent method in order to find a local minimum. A *discrete* objective function is defined,

$$F(D) = \sum_{j=1}^{n} (T(x_j; D) - \tilde{T}(x_j))^2$$

where T is a vector containing the evaluation of the approximate solution (corresponding to the approximated parameter D) in each node of the discretisation and $\tilde{T}$ is a vector containing the evaluation of the exact solution in the same points.
At each step we compute the gradient using the formula

$$d_D F^{(k)} = \frac{F(D^{(k)} + \varepsilon) - F(D^{(k)} - \varepsilon)}{2\varepsilon}$$

and we then update our parameter value according to

$$D^{(k+1)} = D^{(k)} - \gamma \cdot d_D F^{(k)}.$$

Since the convergence speed of this method can be very slow for a constant step, it is possible to add an iterative method to better adapt the step length γ. One such possibility is through the *backtracking line search,* which is a good compromise between the two opposite goals of obtaining a step size γ which substantially reduces F and decreasing computational cost. A sample algorithm which geometrically reduces γ is the following:

choose $\gamma_0 > 0$ (generally $= 1$); $\rho, c \in (0, 1)$;
set $\gamma = \gamma_0$;
while $F(D^{(k+1)}) \geq F(D^{(k)}) + c \cdot \gamma \cdot d_D F^{(k)}$ **do**
  set $\gamma = \rho\gamma$;
end

5.3 Adjoint method

The adjoint method is a way of significantly decreasing the computational cost of calculating the gradient. An introduction to it will be presented here, more details can be found in the book of Vogel (2002).

Let $T = \begin{pmatrix} T_1 \\ T_2 \end{pmatrix} = \begin{pmatrix} T_1 \\ \dot{T}_1 \end{pmatrix}$. Equation (15) can then be written as $\dot{T} = \begin{pmatrix} T_2 \\ \frac{1}{D} \end{pmatrix}$ and the inverse problem can be stated as follows:

$$\begin{aligned}
\underset{D}{\text{minimize}} \quad & F(T; D) = \int_0^1 f(t, D, x)dx, \text{ where } f(T, D, x) = \int_0^1 (\tilde{T}(x) - T(x))^2 dx \\
\text{subject to} \quad & h(T, \dot{T}, D, x) = \dot{T} - \begin{pmatrix} T_2 \\ \frac{1}{D} \end{pmatrix} = 0, \\
& g_1(T(0), D) = T(0) - \begin{pmatrix} 1 \\ T_2(0) \end{pmatrix} = 0, \\
& g_2(T(1), D) = T(1) - \begin{pmatrix} 0 \\ T_2(1) \end{pmatrix} = 0.
\end{aligned}$$

Here D is a vector of unknown parameters, T is a function of x, $h(T, \dot{T}, D, x) = 0$ is an ODE in implicit form and $g_1(T(0), D) = 0$, $g_2(T(1), D) = 0$ are the boundary conditions, which are functions of some of the unknown parameters. Being a gradient-based optimization algorithm, the gradient

$$d_D F(T, D) = \int_0^1 [\partial_T f d_D T + \partial_D f]dx$$

has to be calculated. Unfortunately, it is often expensive to compute $d_D T$. The first step in solving this problem is to introduce the Lagrangian corresponding to the optimization problem defined above,

$$\mathcal{L} = \int_0^1 [f(T, D, x) + \lambda^T h(T, \dot{T}, D, x)]dx + \mu_1^T g_1(T(0), D) + \mu_2^T g_2(T(1), D).$$

Here λ is a vector of Lagrange multipliers depending on x, and μ_1 and μ_2 are vectors of multipliers corresponding to the boundary conditions. Since h, g_1, and g_2 are zero everywhere by definition, λ, μ_1 and μ_2 can be chosen

freely and we have $d_D\mathcal{L} = d_D F$. The main idea is to choose the values of the multipliers in such a way that the total derivative $d_D\mathcal{L}$ is easy to compute. Thus, the derivative of the Lagrangian is

$$\begin{aligned} d_D\mathcal{L} = \int_0^1 &[\partial_T f d_D T + \partial_D f + \lambda^T(\partial_T h d_D T + \partial_T h d_D \dot{T} + \partial_D h]dx \\ &+\mu_1^T(\partial_{T(0)} g_1 d_D T(0) + \partial_D g_1) + \mu_2^T(\partial_{T(1)} g_2 d_D T(1) + \partial_D g_2). \end{aligned} \tag{16}$$

The integrand contains the terms $d_D T$ and $d_D \dot{T}$, which are both hard to calculate. For the second term, we apply integration by parts

$$\int_0^1 \lambda^T \partial_{\dot{T}} h d_D \dot{T} dx = \lambda^T \partial_{\dot{T}} h d_D T\Big|_0^1 - \int_0^1 \left[\dot{\lambda}^T \partial_{\dot{T}} h + \lambda^T d_x\left(\partial_{\dot{T}} h\right)\right] d_D T dx.$$

Substituting this in (16) we obtain the expression

$$\begin{aligned} d_D\mathcal{L} \quad = \quad & \int_0^1 \left[\partial_T f + \lambda^T\left(\partial_T h - d_x \partial_{\dot{T}} h - \dot{\lambda}^T \partial_{\dot{T}} h\right)\right] d_D T \\ & +\partial_D f + \lambda^T \partial_D h dx \\ & +\mu_1^T\left([\partial_{T(0)} g_1 + \lambda^T \partial_{\dot{T}} h]_0 d_D T(0) + \partial_D g_1\right) \\ & +\mu_2^T\left([\partial_{T(1)} g_2 + \lambda^T \partial_{\dot{T}} h]_1 d_D T(1) + \partial_D g_2\right). \end{aligned}$$

Since we are free to choose the multipliers λ, μ_1 and μ_2, let us take

$$\mu_1^T = \lambda^T \partial_{\dot{T}} h|_0 (\partial_{T(0)} g_1)^{-1}$$

$$\mu_2^T = \lambda^T \partial_{\dot{T}} h|_1 (\partial_{T(1)} g_2)^{-1}.$$

This ensures that the first parts of the last two terms vanish. Furthermore, we choose λ such that

$$\partial_T f + \lambda^T(\partial_T h - d_x \partial_{\dot{T}} h) - \dot{\lambda}^T \partial_{\dot{T}} h = 0, \tag{17}$$

which saves us from having to calculate $d_D T$. With this choice of values for the multipliers we obtain

$$d_D\mathcal{L} = \int_0^1 [\partial_D f + \lambda^T \partial_D h]dx + \mu_1^T \partial_D g_1 + \mu_2^T \partial_D g_2. \tag{18}$$

The first order linear ODE h from the optimisation problem stated above can be rewritten as

$$h(T, \dot{T}, D, x) = \dot{T} - A(D)T - b(D),$$

where $A(D) = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$ and $b(D) = \begin{pmatrix} 0 \\ D^{-1} \end{pmatrix}$. It follows easily that $\partial_T h = A(D)$ and $\partial_{\dot{T}} h = I$, and $\partial_D h = \begin{pmatrix} 0 \\ D^{-2} \end{pmatrix}$. Furthermore, $\partial_D f$, $\partial_D g_1$ and $\partial_D g_2$ are zero. Substituting these information into (18) leads to

$$d_D \mathcal{L} = \int_0^1 \lambda^T \partial_D h dx = \int_0^1 \lambda_2(x) D(x)^{-2} dx.$$

and the adjoint equation becomes $\partial_T f - \lambda^T A(D) - \dot{\lambda}^T = 0$.

6 Conclusion

6.1 Summary of results

We have investigated several different ways to attack the problem of modelling hearing damage. They all show promise for further investigation.

Regarding the analysis of the given model, it is obvious that Fourier transforming the differential equations has several advantages. For instance, analysis in the frequency domain is often more natural when working with sound. Furthermore, solving the system is now easier, which can be very helpful for implementing a parameter estimation scheme. It is also worth noticing that parameter estimation may not be needed at all. This is because the motivation behind trying to estimate where the cochlea is damaged is to allow us to alter the sound entering the ear in a controlled way, by means of a hearing aid, so that the ear processes an input that is as close to normal as possible. Therefore, instead of trying to locate where the damage lies, it might be more fruitful to be able to calculate this corrected input signal directly. This may be a significantly hard problem in the time domain, perhaps harder than the parameter estimation itself, but using the frequency domain we showed there are ways to attack it.

The gradient descent method discussed in section 5 is one of the simplest ways of solving an inverse problem. Regardless whether the gradient itself is obtained through finite differences or the adjoint method, the choice of algorithm itself is also very important. Simple methods like gradient descent can have problems such as slow convergence or getting stuck at a local minimum. There exist however more sophisticated methods that may be used to remedy these problems. For example, the conjugate gradient method that keeps track of previous step directions could be tried. Additionally, attention should be paid to quasi-newton methods like BFGS that determine the objective function's Hessian.

6.2 Recommendations for future work

- Investigate whether choosing another function m to fit the experimental data leads to a better model of a damaged cochlea.
- Add the calculation of the OAE to the Fourier-based approach.
- Check whether the input function proposed in section 4 works as expected for real OAE measurements.
- Perform parameter estimation with the current cochlear model using the framework laid out in this paper.

References

C. R. Vogel. *Computational Methods for Inverse Problems.* Number 23 in Frontiers in Applied Mathematics. SIAM, Philadelphia, 2002.

George Zweig. Finding the impedance of the organ of corti. *Journal of the Acoustical Society of America*, 89:1229–1254, 1991.

Index of Authors