Proceedings of the 114th European Study Group
Mathematics with Industry

SWI 2016

Nijmegen, January 25 – 29, 2016

Editors:
Eric Cator
Ross J. Kang

Cover design by Melissa Oliemeulen

Contents

Preface

These are the scientific proceedings of the 114th Study Group Mathematics with Industry (Studiegroep Wiskunde met de Industrie) held at Radboud University in Nijmegen, January 25th to 29th, 2016.

In this volume, the participants of SWI 2016 have provided their account of the week's developments, aimed at a scientific audience. Each of the six groups has prepared a contribution that presents the problem they worked on, the approaches they attempted or used, and the results that they obtained.

In a companion popular proceedings written by Arnout Jaspers, an account meant for a general audience is given in Dutch.

The organisers of SWI 2016
Eric Cator and Ross J. Kang

Flower power: Finding optimal flower cutting strategies through a combination of optimization and data mining

Han Hoogeveen
Utrecht University
j.a.hoogeveen@uu.nl *

Jakub Tomczyk
University of Sydney
jstomczyk@gmail.com

Tom C. van der Zanden
Utrecht University
T.C.vanderZanden@uu.nl

Abstract

We study a problem that plays an important role in the flower industry: we must determine how many mother plants are required to be able to produce a given demand of cuttings. This sounds like an easy problem, but working with living material (plants) introduces complications that are rarely encountered in optimization problems: the constraints for cutting such that the mother plant remains in shape are not explicitly known.

We have tackled this problem by a combination of data mining and linear programming. We apply data mining to infer constraints that a cutting pattern, stating how many cuttings to harvest in each period, should obey, and we use these constraints in a linear programming formulation that determines the minimum number of mother plants necessary. We then consider the problem of maximizing the total profit given the number of mother plants and show how to solve it through linear programming.

Keywords: data mining, linear programming, cutting patterns, column generation.

Dümmen Orange is a leading company in breeding and development of cut flowers, potted plants, bedding plants and perennials with over a century of experience in the horticultural industry. In addition to a large marketing and sales network, Dümmen Orange has a strong network of production locations. In these production centra so-called mother plants are planted and grown for a large number of varieties. When these mother plants are ready, cuttings are harvested during a period of approximately 16 weeks, after which the mother plants are removed.

[0]In this paper we report on the project carried out for Dümmen Orange in the context of the study group 'Wiskunde met de Industrie' (Math with Industry). Next to the authors of the report, the group consisted of Yella Klemm, Kevin Laros, Jan Nelissen, and Brian Wismans from Dümmen Orange, Norbert Mikolajewski (RU Nijmegen), and Jagna Wiesniewska (VU Amsterdam).

*corresponding author

These cuttings are sold to growers, who either place orders beforehand, or place orders during the harvesting. For each variety, the majority of sales takes place in the 'peak weeks', which is a period of approximately 10 weeks; the company has reasonably accurate demand forecasts per week available.

Dümmen Orange experienced the following problem. For each variety, the number of mother plants to be planted is decided on the basis of sales forecasts to which a buffer of 10% is added. When orders come in, contracts are concluded with the growers guaranteeing that the required number of cuttings will be delivered at the desired time. When the harvesting starts, at some point in time the availability of the buffer of 10% is reported to the sales agents, who then try to acquire orders for selling these additional cuttings. Unfortunately, when they are very successful, too many cuttings are required, and the mother plants cannot keep up this pace for too many weeks in a row, which results in a shortage in later periods. This led Dümmen Orange to the question of when to report the availability of the buffer, and possibly to change its size.

Dümmen Orange posed this problem at the study group 'Wiskunde met de Industrie' SWI2016 . In close contact with Dümmen Orange we figured out that we had to address the following research questions:

1. Model how the number of cuttings harvested in previous weeks influences the potential number of cuttings that can be taken from a mother plant in the current and future weeks.

2. Determine how many mother plants should be planted to meet the predicted demand.

3. Determine how many cuttings to offer for sale in each week (and thus how many to cut).

We have looked at this problem for just a single variety of plant in isolation, where we ignored any random disturbances initially. For the variety that we studied, we were provided with the predicted demands and the average number of cuttings per mother plant for each week from 2005 onwards. Unfortunately, detailed information concerning the effects of taking cuttings on the potential mother plants was not available.

This paper is organized as follows: In Sections 1 and 2, we describe how to answer the second and third question by linear programming, for which we need the answer to the first question, which is solved in Section 3 using techniques from data mining. We conclude by providing computational experiments in Section 4 and draw conclusions in Section 5.

[0]see http://www.ru.nl/math/research/vmconferences/swi-2016/

1 Solution approach: linear programming

Since the number of cuttings taken from the mother plants in previous weeks influences the potential yield for the current week in a way that was unknown to us, we decided to work with feasible *cutting patterns*. Here, a cutting pattern describes for each of the 16 weeks the average number of cuttings that are taken from a mother plant; since it is an average (taken over all mother plants), this number can be fractional. For the variety that we studied the typical yield per week was 2 or a little less; as an example a possible cutting pattern could be $\{2.0; 1.8; 1.9; 2.0; \ldots\}$, which indicates that in the first week on average 2.0 cuttings are taken, in the second week 1.8, etc. To be a bit more general, from now on we use T to denote the number of weeks during which we take cuttings. After consulting the experts from Dümmen Orange we found out that the time at which the mother plants were planted made no difference with respect to their potential yield of cuttings, and therefore we do not need to make the cutting patterns depend on the time of planting. Observe the close resemblance between our cutting problem and the standard cutting stock problem (see for example Gilmore and Gomory (1961) and Gilmore and Gomory (1963)). In the cutting stock problem, however, we consider items with different lengths, whereas we now have identical items that are cut in different periods.

Suppose that we know the set of all n possible, feasible cutting patterns. In that case we can solve the problem of determining the required number of mother plants by formulating it as a linear programming problem. We represent cutting pattern j by the parameters a_{jt} that indicate the average number of cuttings harvested in week t ($t = 1, \ldots, T$), when a plant is cut according to pattern j, for $j = 1, \ldots, n$. Define x_j ($j = 1, \ldots, n$) as the number of mother plants that are cut according to cutting pattern j. If we denote the expected demand in period t by b_t ($t = 1, \ldots, T$), then we can formulate the problem of determining the minimum number of mother plants as a linear programming (LP) problem as follows:

$$\begin{aligned} &\min x_1 + \ldots + x_n \\ &\text{subject to} \\ &\textstyle\sum_{j=1}^{n} a_{jt} x_j \geq b_t \quad \forall t \\ &x_j \geq 0 \quad \forall j \end{aligned}$$

The solution of this LP program gives you a lower bound on the number of mother plants that have to be planted. Dümmen Orange can decide to add more (for example to have a buffer to guard against disturbances in the production and/or sales process). Note that, although the x_j variables should attain integral values only since these correspond to numbers, it is sufficient to solve the problem by solving the LP-relaxation (where the integrality constraints are relaxed) and round up the outcome values, since the total of the x_j values is big and at most T of them will get a value different from zero (we will see later that we need only one cutting pattern in an optimal solution). Moreover, if the time of planting the mother plants would make a difference with respect to the yield of cuttings, then this can easily be incorporated

in this model by making the x_j variables dependent on the time of planting.

Suppose that the management of Dümmen Orange has decided on the number M of mother plants to be planted. We can then solve the problem of determining how many cuttings to offer for sale per week in a similar fashion by formulating it as an LP again. We assume here that we know for each week t how many cuttings we can sell additionally (which we denote by D_t) and the profit p_t that we gain per cutting sold additionally. Next to the decision variables x_j, we introduce decision variables y_t $(t = 1, \ldots, T)$ that will indicate the number of additional cuttings to be sold in period t. Just like we did for x_j, we ignore the integrality of the y_t variables. We then get the following LP formulation:

$$\begin{gathered}
\max \textstyle\sum_t p_t y_t \\
\text{subject to} \\
\begin{aligned}
\textstyle\sum_{j=1}^{n} a_{jt} x_j - y_t &\geq b_t \quad \forall t \\
\textstyle\sum_{j=1}^{n} x_j &\leq M \\
0 \leq y_t &\leq D_t \quad \forall t \\
x_j &\geq 0 \quad \forall j
\end{aligned}
\end{gathered}$$

If we solve this LP, then we find the cutting strategy that maximizes the total profit given the number of mother plants M. This LP can also be used to find the value of M that maximizes the total profit; we can then simply make M a decision variable, but we have to include the cost of planting M mother plants in the objective function. Furthermore, we can refine the model in case the profit per additional cutting sold is not constant but decreases when more get sold.

2 Generating cutting patterns

In our derivation of the LP problems of the previous section we have assumed that we know all n possible, feasible cutting patterns. Even if we restrict ourselves to a_{jt} values that are multiples of 0.1, there are so many possible cutting patterns that it is neither feasible, nor efficient to generate them all. Fortunately, we can apply the technique of *column generation*, which was invented by Ford and Fulkerson L. R. Ford and Fulkerson (1958) and Gilmore and Gomory Gilmore and Gomory (1961, 1963). Here we solve the LP problem while taking only a small number of feasible cutting patterns into account; we can start with any subset of the cutting patterns, as long as the feasible region is non-empty. After having solved the current LP, we add variables (which correspond to feasible cutting patterns) that will improve the quality of the solution, until we can guarantee that we have found the optimum of the LP for the entire set of feasible cutting patterns.

We will work this out for the first LP, in which we minimize the number of mother plants needed to cover the demand. It is well-known from the theory of linear programming that in case of a minimization problem adding a new variable x_j will improve the quality of the solution only if its *reduced cost* is negative. When we solve the current LP, then we find the non-negative shadow prices; let π_t denote the shadow price

corresponding to the constraint that we produce at least b_t cuttings. The reduced cost of a variable x_0 that corresponds to using a given cutting pattern $(a_1, \ldots, a_T)$ is equal to

$$1 - \sum_{t=1}^{T} a_t \pi_t,$$

where the 1 corresponds to the cost coefficient of x_0 in the objective function. Instead of just checking for each feasible cutting pattern $(a_1, \ldots, a_T)$ whether its reduced cost happens to be negative (for which we need to know all feasible cutting patterns), we solve the so-called *pricing problem*, the goal of which is to construct a feasible cutting pattern $(a_1, \ldots, a_T)$ with minimum reduced cost. Note that the values a_t $(t = 1, \ldots, T)$ have become decision variables, and we must choose these such that their combination forms a feasible cutting pattern. To that end, we need a way to describe when a set of values $(a_1, \ldots, a_T)$ constitutes a feasible cutting pattern. Moreover, this knowledge should be cast in such a format that we can use it to solve the pricing problem efficiently. To that end, we infer these constraints by applying techniques from *data mining* to the data on the average number of cuttings harvested per week in the years 2006-2015.

3 Data mining

Data mining is used to retrieve relations from the data. There is a large interaction between data mining and operations research, but it is mainly a one way connection: techniques and algorithms from operations research are applied in data mining Olafsson et al. (2008). We want to apply data mining to *learn constraints* that will be incorporated in the model explicitly, after which we can apply the techniques from operations research. As far as we know, such an approach has not been conducted before. For example, Li and Olafsson Li and Olafsson (2005), who use data mining to derive dispatching rules for a complex production scheduling problem, state that *the idea of this data mining approach to production scheduling is to complement more traditional operations research approaches.*

The domain expert at Dümmen Orange gave several constraints on what constitutes a feasible cutting pattern. For instance, for the variety that we consider one can obtain a maximum of 2.0 cuttings per mother plant in a given week; hence, we find the constraint that $a_t \leq 2.0$ for all $t = 1, \ldots, T$. After having harvested the maximum of 2.0 cuttings in week t, the mother plants have to recover, which can be formulated in the constraint that $a_t + a_{t+1} \leq 3.9$ for all $t = 1, \ldots, T-1$. Furthermore, a pattern that alternates between cutting near the maximum and not cutting very much (e.g. a pattern such as $\{2.0; 1.4; 2.0; 1.4, \ldots\}$) is not feasible either; it turned out later that we must introduce a constraint of the form $a_t + a_{t+2} + a_{t+4} \leq 5.71$.

It is apparent that the number of constraints required is very large, and a different set of values is needed for every species. Since obtaining these values from domain experts would be very time consuming, we experimented with data mining

to automatically derive the constraints. To this end, Dümmen Orange provided us with data specifying the average number of cuttings harvested per mother plant in the period 2006-2015. We scanned the data to identify all constraints of the form $a_{t_1} + a_{t_2} + \ldots + a_{t_k} \leq X$, for all k-tuples $(t_1, \ldots, t_k)$ with $t_1 < t_2 < \ldots < t_k$, where $k \leq 6$ and $t_k - t_1 \leq 10$; here X is set to the maximum value that is observed in the historical data for the left hand side.

Even though the bound of each constraint is set to the maximum value observed, the fact that very many such constraints work together ensures that only realistic cutting patterns will satisfy the constraints. The domain expert confirmed that the cutting patterns that we identified in this way appeared feasible. Note that to obtain more conservative constraints one can take X to be the k-th percentile instead of the maximum of the observed values. However, because our data sets were of limited size taking this approach was not necessary, and would have resulted in overly conservative estimates. However, it could be useful in case a larger training data set is used (which may contain more outliers).

Another possible shortcoming of our data mining model might be that we do not have the data available that we need. We used the data concerning the number of cuttings that were actually harvested instead of the maximum number of cuttings that could have been harvested. Hence, the constraints that were inferred might be too restrictive: it might not consider a certain feasible cutting pattern, simply because this cutting pattern has not been used before. We leave these issues to the experts, who if necessary can perform some experiments to test cutting patterns.

Below, we have listed a small excerpt of the list of constraints that we obtained using data mining.

$$
\begin{array}{lcl}
a_t & \leq & 2.0 \\
a_t + a_{t+1} & \leq & 3.9 \\
a_t + a_{t+2} & \leq & 3.85 \\
a_t + a_{t+1} + a_{t+2} & \leq & 5.75 \\
a_t + a_{t+2} + a_{t+4} & \leq & 5.71 \\
a_t + a_{t+1} + a_{t+2} + a_{t+3} & \leq & 7.6 \\
a_t + a_{t+1} + a_{t+3} + a_{t+4} & \leq & 7.61 \\
a_t + a_{t+1} + a_{t+2} + a_{t+3} + a_{t+4} & \leq & 9.46 \\
a_t + a_{t+1} + a_{t+3} + a_{t+4} + a_{t+5} & \leq & 9.21 \\
a_t + a_{t+1} + a_{t+2} + a_{t+3} + a_{t+4} + a_{t+5} & \leq & 11.15 \\
a_t + a_{t+1} + a_{t+3} + a_{t+4} + a_{t+5} + a_{t+6} & \leq & 10.9
\end{array}
$$

Note that we have linear constraints only. Hence, the resulting pricing problem of finding the minimum reduced cost, which was equal to

$$1 - \sum_{t=1}^{T} a_t \pi_t,$$

subject to the constraints we identified using data mining is just another linear program, and hence can be solved very efficiently. Since the feasible region described

by the constraints is convex, we have that each convex combination of a set of cutting patterns satisfies these constraints, and hence corresponds to a feasible cutting pattern again.

Theorem 3.1. *Let $x^* = (x_1^*, \ldots, x_n^*)$ denote an optimal solution to the linear program of minimizing the number of mother plants. Then there exists an equivalent solution in which we use only one cutting pattern $C = (C_1, \ldots, C_T)$.*

Proof. Define $M = \sum_{j=1}^{n} x_j^*$. We construct this cutting pattern C by taking the weighted average of all cutting patterns, where we use x_j^*/M as our weight function, for $j = 1, \ldots, n$. Hence, we have that

$$C_t = \sum_{j=1}^{n} a_{jt} x_j^* / M.$$

Since all weights are non-negative and add up to 1, this is a convex combination, and therefore C is a feasible cutting pattern. If we cut all M mother plants according to this cutting pattern, we get the same yield as we get for the optimal solution x^*. □

As a result, we can solve the LP of minimizing the required number of mother plants in a more efficient way using binary search. In each iteration we test whether harvesting b_t cuttings in period t $(t = 1, \ldots, T)$ from a given number Q of mother plants corresponds to a feasible cutting pattern. The resulting cutting pattern has $a_t = b_t/Q$ $(t = 1, \ldots, T)$, and all that is left is to check whether it satisfies the constraints. If this is the case, then we decrease Q, and if it fails the test, then we increase Q.

In fact, we do not even need binary search. Recall that we have to check whether the values $a_t = b_t/Q$ $(t = 1, \ldots, T)$ satisfy the constraints, like $a_t + a_{t+1} \leq 3.9$. This is equivalent to checking whether $Qa_t + Qa_{t+1} = b_t + b_{t+1} \leq 3.9Q$, which implies that Q must be greater than or equal to $(b_t + b_{t+1})/3.9$. For each constraint from data mining we can obtain a lower bound on Q in this way, from which we find that the minimum number of mother plants required is equal to the maximum of these lower bounds.

Note that this approach works only if we can guarantee that a convex combination of a set of cutting patterns is feasible. If we would need additional non-linear constraints to describe a feasible cutting pattern, then we have to resort to column generation again. The pricing problem would then not be solvable as an LP any more, but we could apply an approach such as Constraint Programming.

Now we consider the second LP, in which we optimize the choice of the number of cuttings that must be harvested in period t. The LP-formulation is as follows:

$$\begin{gathered}
\max \textstyle\sum_t p_t y_t \\
\text{subject to} \\
\textstyle\sum_{j=1}^{n} a_{jt} x_j - y_t \geq b_t \quad \forall t \\
\textstyle\sum_{j=1}^{n} x_j \leq M \\
0 \leq y_t \leq D_t \quad \forall t \\
x_j \geq 0 \quad \forall j
\end{gathered}$$

We can use Theorem 3.1 again to show that we can use a single cutting pattern C. As a consequence, we can once again solve this problem without generating cutting patterns. We introduce the variables z_t that indicate the number of cuttings that we harvest in period t ($t = 1, \ldots, T$); we must have that $z_t \geq b_t$ and we sell the remainder at a price of p_t per cutting. Since we use a single cutting pattern, we cut $a_t = z_t/M$ cuttings per mother plant. Then we can rewrite the LP as

$$\max \textstyle\sum_t p_t z_t$$
$$\text{subject to}$$
$$b_t \leq z_t \leq D_t \quad \forall t$$
$$z_t = M a_t \quad \forall t$$
$$\text{'the variables } a_t \text{ form a feasible cutting pattern'}$$

Notice that we have to subtract the constant $\sum p_t b_t$ from the objective value to make the outcome values equal. Even in the case where M is a decision variable, the problem can still be reformulated so as to avoid column generation by working with the variables $z_t = M a_t$ only. To that end, we multiply the constraints describing the cutting patterns like $a_t + a_{t+2} \leq 3.85$ with M, such that we obtain the constraint $z_t + z_{t+2} \leq 3,85M$. Obviously, we have to include the cost of growing M mother plants to the objective function. We further remark that our approach can also be used in case we refine the model by offering the possibility of selling up to $b_{t,1}$ cuttings for price $p_{t,1}$, up to $b_{t,2}$ cuttings for price $p_{t,2}$, etc.

4 Computational experiments

4.1 First approach

To make our mathematical formulation more tangible for the domain experts, we created a graphical user interface around the LP formulation, which allows the user to enter a set of cutting patterns as training data (note that we used the historic data to that avail), and then experiment with various scenarios. The user can specify a number of mother plants and the (predicted) demand levels for each week, and then see whether the demands can be met given this number of mother plants, and how much (if any) additional capacity there is in each week. The software can also calculate the minimum number of mother plants required to meet a specific set of demands.

The red line shows the demands entered by the user, while the green line shows the maximum number of cuttings we could take each week, while still being able to meet the demands. The gray line shows the absolute maximum number of cuttings available in a single week, but note that it is never feasible to take this many cuttings, except for in the last week (when the demand has dropped to zero).

We also implemented an interface for the harvesting stage, which aids in determining how many additional cuttings to offer for sale (on top of the amounts that have already been (pre-)ordered). This is depicted in Figure 2. For each week, the

Figure 1: GUI for the planting stage.

user can enter how many cuttings have been ordered so far, as well as (an estimate of) the number of cuttings for which there is additional demand. Additionally, the user can enter (for each week) a profit for each additional cutting sold, and a penalty for not delivering cuttings that have already been ordered. Given these values, the program calculates an optimal strategy for selling the additional cuttings.

The red and green lines have the same meaning as before, while the blue line represents our program's advice on selling additional cuttings.

We found that this implementation was a quite powerful tool for conveying our mathematical model to the domain experts.

Figure 2: GUI for the selling stage.

4.2 Second approach

A limitation of the model proposed so far is that it does not take into account unpredictable effects, such as the effects of weather or disease. We performed an experiment where the right hand sides of the constraints were perturbed randomly as a potential approach to getting more robust solutions. However, even though it is possible to determine a good relation between environmental conditions (sun, rain, etc.) and the condition of the mother plants, we cannot use this in our computation of the number of mother plants to plant, since the mother plants have to be planted in advance, and it is impossible to give a reasonable prediction of the environmental conditions at the time of harvest. On the other hand, when we get more data, then we could estimate the fluctuations in outcomes. Therefore, we propose an expert-based approach as well.

In the following Matlab based GUI, the user can infer constraints from data mining results or experience and estimate their variability (error). The estimated variability is used to generate scenarios, which are essential to provide confidence intervals. To generate scenarios we sample from a uniform probability distribution, where the range depends on the estimated variability. Intuitively, variability should decrease with number of summed values in constraints. Obviously, we do not take into account rare, but possible events like wars, droughts, volcanic eruptions (which may block deliveries) or plant diseases.

Based on the provided constraints and variability data, possible scenarios are simulated. They are used to calculate the number of mother plants and a buffer (in percentages) needed to cover, for instance 98% of scenarios. See Figure 3.

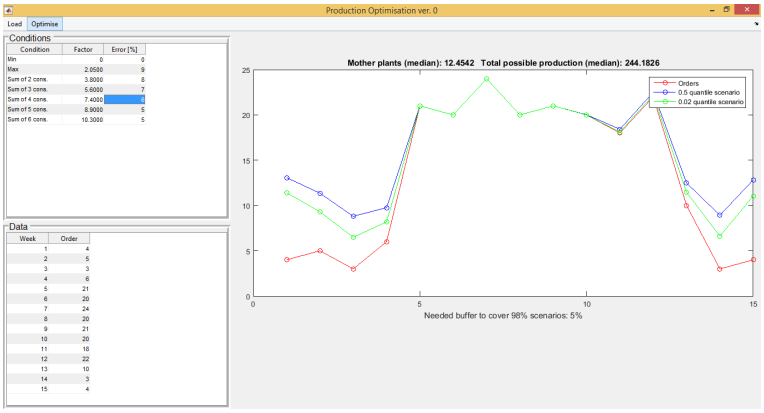

Figure 3: Print screen of Matlab based GUI which is used to estimate number of mother plants and buffer needed to cover 98% of scenarios.

In order to automate a process of estimating the buffer and number of mother plants one has to use history in combination with the advice of experts at least once at the beginning. Due to high number of varieties (thousands) it would be very

time consuming. Moreover, estimation of variability from historical data might be challenging. We believe that expert judgement is essential in this approach.

5 Conclusions

The problem of Dümmen Orange is quite different from other applications because of the laws of nature that have to be obeyed: the output is not constant, but decreases over time if you require too much in the beginning. We have attacked this problem by techniques from mathematical programming, where we use techniques from data mining to cover the lack of technical constraints. Especially this latter part seems to be new and very useful for dealing with these kinds of problems. The linear programs are very flexible and easily solvable, which offers great potential for future use by Dümmen Orange, especially when looking at combinations of varieties.

Finally, we want to thank the organizers of the study group 'Wiskunde met de Industrie' for their work, which has resulted in our collaboration with Dümmen Orange.

References

P. C. Gilmore and R. E. Gomory. A linear programming approach to the cutting-stock problem. *Operations Research*, 9(6):849–859, 1961. doi: 10.1287/opre.9.6.849. URL `http://dx.doi.org/10.1287/opre.9.6.849`.

P. C. Gilmore and R. E. Gomory. A linear programming approach to the cutting stock problem: Part ii. *Operations Research*, 11(6):863–888, 1963. doi: 10.1287/opre.11.6.863. URL `http://dx.doi.org/10.1287/opre.11.6.863`.

J. L. R. Ford and D. R. Fulkerson. A suggested computation for maximal multi-commodity network flows. *Management Science*, 5(1):97–101, 1958. doi: 10.1287/mnsc.5.1.97. URL `http://dx.doi.org/10.1287/mnsc.5.1.97`.

X. Li and S. Olafsson. Discovering dispatching rules using data mining. *Journal of Scheduling*, 8(6):515–527, 2005. ISSN 1099-1425. doi: 10.1007/s10951-005-4781-0. URL `http://dx.doi.org/10.1007/s10951-005-4781-0`.

S. Olafsson, X. Li, and S. Wu. Operations research and data mining. *European Journal of Operational Research*, 187(3):1429 – 1448, 2008. ISSN 0377-2217. doi: http://dx.doi.org/10.1016/j.ejor.2006.09.023. URL `http://www.sciencedirect.com/science/article/pii/S037722170600854X`.

Predicting Early Bulking in Potatoes

Fetsje Bijma Alessandro Di Bucchianico* Eric Cator
Henk Don Patrick Hafkenscheid Jakub Nowotarski
Bijan Ranjbar-Sahraei

Abstract

Early bulking of potatoes is important for potato breeders for several reasons, including flexibility in scheduling and less influence of weather conditions. In this paper we use statistical models to model tuber growth in order to identify which existing varieties allow for early bulking. We also investigate which genetic properties (SNP's) may be important for early bulking.

KEYWORDS: early bulking, SNP, variance stabilizing transformation, linear regression, sparse data, elastic net

1 Introduction

In this section we provide the necessary background for the problem, and state the main research questions.

1.1 Company background

HZPC (`www.hzpc.nl`) is the world leading developer and seller of high quality seed potatoes. It is an internationally operating Dutch company with head quarters in Joure. HZPC has 320 employees, 800 growers and 55 breeders on 19 locations. To serve its customers better, HZPC has an R&D department in Metslawier. The main goal is to develop new varieties of potatoes that meet the needs of consumers and industrial partners by advanced data-driven breeding techniques.

1.2 Problem description

Tuber bulking is the 4th growth stage in the development of a potato (see Figure 1). Tuber cells expand with the accumulation of water, nutrients and carbohydrates. Tubers become the dominant site for deposition of carbohydrates and mobile inorganic nutrients.

*Corresponding author.

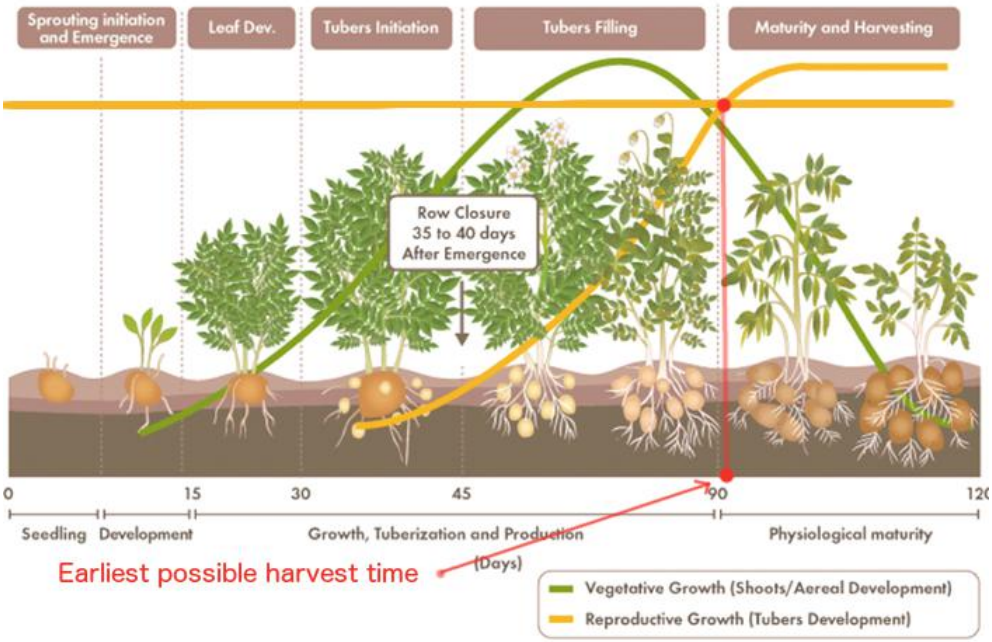

Figure 1: Early bulking (source: `www.sqm.com`).

HZPC wishes to breed early bulking varieties in order to be able to harvest as early as possible high quantities of tubers with desirable sizes . The benefits of early bulking are the opportunities to have new harvests as early as possible, more flexibility with scheduling (since it takes less until harvest), and less influence of climate factors such as rain and humidity.

In order to search for early bulking varieties in an efficient way, there is a need for a statistical model that predicts the tuber filling in length and volume in time per variety and to find the genetic parameters that have a significant effect on early bulking performance. Furthermore, a simple and efficient strategy should be designed for selection of early bulking varieties. More concretely, we will address the following research questions:

Research questions:

Question 1 How to model tuber growth and predict which varieties are more likely to bulk early?

Question 2 How to identify important genetic properties for early bulking?

With respect to Question 1, HZPC is interested in the mass of harvested tubers with

tuber size 45 mm or more as well as subtraits of varieties like tuber filling (length and diameter of tubers) and the number of tubers per plant. Tuber size is commonly defined by potato breeders in terms of "square size", i.e. the length of the side of the smallest square in which the tuber fits. A complication for the development of models for Question 1 is that the number of tubers and sizes of the tubers are correlated (if there are more tubers, then they are likely to be smaller).

For Question 2, the goal is to find models with causal explanations in terms of DNA differences so that one effectively and efficiently measure early bulking in breeding programs. We note that a complication here is that important traits are usually determined by several genes simultaneously.

2 Available data

In this section we describe the data that we could use to address the research questions. In Subsection 2.1 we describe the field data for Question 1, while in Subsection 2.2 we describe the genetic data for Question 2.

2.1 Tuber data

Data of trial fields of the the years 2011-2015 were made available to us by HZPC. These trial fields were laid out using the following experimental design (see also Figure 2):

- 100 varieties of tubers.
- 4 different harvest times.
- 2 replicates per harvest time.

The data set of the year 2015 is very detailed and contains for every individual tuber length, width, height, square size (as defined in Subsection 1.2, weight and volume. For the previous years (2011–2014) only summarized data were available through the number of tubers and total weight for each field plot, square size category and harvest time.

2.2 Genetic data

The genetic data set made available by HZPC is in the form of frequency counts of SNP's (single nucleotide polymorphisms, pronounced "snips"). A SNP is a genetic variation at a specific position in the genome in the form of the replacement of a single nucleotide at a specific base location. The SNP's in the data set only allow two different alleles (i.e., two different nucleotides), so a 0 indicates no variation (the most frequent nucleotide) and a 1 indicates the genetic variation (the alternative nucleotide, which must occur in at least 1% of the population. The values in the data set are integers from 0 to 4, since the SNP's are determined for the 4 chromosomes of a tuber

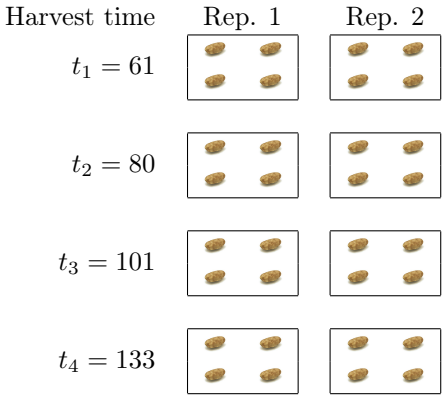

Figure 2: Experimental design

(2 from the father and 2 from the mother). SNP data are available for 113 varieties for the years 2011-2014, and for only 12 of the varieties of the 2015 field trial.

3 Tuber growth modelling

Before we try to make a statistical model for tuber growth, we performed a small exploratory data analysis to check for data quality issues, variation between individual tubers as well as get an idea of the time evolution of tuber growth. Figure 3 shows that there is considerable variation between the individual tubers within varieties. There is no clear difference between the two replicates (indicated by different colours). The main interest of HZPC is the weight of tubers with a square size of at least 45 mm. After examining various plots, we found that a log-log relationship seems to be a suitable model for tuber size and tuber weight since the data points in the plots lie reasonably well on a straight line and the deviations from the straight line are less than for the other standard relationships that we tried out (see Figure 4). So the log-log transformation is also a variance stabilizing transformation. Other plots showed a moderate plot effect, i.e., there is some variation between the weights of tubers of the same variety but planted on different parts (plots) of the experimental field. Since the main interest of HZPC is the total weight of tubers with square size at least 45 mm, we decided to fit a joint model for log-weight and log-square size as function of time and number of tubers instead of model for weight and time. This model allows us to predict yield as function of time. In view of the considerable variation between tubers, we decided to model every tuber individually. Based on Figure 3 we assume a quadratic function as a simple form for the time evolution. To be more precise, we

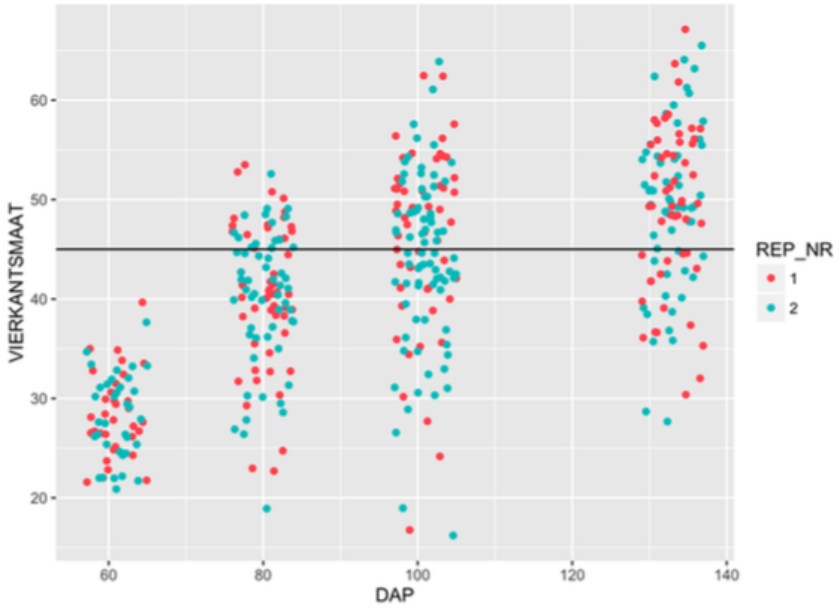

Figure 3: Scatter plot of tuber size ("square size") as function of days after planting

fitted the following linear regression model[1]. Define for each variety v the following quantities:

- $Y_1^v(t)$ = log of **square size** of the tubers at time t
- $Y_2^v(t)$ = log of **weight** of the tubers at time t
- $N^v(t)$ number of tubers belonging to the same potato plant at time t.

Then our model is

$$(Y_1^v(t), Y_2^v(t)) = \begin{pmatrix} 1 & t & t^2 & N^v(t) \end{pmatrix} \begin{pmatrix} \beta_{11}^v & \beta_{12}^v \\ \beta_{21}^v & \beta_{22}^v \\ \beta_{31}^v & \beta_{32}^v \\ \lambda_1^v & \lambda_2^v \end{pmatrix} + (\varepsilon_1^v(t), \varepsilon_2^v(t))\,., \qquad (1)$$

where $(\varepsilon_1^v(t), \varepsilon_2^v(t)) \sim \mathcal{N}((0,0), \Sigma^v)$. We obtained estimates for the parameters β, λ and Σ by using maximum likelihood.

In order to predict the total weight of a potato plant, we multiplied the estimates $N^v(t)$ into the model and compute for each t the expected total weight of big tubers (e.g., tubers with square size at least 45 mm). A graphical representation of our results is presented in Figure 5.

[1]Note that although the time evolution is described as a quadratic function, the parameters appear in a linear way in the regression function.

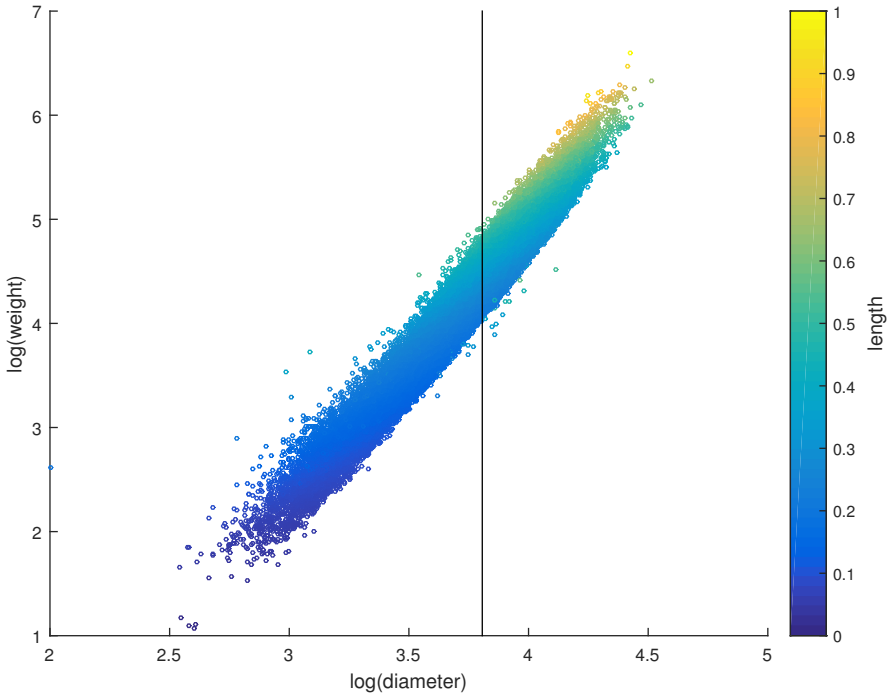

Figure 4: Log-log plot of tuber size ("square size") and weight as function of days after planting

4 Genetic properties and early bulking

In the previous section we made models to predict the early bulking properties of existing varieties. In order to develop new varieties with favourable early bulking performance, it is important to study the genetic properties of early bulking varieties. Therefore we now turn to the genetic data described in Subsection 2.2. Our approach consists in trying to build a linear regression model with the SNP's as independent (explanatory) variables and the total weights per variety of the tubers with square size at least 45 mm. Since the data set contains 113 varieties and 11763 SNP's, we have many more parameters than observations. Thus we cannot perform an ordinary linear regression. However, we may safely assume that only a few SNP's may influence the early bulking performance of a variety. In other words, a sparse model may be appropriate. Sparse models may be fitted using special variants of linear regression, in which the least squares criterion is replaced by another criterion that puts an extra penalty on the number of selected explanatory variables. These variants are sophisticated counterparts of the traditional backward and forward model selection methods. The first example of such a method is the lasso introduced by Tibshirani (see Tibshirani (1996), which makes use of an ℓ_1-criterion rather than the standard

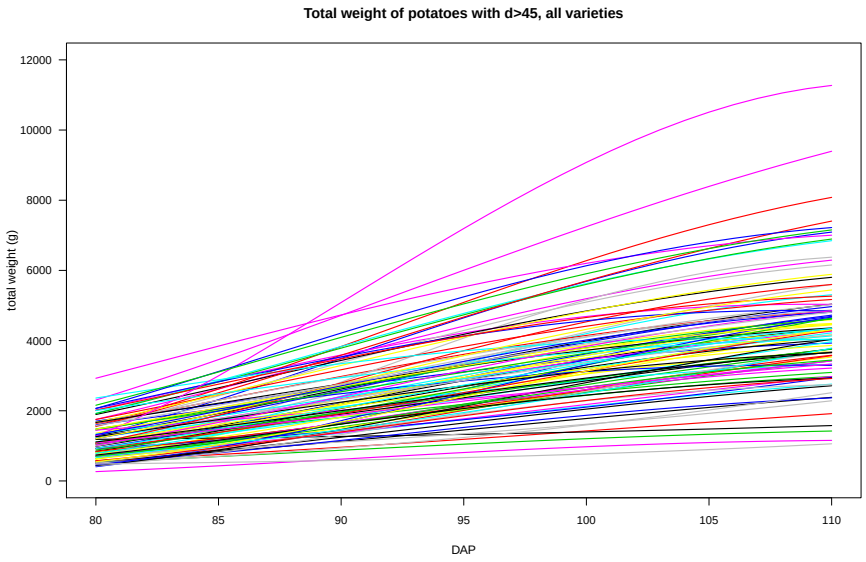

Figure 5: Time profile of total weight of big tubers

ℓ_2-criterion used in the least squares approach. Further refinements are the elastic net in which the criterion involves both an ℓ_1-term and an ℓ_2-term, with an automatic choice of the relative weights of these terms (see Zou and Hastie (2005)) and least angle regression which features a continuous way of including explanatory variables (see Efron et al. (2004)). We refer to Hesterberg et al. (2008) for a gentle and lucid introduction to these advanced regression methods and to Hastie et al. (2015) for an accessible monograph on methods for sparse data like in our case (i.e., we expect that only a few SNP's will influence early bulking performance).

For our analysis we used the data of the $2011-2014$ field trials since they contain SNP data for 113 varieties. It should be noted however, that there are several missing values. Certain SNP's may be difficult to obtain since the maximum number of missing values per SNP equals 51 and there are 266 SNP's with more than 10% missing values.

We followed a two-step approach:

1. apply multiple imputation to fill in missing values

2. apply elastic net to preselect important SNP's

The elastic net regression method requires complete cases. One could leave out the SNP's with missing values, but that would lead to an underestimation of the standard error. Therefore we decided to apply imputation. One should choose a suitable imputation method by considering the possible mechanism causing the missing data. In our case the SNP observations were obtained per variety using a complicated

procedure to extract the relevant genetic data. Due to the complicated nature of the extraction procedure, determination of SNP's may fail at certain locations in the genome. Since there is no indication that this depends on the variety, the missing data mechanism that is appropriate is "missing completely at random" as introduced in Rubin (1976). We chose "predictive mean matching" as imputation method, since it is likely that varieties with similar SNP values for non-missing data entries will have similar SNP entries for missing data (see Van Buuren (2012) for a comprehensive overview of both theoretical and practical issues related to imputation). The analysis was performed using the statistical software `R` with the following packages:

1. the mice package for Step 1
2. the glmnet package for Step 2

In our analysis we used the following linear regression model:

$$\begin{pmatrix} W_1 \\ \vdots \\ W_{113} \end{pmatrix} = \begin{pmatrix} 1 & x_{1,1} & \dots & x_{1,11673} \\ \vdots & \vdots & & \vdots \\ 1 & x_{113,1} & \dots & x_{113,11673} \end{pmatrix} \begin{pmatrix} \gamma_1 \\ \vdots \\ \gamma_{11673} \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_{113} \end{pmatrix}, \qquad (2)$$

where

- W_i is total weight of all tubers with square size 45+ of variety i ($i = 1, \dots, 113$).
- $x(i, j)$ is the value for SNP-j and variety i.

We selected elastic net as regression method instead of the lasso, since the lasso can only select as many variables as there are observations and it does not behave well in case of correlated independent variables (cf. (Hastie et al., 2015, Section 4.2)) so that is hard to make valid statements about which SNP's are important indicators for early bulking performance. Although there is SNP data for 113 varieties, we could only use the 69 varieties because of lacking early bulking data. We used elastic net with $\alpha = 0.5$ in order to have an equal weight of the ℓ_1- and ℓ_2-penalties, since this gave the best result. Cross-validation was used to obtain an optimal value of the λ parameter in the elastic net. After performing the elastic net analysis, we first removed all parameters (SNP's) that had a zero estimate which yielded a list of 140 SNP's worth investigating further. A further look at the results revealed some spurious effects caused by the effect that some SNP's had only 1 observation for a certain value and the remaining observations for one other value or for which there was only value of the SNP (in other words, such SNP's were constant for all varieties and thus no inference could be made for the effect of these SNP's). We removed these SNP's after doing the imputation and the elastic net analysis because it was much easier to remove this SNP's in the relatively small list of SNP's with complete data that remained. Note that we decided not to remove several SNP's with only 2 possible values, one value of which has only 2, 3 or 4 observations or SNP's with several values, but one which has only 1 observation (these SNP's could also lead to spurious effects, see e.g. Figure 6 for an illustration of the possible leverage effect in the form of box plots).

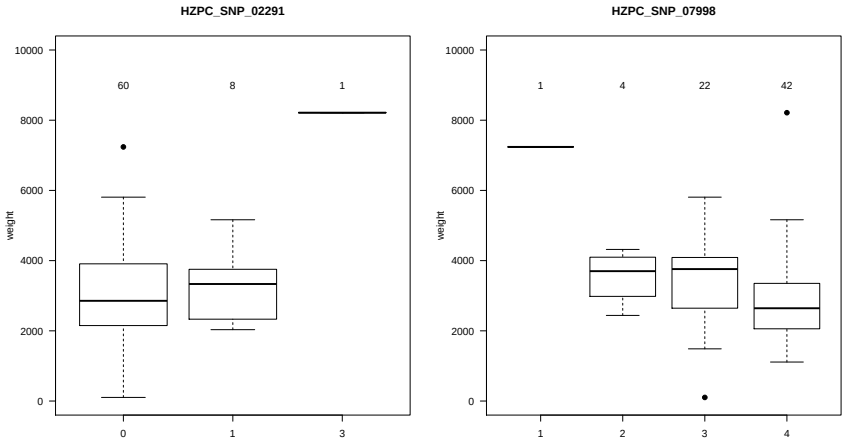

Figure 6: Selected SNP's, possibly spurious effects

After these steps, the list of potentially interesting SNP's reduced to 35 SNP's, only 1 of which has a positive effect (higher number of chromosomes with a modification give a higher weight) and the remaining 34 have a negative effect. So the data is indeed sparse, since this means that at most 1% of the SNP's seem to influence the early bulking performance. In view of possible correlations between the SNP's, one should be careful in identifying which SNP's influence early bulking performance.

5 Discussion

In this section we summarize our main conclusions and results. Based on these conclusions and results, we also indicate we also give some recommendations for future research.

5.1 Key insights

We list our key insights for the questions separately.

Question 1 How to model tuber growth and predict which varieties are more likely to bulk early?

1. There is a linear relation between log-weight and log-square size
2. A $\log - \log$ transformation has a variance stabilizing effect (this is important as constant variance is one of the assumptions of standard linear regression models)
3. There is a moderate plot effect

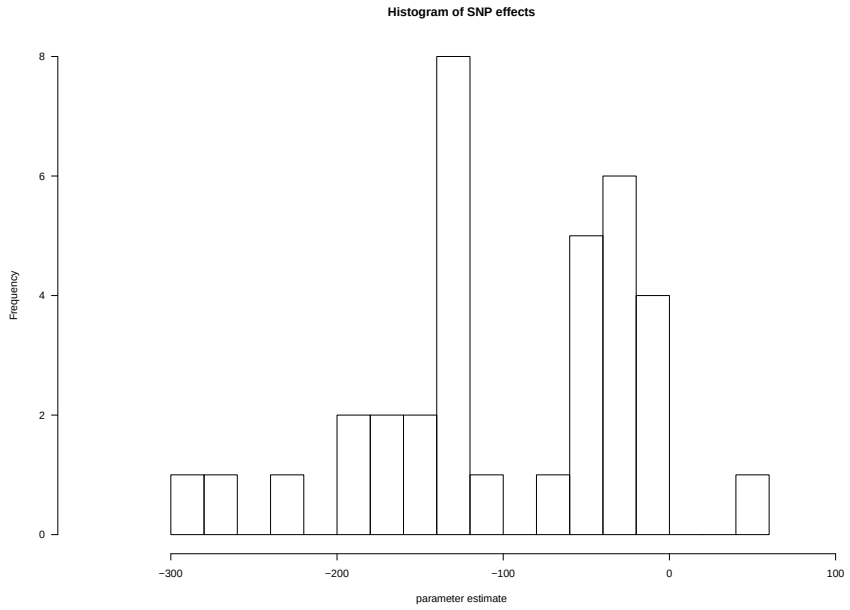

Figure 7: Selected SNPs, positive and negative effects

4. The number of tubers stabilizes after the second harvest time.

Question 2 How to identify important genetic properties (SNP's) for early bulking?

1. Do not include SNP's that are almost constant for all varieties since they may lead to spurious results.
2. A regression analysis with SNP's as predictor variables is possible in spite of the fact that there are many more SNP's than varieties using the elastic net approach
3. At most 1% of the SNP's show a significant effect.
4. Both positive and negative effects occur.

5.2 Future research

There are several ways in which research on the two main questions of this paper could be pursued.

For the growth modeling question, a further investigation of model accuracy should be undertaken and a sensitivity analysis should be performed with respect to harvest times. The growth model should also be enhanced with a plot effect in view of

the observed moderate plot effect. One should explore different shapes for the time profiles, e.g., $\sqrt{t}$.

For the genetic properties question, one should further explore the elastic net model. First of all one should perform modeling diagnostics in particular the normality assumption. In case of normality problems, one could try Box-Cox transformation or model the joint distribution. A further analysis of the relative importance of the significant SNP's is also important. There are several ways to do this, ranging from applying recently developed post-selection inference methods (see e.g., Section 6.3 of Hastie et al. (2015) and Chapter 11 of Bühlmann and van de Geer (2011) to variants of the lasso and elastic net that allow for group effects (i.e., methods that single out groups of highly correlated parameters, see e.g., Bach et al. (2012) for an overview of relevant methods). Apart from these statistical approaches, we also recommend to use more refined genetic data than SNP's.

Acknowledgement We would like to thank Jacqueline Verdijck-Lamers, Hans van Doorn, Rob Klooster, and Pieter-Jelte Lindenbergh of HZPC for introducing us to this interesting problem and for their extensive support to us during the SWI week.

References

F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends® in Machine Learning*, 4(1):1–106, 2012.

P. Bühlmann and S. van de Geer. *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer, Berlin, 2011.

S. van Buuren. *Flexible Imputation of Missing Data*. CRC Press, Boca Raton, Florida, 2012.

B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32(2):407–499, 2004.

T. Hastie, R. Tibshirani, and M. Wainwright. *Statistical Learning with Sparsity*. CRC Press, Boca Raton, Florida, 2015.

T. Hesterberg, N. Choi, L. Meier, and C. Fraley. Least angle and ℓ_1 penalized regression: A review. *Statistical Surveys*, 2:61–93, 2008.

D. Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 58(1):267–288, 1996.

H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

Fog detection from camera images

Roberto Castelli [*] Peter Frolkovič [†] Christian Reinhardt [‡]
Christiaan C. Stolk [§] Jakub Tomczyk [¶] Arthur Vromans [‖]

Abstract

Fog is one of the most dangerous weather types with more fatalities than winter storms. It is in the interest of general public that a precise, predictive and accurate fog density map with high spatial resolution can be created. Currently, the definition of fog as used by national weather services is so detailed and technical that the fog can be identified only at a few locations by means of the prescribed light scattering experiments. With the rising availability of cameras in public places such as airports, streets and highways, a large amount of data on the occurrence of fog becomes available to researchers. In this article we describe methods for determining not necessary only the existence of fog, but sometimes a visibility distance - a type of optical penetration length - as well. We will show that digital cameras can be a reliable alternative or complementary method for creating fog visibility maps when processing of image data is used.

Keywords: fog detection, Dark Channel Prior, edge detection, colour detection, visibility distance

1 Introduction

Fog is the weather phenomenon of light scattering particles - usually water droplets - suspended in air causing an attenuation of light and therefore a severely reducing a visibility of objects. The sudden appearance of fog - especially a dense fog - can lead to such reduced visibility that transportation networks can be affected or even fully compromised: for example massive car collisions resulting in long traffic jams, grounding of airplanes or even closing of airports and reduced speed of trains to

[*]VU Amsterdam
[†]STU Bratislava
[‡]VU Amsterdam, corresponding author
[§]UvA Amsterdam
[¶]Univ. Sydney
[‖]TU Eindhoven

prevent derailment. Some of these effects can be alleviated or even prevented when a transportation network can adjust to a fog density map of high spatial resolution accuracy by issuing warnings or decreasing the speed limit. Unfortunately such a density map needs a dense network of sensors that are capable of detecting the fog and measuring the visibility distance, a network weather services are now lacking.

Current fog detection systems measure the amount of scattering of a collimated beam of infrared light to determine the Meteorological Optical Range (MOR): the distance at which a collimated beam of incandescent light with a light colour of 2700K has reduced to an amount of 5% of the emitted flux. In the Netherlands there are 25 sites, see Figure 1 for the locations, capable of determining the MOR resulting in a spatial resolution that is significantly larger than typical length scales on which fog varies that can be as low as a few meters in the neighbourhood of surface water. Therefore a new and complementary method based on new data sources is needed.

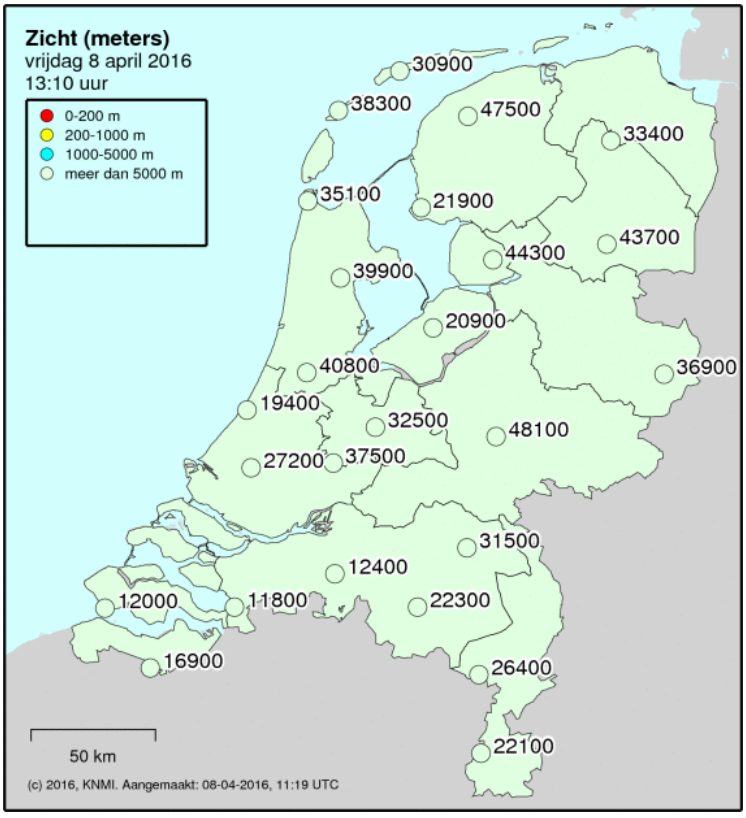

Figure 1: Map of the Netherlands showing the Meteorological Optical Range (MOR) measured at 25 sites capable of determining the MOR. Image from the real time updated public KNMI website: http://knmi.nl/nederland-nu/weer/waarnemingen.

A possible new source of data are public cameras. The rising spread of public cameras for control, security and safety allows for a much denser network of fog detecting sensors. For example The Netherlands had about 2200 state owned traffic cams der Staten Generaal (2010) in 2010, which would yield a spatial resolution of about 2.5 km if the cameras are distributed uniformly. Unfortunately the meteorological definition of fog is incompatible with the data gathered by cameras. Cameras do not see a constant light colour of 2700 K nor have a reference level of the emitted flux. Therefore different properties of fog have to be used in a camera involved in a fog detection system. For validation such fog system must correlate to the MOR and to the fog detection and classification based on human perception.

Properties of fog and their measurement As we stated before fog is the weather phenomenon of light scattering particles, suspended in air causing an attenuation of light and therefore a severe reduction of the visibility of objects. This description already hints to several characteristic properties. The most important parts of this description are the “light scattering particles" and the “reduced visibility of objects". The first part implies that the light of a source can be seen from a direction different than the source direction. As a result the total amount of light scattered into one specific direction will lead to a shift of an object colour towards white or grey. This property will be called Colour Level Shift. Furthermore, the attenuation due to the scattering leads to a gradual change of the fog colour from white to black depending on the attenuation length of the fog, the thickness of the fog layer and the intensity of the light source.
The second part of the description indicates a loss of resolution. It indicates that visibility is a relative quantity depending on the no fog perception of an object. An object becomes “fuzzy" and less detailed. This property will be called Shape Level Decrease.
In general both the Colour Level Shift and the Shape Level Decrease can be interpreted as some combinations of smearing and averaging effects. The smearing implies the existence of a diffusion process like scattering, which is the reason why objects are perceived “fuzzy" and with shifted colour levels, and the averaging indicates the direction of the Colour Level Shift: towards a specific grey level.
The MOR detection method is fully based on scattering and therefore it is by definition a colour level method: a decrease of flux in a specific small wavelength interval will indicate a colour level shift. The MOR detection method does not determine the loss of resolution or the absolute change of colour. Therefore camera data can complement the MOR detection method by determining both the loss of resolution and the absolute change of colour. A loss of resolution can be quantified by edge detection realized e.g. via gradient thresholding, high level wavelet transforms or total variability measures. The colour shift can be quantified by comparison between the RGB-channels of camera.

2 Fog detection based on Dark Channel Prior

In this chapter we introduce fog detection methods that are based on so called Dark Prior Channel from RGB colour images.

2.1 Description of available data

Although in general a video data can be available for our purposes, we note that the video footage is in principle a sequence of photographs, where each photograph is only shown for a very short time interval, usually too short to be perceived as a single photograph. Therefore we will only discuss the datasets consisting of single pictures. Each colour picture consists of three channels - Red, Blue and Green (RGB) - where each channel is a picture: an intensity map of the light received after passing through a specific band pass filter. The combination of the three channels yields the real life colour picture. Current camera technology is typically based on digital data obtained from a CCD (Charged Coupled Device), where the CCD is an array of integrating capacitors Rieke (2009). Each pixel of the picture is identified with a single integrating capacitor. The amount of charge collected by the capacitor is a direct measure of the intensity of the light. The relation is linear except for high values of charge. Current CCDs use a pixel with three integrating capacitors, one for each of the RGB channels Kitchin (2009). A CCD will give an electronic signal, the read-out signal, that consists of sequence of voltage spikes, one for each pixel channel. The data is therefore immediately in an analogue format which is easily and automatically converted into a digital signal.
Camera data will therefore consists of three digital RGB channel data sets in our study. In particular, the methods that will be discussed in this paper are applied to pictures provided by Koninklijk Nederlands Meteorologisch Instituut (KNMI), see Figures 2 or 4 later for an illustration. The pictures obtained from camera images are taken from a single location - KNMI institute terrain at De Bilt, Utrecht, the Netherlands (52.0990 N, 5.1766 E) - and pointed in a single steady direction towards the horizon (NNE). The pictures have a size of about 60 degrees wide and 40 degrees high with the horizon centered at about 18 degrees from the bottom, in pixel sizes 768×562. The temporal resolution is 10 minutes.
Complementary to the camera data the KNMI provided the Meteorological Optical Range (MOR) values of the same weather station location at the same times. The MOR values are in meters and are determined with the same temporal resolution. However the MOR is determined for the air directly at the location of the detector, while the camera has a solid angle to probe with a certain angular resolution resulting in multiple probes of fog of locations at least several tens of meters away from the camera. We assume that the fog is spatially homogeneous on the visible length scales and therefore probed in the same way by the MOR detector and the camera.

Transmission and Dark Channel Prior If we see the i-th image channel as an intensity density mapping I_i, $i = r, g, b$, then the density mapping can be decomposed into two mappings: the transmission mapping and the air scattering mapping, see e.g. Fattal (2008); Narasimhan and Nayar (2000, 2002). The transmission mapping is the perfect visibility image (or the scene radiance) J_i weighted with a transmission density t indicating the amount of transmission of the medium - in this case air. The air scattering mapping is the additive complement of the transmission mapping depending on the global atmospheric radiance A_i indicating the amount of intensity of air radiance being scattered in the direction of the camera.

$$\mathbf{I}(\mathbf{x}) = \mathbf{J}(\mathbf{x})t(\mathbf{x}) + \mathbf{A}(\mathbf{1} - t(\mathbf{x}))), \tag{1}$$

Mathematically speaking, the mappings $\mathbf{I} = [I_r, I_g, I_b]^\top, \mathbf{J} = [J_r, J_g, J_b]^T, \mathbf{A} = [A_r, A_g, A_b]^\top$ are defined on $[1, n] \times [1, m]$, the image of size $n \times m$ pixels, and with their values in $[0, 1]^3$, the relative colour intensities for each RGB channel. Remark that the RGB intensity is rescaled to 1 instead of the to the usual value of $2^B - 1$ for B-bits colour coding.
We are interested in the transmission coefficient $t \in [0, 1]$, since $1 - t$ is a measure of the amount of fog at the location depicted by the image pixel. Therefore one must be able to remove the fog from the image and create the scene radiance image $\mathbf{J}$. The procedure for doing this is called dehazing, because it is the inverse operation of applying fog or haze to an image He et al. (2011).
The objective of dehazing is to estimate $\mathbf{J}$, $\mathbf{A}$ and t in (1) from a single image $\mathbf{I}$. Naturally this procedure is a priori ill-posed since the output is 7/3 times greater than the input from the image. Therefore the relation (1) cannot be solved without extra constraints.

It was empirically observed in He et al. (2011) that patches in haze-free outdoor (day) images in the non-sky regions have very low intensities in at least one channel at some pixels belonging to the patch. These very low intensity pixels are due to large deviations in the intensity of a channel, which is by itself a measure of object resolution (pixel to pixel deviations) and transmission (channel to channel deviations). One expects that for foggy (day) images the scattering causes both a decrease in the object resolution as well as a colour shift to white or grey. Note that the grey scale colours are by a definition unbiased to any of the RGB channels. Therefore the RGB channels must have small deviations in the intensity of the channels, which can be interpreted as a loss of resolution (pixel to pixel) and colour shift (channel to channel).

Consequently, we can introduce the dark channel J_{dark}, which is the minimum over all channels of the minimum of all pixels in a (small) neighbourhood, a patch $\Omega(\mathbf{x})$, centered at a pixel $\mathbf{x}$,

$$J_{dark}(\mathbf{x}) := \min_{c \in \{r,g,b\}} \left(\min_{\mathbf{y} \in \Omega(\mathbf{x})} (J_c(\mathbf{y})) \right), \tag{2}$$

The dark channel is therefore a prior knowledge for dehazing. Note that this Dark Channel Prior is depending on the choice of a patch. If the patch is too large, then

the Dark Channel Prior will be almost uniform in the image, while a too small patch will go beyond the effective resolution of the image causing a Dark Channel Prior with almost the same variability as the original RGB channels.

The scene radiance image is the "no fog" transmission image, which is assumed to have zero values in the dark channel prior, i.e. $J_{dark} = 0$. Therefore the minimal values over all channels for the observed image are fully caused by the scattering mapping. Thus using (1) and supposing an estimate $\hat{\mathbf{A}}$ of $\mathbf{A}$ is known, we can estimate the transmission density mapping by

$$\hat{t}(\mathbf{x}) = 1 - \omega \min_{c \in r,g,b} \left(\min_{\mathbf{y} \in \Omega(\mathbf{x})} \left(\frac{\mathbf{I}_c(\mathbf{y})}{\hat{\mathbf{A}}_c} \right) \right), \tag{3}$$

where $\omega \in [0, 1]$ is a constant parameter introducing a small amount of haze to preserve a correct perception of distant objects. The haze indicated by the factor $1 - \omega$ can be attributed to other effects than scattering by water vapour, such as Rayleigh scattering of air, thermal deviations of the refractive index, or lens problems such as defocussing, chromatic aberration and astigmatism F.L. Pedrotti (2007). In our applications we set $\omega = 1$, since the camera is assumed to have no lens problems and the unobstructed view distance of 250 meter (a typical value in our test images) is assumed to be too small to allow other natural scattering effects.

To determine $\hat{\mathbf{A}}$ we pick the top 0.1% brightest pixels in the dark channel and then the pixels with highest intensity in the input image $\mathbf{I}$ to estimate the atmospheric light $\hat{\mathbf{A}}$, see He et al. (2011) for more details

In the following fog detection methods we make use of smoothed transmission t. The smoothing is permormed using Guided Filter, where we filter $\hat{t}$ and the filtering process is guided by $\mathbf{I}$ He et al. (2011).

In next sections we present particular fog detection methods based on Dark Channel Prior and transmission image. To obtain them for images in our computations we have used an available Matlab implementation, see Tierney (2014).

2.2 Classification Tree methods for fog detection

We assume we have obtained a smoothed transmission t for all pixels of the image from RGB data. Afterwards we compute the average of the smoothed transmission for each row. The resulting function of one variable indicates the transition between sky/air and the ground. One expects that a fog will create a smooth transition between the two, while clear days will have a sharp distinction between the two. In Figures 2 - 5 one can see examples how a clear day and a foggy day will change the transmission function. Hence this horizontal averaged smooth transmission function can be a good indicator for fog.

To test this approach we compute horizontal average for smoothed transmissions t for 4458 images from October 2015 (the dataset *Oct15*) and for 2554 images from

Figure 2: Image with a fog.

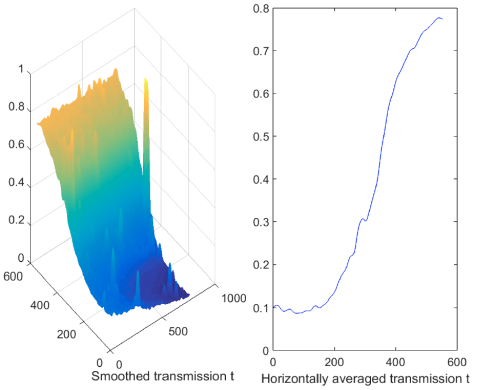

Figure 3: Smoothed transmission t and horizontal averaged function.

November 2015 (the dataset *Nov15*). Thanks to MOR method we have accurate estimation of visibility for these two datasets. We discard images for which visibility measurement is not available. Our goal is to be able to distinguish 3 classes:

- Class 1 - visibility $\leq 250m$,
- Class 2 - visibility $> 250m$ and ≤ 1000 m,
- Class 3 - visibility >1000 m.

Based on the MOR data one can easily determine to which class an image belongs, see Table 1.

Table 1: Number of images for each class and each dataset based on MOR data

	Class 1	Class 2	Class 3
Oct15	171	194	4092
Nov15	3	17	2451

However to determine these classes we intend to use the image datasets only. Therefore to distinguish between the three classes we use machine learning techniques on image data only and the predetermined partition of the images by the MOR data.

We proceeded as follows. We randomly partition the images into training and validation sets for each dataset. The 50% of each dataset is used for the training and the rest for the validation. We report in Figures 6 - 9 the results for two machine learning techniques: Single Classification Tree (SCT) and Bagged Classification Trees (BCT) Breiman et al. (1984) .

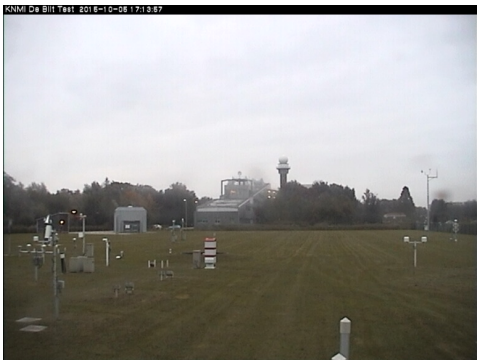

Figure 4: Image without fog.

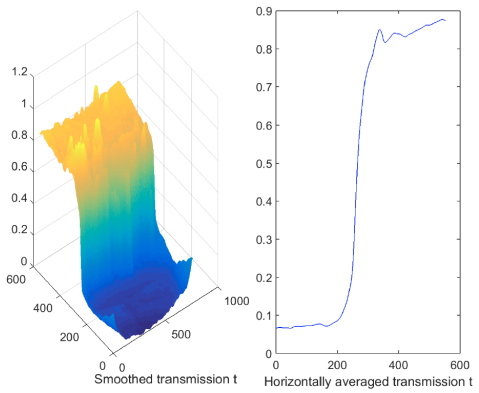

Figure 5: Smoothed transmission t and horizontal averaged function.

We find that the BCT outperforms the SCT for the November 2015 dataset in both Class 1 and 3. In Class 2 both methods are equally bad with 1 in 4 images wrongly classified.

For the October 2015 dataset the both methods are equally correct with a wrong classification of only 1 in 6 of the Class 1 (dense fog) images, 2 in 5 of the Class 2 (moderate fog) images, and 1 in 100 of the Class 3 (no fog) images. However due to the low amount of images with Class 1 and 2 classification it is premature to conclude that the methods are useful for fog classification.

We note that from the point of safety it is not problematic if a method has a bias for a higher probability on false positives towards lower Classes (more fog) images. However from the point of view of disruption, public awareness, believability and costs for society such a method is problematic if the bias is significant. Therefore a machine learning method should be combined with another fog detection method to decrease false positives and false negatives.

2.3 Transmittance Method

A second method exploiting the transmission function, called the transmittance method, is based on two consequences of the model (1).

The first one is the diffusion of the air region in the image into the ground region. This can cause an effective lowering of the horizon in pixel height. In the transmission function this effect can be seen as a shift of the location of the largest transmission jump to lower pixel height values.

The second consequence is the smoothing of the colour level due to the air intensity

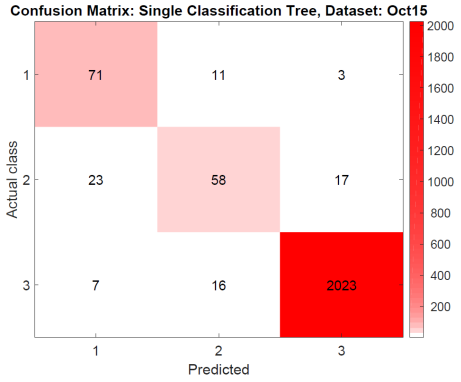

Figure 6: The SCT for dataset *Oct15*.

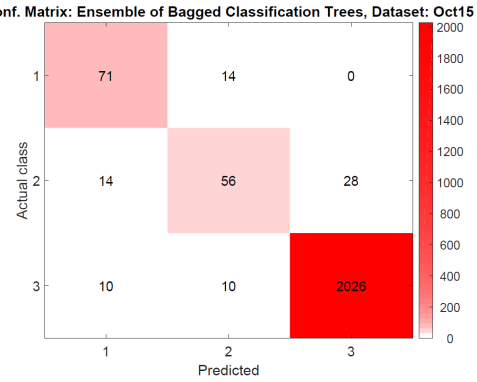

Figure 7: The BCT for dataset *Oct15*.

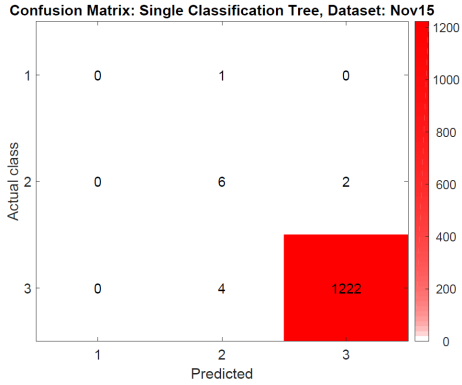

Figure 8: The SCT for dataset *Nov15*.

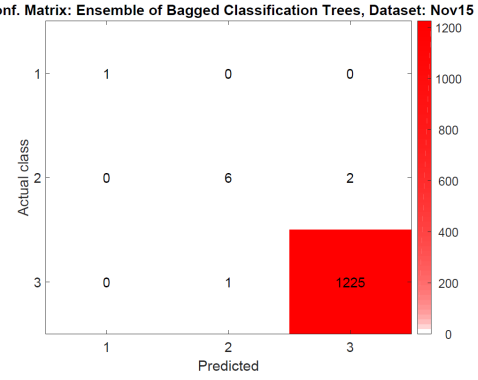

Figure 9: The BCT for dataset*Nov15*.

mixing with the strength $1-t$ in (1). This smoothing of color level can easily be seen by applying a column average of the transmission. The obtained function will be very noisy if there is no fog, while it will be smooth if there is a homogeneous fog.

A clear problem with the determination of the jump location is the smoothing itself. The smoothing implies smaller and more gradual jumps due to the horizon as the horizon itself becomes fuzzier and less clear. However large objects can still create large local deviations resulting in contamination of the jump location. The jump location can therefore only be used for extreme cases (dense fog or no fog conditions). As a result one can see that the jump location is usually at large values when there is fog and at small values when there is no fog. Unfortunately still a significant fraction of fog conditions according to both MOR and total variability data has a small jump location value.

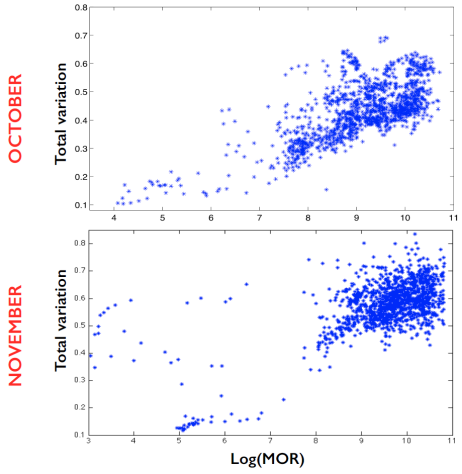

Figure 10: Total variability against the logarithm of MOR for the Oct15 and Nov15 data sets.

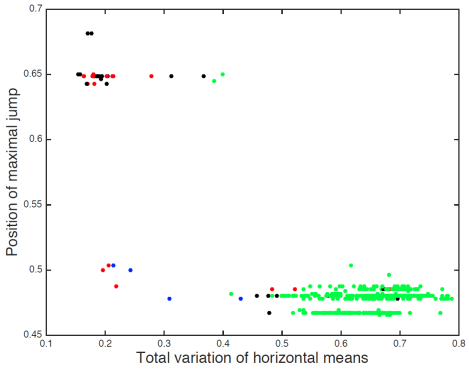

Figure 11: Jump location as the percentage of the height of image (from the top of image) against the total variability for 500 day pictures. Dot colour indication: black for MOR $<$ 200m, red for 200m$<$MOR$<$500m, blue for 500m$<$MOR$<$1km and green for MOR $>$ 1km.

A problem with the usage of total variability method is its dependence on the variability of no fog image. If a camera is pointed at a low variability location such as a snowy landscape or a calm sea, then foggy and clear weather conditions can be difficult to distinguish. Furthermore, the total variability method is only applicable when the camera is focussed at infinity. If a camera is focussed at a nearby location such as an object on the lens, then the resulting defocussing of the background will directly imply low variability, while the actual weather condition might be a clear day.

2.4 Fog Indicator method

The third method related to the transmission function will be called the Fog Indicator method. This method applies the horizontal averaging to the estimated transmission. The method combines the slope change of the obtained horizontal averaged transmission function f with the location of the biggest jump. An elementary observation is the large difference between the transmission values of the sky and the ground. Furthermore, the horizon is a sharp drop during clear days and a shallow drop during foggy days. Thus the horizontal averaged transmission function f for clear days looks more like a step function than in the case of foggy days. Hence the Fog Indicator F_{ind} can be suggested as the squared L^2 norm of the difference between the (discrete) horizontal averaging function f and the fitted step function S_f with respect to f, see

Figures 13 and 12 for an illustration.

Figure 12: An image without fog.

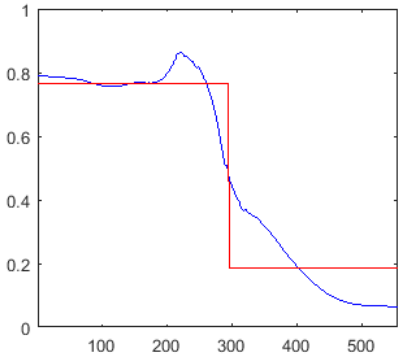

Figure 13: The horizontal averaged transmission function f (blue) and the fitted step function S_f (red).

Consequently, the low values of F_{ind} indicate clear days, while high values indicate foggy conditions. We summarize the obtained results for available data sets in Figures 14 and 15.

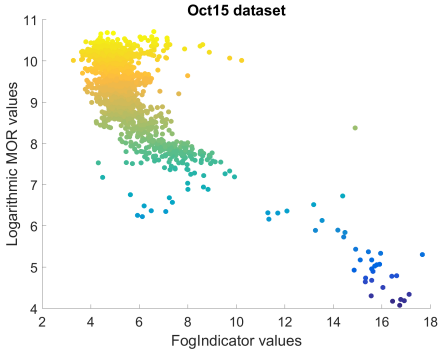

Figure 14: Logarithmic MOR values against Fog Indicator values for the Oct15 dataset. The colours are directly related to the logarithmic MOR values.

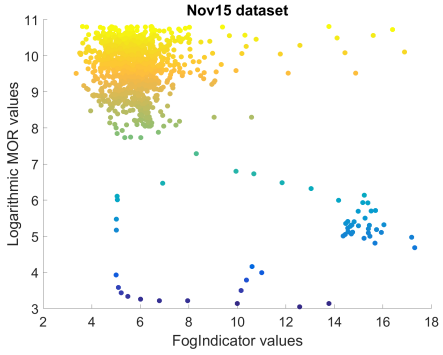

Figure 15: Logarithmic MOR values against Fog Indicator values for the Nov15 dataset. The colours are directly related to the logarithmic MOR values.

When considering the data of October 2015 the Fog Indicator seems like a good predictor for fog or even MOR values. However for the data of November 2015 one can see two additional groups of points that indicate a discrepancy between the two methods. One group (the blue points in the lower and left part of graph) is probably

caused by a faulty MOR reading as the images do not show fog (as indicated by the low Fog Indicator value as well). The other group (the orange points in the top and right part of graph) is due to dark images because of the decreased length of days in winter in the Netherlands. It is natural that the second group did not occur in October as the days are not yet short enough to cause problems for the daily time intervals we are looking at. For typical representative pictures of each group see see Figure 16 and 17.

Figure 16: An image with a faulty low MOR value.

Figure 17: An image with a faulty large Fog Indicator value.

Clearly, the Fog Indicator method has some issues. Dark and nightlike images will yield wrong values. Moreover, the amount of clouds in the sky will influence the Fog Indicator value even though they do not cause fog. In general the Fog Indicator method depends on the visibility of a horizon in the image. If a camera is pointed towards the ground in such a way that the horizon is not visible, then the Fog Indicator method may fail for clear day images.

3 Fog detection methods from shapes in images

Fog is known to affect visibility by reducing the contrast. Multiple methods are possible for determining an effective value for the contrast. We were not successful to create fog detection methods with some of them. For example Fourier analysis methods did not show clear characteristics for discriminating between fog and clear conditions. Other methods like wavelet analysis were relatively complex and computationally expensive compared to the reliability of the data. Therefore we have chosen to do only two related methods: gradient thresholding and local contrast correlation.

3.1 Gradient Threshold method

This method is based on the assumption that the fog has a tendency to smooth colour values, creating a less pronounced colour gradient between an object and a background, the edge.

For simplicity we have converted the RGB colour images into grey scale images. Having this grey scale image we calculate the local gradient vector and its norm. Afterwards we count the number of pixels with a local gradient norm larger than a certain threshold. Our experience is that for a well chosen threshold one can distinguish for the chosen image between fog and clear days, see Figures 18 and 19 for an illustration.

Figure 18: The chosen image for gradient treshold method.

Figure 19: An image representation of gradient thresholding: the white pixels denote locations with sufficiently large gradients (the detected edges), while the black pixels indicate gradients below the threshold.

To compute the gradient we apply a finite difference method, usually a second order one, which implies that the local gradient is a patch size dependent given by the order of the finite difference method.

The proposed Gradient Threshold method seems to be a good fog indicator for at least the presence of fog, see Figures 20 and 21. Concerning the visibility distance we still see a large spread that can give unacceptably large deviations in the visibility distance.

The gradient method is still sensitive to defocussing, a presence of objects on the lens, and ground fog when it can give errors and incorrect interpretations of the data, but it is quite robust in the sense that the problems like astigmatism, chromatic aberration or night images will affect the method far less then colour dependent methods.

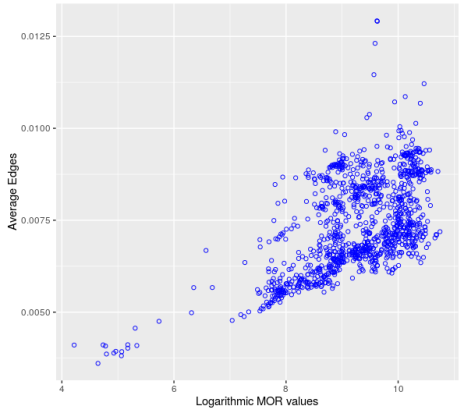

Figure 20: Average edges against the logarithmic MOR for October.

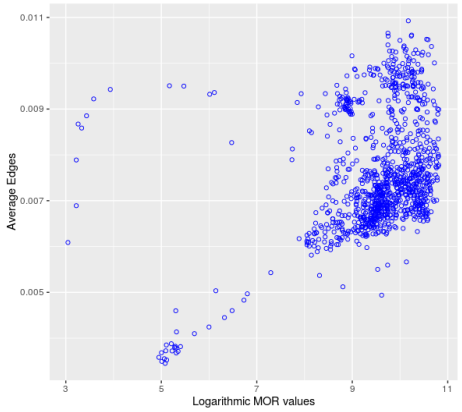

Figure 21: Same as Figure 20 for November.

3.2 Local Contrast Correlation method

In the Local Contrast Correlation method we make use of the fact that we have a sequence of images taken from the same viewpoint. The method proceeds in two phases. First we analyze a set of reference images from the past and determine specific, small-scale, contrast-rich features that are present in this set of images. Secondly, we take the current image, and determine whether the features that have been identified in previous images can be observed in the current image. In other words, the Local Contrast Correlation method tests specifically for the presence of certain contrast-rich features identified from reference images, and not directly for the presence of fog. But since fogs blurs the features of the image one might expect a good correlation with the presence of fog. The first phase will be called the analysis phase, the second phase the test phase. Analogous to the Gradient Threshold method we convert the RGB images to grey scale images. In the examples we only used daytime images.

The main motivation for this method, as compared to the previous, gradient threshold method, is to address the issue of contamination on the shape level, see section 4 below. For example, theoretically it is possible that in a situation of fog, a bird flies close the camera and introduces a lot of extra contrast, compensating for the loss of contrast elsewhere in the picture due to the fog. In the local contrast correlation method, the contrast from the bird will be discarded because it was not present in the set of reference images.

We next describe the analysis phase. For each reference image, we identify the small-scale, contrast-rich features as follows. We subdivide the image into a specified number of patches of a certain size. In each patch, we set the zeroth and first moments of the local grey scale expansion to zero. The constants are chosen to equal the average

constant value and gradients within a patch. The remaining grey scale image for a certain patch will have highly pronounced edges (if edges are present in the patch). The corresponding patches obtained from the different reference images are averaged, to keep only contrast present in many of the images. A set of patches is selected where contrast is above a certain threshold. These will be used for testing the presence of specific features in the test image. Patches can e.g. be of size 16×16 and 100 patches can be selected. After moment removal, the patches where normalized, so that, as a vector, they had unit length. We denote by $\hat{R}_{\alpha,\beta}(j,k)$ the patches after moment removal, averaging and normalization, with α, β the index of the patch and (j,k) the coordinates of each pixel in the patch, and by $S = \{(\alpha_1,\beta_1),(\alpha_2,\beta_2),\ldots,(\alpha_M,\beta_M)\}$ the set of patches selected for testing. See Figure 22 and 23 for an illustration.

In the test phase, one subdivides the test image $I(j,k)$ in patches $I_{\alpha,\beta}(j,k)$ in the same way as for the analysis phase. To find whether a certain feature is present in the image, we consider the inner product (correlation)

$$t_{\alpha,\beta} = \sum_{j,k} I_{\alpha,\beta}(j,k)\hat{R}_{\alpha,\beta}(j,k).$$

A large value of $t_{\alpha,\beta}$ means that the detailed features, observed in the reference images, are present in the test image. Small values for $t_{\alpha,\beta}$ can mean either that there is no contrast present in the specific patch, or that there is an altogether *different* contrast present, e.g. due to an object with a different shape that is present in the image. The logarithms of the values $t_{\alpha,\beta}$ can be summed to give a first indication of the "fogginess".

Figure 22: An RGB reference image used for the analysis phase of the Local Contrast Correlation method.

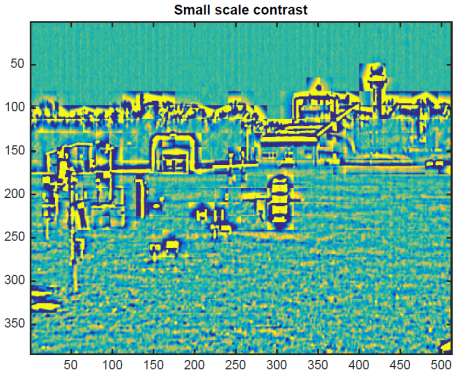

Figure 23: An outtake of the greyscale image of which the zeroth and first moments are removed in grid patches. The colouring is from dark blue for value -1 to bright yellow for value +1.

The indicator value of the proposed Local Contrast Correlation method is actually a

good indicator for the October 2015 data with no deviations from a specific functional form with respect to the MOR data. However for November 2015 we see again a group of points deviating from the October functional dependence. Again these points seem false positives of the MOR data as the values of the edge detection are similar to the values of the clear days. It is highly likely that this group is the same false positive group as found by the Fog Indicator method.

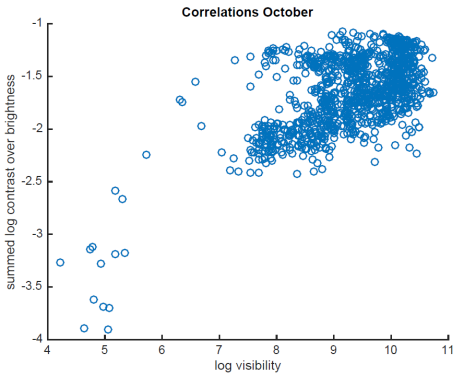

Figure 24: Local contrast correlation indicator values against MORvalues for images of October 2015 dataset.

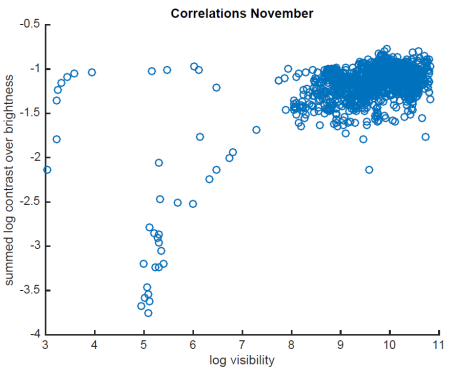

Figure 25: Local contrast correlation indicator values against MOR values for images of November 2015 dataset.

The Local Contrast Correlation method shares some properties of Gradient Tresholding method. But there is also a number of differences. An important difference is the use of reference images, showing what was previously visible at the site. As explained, this could address the issue of shape contamination. In principle, using more of the available information should lead to a reduction of the statistical uncertainty. However, it is unclear whether this is the case when including information from reference images, because there are also potential complications. The use of patches in principle allows the use of different statistics than simply summing the values $t_{\alpha,\beta}$. Each low $t_{\alpha,\beta}$ value is an indicator of reduced contrast in some part of the image which could be caused by fog. However, it is not easy to say anything in general about the relation between the $t_{\alpha,\beta}$ and the fog conditions. Reasons for this are for example that the depth- and height-maps of the pictures are unknown, that the distribution of contrast over the picture can vary and that fog can have different characteristics.

4 Error sourcess in fog detection methods

All methods for determining the existence of fog are subjected to three types of error sources: difficult weather conditions, camera errors and errors in the method itself.

We assume that all errors in the method itself, for example ill-posed matrices for inversion or other problems resulting in non-uniqueness, are negligible, while all other errors are due to the difficult weather conditions or camera errors.

In what follows we discuss contaminating conditions in fog detection methods.

Contamination on the colour level The colour level depends on the properties of the objects seen in the picture, the intensity of the light, the position of the sun and the weather conditions. Several combinations of these factors can lead to false positives in the colour level methods for fog detection. Due to the variable nature of weather, light intensity and solar position in the sky one expects the colour level contamination to be highly variable with predictable time dependencies.

Contamination on the shape level The shape level depends on the local variability in colour (or grey scale), which ultimately depends on the resolution and the intensity of the light. Therefore the same contamination problems as with colour level are present. If the objects in the images vary with time, as it is the case for traffic cameras, then the shape level depends on the amount, shape, size and colour of objects visible in the clear image as well. An empty road should not be classified as foggy just because no car is visible, while a foggy traffic jam should not be classified as clear just because a lot of cars are visible. Therefore only not hidable stationary objects should be used in the determination of the shape level, e.g.one does not want to use the lines on a highway as objects, because they can be hidden from the camera by another objects such as a car or a truck.

Weather conditions A crucial part in the colour level methods is the Dark Channel Prior. By definition this Dark Channel Prior is the minimal value over all channels for all pixels within a patch. As a consequence the Dark Channel Prior is biased towards dark images: darker images are perceived as clearer pictures. Therefore every weather condition that creates dark images will artificially be interpreted as clear. Thick rain clouds, clouds during dusk or dawn and night images can all give the wrong interpretation of fog. Night images are even more problematic because of the vanishing horizon.

A second problem is the imitation of fog by other weather conditions. Heavy precipitation in any form will create a decrease in the visibility distance, but it is not fog and therefore not seen by the MOR (if the sensor is encased for protection purposes). Therefore false positives of the image methods with respect to fog and false negatives of the MOR method with respect to visibility distance will occur.

One can pose the question what the ultimate goal of the KNMI (or any other user of these tools) is: fog detection or visibility distance determination? This question is

crucial in the evaluation of the data, for example in the determination of the influence of the MOR values compared to the image data indicators.

A third problem is the inhomogeneous distribution of fog. If fog is only present in a part of the image, then even dense fog can be classified as moderate fog. Clear examples are thin layers of dense ground fog. They can result in multiple effective horizons or a high spatial discrepancy in edge detection.

Camera errors Camera errors are systematic errors that effect all image data indicators. The most prominent problems are defocussing, chromatic aberration and astigmatism.

Defocussing implies a large scale averaging on the entire perfect image resulting in an artificial dense fog condition. Almost all colours of the image are mixed resulting in a fog like Dark Channel Prior. Chromatic aberration is an effect when a lens does not work equally for all wavelengths. Chromatic aberration occurs in two types: different focal lengths or different foci. Different focal lengths imply different magnifications for different colors. Therefore edges become less pronounced and blurring occurs on the edges of the image resulting in artificial fog. Different foci implies a homogeneous defocussing of different strength for different colours. This results in a local averaging for different colours implying higher values for the Dark Channel Prior, hence the method can classify the image as slightly foggy. Astigmatism is an effect when different lens axes have different foci, resulting in an effective blurring of the image. Naturally the Dark Channel Prior can again classify the image as slightly foggy.

Miscellaneous errors There are other reasons why errors are introduced. For example animal interference. If a spider creates a cobweb on a camera, then it will influence the determination of fog or visibility distance. Similarly wind can blow leaves or cloths on the camera, while animals like bugs or arachnids can stay on the lens. Precipitation can cause blurring as well just by exposure of the lens to weather. Water droplets or ice severely influence the viewing angle or distort the view. Furthermore, dew and crystallization of moisture can cause fast changing operating conditions with total blockage of the camera lens in extreme cases. Aging of the lens due to degrading lens surfaces by sand, salt or other effects can create optical effects that persist. Moreover, CCD aging due to long exposures to high intensities of light or other effects can create permanent artifacts on images.

5 Conclusions

In this paper we present several fog detection methods based purely on processing of image data obtained from digital cameras. We can classify these methods into two groups. The first group is based on well established Dark Channel Prior method that

was originally proposed to dehaze foggy images, and it includes Classification Trees methods, Transmittance methods, and Fog Indicator method. The second group works with converted grey scale images, and it includes Gradient Tresholding and Local Contrast Correlation method.

We note that not only the fog detection but also the visibility correlated to the current meteorological standard for visibility ranging can be obtained from camera images in several cases. For practical usage one may propose a combination of presented methods supported by proper statistical tools to create a fog detection method that is robust against false positives and false negatives of individual methods. In fact, using proposed fog detection methods we could recognize faulty data in Meteorological Optical Range (MOR) measurements.

References

L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. CRC press, 1984.

T. K. der Staten Generaal. Vragen van het lid jansen (sp) aan de minister van verkeer en waterstaat over cameras langs de wegen (ingezonden 21 juli 2010). antwoord van minister eurlings (verkeer en waterstaat) (ontvangen 17 augustus 2010). *Kamervragen (Aanhangsel)*, 2009-2010(3118), 2010.

R. Fattal. Single image dehazing. *SIGGRAPH*, pages 1–9, 2008.

L. P. F.L. Pedrotti, L.S. Pedrotti. Introduction to optics. *Pearson - Prentice Hall*, 3rd edition:355–357,438–456, 2007.

K. He, J. Sun, and X. Tang. Single image haze removal using dark channel prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(12):2341–2353, Dec 2011. ISSN 0162-8828. doi: 10.1109/TPAMI.2010.168.

C. Kitchin. Astrophysical techniques. *CRC Press*, 5th edition:17–30, 2009.

S. Narasimhan and S. Nayar. Chromatic framework for vision in bad weather. *CVPR*, pages 598–605, 2000.

S. Narasimhan and S. Nayar. Vision and the atmosphere. *IJCV*, 48(233-254), 2002.

G. Rieke. Detection of light - from the ultraviolet to the submillimeter. *Cambridge University Press*, 2nd edition digitally transferred:151–175, 2009.

S. Tierney. Matlab implementation of "single image haze removal using dark channel prior". https://github.com/sjtrny/Dark-Channel-Haze-Removal, 2014.

Modelling a long production line as a train with coupled carriages

Nick Gaiko Thomas de Jong Vivi Rottschäfer

Abstract

In this paper, we model an active tension control (ATC) of a long production line offered by Marel Stork Poultry Processing. The production line consists of trolleys with poultry product which move along a rail. The trolleys are connected by a chain. The motor drives the chain by pushing the trolleys forward. However, when the motor is spinning too quickly the tension in the chain, right after the motor can drop which leads to collisions of the trolleys or entanglement of the chain. Hence, control of the tension is necessary. This ATC method works by means of a dead weight, pulling on the chain after the motor, the so-called 'dancer'. The dancer maintains the tension in the chain when the weight is sufficiently high. However, if the dancer is too heavy it will shorten the life-time of the chain. By modelling the motion of the trolleys between the first motor and the following dancer we numerically compute the optimal force for the dancer. Furthermore, we suggest extensions of our model and a novel modification to control the tension.

1 Introduction

In this paper we consider a problem proposed by Marel Stork Poultry Processing at SWI 2016. Marel Stork offers solutions for in-line poultry processing in accordance with a desired automation level and production capacities. Marel Stork systems are modular, that is, they can be combined with other equipment and with manual processes. We consider only one modular part of the whole processing system, namely a long production line transporting poultry through a chilled storage room. This closed-loop line is called an overhead conveyor.

The overhead conveyor consists of a long chain, the maximum length of which is 5500 m, driven by the up to 40 electric motors along the rail with a constant speed (0.6-0.8 m/s), see Figure 1.

Every 6 inches, a trolley is attached to the chain. These trolleys move over an overhead rail. Each trolley contains a shackle, hanging downwards, which suspends poultry products, see Figure 1. Each poultry product weighs between 1.5 and 2.5 kg. We will refer to a trolley containing poultry product as a carriage. The weight of a fully loaded production line is 68000 kg. The hanging poultry in the shackle can swing in all directions; also, the products can almost touch each other with their wings.

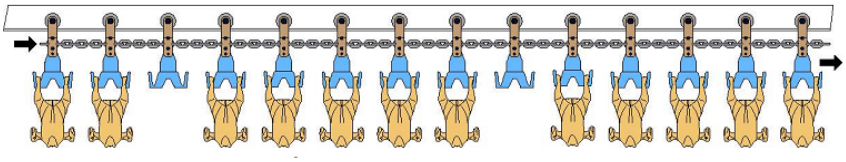

Figure 1: Schematics of the overhead conveyor with poultry. Courtesy of Marel Stork Poultry Processing

From the characteristics above one can see that the production line is very heavy and has a relatively large travelling speed. A problem occurring in the conveyor is slack of the chain. Slack has many causes, however, it is most often caused by the motor. The motor contains uniformly placed slots through which it transports the carriages, see Figure 2. If the motor is spinning quickly and if the chain is stretched due to wear such that the length of the chain between the carriages is longer than the distance between the slots then the chain will slack when carriages leave the motor. This slack could spread throughout the chain. If the chain slacks too much

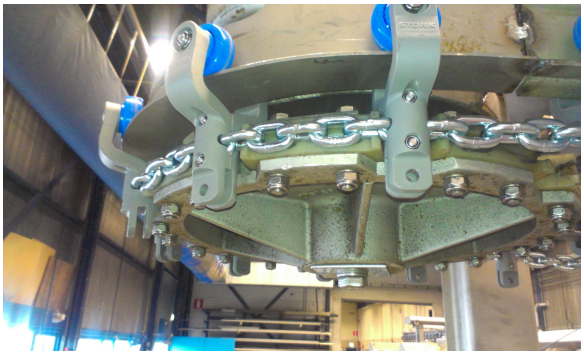

Figure 2: A motor of the overhead conveyor. Courtesy of Marel Stork Poultry Processing

the carriages can collide or the chain could entangle. This leads to a production stop and could even cause permanent damage of the production line. Hence, whenever too much slack is observed the production line is stopped.

The slack is controlled by an active tension control whose simplified schematic is displayed in Figure 3. Since the chain is very long many motors are needed to drive the chain. The first motor, the master motor, turns at a fixed speed. Each motor is followed by a dead weight, the so-called 'dancer'. The dancer is a pulley with a heavy weight. The pulley is fixed to a rail whose length determines the range of the motion of the dancer. Since the dancer applies a force it pulls the carriages in its direction and, consequently, reduces the slack. Hence, it is very similar to a train locomotive

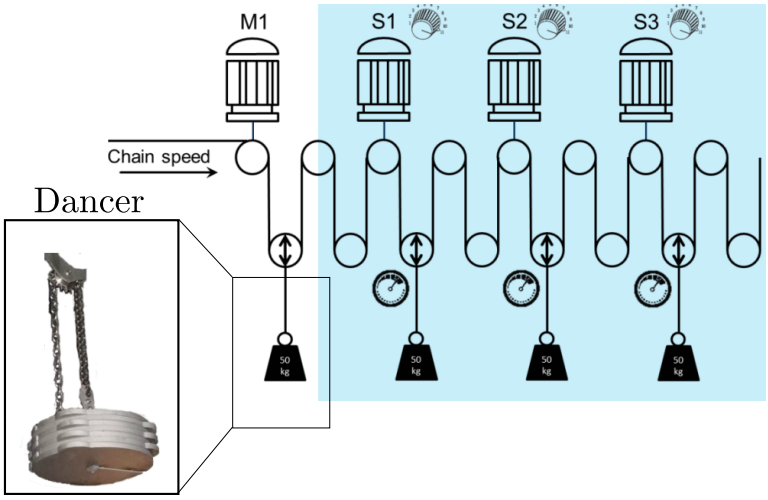

Figure 3: Active tension control of the overhead conveyor. M1, S1, S2 and S3 denote motors. The motor M1 is the master motor which turns at a fixed speed. Each motor is followed by a dead weight, the 'dancer'. The dancer is a pulley with a heavy weight. Since the dancer applies a force it pulls the carriages in its direction and, consequently, reduces the slack. Observe that a quantitative measure of the slack in the chain between a motor and the following dancer is given by the position of the dancer. The motors S1, S2 and S3 are the slave motors. Their speed is controlled by the position of the dancer. Courtesy of Marel Stork Poultry Processing

which pulls its carriages forward. However, contrary to the carriages of a locomotive, these carriages will move past the pulling force. Observe that a quantitative measure of the slack in the chain between a motor and the following dancer is given by the position of the dancer. Hence, the speed of the motors after the master motor are controlled by the position of the dancer. Thus, the motors after the master motor are called slave motors.

Observe that the dancer does not make the slack vanish instantaneously. Hence, the force of the dancer should be large enough such that it counteracts the slack caused by the motor. However, the problem with the dancer is that it applies a constant force on the chain which leads to stretching of the chain over time, in turn, this leads to more slack. Furthermore, when the chain is stretched for about 3%, the chain is classified as worn out and needs to be entirely replaced. During this maintenance process no products can be processed. Thus, it is of importance to maintain a tension in the chain which prevents its slack and reduces its wear.

In Section 2 we set up a model for the motion of the carriages where the slack is caused by the motor. More specifically, we focus on the motion between the master

motor and the following dancer. In Section 3 we present the results of numerical simulations of the model constructed in Section 2. Finally, we present the conclusions and recommendations in Section 4. More specifically, we explain how the numerics validates our model, how the model can be extended to model the system better and how the mathematical results might lead to an improvement of the current system.

2 Modelling the production line as a train with carriages

In this section we formulate the model. We consider a chain that is fully loaded with product. However, we only study part of the complete chain and restrict our model to the study of the carriage motion between the master motor and the next dancer, see Figure 4.

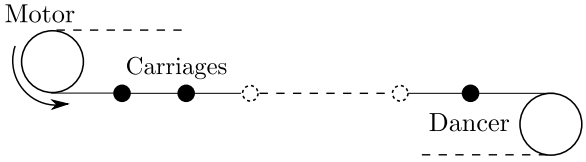

Figure 4: The chain between the master motor and the dancer. The carriages are depicted by the black dots.

2.1 Modelling assumptions

To set up the equations of motion we consider some modelling assumptions:

Carriage:
1. Carriages are point masses with equal mass.
2. The friction of the wheels with the rails is linear.
3. When the chain between two carriages has slack the carriages do not exert forces on each other. However, when the chain is tight the forces on both carriages are equal.

Dancer:
1. The position of the dancer is fixed.
2. The dancer applies a constant force on the chain.

Chain:
1. There is no mechanical energy loss through the chain.

Motor:
1. The motor supplies carriages with a constant rate.
2. The carriages that leave the motor have a prescribed initial velocity. This velocity is sufficiently low such that the chain between the motor and the first carriage is never pulled tight whenever this carriage is unaffected by the dancer. Hence, the motor causes slack.

The model will consist of equations of motion for each of the carriages. Here we measure their position by the distance to the motor where the motor is located at $x = 0$ and and the dancer is located at $x = \ell$. Based on the assumptions above we can introduce the parameters for the equations of motion:

ℓ: the length, in meters, from the motor to the dancer

ℓ_{ct}: the length, in meters, of the chain between two carriages when it is pulled tight

m: the mass, in kilograms, of the (loaded) carriage

c: the friction coefficient, in kilograms per second, between the wheels of the hangar with the rails

f_{mot}: the number of carriages that enter per second through the motor

v_{mot}: the velocity, in meters per second, of the carriages that enter through the motor

F_{dan}: the force, in Newton, which is being applied by the dancer

We assume that all the constants are non-zero.

Denote by $x(t)$ the position of a carriage. Then, if the carriages is unaffected by the force of the dancer the equation of motion is given by

$$m\ddot{x} = -c\dot{x}, \tag{1}$$

where we used the short-hand notation $\dot{x} = dx/dt$ and $\ddot{x} = d^2x/dt^2$. If the carriage is part of W carriages whose chains are pulled tight the total mass is Wm and the total friction is Wc. Hence, if these carriages are affected by the force of the dancer the equation of motion is given by

$$Wm\ddot{x} = -Wc\dot{x} + F_{\text{dan}}. \tag{2}$$

The equations (1) and (2) are standard ODEs of the form

$$\ddot{y} = a\dot{y} + b, \tag{3}$$

with $a, b \in \mathbb{R}$ of which solutions are given by

$$y(t) = \frac{c_1 \mathrm{e}^{at} - bt}{a} + c_2, \tag{4}$$

with $c_1, c_2 \in \mathbb{R}$ constants determined by the initial conditions. It will turn out that all the ODEs in this paper are of the form (3).

Observe that assumption 2 for the motor is an assumptions on the parameters. Hence, let us formulate this assumptions in terms of the parameters using (1):

Motor: 2. Consider the equations of motion in (1). Let x satisfy the initial conditions

$$x(0) = 0,\ \dot{x}(0) = v_{\text{mot}}.$$

At $t = 1/f_{\text{mot}}$ a new carriage will enter via the motor. Hence, we assume that v_{mot} is sufficiently small such that

$$x(1/f_{\text{mot}}) < \ell_{ct}.$$

2.2 Set-up: possible events

We assumed that when the chain between two carriages is loose the carriages do not exert forces on each other. However, when the chain is tight the forces on both carriages are equal. In formulating the model it turns out that the configuration of the carriages is very important. As we explained before the chain between the carriages can hang loose or be tight and we have to take that into account. So, by configuration we mean the number of carriages between the motor and dancer, and also between which carriages the chain is tight and loose.

For a given configuration the equations of motion remain unchanged. However, when the configuration changes the equations of motion also change. The configuration changes when one of the following events occur:

P: a loose chain is pulled tight,

C: a collision occurs between two carriages,

E: a new carriage enters via the motor,

L: the last carriage leaves by passing the dancer.

We shall abbreviate these events by the capital letters above. Note that several of the above events could also occur at the same time. Hence, all possible events are given by all the combinations of (P, C, E, L). When an event occurs the equations of motion change and we have to prescribe the new equations of motion until the next event. The event C is special in this respect since in this case the production line should be stopped so our model should end too.

2.3 The event map

In this section we will present a general procedure which gives the equations of motion as the system undergoes a configuration change due to an event.

First, we assume that we know the equations of motion at $t = 0$ and that at $t = 0$ there are $N_0 > 0$ carriages between the motor and dancer. We denote the position of the ith carriage by $x_{0i}(t)$ for $i = 1, 2, \ldots, N_0$. Here we order the carriages in such a way that the 1st carriage is closest to the motor and the N_0th carriage is closest to the dancer, see Figure 5.

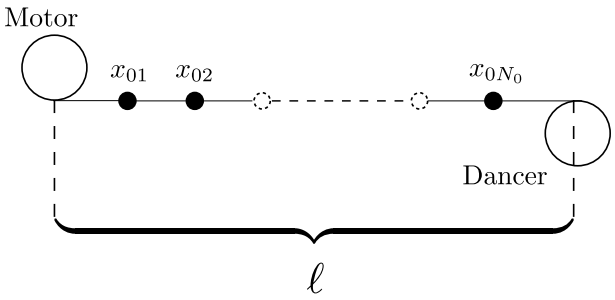

Figure 5: Carriage position at $t = 0$. The carriages are depicted by black dots. The 1st carriage is closest to the motor and the N_0th carriage is closest to the dancer.

Now, assume that the first time that an event occurs is at $t_1 > 0$. Then, a full description of the carriages for $0 \leq t < t_1$ is given by the tuple

$$(T_0, X_0), \tag{5}$$

with

$$T_0 := [0, t_1), \quad X_0(t) = (x_{01}, x_{02}, \cdots, x_{0N_0}).$$

The tuple (5) will be called the carriage motion on T_0. Denote the event at t_1 by A. Then we want to construct a map Ψ such that

$$\Psi(T_0, X_0) = (T_1, X_1),$$

with

$$T_1 := [t_1, t_2), \quad X_1(t) = (x_{11}, x_{12}, \cdots, x_{1N_1}),$$

where t_2 is time when the next event occurs, N_1 is the number of carriages between the motor and dancer and $x_{1i}(t)$ is the position for the ith carriage for all $t \in T_1$. Again the carriages are ordered in such a way that the 1st carriage is closest to the motor and the N_1th carriage is closest to the dancer. Observe that Ψ maps (T_0, X_0) into the carriage motion until the next event. Hence, we call Ψ the event map and $\Psi(T_0, X_0)$ is called the carriage motion from the 1st to the 2nd event of (T_0, X_0). Similarly, $\Psi^n(T_0, X_0)$ is called the carriage motion from the nth to the $(n + 1)$th event of (T_0, X_0).

2.4 Initial carriage motion

To construct a general Ψ is a very lengthy exercise. However, by restricting to a specific initial carriage motion the construction of Ψ simplifies. We consider (T_0, X_0) the tuple (5). As the initial configuration we assume that the chain is tight between all the carriages. Observe that we must require that $N_0\ell_{ct} \leq \ell < (N_0 + 1)\ell_{ct}$ when all

the chains are tight. Then, it follows from (2) that the equations of motion for the carriages are given by

$$N_0 m\ddot{x}_{0j} \quad = \quad -N_0 c\dot{x}_{0j} + F_{\text{dan}}, \;\; j = 1, 2, \cdots, N_0 \,, \tag{6}$$

with initial conditions

$$x_{0j}(0) \quad = \quad (j-1)\ell_{ct} + d_0, \qquad \dot{x}_{0j}(0) \quad = \quad v_1 > 0, \;\; j = 1, 2, \cdots, N_0 \,, \tag{7}$$

where $d_0 \in [0, \ell_{ct}]$. Since the chain between the carriages is tight, the initial distance between two adjacent carriages is ℓ_{ct}. Observe that d_0 is the distance between the motor and the first carriage. Hence, the chain between the motor and the first carriage can be loose ($d_0 < \ell_{ct}$), tight ($d_0 = \ell_{ct}$) or the first carriage starts at the position of the motor ($d_0 = 0$). Since the N_0 carriages move as a whole all the carriage have the same initial velocity.

Next, we look at which event can occur at t_1. First, let us make a distinction between two different P events:

P_c: The chain between two carriages is pulled tight

P_{mot}: The chain between the motor and the first carriage is pulled tight

Then, the events C and P_c cannot occur at t_1. Hence, the events E, L and/or P_{mot} can occur at t_1. We assume that the motor is switched on at $t = 0$. Hence, at $t = 1/f_{\text{mot}}$ the first carriage will enter via the motor. Thus, t_1 is given by

$$t_1 = \min(\tau_1, \tau_2, 1/f_{\text{mot}}), \tag{8}$$

where $\tau_1, \tau_2 > 0$ are the smallest τ_1, τ_2 such that

$$\begin{aligned} x_{0N_0}(\tau_1) \quad &= \quad \ell, \;\; (\text{event } L) \\ x_{01}(\tau_2) \quad &= \quad \ell_{ct} \;\; (\text{event } E). \end{aligned}$$

2.5 Carriage motion from the nth to the $(n+1)$th event

It turns out that we chose the initial carriage motion, (T_0, X_0), in such a way that the following holds for $\Psi^n(T_0, X_0) = (T_n, X_n)$ for any $n \in \mathbb{N}$:

Property 1. The carriages can be divided into two connected parts: the loose part, which consists of all the carriages with a loose chain in between them, and the tight part, which consists of all the carriages with a tight chain between them. Denote the number of carriages of the loose part by K_n. If the loose part contains more than 0 carriages, so $K_n > 0$, then the first carriage of the loose part connects to the motor. If the tight part consists of more than 0 carriages then the last carriage of the tight part is next to the dancer, see Figure 6.

Property 2. If $K_n > 0$ then the speed of the carriage closest to the motor is largest and the speed of the carriages decreases when moving away from the motor. Hence, for all $t \in T_n$ the following inequality holds:

$$\frac{dx_{n1}}{dt}(t) > \frac{dx_{n2}}{dt}(t) > \cdots > \frac{dx_{nK_n}}{dt}(t). \tag{9}$$

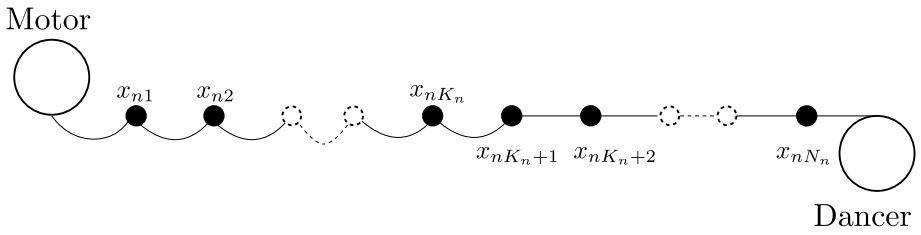

Figure 6: The loose part and tight part. Denote the number of carriages by N_n and the carriages of the loose part by K_n. The carriages can be divided into a loose part with K_n carriages which are connected to the motor and a tight part of $(N_n - K_n)$ carriages which are connected to the dancer.

These two properties are important for the construction of Ψ. We will now construct Ψ and prove the above properties by induction. Observe that property 1 and 2 hold for (T_0, X_0).

Now, we assume that (T_k, X_k) is known. Furthermore, we assume that (T_k, X_k) satisfies property 1 and 2. Then, we want to prescribe the carriage motion from the $(k+1)$th to the $(k+2)$th event of (T_0, X_0), $\Psi(T_k, X_k) =: (T_{k+1}, X_{k+1})$, and prove that property 1 and 2 hold. We denote $T_k = [t_k, t_{k+1})$ and N_k is the number of carriages. As before, we index X_k in the following way:

$$X_k = (x_{k1}, x_{k2}, \cdots, x_{kN_k}).$$

First we determine which events can occur at t_{k+1}. We will present mathematical equivalents for the events. Using property 1 and 2, we first make some observations:

- If P_{mot} occurs at t_{k+1} then all the chains are pulled tight.

- If P_c occurs at t_{k+1} then the chain between the loose part and the tight part of the carriages is pulled tight.

- If C occurs at t_{k+1} then the last carriage of the loose part and the first carriage of the tight part collide.

Denote the number of carriages of the loose part by K_k. Then the events are described by

$$P_{\text{mot}} \text{ a } t_{k+1} \iff x_{k1}(t_{k+1}) = \ell_{ct}, \tag{10}$$

$$L \text{ at } t_{k+1} \iff x_{kN_k}(t_{k+1}) = \ell, \tag{11}$$

$$E \text{ at } t_{k+1} \iff t_{k+1} = p/f_{\text{mot}} \text{ with } p = \min_{\substack{p_0/f_{\text{mot}} - t_k > 0, \\ p_0 \in \mathbb{N}}} p_0, \tag{12}$$

and if $K_k > 0$ then, in addition, the following events could occur:

$$P_c \text{ at } t_{k+1} \iff x_{kK_k+1}(t_{k+1}) - x_{kK_k}(t_{k+1}) = \ell, \tag{13}$$

$$C \text{ at } t_{k+1} \iff x_{kK_k+1}(t_{k+1}) - x_{kK_k}(t_{k+1}) = 0. \tag{14}$$

If C occurs at t_{k+1} then there is no carriage motion after the $(k+1)$th event of (T_0, X_0). At t_{k+1} either a single event or a combination of $(L, E, P_{\text{mot}}, P_c)$ can occur. Next, we will prescribe X_{k+1} based on which event occurs:

Event L:

The last carriage leaves. We have that

$$X_{k+1} = (x_{k1}, x_{k2}, \cdots x_{kK_k}, y_{K_k+1}, y_{K_k+2}, \cdots, y_{N_k-1}), \tag{15}$$

where y_j with $j = K_k + 1, \cdots, N_k - 1$, satisfies

$$m(N_k - 1 - K_k)\ddot{y}_j = -c(N_k - 1 - K_k)\dot{y}_j + F_{\text{dan}}, \tag{16}$$

and initial conditions

$$y_j(t_{k+1}) = x_{kj}(t_{k+1}), \;\; \dot{y}_j(t_{k+1}) = \dot{x}_{kj}(t_{k+1}). \tag{17}$$

Event E:

A new carriage enters. Hence, we have

$$X_{k+1} = (z, x_{k1}, x_{k2}, \cdots, x_{kN_k}), \tag{18}$$

with z satisfying

$$m\ddot{z} = -c\dot{z}, \tag{19}$$

and initial conditions

$$z(t_{k+1}) = 0, \;\; \dot{z}(t_{k+1}) = v_{\text{mot}}. \tag{20}$$

Events E and L:

We have that

$$X_{k+1} = (z, x_{k1}, x_{k2}, \cdots x_{kK_k}, y_{K_k+1}, y_{K_k+2}, \cdots, y_{N_k-1}), \tag{21}$$

where z satisfies Equation (19) and (20) and where y_j with $j = K_k + 1, \cdots, N_k - 1$, satisfies (16) and (17).

Event P_{mot}:

We have that

$$X_{k+1} = (w_1, w_2, \cdots, w_{N_k}),$$

where w_j with $j = 1, 2, \cdots, N_k$, satisfies

$$\dot{w}_j = 0, \tag{22}$$

and initial conditions

$$w_j(t_{k+1}) = x_{kj}(t_{k+1}). \tag{23}$$

Events E and P_{mot}:

We have that

$$X_{k+1} = (w_0, w_1, w_2, \cdots, w_{N_k}),$$

where w_j with $j = 0, 1, 2, \cdots, N_k$, satisfies Equation (22) and (23).

Events L and P_{mot}:

We have that

$$X_{k+1} = (w_1, w_2, \cdots, w_{N_k-1}),$$

where w_j with $j = 1, 2, \cdots, N_k - 1$ satisfies (22) with (23).

Events E, L and P_{mot}:

We have that

$$X_{k+1} = (w_0, w_1, w_2, \cdots, w_{N_k-1}),$$

where w_j with $j = 0, 1, 2, \cdots, N_k - 1$, satisfies (22) and (23).

Event P_c:

This only happens when $K_k > 0$. The K_kth carriage which was in the loose part during T_k will become part of the tight part at t_{k+1}. We then have that

$$X_{k+1} = (x_{k1}, x_{k2}, \cdots, x_{kK_k-1}, u_{K_k}, u_{K_k+1}, \cdots, u_{N_k}), \tag{24}$$

where u_j with $j = K_k, K_k + 1, \cdots, N_k$, satisfies

$$m(N_k - K_k + 1)\ddot{u}_j = -c(N_k - K_k + 1)\dot{u}_j + F_{\text{dan}}, \tag{25}$$

and initial conditions

$$u_j(t_{k+1}) = x_{kj}(t_{k+1}), \quad \dot{u}_j(t_{k+1}) = \dot{x}_{kj}(t_{k+1}). \tag{26}$$

Events E and P_c:

Again, $K_k > 0$ and it follows that

$$X_{k+1} = (z, x_{k1}, x_{k2}, \cdots, x_{kK_k-1}, u_{K_k}, u_{K_k+1}, \cdots, u_{kN_k}), \tag{27}$$

where z is satisfies (19) and (20) and where u_j with $j = K_k, K_k + 1, \cdots, N_k$ satisfies (25) and (26).

Events L and P_c:

Then $K_k > 0$ and

$$X_{k+1} = (x_{k1}, x_{k2}, \cdots, x_{kK_k-1}, \hat{u}_{K_k}, \hat{u}_{K_k+1}, \cdots, \hat{u}_{N_k-1}), \tag{28}$$

where $\hat{u}_j$ with $j = K_k, K_k + 1, \cdots, N_k - 1$ satisfies

$$m(N_k - 1 - K_k)\ddot{\hat{u}}_j = -c(N_k - 1 - K_k)\dot{\hat{u}}_j + F_{\text{dan}} \tag{29}$$

and initial conditions

$$\hat{u}_j(t_{k+1}) = x_{kj}(t_{k+1}), \quad \dot{\hat{u}}_j(t_{k+1}) = \dot{x}_{kj}(t_{k+1}). \tag{30}$$

Events E, L and P_c:

Then $K_k > 0$ and

$$X_{k+1} = (z, x_{k1}, x_{k2}, \cdots, x_{kK_k-1}, \hat{u}_{K_k}, \hat{u}_{K_k+1}, \cdots, \hat{u}_{N_k-1}), \tag{31}$$

where z is satisfies (19) with (20) and where $\hat{u}_j$ with $j = K_k, K_k + 1, \cdots, N_k - 1$ satisfies (29) with (30). The ODEs above are solved by (4).

We have given X_{k+1}. We denote $T_{k+1} = [t_{k+1}, t_{k+2})$ and N_{k+1} is the number of carriages. As before, we index X_{k+1} in the following way:

$$X_{k+1} = (x_{k+1\ 1}, x_{k+1\ 2}, \cdots, x_{k+1\ N_{k+1}}).$$

It remains to give t_{k+2}. From (10), (11), (12), (13), (14) it follows that

$$t_{k+2} = \begin{cases} \min(\tau_1, \tau_2, \tau_3) & \text{if } K_{k+1} = 0, \\ \min(\tau_1, \tau_2, \tau_3, \tau_4\tau_5) & \text{if } K_{k+1} > 0, \end{cases}$$

where $\tau_i > t_{k+1}$ with $i = 1, 2, 3, 4, 5$ are the smallest τ_i such that

$$\begin{aligned} x_{k+1\ 1}(\tau_1) &= \ell_{ct}, \\ x_{k+1\ N_{k+1}}(\tau_2) &= \ell, \\ \tau_3 \bmod 1/f_{\text{mot}} &= 0, \\ x_{k+1\ K_k+1}(\tau_4) - x_{k+1\ K_k}(\tau_4) &= \ell, \\ x_{k+1\ K_k+1}(\tau_5) - x_{k+1\ K_k}(\tau_5) &= 0. \end{aligned}$$

We have given T_{k+1}. We are left with proving property 1 and 2, but this follows directly from considering the equations of motion for all the cases.

3 Numerical simulation

We implemented the model in Mathematica. It turns out that the dynamics of the slack caused by the motor does not change if we consider a chain with many carriages. Hence, for convenience we consider a chain starting with only 4 carriages. At $t = 0$ we assume that the system is at rest, which means that the velocity of the carriages is zero. In the physical system many carriages per minute enter via the motor. Hence, we will take f_{mot} large. We consider the following parameters:

$$\ell = 4,\ \ell_{ct} = 1,\ c = 1/100,\ m = 1,\ f_{\text{mot}} = 100,\ v_{\text{mot}} = 10, \tag{32}$$

and for the following initial conditions:

$$X_0(0) = (1/2, 3/2, 5/2, 7/2),\ \dot{X}_0(0) = (0, 0, 0, 0). \tag{33}$$

We will vary the force F_{dan}. The results for $F_{\text{dan}} = 150$ and $F_{\text{dan}} = 250$ are displayed in Figure 7 and Figure 8, respectively. In Figure 7 and Figure 8 the carriages are labelled with numbers so that the motion of each carriage can be followed over time.

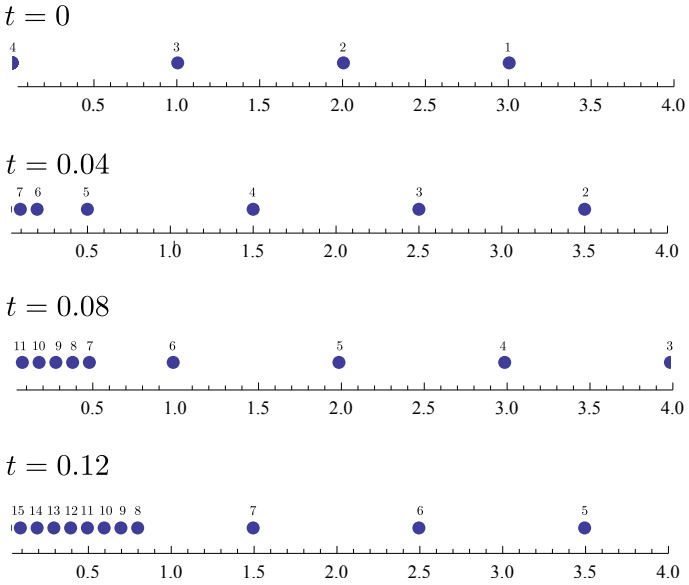

Figure 7: Time frames of the numerical simulations with initial conditions (33) and with parameters (32), $F_{\text{dan}} = 150$.

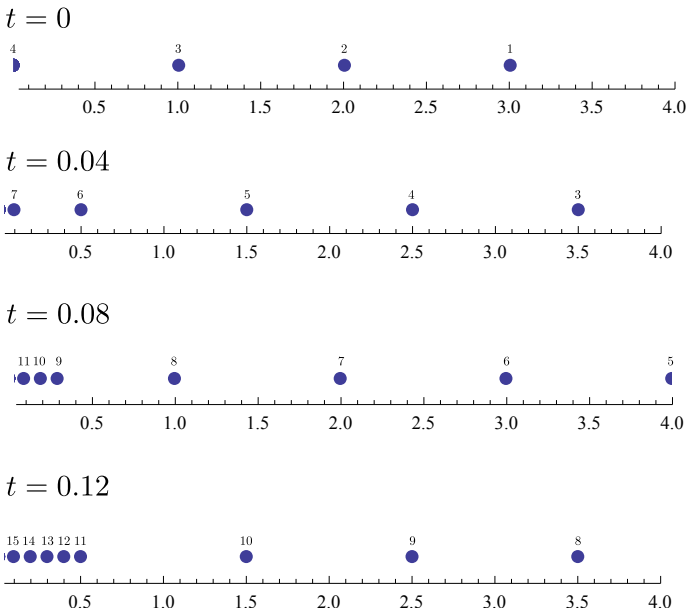

Figure 8: Time frames of the numerical simulations with initial conditions (33) and with parameters (32), $F_{\text{dan}} = 250$.

In Figure 7 and Figure 8 the carriages with loose chain between them are pulled tight by the dancer. We observe that the time it takes for the chain between the loose part and the tight part of the carriages to be pulled tight is too large. This can be seen from the fact that the loose carriages accumulate over time. We find that for $F_{\text{dan}} = 390$ the carriages have the same configuration as for $F_{\text{dan}} = 250$ at $t = 0.12$ at a later time, namely at $t = 2$, see Figure 9.

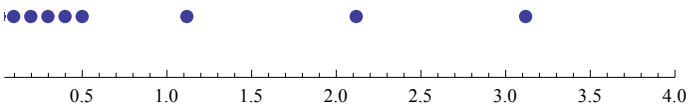

Figure 9: The numerical simulation at $t = 2$ with initial conditions (33), parameters (32) and $F_{\text{dan}} = 390$.

If we take $F_{\text{dan}} \geq 400$ then we find that the events P_{mot} and L are the first events that occur. Recall that when P_{mot} occurs all the velocities become zero. The event $P_{\text{mot}}L$ is followed by the event E after which we are back in the starting configuration and the process repeats. Consequently, there is no accumulation of carriages with loose chain.

4 Conclusions and recommendations

In this paper we constructed a mathematical model for the carriage motion between the master motor and the next dancer. In Section 3 we found that if the force applied by the dancer is large enough then the slack in the chain that is caused by the motor will not spread over the whole chain. This is also observed in the physical system. Alternatively, we could have modelled the chain as a moving continuum (e.g., string Chen (2005)) or a harmonic oscillator (spring-mass system). However, we looked into the approaches and neither of them yielded satisfying results.

The model we formulated is only a first step in the study of this system. There are several ways in which it can be extended. As next steps, we recommend the following extensions of our model:

- **General initial configurations:** Our model can only be used for the initial configuration when the chain is tight between all the carriages. For a more general initial configuration the possible events will increase. Consequently, the event map Ψ will become more complicated.

- **Moving dancer**: The distance between the motor and dancer is assumed to be constant in our model. However, in the physical system the dancer can move and this can be incorporated in our model.

- **Control problem for coupled dancers and motors:** In the original problem the master motor is followed by motors whose speed is coupled to the position of the dancer. Using our model we can formulate a control problem for this system.

- **Not fully loaded chain**: When there is not a product on every trolley but several are empty we should consider a partially filled chain. This can be done by varying the mass of the carriages which enter via the motor.

- **Realistic parameters**: In Section 3 we only consider very specific parameters. Parameters which better fit the reality should be studied.

The numerical simulation in Section 3 is aimed at finding the lowest force on the dancer such that the carriages with loose chains between them do not accumulate. We call this the optimal force of the dancer. Recall that the greater the force of the dancer the shorter the life-time of the chain. A topic for future work is a further improvement of the life-time of the chain. More specifically, by modifying the system we want to take the force of the dancer lower than the optimal force of our model while ensuring that the entire chain will not slack over time. If we could reduce the speed of the carriages with loose chain that leave the motor then the dancer has more time to pull the chains between the carriages tight. This might be accomplished by placing a high friction mat on the rails close to the motor. This reduces the speed of the carriages close to the motor. By using our model it is possible to test whether this might work.

References

L. Chen. Analysis and control of transverse vibration of axially moving strings. *Applied Mechanics Reviews*, 2005.

Courtesy of Marel Stork Poultry Processing.

Energy Consumption of Trains

Tugce Akkaya * Ivan Kryven † Michael Muskulus ‡
Guus Regts §

Abstract

In this report, we consider a problem on energy minimisation of trains proposed by Nederlandse Spoorwegen (NS). Our results include a quick heuristic to compute the energy consumption for a given time table as well as a heuristic to find a timetable which is more energy efficient.
Keywords: Energy minimisation, Timetabling, Heuristic algorithm

1 Introduction

We consider the problem proposed by Nederlandse Spoorwegen (NS) at the Study Group Mathematics with Industry 2016, held at Radboud University, Nijmegen. NS is a Dutch passenger railway operator and provides domestic and international rail services, which makes the company one of the largest consumers of electricity in the Netherlands. Due to environmental considerations and the quality of service for the passengers, NS seeks methods to reduce carbon dioxide CO_2 emissions and to improve the efficiency of the railway system.

Figure 1 shows that the energy optimal way of going from one station to the next (when there are no intermediate constraints). The behaviour of a train is described by four driving regimes: *accelerating*, *cruising* (maintaining constant speed), *coasting* (driving without using energy), *braking*. This is derived using Pontrayagin's Maxium Principle Pontryagin et al. (1962) cf. Howlett (1996); Khmelnitsky (2000); Liu and Golovitcher (2003); Scheepmaker and Goverde (2015a). To find the energy optimal profile one then needs to determine the points x_1, x_2 and x_3 depending on how much time is scheduled to go from one station to the next.

In this project, the main objective is to obtain understanding of how modifications in timetabling can even out the electricity demands, and hereby increase energy efficiency. In fact, this consists of (at least) two subproblems.

- Problem 1: Given a timetable, find the the most energy efficient way for the trains to drive from station to station.
- Problem 2: Find a timetable that uses least energy.

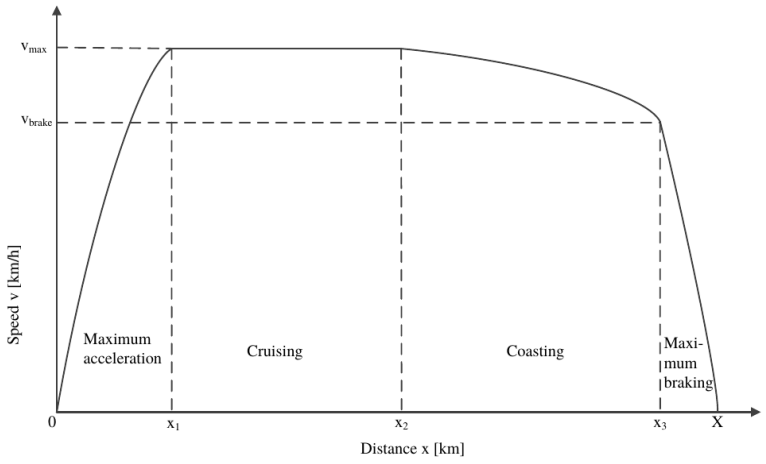

Figure 1: Optimal velocity profile of a basic energy-efficient driving strategy on a level track with switching points between driving regimes at x_1, x_2 and x_3. Courtesy of Gerben M. Scheepmaker(Scheepmaker, 2013).

In view of Figure 1, it looks that one just has to determine the x_i to solve Problem 1 for a given timetable. However, for a journey between two stations there are a lot of additional constraints that are not visible in the public timetable. For example, there are constraints saying that two trains may not pass the same point within 3 minutes.

The paper is organized as follows: In Section 2, we formulate these problems concretely. In the remainder of the paper we focus on our attempts to find a solution to these problems. In Section 3, we look at a heuristic solution for computing the optimal energy profile; i.e., a solution to Problem 1. This heuristic is also tested on a realistic data set from NS. In Section 4, we take a numerical approach to compute the optimal energy profile for a realistic data set from NS and use this to find an improved timetable. We close with discussion in Section 5.

2 Formulation of the problem

In this section, we consider a basic energy-efficient train control model which is the problem of driving along a flat track within a given time T. The train speed $v(t)$ at time t is governed by an energy functional $F(t)$ and a resistance force $r(v)$ according

0

*Delft University of Technology
†University of Amsterdam
‡Norwegian University of Science and Technology
§University of Amsterdam. Email: g.regts@uva.nl

to the Newton force equilibrium

$$\rho m v' = F(t) - r(v(t)), \tag{1}$$

where $v' = \frac{\mathrm{d}v}{\mathrm{d}t}$ is the derivative of velocity to time, m is the train mass, ρ the dimensionless rotating mass factor (Brünger and Dahlhaus, 2007). The resistance force $R(v)$ is given by the Davis equation

$$r(v) = r_0 + r_1 v + r_2 v^2. \tag{2}$$

Here r_1, r_2 and r_3 are non-negative coefficients (Davis, 1926). The energy consumption to be optimised is given by

$$E = \int_0^T F^+(t) v(t) dt, \tag{3}$$

where F^+ denotes the nonnegative part of F. That is we do not assume that the train can gain energy from braking, contrary to e.g. Scheepmaker and Goverde (2015b).

As mentioned in the introduction, if there are no further constraints between station A and B, then Figure 1 gives the energy optimal speed profile. However, generally there are additional constraints to be met between station A and B. A journey consists of *events*. An event should be thought of as 'train α passes junction x' or 'train β arrives at station y' etc. For each event i, there is a variable t_i saying at what time in minutes this event takes place. There is one catch however. Since the timetable should be periodic, these times are to be prescribed modulo 60 minutes. Then there are constraints prescribing how certain events relate to each other; they are all of the form

$$l_{i,j} \leq (t_i - t_j) \mod 60 \leq u_{i,j}, \tag{4}$$

saying that event j should take place at least $l_{i,j}$ minutes later than event i and not later than $u_{i,j}$ minutes after event i. For example, this could encode that the time that train β passes junction x should be at least 3 minutes later than the time that train α passes junction x. When designing a timetable it is exactly the modularity of the constraints that makes this a really difficult task. So one usually modifies a feasible solution to obtain a better solution. In particular, fixing a feasible solution, i.e. a timetable that satisfies the constraints, one can get rid of the modularity constraints and then the constraints (4) all of a sudden look much nicer: they are totally unimodular; see Schrijver (1998) for details on totally unimodularity and its use in optimisation.

3 Heuristic solution

In the case of a single segment the optimal solution consists of four different phases: acceleration, cruising, coasting and braking, in this order (Howlett and Pudney, 1995). The optimal length of each phase can be found by a simple line search (e.g., using the cruising speed as parameter).

The optimal solution in the case of multiple segments along a railway track is fundamentally different and more difficult to obtain, especially if one is interested in a computationally efficient solution. In the following we will suggest an approximate solution based on heuristic reasoning. The motivation for this is the following theorem, which for a lack of better name we call the *friction theorem*.

3.1 The friction theorem

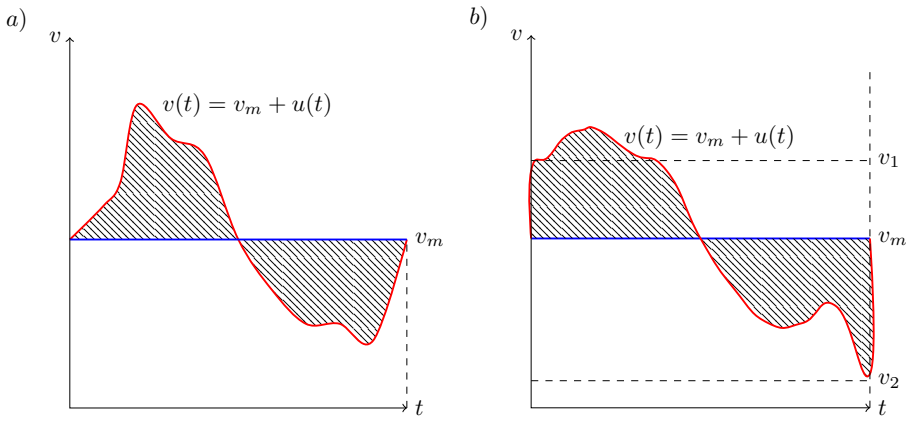

Figure 2: Illustration of the friction theorem. a) In velocity space the area under a curve corresponds to the distance travelled. The constant trajectory with the mean velocity v_m (blue curve) needs less energy than any other equal-area trajectory $v(t) = v_m + u(t)$ (covering the same distance) starting and ending with v_m (red curve). b) A consequence of the friction theorem is a bound on the maximum energy that can be saved by an equal-area trajectory starting at velocity $v_1 > v_m$ and ending at $v_2 < v_m$.

Theorem 3.1. *The optimal way of getting across a distance x_1 in time t_1, when nonlinear friction is acting and when starting and ending with the average speed $v_m = x_1/t_1$, is by traveling all the way at the average speed.*

The proof is based on the intuition that nonlinear friction forces do not average out across the trajectory. We will show this for the Davis model of friction (2) that is relevant for railway problems.

Proof. We decompose the work dW performed on the system by external forces into a contribution dR due to the frictional resistance and a contribution dT used to raise or lower the kinetic energy: $dW = dR + dT$. The kinetic energy is the same at the beginning and at the end of the trajectory, therefore $\int dT = \Delta T = 0$. The total work

done on the system is therefore equal to the work done against friction, and amounts to

$$W = \int \mathrm{d}W = \int \mathrm{d}R = \int_0^{x_1} r(v)\,\mathrm{d}x = \int_0^{t_1} r(v)v\,\mathrm{d}t. \tag{5}$$

Decompose the trajectory in velocity space into $v(t) = v_m + u(t)$, where $v_m = x_1/t_1$ is the mean velocity (Figure 2a). Compared with the mean trajectory $v(t) = v_m$, the difference in energy expended is

$$\begin{aligned}\Delta W &= \int_0^{t_1} r(v_m + u)(v_m + u)\,\mathrm{d}t - \int_0^{t_1} r(v_m)v_m\,\mathrm{d}t, \\ &= r(v_m)\int_0^{t_1} u\,\mathrm{d}t + \int_0^{t_1} (r_1 u + r_2 u^2 + 2r_2 u v_m)(v_m + u)\,\mathrm{d}t.\end{aligned} \tag{6}$$

The first term is zero due to the constraint on the distance travelled (which is equal to the area under the velocity trajectory),

$$\int_0^{t_1} (v_m + u)\,\mathrm{d}t = x_1 = \int_0^{t_1} v_m\,\mathrm{d}t \quad \Rightarrow \quad \int_0^{t_1} u\,\mathrm{d}t = 0. \tag{7}$$

The remaining term amounts to

$$\begin{aligned}\Delta W &= \int_0^{t_1} (r_1 u + r_2 u^2 + 2r_2 u v_m)(v_m + u)\,\mathrm{d}t, \\ &= (r_1 + 3r_2 v_m)\int_0^{t_1} u^2\,\mathrm{d}t + r_2 \int_0^{t_1} u^3\,\mathrm{d}t,\end{aligned} \tag{8}$$

where we have used Eq. 7 again to simplify. Writing the remainder as

$$\Delta W = r_1 \int_0^{t_1} u^2\,\mathrm{d}t + r_2 \int_0^{t_1} (u + 3v_m)u^2\,\mathrm{d}t, \tag{9}$$

and using that $|u| \leq v_m$, shows that $\Delta W \geq 0$. □

3.2 Consequences of the friction theorem

Theorem 3.1 has important consequences, in combination with the constraint on distance travelled. Consider first the journey along a single segment or track, that starts at a velocity below the mean velocity v_m and is supposed to finish at a velocity similarly below v_m. Because of the constraint on distance travelled, there needs to be some acceleration in between and the trajectory follows the well-known optimal shape with up to four phases (acceleration, cruising, coasting, braking) in succession.

A trajectory that includes coasting needs to accelerate longer and the final velocity at the end of the segment will be lower than when only cruising (Figure 3). If the loss in kinetic energy due to coasting leads to the coasting ending on the braking curve, some energy has been saved. However, if the loss in kinetic energy due to

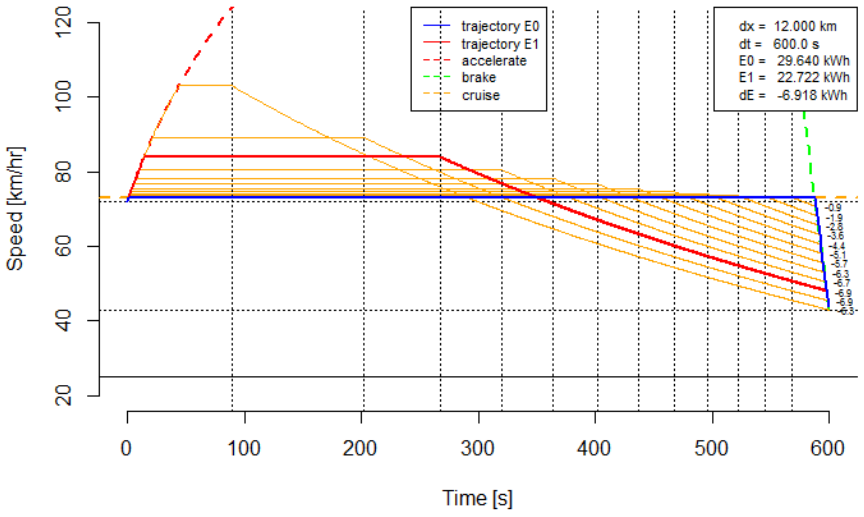

Figure 3: Increasing the energy efficiency by coasting, when there is braking at the end. The blue curve corresponds to travel without coasting. The red and orange curves show alternative trajectories that use coasting to reduce the energy expenditure. Longer coasting needs higher initial acceleration and results in lower velocities. The energy saved with respect to the blue curve is given on the right (in kWh) for each of these curves. In this example about 15 percent of the energy can be saved.

coasting needs to be compensated, i.e., if an additional acceleration is (during this or a following segment) needed because of the coasting, then the friction theorem tells us that this is energetically unfavourable. It is better then to reduce the amount of coasting (by reducing the cruising speed and increasing the cruising phase) until the loss in velocity has no consequences. In other words: coasting can be used to reduce the energy expenditure only when it *replaces braking*, not when it incurs additional acceleration later on. Note that in practice, the potential gain of this is eventually limited by the increasingly unfavourably loss due to the nonlinear behaviour of the friction, cf. Figure 4.

This is the main difference with the situation where only a single segment needs to traversed. In that case, coasting could potentially reduce the energy to zero, if this would result in exactly the right distance travelled. In the case of multiple segments, however, coasting should only reduce the kinetic energy if the train is travelling too fast for the next segment anyway, such that braking would be needed otherwise.

The friction theorem also gives us a bound on the maximum energy saving:

Corollary 3.2. *The energy that can be saved by coasting during a trajectory starting at $v_1 \geq v_m$ and ending at $v_2 \leq v_m$ is at most equal to the kinetic energy difference because of the difference in starting and ending velocities,*

$$\Delta E \leq \frac{1}{2} m \left(v_1^2 - v_2^2 \right). \tag{10}$$

Proof. The friction theorem shows that $\Delta E \leq 0$ for a modified trajectory that includes a (hypothetical, instantaneous) initial and final acceleration from v_m to v_1 and from v_2 to v_m, respectively (Figure 2b). Subtracting the difference in kinetic energy results in Eq. 10 for the trajectory starting at v_1 and ending at v_2. □

What is the optimal amount of coasting? There is no simple, definite answer to this, as it depends on the interplay between the nonlinearities in the friction $r(v)$ and the geometric properties of the trajectory. The optimal trajectory balances replacing as much cruising (work against frictional losses) as possible with coasting (no work) with the increased work during the initial acceleration and (shorter) cruising phase. In practice it seems often to be the case that close to the least amount of cruising leads to the best energy balance (Figure 3).

This leads to the following heuristic, where the phases in brackets can be missing:

- Where possible, replace *cruising + braking* with *accelerating + (cruising) + coasting + (braking)*. If the best curve to follow cannot be determined (e.g., because of the need for a highly efficient method that cannot optimise the cruising speed), use the highest cruising velocity ending on the braking curve.

What happens if the train travels too fast initially? It is always possible to satisfy the constraint on distance by first braking, then cruising, followed by accelerating or braking, as necessary. Similar to the the first case, it is possible to relax this solution by the following heuristic, thereby also improving energy efficiency (Figure 4):

- Replace *braking + cruising* with *(braking) + (cruising) + coasting + accelerating*. If the best curve cannot be determined, use the one with the highest cruising speed (and thereby the lowest speed immediately after coasting).

The energy saving in this case is typically much lower than for the first case and only significant when the acceleration at the end of the segment is very large.

3.3 A reference solution for multiple segments

Solving for the optimal cruising velocities in the above cases of a single segment (Figure 3-4) is not difficult. A straightforward algorithm uses a double loop where the outer loop optimises the cruising speed for the best saving in energy and the inner loop searches for the corresponding length of the cruising phase, in order to fulfill the constraint on distance travelled. As both loops search for a minimum in one dimension, Brent's algorithm or a variant thereof can be used (Press et al., 2007).

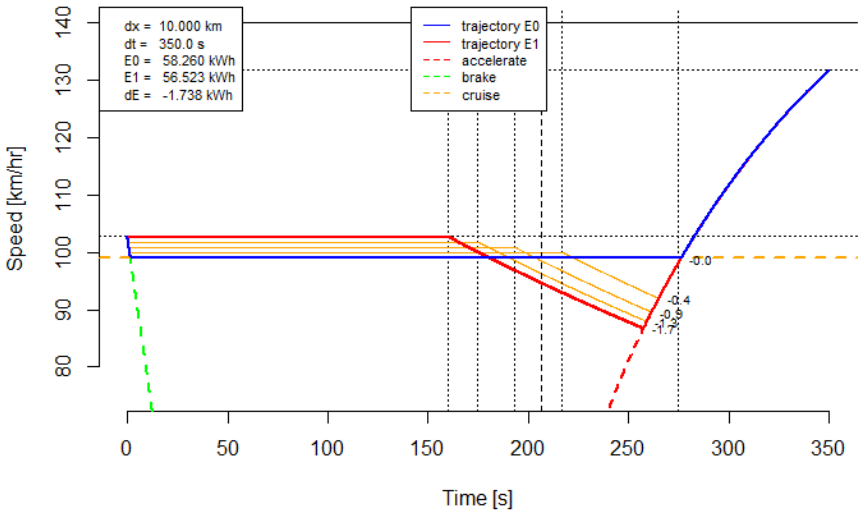

Figure 4: Increasing the energy efficiency by coasting. This is the case with acceleration at the end. This results in the need for an initial drop (braking) for the reference curve (blue) without coasting. The red and orange curves show alternative trajectories that use coasting to reduce the energy expenditure. Longer coasting needs higher initial velocity and results in lower velocities. The energy saved with respect to the blue curve is given on the right (in kWh) for each of these curves. In this example about 3 percent of the energy can be saved.

The main question is how to optimise the energy across multiple segments with intermediary constraints on times and distances. The above suggests a simple, heuristic solution:

1. The first segment is treated in a special way. The train accelerates to the velocity needed to cross the rest of the segment just by cruising. The rest of the segment is then treated as a new segment according to the following procedure.

2. Each segment starts with the mean velocity needed to cross it only by cruising, which would be optimal if not for the differences in mean velocity between segments.

3. Each segment anticipates the subsequent segment and at its end either accelerates or brakes the train to the mean velocity of the following segment.

4. If this cannot be achieved (due to time/distance constraints), then the train accelerates or brakes as much as possible, and the next segment is split into two phases. In the first part the train continues to accelerate or brake until the mean velocity for the remaining second part is reached. (The point where this happens needs to be calculated in an iterative way, since shortening the second part changes its mean velocity). The second part is then treated as a new segment.

5. If braking is needed at the end of the current segment, this means that an additional acceleration is needed at the beginning. Coasting is additionally introduced to relax this situation to a more energetically favourable one, reducing the amount of braking (as in Figure 3).

6. If acceleration is needed at the end of the current segment, this means that additional braking is needed at the beginning. Coasting is additionally introduced to relax this situation to a more energetically favourable one, reducing the amount of braking (as in Figure 4).

3.4 Example solution

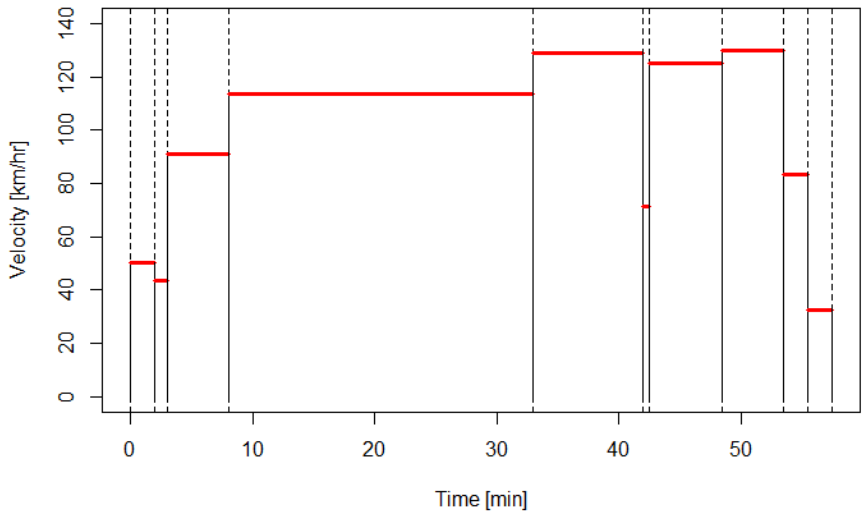

Figure 5: Example track. The mean velocity for each segment is shown. Large differences in these velocities potentially lead to energy-inefficient journeys.

The track between Groningen and Zwolle was used for this example, consisting of in total 10 segments. The mean speed along the segments of the track varies considerably (Figure 5). The train data was compiled from data given by (Scheepmaker, 2013) and NS. The timetable entries were rounded to the minute and are therefore not completely realistic. In fact, the timetable had to be slightly adjusted in order to be feasible.

A reference solution with only cruising needs 582.6 kWh for this track. Solving for the solution with the above algorithm leads to an energy consumption of 550.1 kWh, which is an improvement of 5.6 percent. This value is not the true minimum, but it seems unlikely that the energy expenditure could be further reduced by very much. Most improvements were obtained during the longest segments, where coasting could be used for a significant part of the journey (Figure 6, panels 5 and 8).

3.5 Discussion

This section shows one way of quickly constructing an approximate solution to the most energy-efficient journey along a railroad track with multiple segments (checkpoints). The method is sufficiently fast that it can be used to evaluate thousands of tracks, i.e., a complete timetable, in a reasonable time.

The computations for this section have been made with a simple, straightforward implementation in the system for computational statistics R (R Core Team, 2015). Solving for a single track and plotting the solution takes a few seconds only. Implementing the method in a compiled language and optimizing the code should result in runtimes of a few microseconds per track, which is suitable for applications such as timetable optimisation.

The timetable constrains the solution very much. Especially the occurrence of large differences in mean velocities for different segments of a journey lead to inefficient voyages, due to the need for braking and re-acceleration. Coasting can reduce some of these losses, but often only partially. It seems likely that more energy can be saved by adjusting the timetable (if possible) then by further optimizing the individual journeys for the given timetable beyond what has been shown here. As a next step one should therefore investigate how changes in the timetable affect the energy expenditure.

4 Towards better timetable

4.1 Optimal Energy for a given timetable

Two stops, A and B, are positioned at distance X apart from each other. We consider a train going from A to B in time T. The velocity at A,B is zero, $v(t) = 0,\ t = t_A, t_B,\ T = t_B - t_A$. In the current setup of the problem the timetable is fixed. That is to say a train has to pass prescribed intermediate points at distances x_i from A at specific times $t_i,\ i = 1, \ldots, N$. Without loss of generality we may consider both the journey time and the distance to be unities: $X = 1, T = 1$, so that $t_A = 0, t_B = 1$

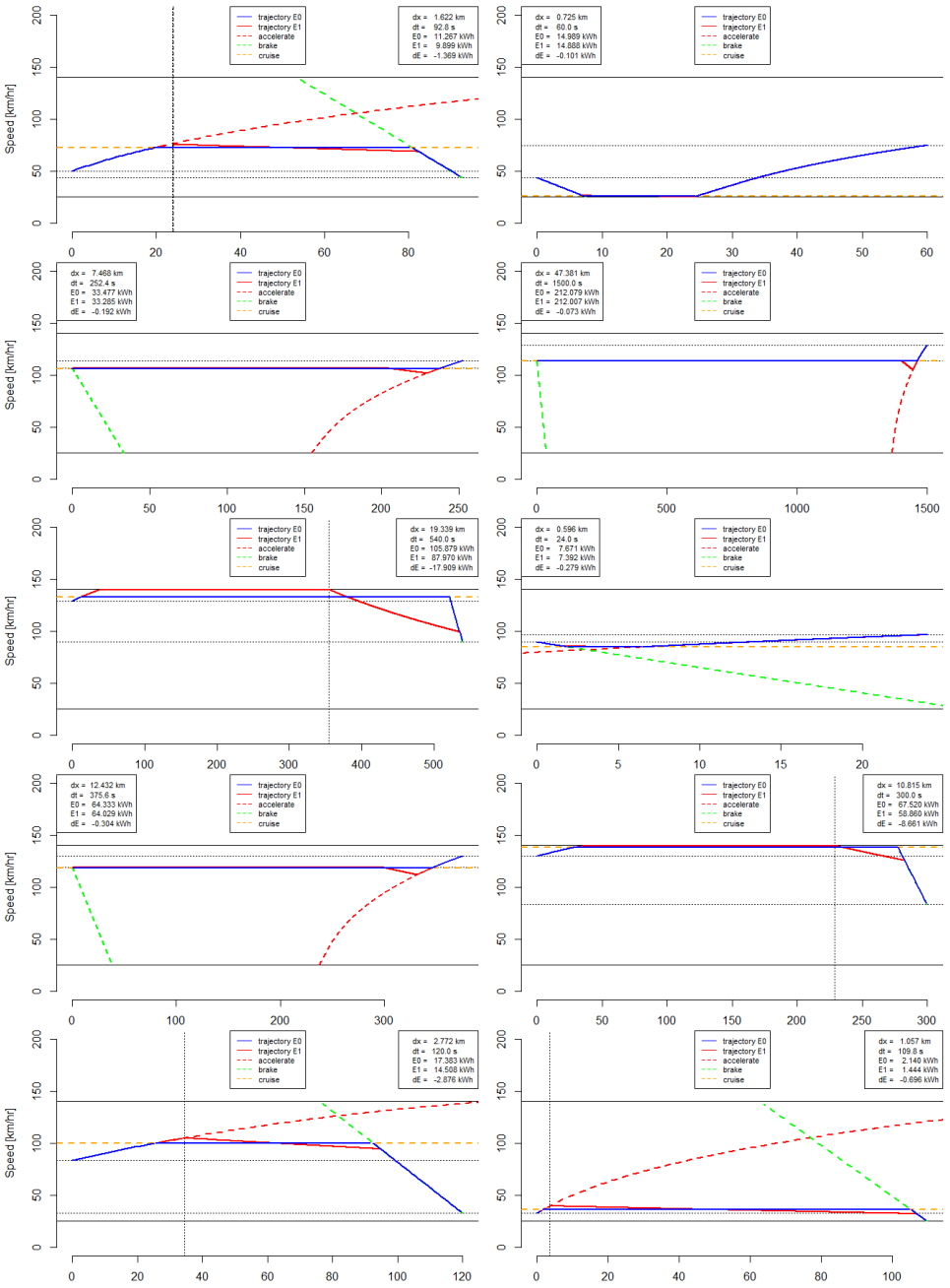

Figure 6: Heuristic solution for example track from Groningen to Zwolle. Each panel shows a segment of the journey. If it is not possible to accelerate enough during a segment (e.g. panel 2 in the top right), an additional acceleration phase is initiated after the segment, adjusting the next segment. These phases are not shown.

and $0 \leq x_i \leq 1$. Then the associated velocity profile $v(t)$ is a continuous function $v(t) \in C[0,1]$ that is restricted by the timetable with the following constraints:

$$v(0) = v(1) = 0 \text{ (full stop at terminal points)}; \tag{11}$$

$$\int_0^1 v(t)\,\mathrm{d}t = 1 \text{ (total distance)};$$

$$\int_0^{t_i} v(t)\,\mathrm{d}t = x_i, \text{ for } i = 1,\ldots,N \text{ (passing } x_i \text{ at time } t_i);$$

$$0 \leq (t) \leq v_{\max} \text{ and } a_{\min} \leq v'(t) \leq a_{\max} \text{ (velocity and acceleration limits)} .$$

The constraints do generally not determine $v(t)$ completely, allowing to search for the specific profile that realises the minimum of the energy functional

$$F(v) = \int_0^1 v[v' + \frac{r(v)}{\rho m}]^+\,\mathrm{d}t, \tag{12}$$

where the nonlinear resistance $r(v)$ is defined in Eq. 2 according to the Davis model.

In order to apply a numerical optimisation algorithm we discretise the continuous function $v(t)$ by means of projection onto the space spanned by a convenient basis:

$$\tilde{v}(t) = \sum_{i=0}^{n} \alpha_i \phi_i(t),\ t \in [0,1].$$

For the sake of simplicity we demonstrate the concept for the piecewise-linear approximation on a uniform grid with step $h = \frac{1}{n}$. That is the approximation coefficients α_i are chosen so that

$$\tilde{v}(\frac{i}{n}) = v(\frac{i}{n}),\ i = 0,\ldots,n,$$

and for $i = 0,\ldots,n$ the basis functions are defined as

$$\phi_i(t) := \begin{cases} 1 - |nt - i|, & \text{if } |nt - i| \leq 1, \\ 0, & \text{otherwise,} \end{cases}$$

that have derivatives

$$\phi_i'(t) := \begin{cases} n, & \text{if } -1 \leq nt - i < 0, \\ -n, & \text{if } 0 < nt - i \leq 1, \\ 0, & \text{otherwise.} \end{cases}$$

In this way, every $\phi_i(t)$ is supported only on interval $[i/n - h, i/n + h]$. Values of $\tilde{v}(t)$ and $\tilde{v}'(t)$ at grid points can be computed as a multiplication of the matrices M, D with coefficient column $\alpha = (\alpha_0,\ldots,\alpha_n)^T$,

$$(M)_{i,j} = \phi_j(\frac{i}{n}),$$

$$(D)_{i,j} = \phi'_j(\frac{i}{n}).$$

The approximation to the energy functional (12) is now expressed as a function of α :

$$\tilde{F}(\alpha) = T\Big([D\alpha + r(\alpha)]^+ \cdot M\alpha\Big), \tag{13}$$

where functions $r(\alpha)$, $[\alpha]^+$ and multiplication $\cdot$ are taken element-wise and T implements appropriate integration quadrature. In the case of a linear basis this is the trapezoidal rule,

$$(T)_{i,j} = \begin{cases} \frac{1}{2(n-1)}, & 0 \le i-j \le 1, \\ 0, & \text{otherwise.} \end{cases}$$

Finally, the cumulative integral of $v(t)$ is approximated by the vector product $q(\tau)^T\alpha$,

$$(q(\tau))_i = \int_0^\tau \phi_i(t)\mathrm{d}t.$$

Now, we are ready to formulate a non-linear optimisation problem that approximates the desired solution $v(t)$:
find a vector $\alpha \in \mathbb{R}^{n+1}$ such that

$$\begin{aligned} &(M\alpha)_0 = 0 \text{ and } (M\alpha)_n = 0; \\ &q(1)^T\alpha = 1; \\ &q(t_i)^T\alpha = x_i; \\ &0 \le (M\alpha) \le v_{\max}; \\ &a_{\min} \le (D\alpha) \le a_{\max}; \\ &\text{and put } \tilde{F}(\alpha) \to \min. \end{aligned} \tag{14}$$

To illustrate the concept let us consider the case when there is only one intermediate constraint, i.e., a train going from A to B has to pass intermediate point x_1 precisely a time t_1. We treat position as fixed, $x_1 = 0.5$, and by varying t_1 obtain a family of velocity profiles $v_{t_1}(t)$ corresponding to minimal energies, as shown in the left panel of Figure 7. One may observe that certain constraints yield optimal velocity profiles with lower energy cost than others, (see Figure 7, right panel). The velocity profile that has the smallest energy within the family is also the optimal velocity profile with no intermediate constraints. This observation can be used to adjust the given timetable in order to achieve even better energy efficiency (see Figure 8).

4.2 Optimisation of train timetable

Here, we assume that a train always travels according to the optimal velocity profile. The main question is: can we alter the existing set of constraints (i.e. timetable)

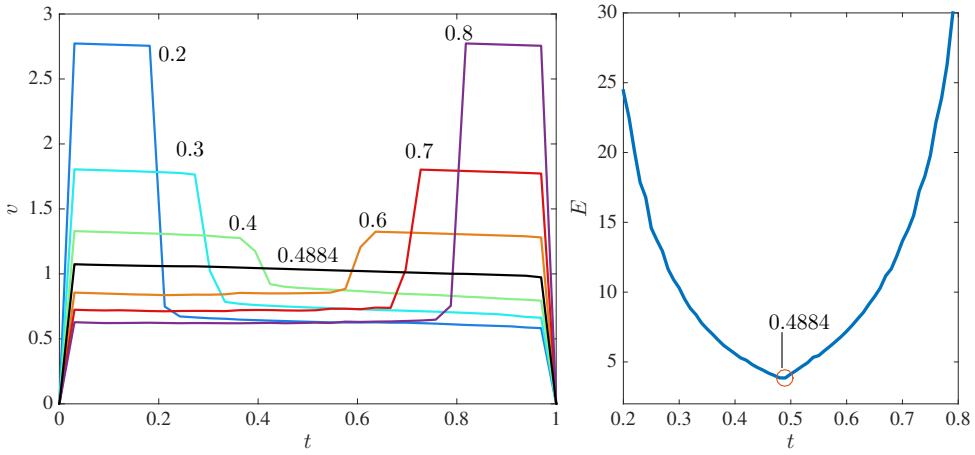

Figure 7: *L*eft: optimal velocity profiles for a single intermediate constraint with position $x = 0.5$ and various passing-time values (indicated). *R*ight: the optimal energy depends significantly on the passing time, t. The smallest optimal energy is reached if constraint's passing time coincides with the passing time of the unconstrainted velocity profile (i.e. 0.4884 for the current value of the constraint's position).

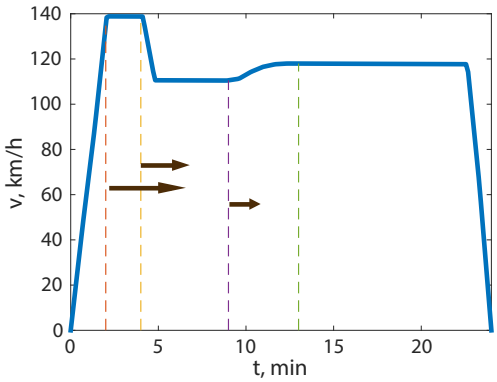

Figure 8: Optimal velocity profile for 4 constraints. The timetable can be improved by moving the constraints towards their optimal place (as if the constraint passing times belong to the unconstrained profile v_o.)

so that the energy consumption is even better? Small adjustments to the timetable (t_i, x_i) are feasible as long the timetable satisfies the periodic event scheduling model,

$$l_{i,j} \leq (t_j - t_i) \mod 60 \leq u_{i,j},$$

where t_i, t_j are event times and $l_{i,j},\ u_{i,j}$ are fixed limitations. In principle it is possible to directly set up an optimisation with an objective function defined as the energy of the timetable $f_{\mathrm{o}} = \tilde{F}(\alpha)$ where α solves the optimal velocity profile problem from the previous section. Such a routine, however, has to deal with a big non-linear optimisation problem and thus requires a good initial guess. We obtain this initial guess by running optimisation with a heuristic objective function. Let $v_{\mathrm{o}}(t)$ be an optimal energy profile with no intermediate constraints. We construct a heuristic objective function $f_{\mathrm{h}}(t_1, \ldots, t_N)$ that measure how far in L^2 norm is the given set of constraints t_i from passing times according to the optimal profile τ_i :

$$f_{\mathrm{h}}(t_1, \ldots, t_N) = \sum_{i=1}^{N} (t_i - \tau_i)^2,$$

where τ_i solves $\int_0^{\tau_i} v_{\mathrm{o}}(t)\,\mathrm{d}t = x_i$. If a train makes stops at $(t_{x,i}, x_{s,i}),\ i = 1, \ldots, M$ we will additionally require the average speed between each pairs of stops be close to the overall average speed, v_{avg} (when calculated between terminal stations),

$$f_{\mathrm{h}}(t_1, \ldots, t_N) = \sum_{i=1}^{N} (t_i - \tau_i)^2 + \sum_{i=2}^{M} (t_{s,i} - (x_{s,i} - x_{s,j-1})/v_{\mathrm{avg}})^2. \tag{15}$$

Such an objective function provides a crude optimality estimate for a timetable. This estimate can be later used as an initial guess for, computationally more expensive, optimisation involving the functional $\tilde{F}(v)$ in 'predictor/corrector' combination. Table 1 depicts results of such an approach applied to a sample timetable. The first column of Table 1 contains information on the current timetable; the second column describes results of heuristic optimisation (CPU time less than 1 sec); the third column contains correction of the heuristic results by energy optimisation according to the functional $\tilde{F}(v)$ (CPU time 1.5 hour). Fragments of the optimal velocity profile for the optimised and original timetables are given in Figure 9.

Type	t_i	Predictor	Δ	Corrector	Δ
D	37	37	0	36.97	−0.03
P	41	41	0	40.99	−0.01
P	42	42	0	41.93	−0.07
A	44	44	0	43.96	−0.04
D	45	45	0	44.98	−0.02
P	47	47	0	47.00	0
P	48	48	0	48.01	+0.01

A	50	50	0	50.00	+0.01
D	51	51	0	50.98	−0.02
P	52	52	0	52.02	+0.02
P	54	54	0	53.94	−0.06
P	55	55	0	55.00	0
P	57	57	0	56.98	−0.02
P	58	58	0	57.96	−0.04
P	59	59	0	58.97	−0.03
P	2	2	0	2.00	0
P	6	5	−1	4.99	−1.01
A	7	6	−1	6.03	−0.97
D	8	7	−1	7.00	−1.00
P	13	13.23	+0.23	13.04	+0.04
P	14	14.23	+0.23	13.97	−0.03
A	21	21	0	21.00	0
D	23	23	0	22.97	−0.03
P	25	25	0	24.98	−0.02
P	27	27	0	26.93	−0.07
P	32	32	0	31.85	−0.15
P	36	36	0	35.83	−0.17
A	47	47	0	46.75	−0.25
D	48	48	0	47.67	−0.33
P	49	49	0	48.60	−0.40
P	50	50	0	49.65	−0.35
P	53	53	0	53.15	+0.15
A	58	58	0	58.07	+0.07
D	0	0	0	0.00	0
P	2	2	0	1.99	−0.01
P	3	3	0	3.00	0
P	4	4	0	3.98	−0.02
P	15	15	0	14.98	−0.02
P	20	19	−1	18.90	−1.10
A	24	23	−1	22.97	−1.03
D	26	25	−1	24.94	−1.06
P	37	35	−1	35.05	−1.95
P	38	36	−1	36.01	−1.99
A	45	46	+1.5	46.34	+1.34
D	41	36	−5	36.22	−4.78
P	43	43	0	43.02	+0.02
A	50	50	0	50.04	+0.04
D	53	52	−1	52.07	−0.93
P	4	4	0	4.01	+0.01
A	6	6	0	5.94	−0.06
D	10	10	0	9.93	−0.07

P	11	11	0	11.24	+0.24
P	23	26	+1	26.05	+3.05
A	25	28	+3	28.08	+3.08
Distance from original			24.0min		26.19min
	Total energy		**87.39%**		**83.96%**

Table 1: A sample of a real timetable with 12 stops and 42 passing constraints. All distances are indicated in km and time in min. The timetable is consequently optimised with heuristic (predictor) and energy-functional (corrector) objective functions. The constraint types are encoded as follows: **D**eparture, **P**assing, **A**rrival. Distance from original indicates the sum of absolute changes in minutes.

4.3 Conclusions

For a given timetable we can find the optimal velocity profile numerically. This information may be presented to train drivers as an advisory. The routine computing optimal velocity profiles and energy is then further used to adjust the existing timetable. Such adjustment is done in two steps: heuristic objective function (cpu time 1sec, reduces energy down to 87.39 on sample data), and energy objective function (cpu time 1.5h, 83.96 on sample data). Even though the energy reduction is quite high, this approach involves numerical non-linear optimisation and does not necessarily lead to global minimum.

5 Conclusion and discussion

In this paper, we have looked at the problem proposed by NS. We considered two approaches. The first approach was primarily aimed at trying to reduce energy consumption while not changing the timetable. This was done by trying to understand what an optimal journey (with respect to energy consumption) looks like. Using this knowledge we developed a simple heuristic to optimise the usage of energy of a single train journey. This heuristic has been applied to a sample of actual train data and resulted in a energy reduction of 5%.

In the second approach, our aim was to compute for a given timetable the optimal energy profile numerically. Using this we applied numerical optimisation to a sample of an actual time table. Since the constraints Eq. 4 are modular this is not an easy task. However, taking the current timetable one can rewrite these constraint to absoute constraints. This resulted in a time table (for the sample) for which the optimal velocity profile yields a 16% energy reduction.

The main conclusion that can be drawn from this work is that energy consumption can in fact be reduced significantly. Not only by more efficient driving, but also by making small adjustments to the timetable allowing for more efficient velocity profiles. We note however that our results have only been applied to small samples of the timetable. To see what happens on a larger scale one should of course apply

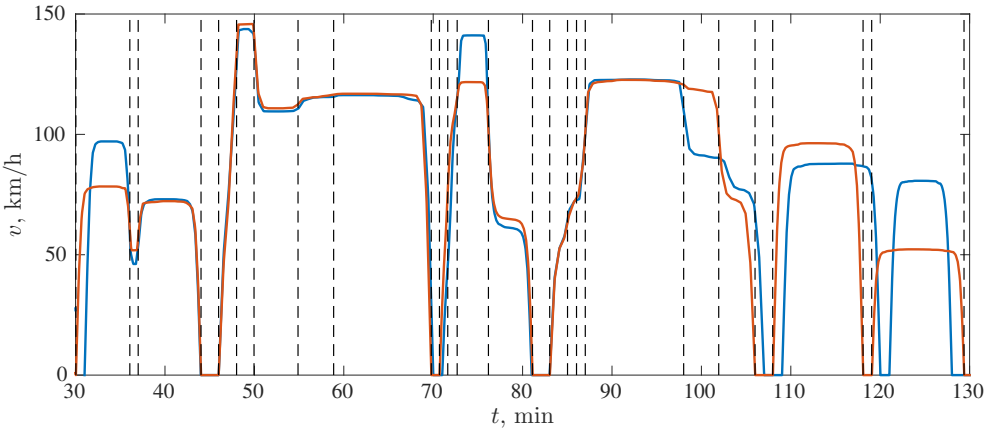

Figure 9: Fragments of optimal velocity profiles for current (*blue*) and improved (*red*) timetables. The vertical lines represent constraints after optimisation.

our results to the entire timetable. One thing that we observed is that prescribing time in minutes appear to make matters a bit complicated. For example the current timetable has some inconstancies, i.e. a train α should be at position x at time t but also on position x' at the same time. So it makes more sense to determine these times more accurately. Also from the point of view of energy reduction this makes sense. Allowing more flexible times values (not just entire minutes) can already lead to significant energy reduction (for the optimal profile).

It is not unlikely that the methods we have used can be improved. In particular, we believe that it would pay off to get a fast direct computation of the optimal velocity profile given a timetable. This could then be used to search for a better timetable with more advanced heuristics than we have currently employed.

Acknowledgements

We thank Guus Berkelmans, Wouter Berkelmans and Majid Salmani who were members of our group that worked on the problem from 25 January until 29 January in Nijmegen. We also thank Gábor Maróti from NS for introducing the problem to us.

References

O. Brünger and E. Dahlhaus. *Running time estimation.In: Hansen IA, Pachl J (eds) Railway Timetabling and Operations.* Eurail Press, Hamburg, Germany, second edition, 2007.

W. Davis. *The tractive resistance of electric locomotives and cars*, volume 29. General Electric Review, 1926.

P. Howlett. Optimal strategies for the control of a train. *Automatica*, 32:519–532, 1996.

P. Howlett and P. Pudney. *Energy-efficient train control.* Springer, London, 1995.

E. Khmelnitsky. On an optimal control problem of train operation. *IEEE Transactions on Automatic Control*, 45:1257–1266, 2000.

R. Liu and I. Golovitcher. Energy efficient operation of rail vehicles. *Transportation Research Part A: Policy and Practice*, 37:917–932, 2003.

L. Pontryagin, V. Boltyanskii, R. Gamkrelidze, and E. Mishchenko. *The Mathematical Theory of Optimal Processes.* Wiley, New York, 1962.

W. H. Press, S. A. Teukolsky, and W. T. Vetterling. *Numerical recipes: The art of scientific programming.* Cambridge University Press, Cambridge, 2007.

R Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2015. URL `http://www.R-project.org/`.

G. Scheepmaker. Rijtijdspeling in treindienstregelingen: energiezuinig rijden versus robuustheid. Master's thesis, Delft University of Technology, The Netherlands, 2013.

G. Scheepmaker and R. Goverde. Running time supplements: energy efficient train control versus robust timetables. In *Proceedings 6th International Conference on Railway Operations Modelling and Analysis (Rail-Tokyo 2015)*, 23-26 March 2015a.

G. Scheepmaker and R. Goverde. Effect of regenerative braking on energy-efficient train control. In *Conference on Advanced Systems in Public Transport (CASPT 2015)*, 19-23 July 2015b.

A. Schrijver. *Theory of linear and integer programming.* John Wiley & Sons, 1998.

Frequency decompositions in autoregression models

Wouter Cames van Batenburg [*] Aleksander Czechowski[†]
Joey van der Leer Duran [‡] Bert Lindenhovius [§] Eric Siero [¶]

Abstract

Autoregression models are used by Ortec Finance to forecast the evolution of economic variables, such as interest rates. To distinguish the impact of short, medium and long term fluctuations, the company decomposes their models into three components: month, business and trend, respectively. We answer the question of how to design a model, so that predictions generated for a given frequency band do not overlap with other frequencies. We also discuss several other related matters, i.e. how to address the frequency leaking problem, how to choose the number of frequencies in each band and how our method generalizes to time-dependent models.

KEYWORDS: Fourier filter, autoregression, time series forecast

1 Introduction

This paper contains results on the problem of designing a good filtering method for autoregression models posed by Ortec Finance for the 114th European Study Group Mathematics with Industry. The general setting of the problem is as follows. Suppose we have a time series $\mathbf{r} = \{r_t\}_t$, where r is a quantity of interest (such as interest rate, oil price etc.), or a collection thereof, and $t \in \mathbb{Z}$ is a time parameter which takes discrete steps (representing months, years etc.). We want to make future predictions of r_t, given a historical set of values. A natural approach for forecasting based on data of such a time series is to describe it as a function of its predecessors

$$r_t = f(r_{t-1}, r_{t-2}, \ldots) + \epsilon_t, \tag{1}$$

where f is some function and ϵ_t is a sequence of independently, identically distributed random variables representing the probabilistic nature of future predictions. Vaguely

[*]Radboud University Nijmegen
[†]Jagiellonian University, Kraków
[‡]Utrecht University
[§]Radboud University Nijmegen
[¶]Leiden University

put, the aim is to choose f as simple as possible, while minimizing the deviation of the error terms ϵ_t. Often f is taken to be linear and dependent on only finitely many predecessors, in which case the model is called the AR(k) model:

$$r_t = c + \sum_{p=1}^{k} a_p r_{t-p} + \epsilon_t,$$

with $a_p \in \mathbb{R}$. We will assume that the ϵ_t are independent and identically distributed with mean 0 and the same standard deviation σ. Such a set $\{\epsilon_t\}_t$ is called *white noise*.

The AR(k) model is a special case of the *vector regression model*, where r_t and ϵ_t are both vector valued and the recursive structure is given by

$$r_t = c + Ar_{t-1} + \epsilon_t \tag{2}$$

where A is a matrix and c a vector. We will impose the restriction $||A|| < 1$, where $||A||$ denotes the operator norm of A, which will allow us to discard high powers of A. Intuitively, this condition amounts to stability of the model, but we will not make this statement precise.

By demeaning the data we can take $c = 0$. Let us also assume that the sequence starts at $t = 0$ with value r_0. Then, equation (2) has the following solution

$$r_t = \sum_{l=0}^{t-1} A^l \epsilon_{t-l} + A^t r_0.$$

Note that the expectation $\mathbb{E}(r_t)$ decays to 0 as time goes to infinity, because the ϵ_t have zero mean and $||A|| < 1$. Furthermore, as time becomes large the effect of the initial value diminishes.

Of course one cannot expect a single forecast based on such a rough model to be accurate. The value of the method is that it can be used to quickly generate a large number of scenarios and evaluate probabilities of future states via Monte Carlo experiments.

Ortec's approach to forecasting economic variables via autoregression models is to decompose the time series into a sum

$$\mathbf{r} = \mathbf{r}^T + \mathbf{r}^B + \mathbf{r}^M, \tag{3}$$

where $\mathbf{r}^T$ represents the long term (*trend*) fluctuations, $\mathbf{r}^B$ the medium term (*business cycle*) oscillations, and $\mathbf{r}^M$ the short term (*month*) movements. This is performed via so-called *filters*. We will elaborate more on them in Section 2, but to give some intuition, let us mention that a basic example is a filter is based on a discrete Fourier transform (*the Fourier filter*). For this filter the terms $\mathbf{r}^T$ correspond to low Fourier modes, the terms $\mathbf{r}^B$ to medium ones and $\mathbf{r}^M$ to the high frequencies.

The motivation for the decomposition (3) is that short, medium and long term terms are (to a certain degree) independent from each other, and, as such, their

evolution should also be forecasted independently. However, a naive approach of applying an AR model separately for the trend, business cycle and month components can result in an undesirable effect, where a forecast for shorter fluctuations starts developing long term movements e.g. a forecast for the month term evolves a trend on its own.

For more background information about the intuition and practical applications of the frequency decomposition approach, we refer to Van der Schans and Steenhouwer (2012) and references therein.

In this paper we propose a *filtered AR model*, where long-term predictions are made for each term separately, in such a way that they stay in their own frequency band (which is chosen on the basis of historical data). Given a frequency decomposition, i.e. the choice of the filter, the recipe for such a prediction for a given term is as follows:

1. Firstly, we choose a forecasting period, which we specify by an integer $N \in \mathbb{Z}$, so that the outcome of the prediction will be a set $\{r_1, \ldots, r_N\}$, with initial condition r_0.

2. Secondly, we generate a time series of noise $\{\epsilon_t\}_{1 \leq t \leq N}$ of length equal to the forecasting period N, using the white noise probability distribution.

3. Thirdly, we apply the filter to the sequence $\{\epsilon_t\}_t$, in order to obtain a *filtered noise sequence* $\{\epsilon_t^*\}_t$.

4. Finally, we use the sequence $\{\epsilon_t^*\}_t$ to generate a prediction by the formula (2).

The filter F is typically chosen on the basis of the last N consecutive historical data points, so that both the employed historical data and the prediction are represented by a $N-$dimensional time series, implying that both are in the domain of F. As a consequence, it is possible to apply the *same* filter to both the employed historical data and the prediction, thus facilitating a meaningful comparison between the filtered historical data and the filtered prediction.

In order to apply the recipe, the forecasting period N can be arbitrary. In practice however, it is limited by the amount of historical data we have available.

In Section 2 we show, for the class of filters that are linear, weakly translation invariant and commuting with the autoregression parameter matrix, that such predictions will indeed remain in their own frequency band. Next, we give an example of two linear filters. The Fourier filter, presented in Subsection 2.1 is translation invariant, hence it can be employed in the filtered AR model. Another example is the Christiano-Fitzgerald Band Pass Filter, treated in Subsection 2.2. It is not clear whether this filter is weakly translation invariant. However, its advantage is that it deals with the *frequency leaking* problem, discussed later. In Section 3 we extend this method to regression models with time-dependent parameters.

Another problem we deal with is how to choose a partition into frequency bands. Ortec Finance chooses its decompositions based on heuristic reasoning backed by

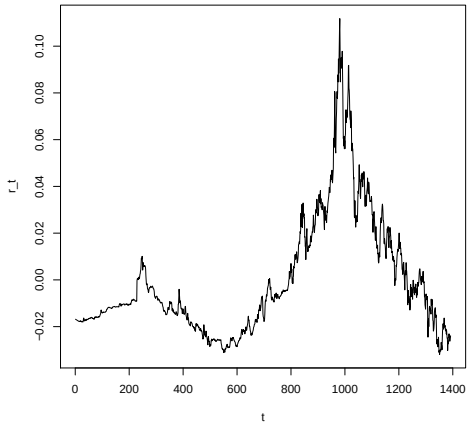

Figure 1: Demeaned interest rate of US bonds with a 10 year term over the past 116 years (in months; x-axis).

economic theories. We propose a different approach, where the allocation of the frequencies to each of the three terms is chosen to minimize the total variance of the given historical time series $\{r_t\}_t$ with respect to the filtered regression model. The rationale behind this is that the variance gives a measure of how well the regression model fits the given data, which is also the reason why least squares methods are often used to estimate the parameters of such models.

Due to analytical difficulties, we only performed a numerical study, and implemented our idea on historical data of the (univariate) interest rate of US bonds with a term of 10 years (see Figure 1). The details are presented in Section 4.

2 Filters

As discussed previously, a time series of interest can sometimes be regarded as a superposition of other time series with different kinds of evolution behaviors. To take that into account, we introduce a filtering process below, whose purpose is to decompose time series into its different constituents.

Let $l^\infty(\mathbb{R}) := \{(x_n)_{n\in\mathbb{Z}} | \sup|x_n| < \infty\}$ be the vector space of bounded sequences, and denote by $L : l^\infty(\mathbb{R}) \to l^\infty(\mathbb{R})$ the shift operator $(L(x))_n = x_{n-1}$. We are mainly interested in finite sets of data, which we regard as a subset of $l^\infty(\mathbb{R})$ by periodic extension. More precisely, we fix an $N \in \mathbb{Z}_{>0}$ and define $V \subset l^\infty(\mathbb{R})$ as the subspace of N-periodic sequences. Obviously, V is invariant under L.

Definition 2.1. A *linear filter* is a linear map $F : V \to V$. We call F *weakly translation invariant* if $L(im(F)) \subset im(F)$.

We think of F as a device that takes a time series and picks out a component with a specific evolution behavior. In particular, time series in the image of F are to be thought of as evolving in this specific way. The question at hand is how to produce a regression model that has as output a time series in the image of F. To this end, we modify the vector regression model (2) as follows. Let $F^1, \ldots, F^d$ be linear filters and define $F := diag(F^1, \ldots, F^d)$, considered as a linear map from $V^{\oplus d}$ to itself. Let ϵ_t^i be a set of random variables, with $1 \leq i \leq d$ and $t \in \{0, \ldots, N-1\}$, put together into a sequence of vectors $\epsilon_t := (\epsilon_t^1, \ldots, \epsilon_t^d)$. We take all ϵ_t^i independent of each other and, for each fixed i, we take the ϵ_t^i identically distributed with zero mean and variance σ_i. We can regard $(\epsilon_t^i)_{1 \leq t \leq N}$ as an element of V, and we will denote it by ϵ^i. Then, we define

$$\epsilon_t^* := (F\epsilon)_t = ((F^1\epsilon^1)_t, \ldots, (F^d\epsilon^d)_t).$$

Simply put, we have d sequences of random variables and d filters, and we apply the filters component-wise. The *filtered vector regression model* is then defined by an initial value r_0, with time evolution given by

$$r_t = Ar_{t-1} + \epsilon_t^*, \tag{4}$$

where, as before, A is a matrix satisfying the stability condition $||A|| < 1$. Note that, in contrast to the non-filtered regression models, we need to specify the prediction period N in advance, in order for (4) to make sense. Indeed, we first need all the ϵ_t in order to apply the filter, after which the regression model can be initiated. The answer to the above question is given by the following proposition.

Proposition 2.1. *If all the F^i are linear and weakly translation invariant and if A commutes*[1] *with $F = \mathrm{diag}(F^1, \ldots, F^d)$, then*

$$\Big(r_t - A^t r_0\Big)_{1 \leq t \leq N} \in Im(F).$$

So, except for the initial value terms $A^t r_0$ that converge to 0, the output of the filtered regression model is contained in the image of F.

Proof. By writing out the definitions and using that $[F, A] = 0$ we get

$$\begin{aligned} r_t &= \sum_{l=0}^{t-1} A^l \epsilon_{t-l}^* + A^t r_0 = \sum_{l=0}^{t-1} A^l (L^l F\epsilon)_t + A^t r_0 = \sum_{l=0}^{t-1} A^l (Fy^l)_t + A^t r_0 \\ &= \Big(F\big(\sum_{l=0}^{t-1} A^l (y^l)\big)\Big)_t + A^t r_0. \end{aligned}$$

[1] To be precise, A is a $d \times d$ matrix which acts naturally on $V^{\oplus d}$, while F acts on $V^{\oplus d}$ in a diagonal way by applying F^i to each component.

In the third equality we used that L preserves the image of F, so that we can find sequences y^l with the property that $L^l F\epsilon = Fy^l$. □

This proposition tells us that if we think of the filter as forcing the noise $(\epsilon_t)_t$ to have a certain time evolution, then the prediction for r will have this time evolution as well (at least in the long run, if we ignore the contribution from the initial value), provided that we use the same filters for those components of r that are interacting with each other (i.e. we need $[A, F] = 0$). We will discuss interactions between evolutions lying in different filters in Section 3.

2.1 The Fourier filter

An example of a linear filter F is the *Fourier filter* defined below. The *discrete Fourier transform* (DFT) of a sequence $x \in V$ is given by

$$\tilde{x}_k := \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x_n e^{\frac{-2\pi i k n}{N}}.$$

The sequence $\tilde{x}_k$ is obviously also N–periodic and the inverse DFT is given by

$$x_n = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \tilde{x}_k e^{\frac{2\pi i k n}{N}}.$$

Given a subset $K \subset \{0, \ldots, N-1\}$ with the property that $k \in K \Leftrightarrow N - k \in K$, we define the Fourier filter with respect to K as the map $x \mapsto Fx =: x^*$, where

$$x^*_n = \frac{1}{\sqrt{N}} \sum_{k \in K} \tilde{x}_k e^{\frac{2\pi i k n}{N}}.$$

Basically, the Fourier filter is given by first applying DFT, then applying a linear projection by forgetting some of the frequencies and then applying the inverse DFT. This example is prototypical for the concept of filter, designed with the purpose of making a separation between different time scales or frequencies, which are expected to have different driving mechanisms. Note that the Fourier filter F is translation invariant, satisfies $F^2 = F$ and $F^* = F$, with respect to the inner product on V given by

$$\langle x, y \rangle := \sum_{n=0}^{N-1} x_n y_n.$$

If $\{0, \ldots, N-1\} = K_1 \cup \ldots \cup K_q$ is a disjoint decomposition, the associated Fourier filters F_{K_i} additionally satisfy: $1 = \sum_i F_{K_i}$ and $F_{K_i} F_{K_j} = 0$ for all $i \neq j$.

2.2 The Christiano-Fitzgerald Band Pass Filter

In this section, we investigate the *Christiano-Fitzgerald filter*, for the following reasons. Firstly, the Fourier filter has the disadvantage of frequency leaking (see also below). For this reason, Ortec Finance is using another filter which is, however, non-linear. The Christiano-Fitzgerald filter might be the most prominent choice of a linear filter that prevents frequency leaking.

The discrete Fourier transform only filters a discrete set of frequencies. However, it may be possible (even plausible) that the frequencies of the input signal do not (perfectly) match the frequencies that are chosen to be filtered by the discrete Fourier transform. The discrete Fourier transform assumes that the input signal is periodic with a certain period, but it could happen that the input signal has a slightly different period. For instance, we want to filter the business cycle component of the interest rate and we assume a period of 8 years. However, the actual period of the rate turns out to be 7 years. It follows that if we use the discrete Fourier filter in order to filter certain frequencies out, we might damp certain eigenfrequencies of the input signal nearby the frequencies we actually want to keep, which is not desirable. This effect is called *frequency leaking*.

Therefore, it is desirable to filter an interval of frequencies. This leads to the *Ideal Band Pass Filter*. Unfortunately, this filter has the disadvantage that it requires the use of an infinite number of input values, whereas data sets are usually finite sets. Hence, an approximation is required, leading to the *Christiano-Fitzgerald Band Pass Filter*. This filter assumes that the historical data follows a random walk pattern (even though in most cases, this is a false assumption).

We start by defining the Ideal Band Pass Filter.

Definition 2.2. Let $(x_n)_{n\in\mathbb{Z}}$ be a time series. Choose $0 < a < b \leq \pi$ and let L be the shift operator sending x_n to x_{n-1}. Then, the Ideal Band Pass Filter is given by

$$B = \sum_{n\in\mathbb{Z}} B_n L^n$$

with

$$B_n = \begin{cases} \frac{b-a}{\pi}, & n = 0, \\ \frac{\sin(nb)-\sin(na)}{n\pi}, & n \neq 0. \end{cases}$$

The sum of all B_n is zero and $B_{-n} = B_n$. Moreover, we have

$$\sum_{n\in\mathbb{Z}} B_n e^{-in\omega} = \begin{cases} 1, & \omega \in (a,b) \cup (-b,-a), \\ 0, & \text{otherwise.} \end{cases}$$

Hence B is a filter that 'accepts' frequencies between a and b. Usually, the data set x_n is split into a 'trend' component t_n and a 'cyclic' component y_n, such that $x_n = y_n + t_n$, where $y_n = Bx_n$ for each $n \in \mathbb{Z}$. By definition, we have

$$y_n = \sum_{k\in\mathbb{Z}} B_k x_{n-k}.$$

Hence we require all x_k in order to calculate y_n. However, usually we only have a finite data set $(x_n)_{n=1}^N$, so the output y_k might not be accurate. We now define the Christiano-Fitzgerald Band Pass Filter (abbreviated by *CF Filter*) C as follows. Let z_n be the solution of minimizing the mean square error

$$E\big((y_n - z_n)^2 | x_1, \ldots, x_N\big).$$

Then, we define $Cx_n = z_n$. For $k = 1, \ldots, N-1$, define

$$\tilde{B}_{N-k} = -\frac{1}{2}B_0 - \sum_{j=1}^{N-k-1} B_j.$$

It is stated in Christiano and Fitzgerald (2003) that for $k \in \{2, \ldots, N-1\}$, we have

$$z_k = B_0 x_k + \sum_{j=1}^{N-k-1} B_j x_{k+j} + \sum_{j=1}^{k-2} B_j x_{k-j} + \tilde{B}_{k-1} x_1.$$

The values of z_1 and z_N are given by

$$z_1 = \frac{1}{2}B_0 x_1 + \sum_{j=1}^{N-2} B_j x_{j+1} + \tilde{B}_{T-1} x_N$$

and

$$z_N = \frac{1}{2}B_0 x_N + \sum_{j=1}^{N-2} B_j x_{N-j} + \tilde{B}_{T-1} x_1.$$

More generally, the CF filter is of the following form (cf. Schleicher (2003)):

$$z_k = \sum_{j=-n_{1,k}}^{n_{2,k}} C_{k,j} x_{k+j}$$

for some coefficients $C_{k,j}$. Clearly, this formula is only translation invariant if $C_{k,j} = C_j$ for each k, which is not the case for the CF filter. It is also not clear, whether the CF filter is weakly translation invariant. Hence, the CF filter might not be a useful filter if one wants to implement the methods that are developed in this contribution. Another filter which deals with frequency leakage is the Hodrick-Prescott filter. It is both linear and translation invariant, therefore it may be better suited to our purposes. For more information about the Hodrick-Prescott filter we refer to (Schleicher, 2003, §2.5.2).

3 Coupling models with different frequencies

In Proposition 2.1 we saw that the filtered regression model produces sequences lying in the image of the filter, provided that the filter is linear and weakly translation invariant in all components *and* commutes with the regression matrix A. Another way

to describe the final condition is: different components that interact with each other need to be filtered in the same way. In practice however, there are situations where components with different time evolution still influence each other in some way. We now give one strategy to incorporate such interactions whilst preserving the conclusion of Proposition 2.1.

We can generalize the regression model (2) a bit if we allow A and the distribution of the ϵ_t's to depend on time as well. For instance, one can consider the AR(1)-model with constant $a = a(t)$ and standard deviations $\sigma = \sigma(t)$ that depend on time. In this way one can incorporate interactions between different models by letting them act via the parameters. Suppose that ϵ_t is a sequence of vector-valued random variables with zero mean and standard deviations σ_t and that A_t is a sequence of matrices with operator norms $||A_t|| < 1$ where $t \in \{0, \ldots, N-1\}$. If $F = (F^1, \ldots, F^d)$ is a filter, we can define, as before, the filtered regression by starting with an initial value r_0 and applying the recursive formula

$$r_t = A_t r_{t-1} + \epsilon_t^* \tag{5}$$

with $\epsilon^* = F\epsilon$. As before, we have

Proposition 3.1. *If F is linear and weakly translation invariant and commutes with A_t for all t, then the solution of the filtered regression (5) lies in the image of F, modulo an initial value term that converges to zero.*

Proof. In this case, the solution of (5) is given by

$$r_t = \sum_{l=0}^{t-1} A_t A_{t-1} \cdots A_{t-l+1} \epsilon_{t-l}^* + A_t \cdots A_1 r_0.$$

From here on the proof is identical to the proof of Proposition 2.1. □

4 A minimal variance approach to band decomposition

In this section we numerically investigate the (optimal) decomposition into frequency bands. For this, we focus on the (univariate) time series of the monthly interest rate of US bonds with a term of 10 years. We restrict our attention to the AR(1) model and a Fourier filter (Section 2.1).
As explained in the introduction, the current decomposition used by Ortec Finance consists of three bands – trend, business cycle and month. For the Fourier filter, the table below explicilty shows which of the frequencies are part of which band.

	period	frequencies
Month	2 months-2 years	$K_M = [N/24, N - N/24]$
Business	2 years-16 years	$K_B = [N/192, N/24) \cup (N - N/24, N - N/192]$
Trend	longer than 16 years	$K_T = [1, N/192) \cup (N - N/192, N - 1]$

Given the equidistant frequency distribution of the Fourier filter, about 92% of the frequency components is in the month component, 7% is in the business cycle component and only 1% is in the trend component. We note that $0 \in K$ corresponds to the mean interest rate which we have (without loss of generality) disregarded. We also note that Ortec Finance currently uses a nonlinear filter which may lead to a different distribution of frequencies.

How does the decomposition compare to other possible partitions in the simple univariate setting? We will compare the different decompositions based on the linear Fourier filter by comparing the total variances from the AR(1) model. As mentioned before, the total variance is a certain measure of fit of the model to the time series, so our idea for the choice of decomposition is to select the one that has the minimal total variance.

Since the interest rate time series is real and we want the filtered time series to be real as well, we impose that $j \in K$ implies $N - j \in K$. Furthermore, to make the computation feasible, we make the reasonable assumption that each of the three parts of the partition is 'connected' in the sense that there exist integers $2 \leq a \leq b \leq \lfloor \frac{N}{2} \rfloor$ such that $K_T = \{1, \ldots, a-1\} \cup \{N-a+1, \ldots, N-1\}, K_B = \{a, \ldots, b-1\} \cup \{N-b+1, \ldots, N-a\}$ and $K_M = \{b, \ldots, N-b\}$.

Write $\mathbf{r} = (F_{K_M}(\mathbf{r}), F_{K_B}(\mathbf{r}), F_{K_T}(\mathbf{r})) = (\mathbf{r}^M, \mathbf{r}^B, \mathbf{r}^T)$ for the decomposition of the interest rate in a month, business and trend component. We initialize the AR(1)-model for each frequency band by using the ordinary least squares method.

To have the best fit with the historical data, we should find (a^M, a^B, a^T) such that the total variance

$$\mathrm{Var}_{\mathrm{tot}} := \frac{1}{N-1} \sum_{t=1}^{N-1} \left|\left| r_t^M + r_t^B + r_t^T - a^M r_{t-1}^M - a^B r_{t-1}^B - a^T r_{t-1}^T \right|\right|^2 \tag{6}$$

is minimal. Instead, in the filtered AR(1) framework based on the least squares method, the parameters a^M, a^B, a^T are chosen separately to minimize the separate variances:

$$\begin{aligned} \mathrm{Var}_{\mathrm{M}} &:= \frac{1}{N-1} \sum_{t=1}^{N-1} \left|\left| r_t^T - a^T r_{t-1}^T \right|\right|^2, \\ \mathrm{Var}_{\mathrm{B}} &:= \frac{1}{N-1} \sum_{t=1}^{N-1} \left|\left| r_t^B - a^B r_{t-1}^B \right|\right|^2, \\ \mathrm{Var}_{\mathrm{T}} &:= \frac{1}{N-1} \sum_{t=1}^{N-1} \left|\left| r_t^M - a^M r_{t-1}^M \right|\right|^2. \end{aligned} \tag{7}$$

The separate minimizers a^M, a^B and a^T of (7) together yield an (almost) minimal value of (6). To see this, we show that $\mathrm{Var}_{\mathrm{M}} + \mathrm{Var}_{\mathrm{B}} + \mathrm{Var}_{\mathrm{T}} \approx \mathrm{Var}_{\mathrm{tot}}$. We make use

of the inner product $\langle \cdot, \cdot \rangle$ on V and orthogonality of the Fourier basis. It holds that

$$\begin{aligned}
(N-1)\mathrm{Var}_{\mathrm{tot}} &= \sum_{t=0}^{N-2} \left|\left| r_t^M + r_t^B + r_t^T - a^M r_{t-1} - a^B r_{t-1} - a^T r_{t-1} \right|\right|^2 \\
&= \langle \mathbf{r}^M + \mathbf{r}^B + \mathbf{r}^T - a^M L\mathbf{r} - a^B L\mathbf{r} - a^T L\mathbf{r}, \mathbf{r}^M + \mathbf{r}^B + \mathbf{r}^T - a^M L\mathbf{r} - a^B L\mathbf{r} - a^T L\mathbf{r} \rangle \\
&\quad - \left|\left| r_{N-1}^M + r_{N-1}^B + r_{N-1}^T - a^M r_0^M - a^B r_0^B - a^T r_0^T \right|\right|^2 \\
&= \langle F_{K_M}\mathbf{r} - a^M F_{K_M} L\mathbf{r}, F_{K_M}\mathbf{r} - a^M F_{K_M} L\mathbf{r} \rangle + \langle F_{K_B}\mathbf{r} - a^B F_{K_B} L\mathbf{r}, F_{K_B}\mathbf{r} - a^B F_{K_B} L\mathbf{r} \rangle \\
&\quad + \langle F_{K_T}\mathbf{r} - a^T F_{K_T} L\mathbf{r}, F_{K_T}\mathbf{r} - a^T F_{K_T} L\mathbf{r} \rangle \\
&\quad - \left|\left| r_{N-1}^M + r_{N-1}^B + r_{N-1}^T - a^M r_0^M - a^B r_0^B - a^T r_0^T \right|\right|^2 \\
&= (N-1)\mathrm{Var}_M + (N-1)\mathrm{Var}_B + (N-1)\mathrm{Var}_T \\
&\quad - \left|\left| r_{N-1}^M + r_{N-1}^B + r_{N-1}^T - a^M r_0^M - a^B r_0^B - a^T r_0^T \right|\right|^2 \\
&\quad + \left|\left| r_{N-1}^M - a^M r_0^M \right|\right|^2 + \left|\left| r_{N-1}^B - a^B r_0^B \right|\right|^2 + \left|\left| r_{N-1}^T - a^T r_0^T \right|\right|^2 .
\end{aligned}$$

After dividing the equality by $N-1$, we see that the error that is made by minimizing the separate variances (7) instead of (6), represented by the terms on the last two lines of the equation, is small if the amount of data N is large.

Since it is computationally much more efficient to optimize three times over a one dimensional set than once over a three dimensional data set, our script minimizes the separate variances (7).

We performed numerical tests on the monthly time series of interest rates from the past 116 years, consisting of 1392 data points (Figure 1). We computed the resulting total variance of all possible frequency decompositions (Figure 2). We found that (in this case) $\mathrm{Var}_{\mathrm{tot}}$ is minimal for a decomposition given by 624 frequencies in the month component, 410 in the business cycle component and 358 in the trend component. This results in the decomposition of the interest rate as shown in Figure 3. The corresponding frequency and period decomposition is shown in the table below.

	frequencies	period
Month	$K_M = [384, 1008]$	2 months-3.6 months
Business	$K_B = [179, 384) \cup (1008, 1087]$	3.6 months-7.8 months
Trend	$K_T = [1, 179) \cup (1087, 1391]$	longer than 7.8 months

so that about 45% of the frequencies are in the month component, 29% in the business component and 26% in the trend component.

5 Concluding remarks

We have proposed two ways to possibly improve the filtered regression models used by Ortec Finance. The first one is a method for generating predictions, which ensures that predictions via regression stay in the same frequency band as the one corresponding to the filtered historical time series. To put it shortly, a sequence of samples from

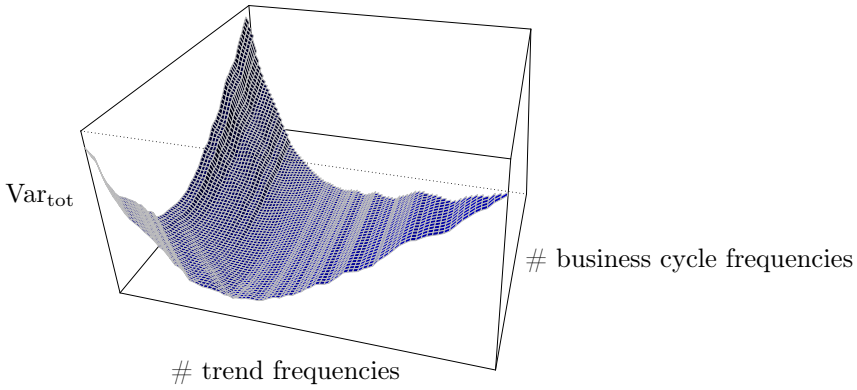

Figure 2: The total variance for all frequency decompositions attains its minimum $(N-1) \times \mathrm{Var}_{\mathrm{tot}} \approx 0.004640976$ for 358 frequencies in the trend component and 410 frequencies in the business cycle component. The total number of frequencies is $N = 1392$, so the remaining ones (624) belong to the month component.

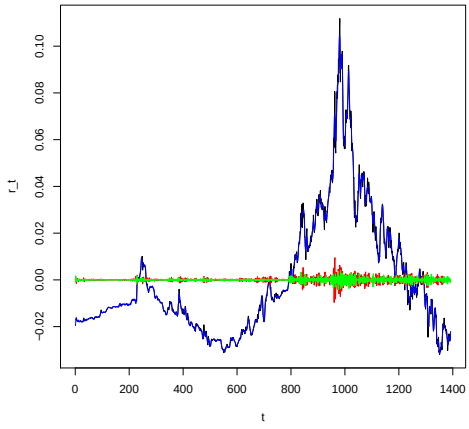

Figure 3: The optimal (minimizing the total variance) decomposition of the interest rate. The month component in green, the business cycle component in red and the trend component in blue.

white noise needs to be generated *a priori* for the whole prediction period, and then filtered, as opposed to sampling the white noise at each time step of the prediction. We identify a group of filters for which the method is applicable – the class of linear, weakly translation invariant filters that commute with the parameter matrix. In particular, the method can be readily applied for scalar, Fourier filtered AR(1) models, and can incorporate time-dependent parameters. However, currently Ortec Finance is using nonlinear filters to address the frequency leaking problem, and further investigation has to be performed to find a weakly translation invariant, linear filter that prevents frequency leaking. In particular, even though the Christiano-Fitzgerald band pass filter is linear and prevents frequency leaking, it is not applicable as it does not possess good translation invariance properties.

Our second contribution is the idea that by optimizing the number of frequencies in each band, one can further reduce the total variance of the model with respect to the given time series. This way, the frequency decomposition can be adapted to the time series, rather than arbitrarily fixed beforehand. Our numerical calculations based on the data set of demeaned interest rates of US bonds and a Fourier filtered AR(1) model indeed shows that there seems to be a clear global minimum for the total variance; see Figure 2. The method is, in principle, independent of the filter and applicable to any filtered autoregression model. We note that in the application to this particular dataset, only 45% of the frequencies entered the month component, as opposed to 92% in the decomposition used by Ortec Finance and consequently the size of the business cycle component, and particularly of the trend component was

much bigger. Perhaps increasing the number of frequency bands (e.g. to four or five) would make a clear narrow trend similar to the one from Ortec's decomposition reveal itself, and what has been captured as trend in the three frequency band setting is in fact a new, intermediate pattern.

References

L. Christiano and T. Fitzgerald. The band pass filter. *International Economic Review*, 44:435–465, 2003.

C. Schleicher. Essays on the decomposition of economic variables. *PhD Thesis, University of British Columbia*, 2003.

M. Van der Schans and H. Steenhouwer. Imposing views on frequency domain factor models, methodological working paper no. 2012-01. *Ortec Finance Research Center*, 2012.

Acknowledgments

The generous financial support from NWO and STW together with the contributions by the problem owners (Dümmen Orange, HZPC, KNMI, Marel Stork, NS, Ortec Finance) made SWI 2016 possible. The success of the meeting stemmed from both the active involvement of mathematicians as well as the close collaboration of industrial partners. A special word of appreciation to Greta Oliemeulen for her excellent and cheerful organisational efforts. Thank you all.

The organizers of SWI 2016

.


NWO
Nederlandse Organisatie voor Wetenschappelijk Onderzoek


STW
Enabling new technology


DÜMMEN
ORANGE
for you


HZPC
Growing with our potatoes


Royal Netherlands
Meteorological Institute
Ministry of Infrastructure
and the Environment


marel


ORTEC
FINANCE