Grups d'Estudi de Matemàtica i Tecnologia

Barcelona, July 2007

Edited by

José Antonio Carrillo (ICREA-UAB)
Enric Fossas (UPC)
Joan Solà-Morales (UPC)

FACULTAT de MATEMÀTIQUES i ESTADÍSTICA

i-math ingenio mathematica consolider ingenio2010

CENTRE DE RECERCA MATEMÀTICA

Centre de Recerca Matemàtica
Apartat 50
08193 Bellaterra, Spain

First edition: June 2008

ISBN: 978-84-612-4771-4
Legal deposit:

Presentation

This booklet collects the problems that were proposed at the *Grups d'Estudi de Matemàtica i Tecnologia* (Study Groups of Mathematics and Technology), GEMT 2007. It also contains the solutions that were given. The problems belong to different areas of Mathematics and were proposed by companies and institutions through different ways. We believe that the results are very satisfactory, and ways were found in all the problems to achieve the type of solution that could be expected from an activity like this one.

The event took place in the premises of the Facultat de Matemàtiques i Estadística of the Universitat Politècnica de Catalunya, Barcelona, from July 9 to 11, 2007. The number of attendants —around thirty researchers— is enough to insure the continuity of this activity for the next years. Also the dimension of the event —three days and three problems— proved to be satisfactory. It is of course clear that with more time available for work and discussions the results could have been more complete. However, we have to keep in mind that the goal of these Study Groups is just a first contact with the problems, that can continue after the event in different ways.

Participation was free of any cost, both for companies and institutions and for individual participants. We are especially grateful to the two institutional co-organizers —the Facultat de Matemàtiques i Estadística of the Universitat Politècnica de Catalunya (FME) and the Centre de Recerca Matemàtica (CRM). They both contributed by strongly supporting these Study Groups. We also thank Ingenio Mathematica (i-MATH) for financial support. Additionally, we have worked in coordination with the *Jornadas de Consulta Matemática para Empresas e Instituciones*, that are organized by CESGA in Santiago de Compostela, an event with the same aims as ours.

Finally, we thank very much all the participants: the companies and institutions that presented the problems and also the researchers that contributed to the discussions and wrote the final reports. We also take this opportunity to invite all of them to join us in future editions.

Barcelona, December 2007

José Antonio Carrillo, Enric Fossas and Joan Solà-Morales

Statement of the Problems

Mathematical modelling of rowing in a Catalan *llagut*

by Àlex de Mingo, President of the rowing team Santa Cristina and member of *Federació Catalana de Rem*

1.1 Introduction

The Catalan *llagut* is a traditional boat that was formerly used for fishing or to transport merchandises. It used to have a Latin sail, but when the weather conditions were unfavourable the fishers would remove the sail and use the oars instead. The aim to be the first to reach the harbour soon turned the way back home in a real competition. Later on it became a tradition to organise rowing competitions on bank holiday days, or on village festival dates.

Later on, engines changed fishers' lives and left the oars to competitions, thus becoming a sport rather than a way of living. Nowadays the *llagut* is the kind of sports boat that is used in the Catalan league (see Figure 1.1).

Figure 1.1: Catalan *llaguts*

1.2 The competition

The Catalan Rowing Federation currently has a section devoted to *llagut* competitions. The boat competitions are races with four boats that must cover a distance of 250 m four times, that is, they must turn three times. The conditions of the location where each race takes place determines the direction in which the regatta field is placed, but it is very common to place the regatta field parallel to the coast, so that the public can watch the race more confortably. As a consequence, it is very common that the depth of the water under each boat is different, suffering differently the influence of waves and wind. Nevertheless, it is still not clear whether a position is more favourable than another, and if rowing closer to the coast is really harder than rowing farther inside the sea.

1.3 Structure of the Catalan *llagut*

The main characteristic that makes *llagut* rowing so different from classical Olympic rowing is that the rowers sit on a fixed seat rather than on a moving seat (see Figure 1.2). One of the reasons may be that boats with a movable

Figure 1.2: Boat with a movable seat

seat seem not to be the best option for competitions in the open sea. Therefore, the movement of the rowers is substantially different to that made by the Olympic rowers. Figure 1.3 shows a sketch of the rowers' movement in both cases. In the case of a movable seat, each rower achieves the maximum strength by propulsing himself or herself through his or her legs. On the

contrary, when the seat is fixed, the rower's legs are fixed and he or she must bend the torso down in order to maintain the row inside the water as much as possible. Each boat fits eight people that sit in couples in four rows plus

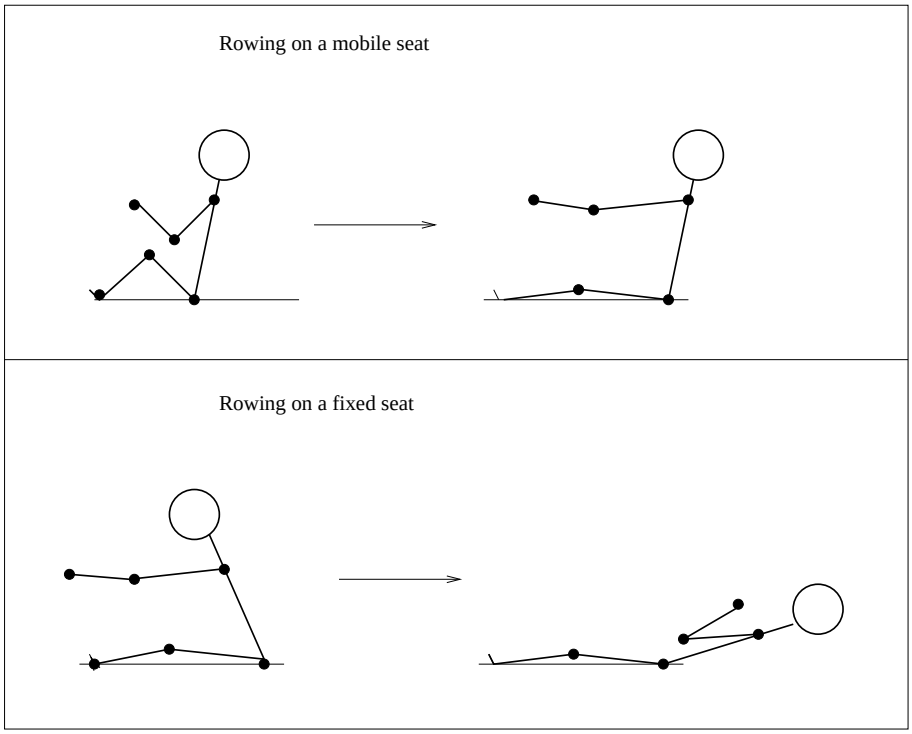


Figure 1.3: Rower's movement on a movable seat and on a fixed seat

the cox. Therefore, rowers sit in couples, side by side, so that one rows on the port side and the other on starboard, and they sit facing backwards. The synchronisation of the rowers is a key point in order to achieve the best performance, although the sea conditions makes it sometimes very difficult. The rowers that seat on the back of the boat, that is, those who are in the first seat, should be those with a better rowing technique because the rest of the rowers must row at the pace that these two set, following their movements as well as possible.

Nowadays the measurements of a *llagut* are those shown in Figure 1.4. The minimum allowed weight is 300 kg. Regarding the oars, there are no restrictions and each team can choose the type of oars that they want to use. Nonetheless, they are usually 3.40 m long and the blade is about 19 cm wide and 57 cm long. The materials used to build the boats and the oars are usually either wood or carbonic fiber, which is lighter but also more expensive.

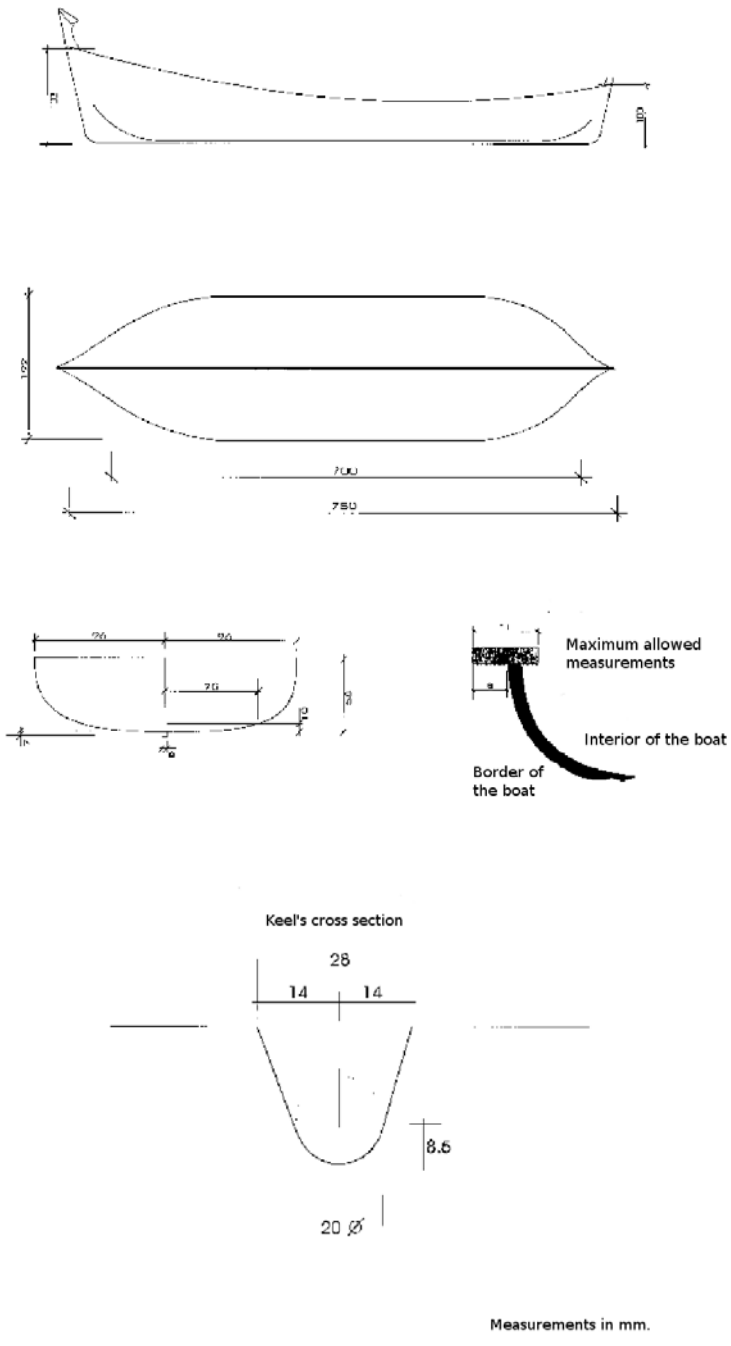


Figure 1.4: Measurements of a *llagut*

1.4 Mathematical modelling and open questions

Since rowing in a fixed sit is not so popular as it is on a mobile sit and it is not an Olympic sport, to our knowledge nowadays it does not exist any mathematical model to analyse any aspect of this sport. Nevertheless, in the Basque Country there is a type of boat called *traineras* that is used in competitions and that is similar to the *llagut*, but instead of eight rowers there are thirteen. This sport is very well known in this area, and several empirical studies have been made based on measurements of the rowers' performance, velocity of the boat, etc. Several questions remain still open. Some of them are the following:

- Throughout the years, the rowing technique has changed quite a lot. At the beginning, most of the strength that the rowers did was made using only their arms. Nowadays it seems to be more correct to bend the torso and use also the abdominal muscles to push harder the oar and to maintain the blade longer time inside the water. It would be very interesting to understand what is the blade efficiency in each part of the length that the blade is inside the water.

- Therefore, another interesting question is whether there is an optimal shape for the blade to achieve the best rowing efficiency. Nevertheless one must take into account that oar blades in the shape of a spoon are not allowed, so the only possibility is to change its length and width.

- In each rowing period, when the oar leaves the water, rowers tend to restart the rowing period quite abruptly in order to put the oar back into the water as soon as possible. But this sudden movement is done against the direction of the velocity of the boat and it seems that a smoother movement would not create such a big inertia. However, if the movement is done too slowly, the velocity seems to decrease. A mathematical model of the effect may give an answer to this question.

- Another issue lies in the distribution of the rowers taking into account their weight and their strength. The big problem here is that usually those who are stronger are also heavier, so finding the best distribution is not an easy problem.

- In each competition, the boat must turn 180 degrees three times. This is not an easy manoeuvre and should be done as fast as possible. To turn the boat, each rower must do a different movement, so the synchrony is lost for a while. A good design of this manoeuvre would also help to achieve higher velocities.

1.5 Some links and references

Some mathematical models on rowing on a mobile sit are:

1. N. Caplan and T. Gardner, A mathematical model of the oar blade-water interaction in rowing. *J. Sports Sci.* **25 (9)** (2007), 1025–1034.

2. N. Caplan and T. Gardner, A fluid dynamic investigation of the Big Blade and Macon oar blade designs in rowing propulsion. *J. Sports Sci.* **25 (6)** (2007), 643–650.

3. D. Cabrera, A. Ruina and V. Kleshnev, A simple 1^+ dimensional model of rowing mimics observed forces and motions. *Hum. Mov. Sci.* **25** (2006), 192–220.

Regarding the fixed sit rowing, there are studies made by José Manuel Francisco, trainer of the rowing team *Astilleros*, in the Basque Country, who probably is the only person who has tried to understand from a medical or scientific point of view different aspects of this sport.

Forecasting wind energy production

by Carles Comas, Ambio S.A.

2.1 Introduction

The use of wind energy has increased in recent years to the extent that it now contributes an average of 10% of the energy that is consumed in Spain, which has become the second biggest market in the world in terms of installed power. Wind energy has to be consumed as it is produced, because the kinetic energy provided by wind cannot be stored like natural gas in tanks or like water in a reservoir and consumed in accordance with demand. Given its intermittent, non-programmed and variable nature (contributing between 3% to 23%) the integration of wind energy into the electricity system involves the challenge of managing the reprogramming or connection/disconnection of the network of power stations depending on the demand, the capacity of the network and wind production. The company in charge of implanting and maintaining the high tension infrastructure required to transport the energy generated at production centres to distribution substations is the *Red Eléctrica Española* (REE). An accurate forecast of wind generation clearly helps both producers and distributors of wind energy to manage affairs in an orderly and programmed fashion. Moreover, following the deregulation of the electricity market, producers and distributors now negotiate electricity prices. The organisation in charge of managing this sales-purchase market is the *Operador del Mercado Ibérico de Energía* (OMEL). Electricity prices are

negotiated on the basis of the forecast of the demand and production. There is a daily market where most operations are made and six inter-daily markets where producers and distributors can adjust the daily market forecasts made the day before. The daily market is closed at 10.00 in the morning to fix the amount and price of electricity for the following day, i.e., 14 hours before the start of the day. Production centres must provide an hour by hour forecast of production for the following day. The intra-daily market works to correct any deviations from this forecast, and is opened for 6 sessions: 16.00, 21.00, 1.00, 4.00, 8.00 and 12.00. In this market, penalties are applied to producers that fail to comply with the forecast, either by falling short of it or exceeding it. Until the year 2004, Kwh of wind energy were paid for at a fixed rate that was established annually by the Spanish government in accordance with certain defined parameters. Since 2004, wind energy has had the option of participating in the electricity market to bring it into line with more 'conventional' production centres. This can mean greater profits for wind energy producers but also penalties should they fail to comply with production forecasts. A wind farm as a production centre, whether it sticks to the fixed rate or applies market negotiations must present production forecasts:

- A wind farm that opts for the fixed rate is obliged to inform the distributor of the production forecast for every hour of the day, at least 30 hours before the start of the day. This forecast can be corrected up to 1 hour before the closure of the 6 intra-daily sessions. The forecast tolerance before the application of penalties is $\pm 20\%$.

- A wind farm that negotiates market prices must provide a daily forecast and intra-daily forecasts in accordance with the protocol of the electricity market. The penalties for incompliance with production are established in the sales-purchase contracts negotiated with market agents.

2.2 Models for forecasting production

The forecast of wind production depends at least on the wind forecast. A wind forecast is an important element of a weather forecast and is made on the basis of physical models, which are known as numerical forecast models. When forecasting 24–48 hours in advance, they are the best possible option. Physical models forecast wind speed and direction, and also air temperature, within a network of 10×10 Km points, which is fine for making general forecasts. There is therefore a clear interest in having tools for forecasting

wind production, not only in the medium term (24–48 hours) but also in the short term (0–6 hours), that can enable producers to operate more reliably in intra-daily electricity market sessions.

2.3 Data for consideration

The first information to take into consideration is the weather forecast. For example, the forecasts made by the *Servei Meteorològic de Catalunya* supply a forecast 48 hours in advance with a temporal resolution of 3 hours for the nodes in a typical 8 Km network. By interpolation, a forecast is obtained for a point inside the network, which coincides with the location of the wind farm's weather station. The forecast can provide the wind speed, wind direction, temperature and the pressure at the height of the wind turbine hub, at the coordinates of the weather station. The wind farm's weather station measures at least the speed and direction of the wind in averages for 10 minute periods at the height of the wind turbine hub. This means that data is available for time periods up to the last ten minutes. Finally, data can also be obtained for the number of wind turbines in operation and the production series for the wind farm as a whole in averages for 15 and 60 minute periods, as well as historical records of the farm's power curve over time, which is important because the conditions in January are not the same as they are in August.

2.4 Objective

To provide an hour-by-hour forecast for a wind farm's production 24 hours or more in advance along with the associated uncertainty.

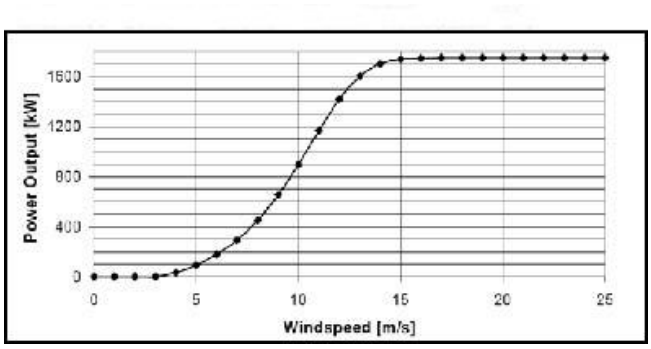


Figure 2.1: Ideal power curve of a typical wind turbine

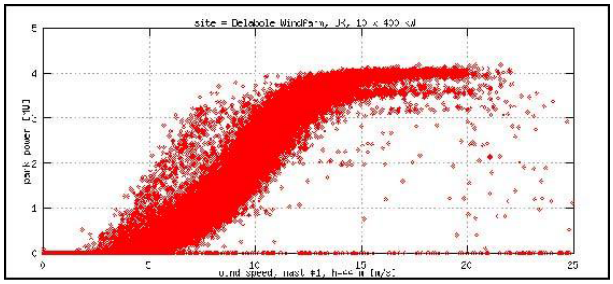


Figure 2.2: Wind farm power curve: contribution of the individual curves of each wind turbine in a multitude of situations

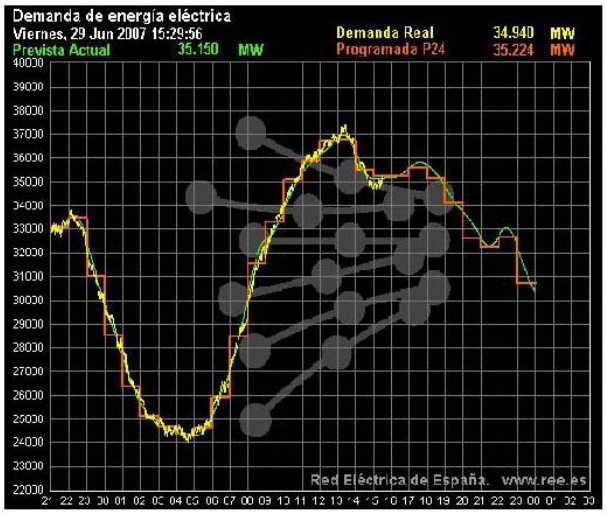


Figure 2.3: Daily electric power curves: forecasted demand, program reduction and real demand

2.5 References and internet documents

1. M. Lange and U. Focken, *Physical Approach to Short-Term Wind Power Prediction*, Springer, 2005

2. www.canwea.ca/images/uploads/File/Resources/FinalGHReport.pdf

3. www.uwig.org/forecst_overview_report_dec_2003.pdf

4. journals.tubitak.gov.tr/engineering/issues/muh-06-30-1/muh-30-1-4-0504-3.pdf

5. ams.confex.com/ams/84Annual/techprogram/paper_68284.htm

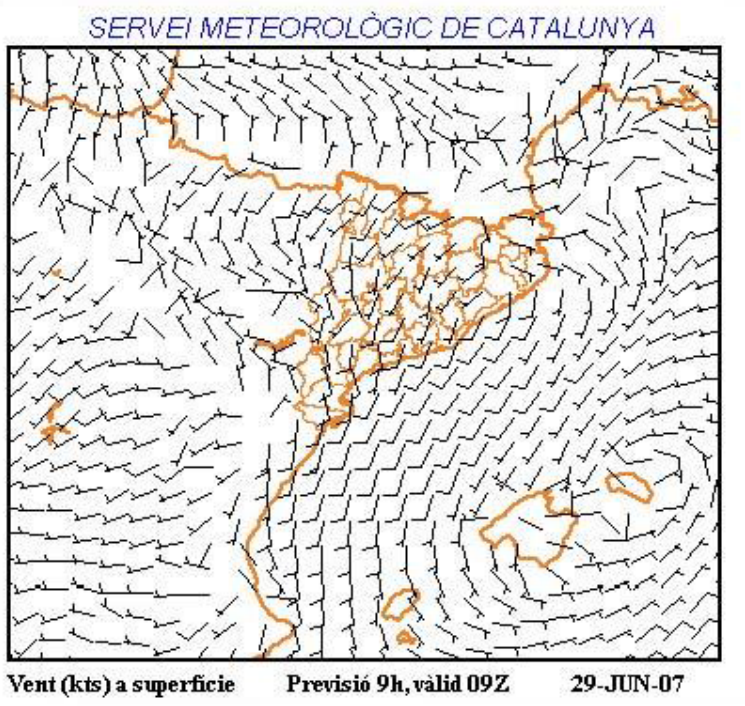


Figure 2.4: Wind map from the Catalan Weather Service

6. anemos.cma.fr/download/publications/
   pub_2003_paper_EWEC03_ComplexTerrain.pdf
7. www.ceem.unsw.edu.au/content/userDocs/
   IssuesPaper-WindEnergyForecasting-v12.pdf

2.6 Links

1. www.omel.es
2. www.ree.es
3. www.garradhassan.com/services/ghforecaster/
4. www.aeeolica.org/html/observatorio_eolico.html
5. www.meteologica.es/meteologica/eolicabil.htm

Diagnosis of multiple faults in the network of rain gauges in the Barcelona sewage system

by Vicenç Puig, Universitat Politècnica de Catalunya, Dept. ESAII

3.1 Problem statement

The Barcelona sewage system includes a network of 23 rain gauges that collect information about rain intensity in real time (every 5 minutes). Figure 3.1 shows the distribution of the rain gauges on a map of Barcelona.

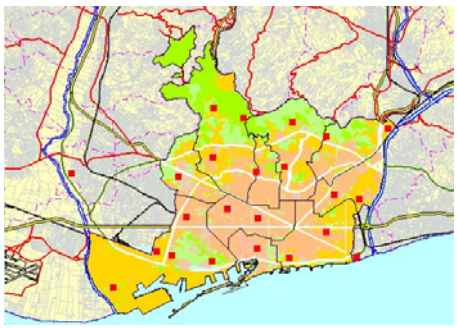


Figure 3.1: Distribution of rain gauges on a map of Barcelona

This information is sent to the control centre that the CLABSA company, which manages the Barcelona sewage system, operates in the Zona Franca district (Figure 3.2). Using the information on rain and the level of the collectors provided by the network of limnimetres (water level gauges), the control system installed at the control centre must decide how to move the sluice gates that are installed in the sewage system in order to prevent the streets from flooding or any overflow into the sea, while at the same time maximising the volume of water treated at the Besòs and Llobregat treatment plants.

Figure 3.2: Picture of the CLABSA Control Centre

The problem presented below is that of designing a diagnosis system based on models that can diagnose multiple sequential faults in rain gauges (Figure 3.3).

The model-based diagnosis involves constructing a model of the rain gauges by exploiting the fact that there is a major degree of correlation between the closest measurements.

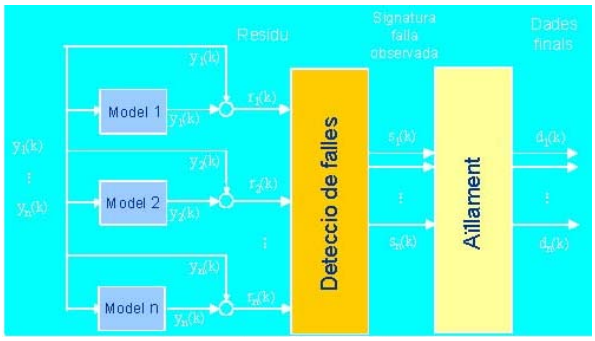


Figure 3.3: Model-based diagnosis

Once the two previous issues have been settled, we now have a structure of rain gauge models that will be used to detect and isolate faults. For example, if employing the criterion of making use of a minimum of 3 rain gauges in the model of each of the rain gauges and the most correlated are selected, we would obtain the incidence matrix shown in Table 3.1.

The rows show us which models intervene in each of the rain gauges while the columns show the rain gauges and which models they are served by. This table is the basis for diagnosing faults. When a set of models show that they are not consistent with the data measured by the rain gauges, the table can show us which rain gauge is failing. For example, if the models

Table 3.1: Incidence matrix

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
P1								x										x			x		
P2												x	x		x								
P3																x	x		x				
P4																			x	x	x		
P5													x	x						x			
P6											x			x		x							
P7			x								x					x							
P8	x														x			x					
P9																	x	x	x				
P10											x			x		x							
P11						x				x						x							
P12		x											x		x								
P13		x			x															x			
P14					x	x					x												
P15	x	x																			x		
P16			x			x					x												
P17			x						x										x				
P18	x								x												x		
P19			x	x													x						
P20				x	x								x										
P21				x											x			x					

of P8, P15 and P18 are inconsistent with the data given by rain gauges P8, P15 and P18, respectively, this implies that the rain gauge that is failing is P1, because this is the only rain gauge that intervenes in all these models. Evidently, so we can isolate the faults in all of the rain gauges, all of the columns in this matrix have to be different. The problem that now arises is that although at first we can see that the columns for the previous matrix are different after the failure of a rain gauge, there is a need to generate a new table by eliminating the file corresponding to the failing rain gauge and replacing it by its model into the other models where it was used.

This means that the crosses in the previous table corresponding to the model of the failing rain gauge are propagated in all the models that use it. Consequently, after several failures columns can start repeating and therefore we may longer be able to isolate faults. Thus, the following questions arise:

- What should the previous table ideally be like in order for the repetition of columns due to the sequential failure of rain gauges to take as long as possible to appear?

- What is the maximum number of sequential faults that can arise before columns start repeating?

A possible answer could be that the number of times a rain gauge is used in each model should be the same (i.e. there are no privileged rain gauges). This would lead to the following question:

- What would the spatial distribution be in order for, when we select the 3 most correlated examples, we end up with an incidence matrix that complies with the condition that the number of times a rain gauge is used by one of the models is the same?

Given that the initial table that we have of the current distribution of rain gauges in the Barcelona network does not comply with the condition that the number of times a rain gauge appears in each model is the same,

- Which rain gauges should we move (the minimum possible number in order to limit financial costs) in order to get the spatial distribution that would lead to the ideal table of incidences?

- Could we achieve the ideal distribution considering the fact that we have a fixed number of rain gauges?

In order to answer the previous questions, we have to be able to model how the correlation varies between the rain gauges and the distance. The following question is:

- How can we describe the variation in the correlation of the rain gauges with distance, in such a way that when they are moved one will know how this will vary the correlations with the remainder? How can this be done if we consider not only the distance but also displacements on the "x-y" plane?

In the literature on fault diagnosis one way of dealing with the case of sequential faults is to transform the incidence matrix by means of a transformation that makes the resulting matrix diagonal. The question is:

- Can this be applied in the case of the rain gauge incidence matrix?

- If we cannot, would it be possible to transform the matrix into a semi-diagonal matrix?

Having overcome all these problems, we can now turn to dealing with the problem of the limnimeter network in the Barcelona sewer system. Limnimetres are sensors that measure the level of the collectors in the sewage system. Like rain gauges, they are connected to the control centre and are used by the control system to decide how to optimally manage the sluice gates that decide where water should be sent. In this case, the diagnosis models are obtained on the basis of the models of the system. The following is a presentation of the matrix of incidences for the case of the limnimetres in the Barcelona network:

Table 3.2:

	f(L03)	f(L07)	f(L08)	f(L09)	f(L11)	f(L16)	f(L19)	f(L27)	f(L39)	f(L41)	f(L47)	f(L56)
r(L03)	1	0	0	0	0	0	0	1	0	0	0	0
r(L07)	0	1	0	0	0	0	0	0	0	0	0	0
r(L08)	0	0	1	0	0	1	0	0	0	0	0	1
r(L09)	0	0	0	1	0	0	0	0	0	0	0	0
r(L11)	0	0	1	1	1	0	0	0	0	0	0	0
r(L16)	0	0	0	0	0	1	0	0	0	0	1	0
r(L19)	0	0	0	0	1	0	1	0	0	0	0	0
r(L27)	0	0	0	0	0	0	0	1	0	0	0	0
r(L39)	0	0	0	0	0	0	0	0	1	0	0	0
r(L41)	0	0	0	0	0	0	0	0	1	1	0	0
r(L47)	0	0	0	0	0	0	0	0	0	1	1	0
r(L56)	0	0	0	0	0	0	0	0	0	0	0	1

The difference here is that these models are dynamic models (described by difference equations). This implies that when a limnimeter fails and we want to substitute it with its model, an observer has to be constructed. But to do this, the resulting system after the loss of a limnimeter has to be observable. Thus the following questions arise:

- Which limnimetres are critical in that if we lose them due to a fault we lose observability?
- Is there any way that we can avoid there being critical limnimetres, by repositioning, or maybe by adding new ones?
- If we could decide how to position the limnimetres from the outset, what would be the ideal location in order for the problem of column repetition to take as long as possible to arise?

That solved, the problems with limnimetres could also be dealt with using sluice gates. The problem now, apart from whether we would be able to isolate sluice gate faults sequentially, is as follows:

- How many sequential sluice gate faults could we have before losing control of the system?
- Where should we place the sluice gates in order to delay the problem of losing control of the system as much as possible?
- What degradation would we experience in the control of the system after the failure of each sluice gate?
- From what moment could we consider that the degradation is so great that control is now unacceptable?
- Where should we position the sluice gates in order to delay the degradation of the control of the system?

Answers to the Problems

Mathematical modelling of rowing in a Catalan *llagut*

by M. Aguareles, X. Mora, M. Noguera, and M. Pellicer

4.1 Introduction

The Catalan *llagut* is a traditional boat that nowadays is used in Catalonia in local rowing competitions. It is similar to the boat used in classical Olympic rowing, but with a main and important characteristic that makes both ways of rowing substantially different: the seats are fixed and cannot move, while in an Olympic rowing boat the seats move. This is why the movement of the boat is achieved by bending the torso rather than bending the legs as it is the case in Olympic rowing.

The rowing in a *llagut* is not a very well understood mechanism. Several different problems were proposed in the GEMT, but we concentrated on two main aspects (developed in the sections below): the stroke efficiency and the effect of the rower movement on the boat in terms of inertia. As there are no models of *llagut* rowing, our approach is based on the work of [1], where the modelling of a sliding seat boat is developed. Despite of the differences between the two types of rowing that we have pointed out before, some parts of the work of [1] can be adapted and used in our case. In some other works, like for instance [2] or [3], the very important issue of the design and performance of the oar's blade is considered. We will not focus on this issue —on the contrary, in this work we are concerned with a more general approach to give some insight into the rower's performance in fixed-seat rowing competition.

The report is structured as follows. In Section 4.2 we analyse an aspect of the stroke efficiency which is the angle of catch in which the oar is inserted into the water. That is, without taking into account limitations coming from the rower, we wonder which is the best angle to insert/take out the oar into/from the water in order to minimise energy losses. In Section 4.3 we face a second aspect on the efficiency of the stroke, namely the relation between

the force made onto the oar and the resulting velocity of the boat. Again, we do not take into account restrictions due to the rower. In Section 4.4 we change our point of view by considering only the rower movement and not the oar, and we study its effect onto the velocity of the *llagut* by analysing the forces that are created. This is important in order to understand the recovery part of the rowing (the part in which the oar is taken out from the water and placed back in its catch position). We try to verify that a smoother recovery is better than a quicker but sudden one.

This report is based on contributions from the discussions held at the Grups d'Estudi, which included the following members: Maria Aguareles, Xavier Alameda, Enric Fossas, Antoni Guillamon, Albert Mas, Xavier Mora, Miquel Noguera, Marta Pellicer, Peregrina Quintela, and Noemí Ruiz.

4.2 Aquatic phase: angles of catch and finish

It is important for the rower to insert the oar at the right time so that no missed or splashed water occurs. As it is pointed out in [1], the way to achieve such an optimal stroke is by inserting or extracting the blade of the oar into or from the water in the precise moment in which the blade's absolute velocity has the same direction as the oar (see Figure 4.1). In this way the insertion and extraction of the blade into and from the water, which is a movement in the vertical plane perpendicular to the water, is done so that the relative velocity of the blade with respect to the water is purely in this direction. This means that the insertion and extraction becomes as smooth as possible and, as a consequence, the amount of energy that is lost in this process is minimal. Of course, this condition is independent of the way the rowing is done, that is to say, it does not depend on whether the seat moves or not. What will make the difference is the way the rower moves in order to achieve the corresponding angles of catch and finish, which may induce a different inertia moment on the boat and might therefore affect the resulting boat velocity. The effect of the inertia of the rower into the boat is discussed in Section 4.4.

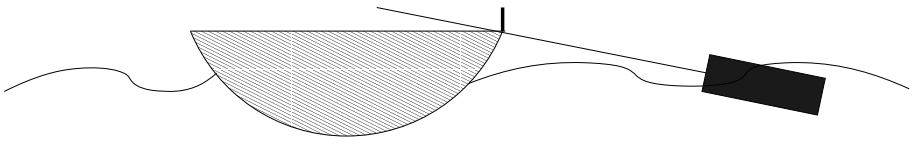


Figure 4.1: Cross-section of a *llagut* with one oar

The absolute velocity $\mathbf{v}_o$ of the oar depends on the velocity $\mathbf{v}_b$ of the boat and also on the relative velocity $\mathbf{v}_{o/b}$ of the oar with respect to the boat through the relation

$$\mathbf{v}_o = \mathbf{v}_b + \mathbf{v}_{o/b}. \tag{4.2.1}$$

In a reference system fixed on the boat, the movement of the blade is purely angular, since its length is fixed and we assume, as a first approximation, that the oar is rigid and cannot bend. Therefore, the velocity $\mathbf{v}_{o/b}$ of the oar (or the blade) with respect to the boat is perpendicular to the oar itself. Hence, the non-splashing condition does actually say that the absolute velocity $\mathbf{v}_o$ of the oar must be perpendicular to the relative velocity $\mathbf{v}_{o/b}$ of the oar with respect to the boat. Then, writing all the velocities in terms of the angle of the oar with respect to the boat (see Figure 4.2), one finds that

$$\mathbf{v}_b = \pm v_b \cos\phi\, \mathbf{e}_\phi + v_b \sin\phi\, \mathbf{e}_r \tag{4.2.2}$$

$$\mathbf{v}_{o/b} = v_{o/b}\, \mathbf{e}_\phi \tag{4.2.3}$$

where $\mathbf{e}_r$ and $\mathbf{e}_\phi$ stand for the unitary vectors in the direction of the oar and in the direction perpendicular to the oar respectively. Upon using (4.2.1) the absolute velocity of the oar reads

$$\mathbf{v}_o = (\pm v_b \cos\phi + v_{o/b})\, \mathbf{e}_\phi + v_b \sin\phi\, \mathbf{e}_r,$$

so the non-splashing/non-missing condition becomes

$$\pm v_b \cos\phi + v_{o/b} = 0, \tag{4.2.4}$$

and thus, while the blade is inside the water $\mathbf{v}_o \cdot \mathbf{e}_\phi > 0$. Hence the optimal angles of the oar with respect to the boat to put it into the water and to take the blade out are

$$\cos\phi_{\max/\min} = \pm \frac{v_{o/b}}{v_b}. \tag{4.2.5}$$

From the experience of the rowers it is commonly accepted that the best way to row is by inserting the blade at an angle that is smaller than the angle of extraction. One of the questions that was asked to the GEMT was if this way of rowing had actually any theoretical base that supported it. The analysis of the angles shows that this way of rowing is indeed consistent with condition (4.2.5), due to the fact that, when the blade is about to be taken out, the rower is in a position to be able to perform the maximum strength and therefore the relative velocity of the oar with respect to the boat is also greater, which produces $\phi_{\max}$ to be greater than $\phi_{\min}$. As the boat accelerates and v_b becomes larger, the angle of catch can be smaller, while the finish angle at which the blade is taken out can be kept large due to the fact that the rower can also move the oar faster (the opposition of the water is weaker).

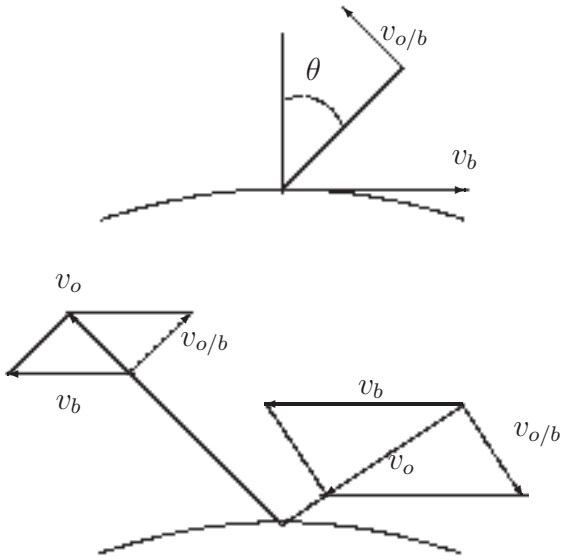


Figure 4.2: Velocities of the oar, the boat and the oar with respect to the boat by means of the angle of the oar with respect to the boat

4.3 Relation between oar force and velocity

In this section we discuss how the force that the oar applies into the water changes the boat's velocity. We again focus on the results in [1] and adapt conveniently the expressions to the fixed-seat rowing modality.

We now write a one-dimensional model for the movement of the boat, where we do not consider effects like the pitch of the boat as it moves, which in a more refined model should definitely be taken into account. This pitch effect is related to the analysis of the inertia driven by the rowers as they bend their torso up and down, which is analysed in Section 4.4, but it is also strongly dependent on sea waves. It is important to remark that this pitch effect, that in a sliding seat competition is not important, may become crucial when considering fixed-seat rowing in the sea.

We use the same notation as in [1]: F_{hand} stands for the force that the rower applies to the oar; $-F_{\text{foot}}$ is the propulsion force that the rower does with his or her feet and abdominal muscles in order to move his or her torso backwards (see Figure 4.3 for a sketch of the rower's movement); F_{lock} is the force that the oar applies on the boat through the oarlock; F_{boat} is the drag force of the boat in the water; $-F_{\text{oar}}$ is the force that the oar applies onto the water (and θ is the oar angle in a plane parallel to the water), which has the direction perpendicular to the blade. We also denote by x_R and m_R the position of the centre of gravity of the rower and his or her mass, while

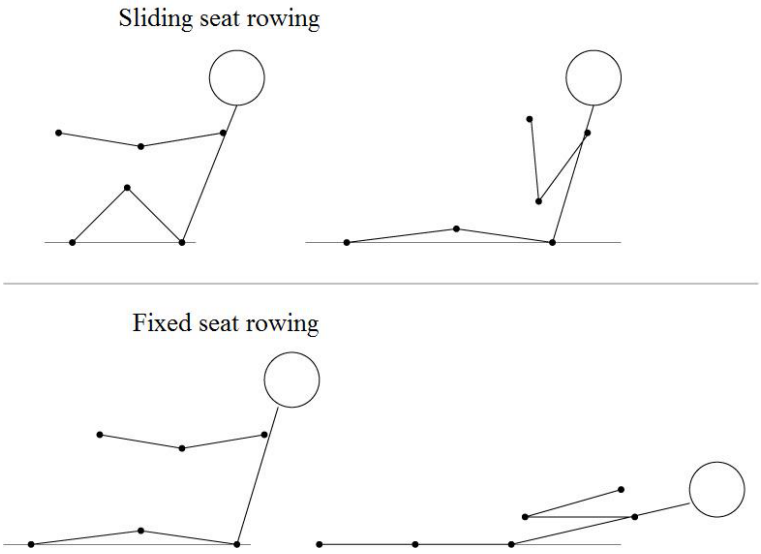


Figure 4.3: Fixed-seat and sliding-seat rowing: sketch of rower's movement

x_b and m_b are the position of the centre of gravity of the boat and its mass, and x_o and m_o are the position of the centre of gravity of the oar and its mass (see Figure 4.4). Having all these magnitudes defined, we follow [1] in writing the force balance equations for the rower, the oar and the boat:

$$-F_{\text{hand}} + F_{\text{foot}} = m_R \ddot{x}_R, \tag{4.3.1}$$

$$F_{\text{hand}} - F_{\text{lock}} + F_{\text{oar}} \cos\theta = m_o \ddot{x}_o, \tag{4.3.2}$$

$$-F_{\text{boat}} + F_{\text{lock}} - F_{\text{foot}} = m_b \ddot{x}_b. \tag{4.3.3}$$

The positions of the oar and the rower can be expressed in terms of x_b, the position of the centre of mass of the boat:

$$x_o = x_b + d_{L/F} + d\sin\theta,$$
$$x_R = x_b + x_{B/F} + r x_{S/B}$$

where $d_{L/F}$ is the horizontal distance between the oarlock and the rower's feet, and d is the distance between the oarlock to the oar's centre of mass. As for $x_{B/F}$ and $x_{S/B}$, they represent the distance between the seat and the feet (which is fixed in this case) and the distance between the seat and the shoulders (see Figure 4.4). Finally, r stands for the ratio of the rower's center of mass height h_R to the shoulder's height h_s, both measured with respect to the seat. These relations can be differentiated twice to find the corresponding

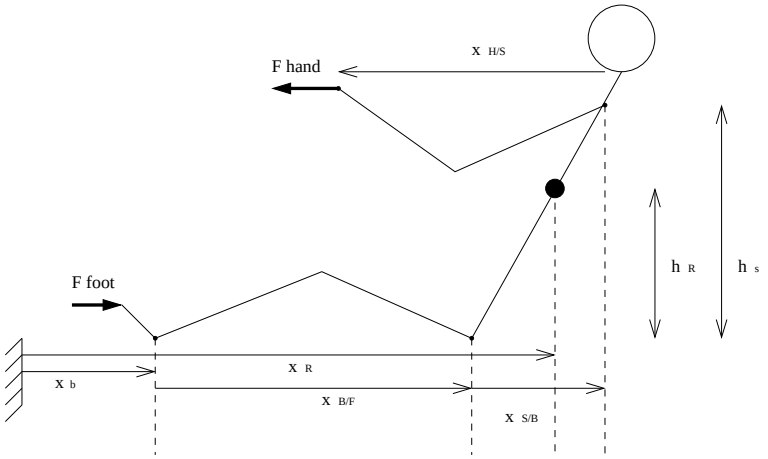


Figure 4.4: Representation of some magnitudes involved in the modelisation

relation between the accelerations in terms of $\ddot{x}_b$, obtaining

$$\ddot{x}_o = \ddot{x}_b + d\ddot{\theta}\cos\theta - d\dot{\theta}^2\sin\theta, \tag{4.3.4}$$

$$\ddot{x}_R = \ddot{x}_b + r\ddot{x}_{S/B}. \tag{4.3.5}$$

The drag force due to the water in this type of problems is usually considered to be

$$F_{\text{boat}} = C_1\dot{x}_b^2, \tag{4.3.6}$$

where C_1 is the boat drag coefficient, which must be determined from drag tests. Hence, combining equations from (4.3.1) to (4.3.6), the following second-order differential equation is obtained

$$(m_R + m_b + m_o)\ddot{x}_b = -C_1\dot{x}_b^2 + F_{\text{oar}}\cos\theta - m_R r\ddot{x}_{S/B} - m_o d(\ddot{\theta}\cos\theta - \dot{\theta}^2\sin\theta). \tag{4.3.7}$$

We can now simulate the rower's performance in one stroke using equation (4.3.7) and considering x_b as the unknown of the problem. As for the other magnitudes, they will be set conveniently according to a prescribed rower's performance along with some other assumptions that we now describe.

As for F_{oar}, we use as a very first approach the one given in [1],

$$F_{\text{oar}} = C_2\left(l\dot{\theta} + \dot{x}_b\cos\theta\right)^2, \tag{4.3.8}$$

where C_2 is the coefficient for the blade force and l stands for the length of the oar that is outside the boat. At this point we note that this might not be the best model for the oar's force when it is being used in the sea,

which might become rough or at least is always rougher and denser than the typical river or channel water where sliding-seat competitions take place. As a matter of fact and from our point of view, in order to obtain a better model for the force of the oar, an experiment should be designed to obtain data that give some clue on the type of interaction blade/water that is really taking place in the case of rowing in the sea.

On the other hand, the angle that the oar forms with respect to the boat is related to the rower's coordination by

$$d_{L/F} - s\sin\theta = x_{B/F} + x_{S/B} - x_{H/S}$$

where s stands for the length of the oar that is inside the boat, that is to say, the distance between the oarlock and the rower's hands (see [1] for details). As for $x_{H/S}$, it represents the distance between the hands and the shoulders (see Figure 4.4). Hence, differentiating the above expression twice with respect to time we obtain

$$s\,\ddot{\theta}\cos\theta = \ddot{x}_{S/B} - \ddot{x}_{H/S} + s\,\dot{\theta}^2\sin\theta, \tag{4.3.9}$$

since, in our case, $x_{B/F}$ is fixed. Using equations (4.3.7), (4.3.8), and (4.3.9), we now simulate the rower's performance in one stroke, meaning that we focus on the evolution of several magnitudes from the catch time until the rower finishes the stroke.

As for the rower's movement under consideration and the hypothesis that have been used for the simulations, they are as follows:

1. The distance between the feet and the seat is constant: $x_{B/F} = 1$ m (the seat does not slide).

2. The rower is initially bent forward with his or her back at an angle of $2/3$ radians and his or her arms are completely stretched. Then the rower moves backwards until he or she lies at an angle also of $2/3$ radians, still keeping the arms stretched. The angle of the body is taken as a linear function of time. This movement takes one second.

3. After this second, keeping the back's inclination, the rower tucks his or her arms. This is modelled by using a linear function in time for $x_{H/S}$ that changes in 0.5 seconds from 0.75 metres to zero.

4. The length of the back is assumed to be $h_S = 0.6$ m.

As initial condition we assume that the boat is already moving at a velocity of 3 m/s, so we are actually simulating the effect of one stroke when the boat has been already accelerated, that is, after the competition has started.

With these hypotheses, we obtain an approximated F_{oar} from equation (4.3.8), and an approximated θ from equation (4.3.9). Then we use equation (4.3.7) with these approximations and obtain the corresponding rower's performance.

The results are shown in Figure 4.5. The upper pictures show that the velocity of the boat increases when the rower tucks his or her arms, which is due to the fact that the oar's force onto the water does also increase very much in this period, as can be seen in the bottom left picture. As for the evolution of the oar's angle (bottom right picture), we find that when the rower performs this movement the oar is moved slower the first second, but then it is moved faster before the stroke finishes. The strokes are very often performed this way, since this last pulling of the oar helps the rower to achieve the inertia to then stand up and start the next stroke. But the picture for the velocity (upper right) also shows that by doing this non-smooth movement the velocity at some point decreases, although at the end of the stroke a higher velocity is achieved. Therefore it would be interesting to model some other rower's performance that was smoother to see whether the boat achieved a higher velocity at the end of the stroke. In Section 4.4 we will see how this non-smooth movement affects the velocity of the boat.

This analysis does only consider the phase of the period where the oar is inside the water. A complete analysis should couple equation (4.3.7) with a model of the forces and velocities when the oar is outside of the water. This second analysis should take into account the rower's change in his or her body's acceleration.

4.4 The effect of the rower movement onto the *llagut* velocity

In this section we are concerned with the influence of the rower movement on the boat displacement. Thus, from now on we will imagine a *non-rowing rower*, that is, a rower without oars but only moving himself or herself in a certain given way. Moreover, we will think of the rower as a unique point with its centre of gravity moving on a line (forward and backwards).

Although it may seem strange, this simplification makes sense since we are concerned with a better understanding of how the recovery movement has to be done, which is one of the questions we were asked for. The recovery is the aerial phase of the stroke, when the oars are taken out from the water and the rower bends *forward* (opposite to the advance of the boat) in order to be ready for the next stroke. Since during this process the oars are not

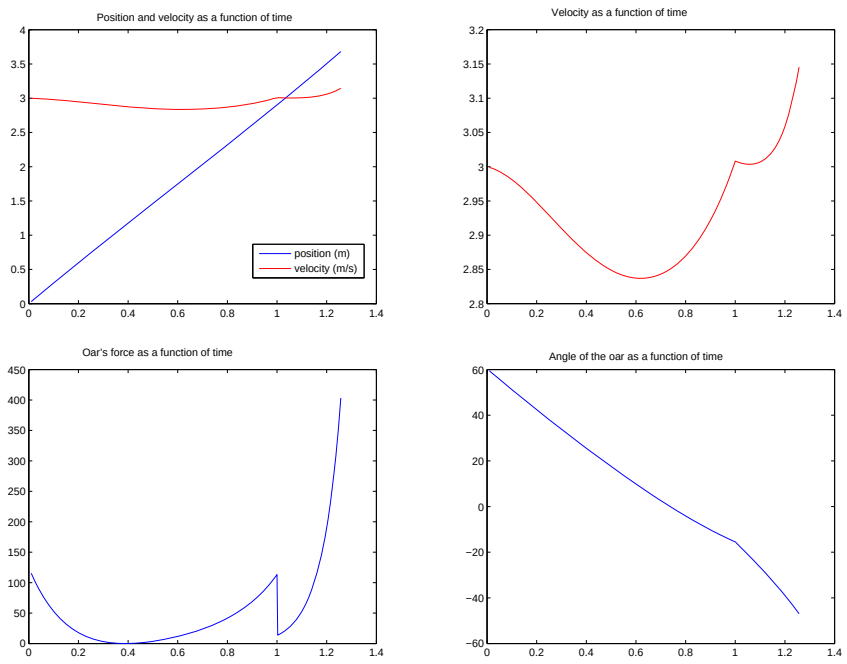


Figure 4.5: Upper pictures: (left) evolution of the velocity and position of the boat, (right) velocity of the boat. Bottom pictures: (left) oar force, (right) oar angle

making any force to progress, the rower has to do this movement as quick as possible. But doing it too quickly introduces an acceleration against the inertia of the boat. Consequently, we have to achieve a certain balance between both factors.

The relation between the rower's movement and the boat displacement is given by equation (4.4.3). We are going to explain this model, which also follows from the one given in [1] when the present assumptions are used.

If we denote by x_R the absolute position of the rower and by m_R his or her mass, the linear momentum balance for the rower gives

$$m_R \ddot{x}_R = -F \tag{4.4.1}$$

where F stands for the force that the rower is doing against the boat. Denoting by x_b the absolute position of the boat and by m_b its mass, the linear momentum balance for the boat gives

$$m_b \ddot{x}_b = F - C_1 \dot{x}_b |\dot{x}_b| \tag{4.4.2}$$

where C_1 is the boat drag coefficient. Observe that, since in this part we may consider forward and backward movement of the boat, and that since the drag force always acts against this, we have to consider $\dot{x}_b|\dot{x}_b|$ instead of $\dot{x}_b^2$ as in the previous section. Here we assume that the only forces acting on the boat are the reaction force from (4.4.1) and the drag force, which is typically taken in the form given in equation (4.4.2). Note that in both equations the oars do not play any role.

We now consider $x_{R/b}$ as the position of the rower relative to the boat,

$$x_{R/b} = x_R - x_b.$$

This new variable allows to combine equations (4.4.1) and (4.4.2) into the following one, that determines the motions of the boat when the motion of the rower is given:

$$m_b\ddot{x}_b = -C_1\dot{x}_b|\dot{x}_b| - m_R\ddot{x}_{R/b}. \tag{4.4.3}$$

Our aim is to check with some examples whether a too quick recovery movement has a negative effect, as it seems to be commonly accepted. Using equation (4.4.3), we now look at the distance covered by the boat for different kinds of rower actions. More specifically, we have computed this distance during a fixed interval of time for four kinds of rower actions.

(a) **Symmetrical catch and finish.** We consider a rower doing the recovery movement at the same velocity as the catch one, that is, moving forward and backwards at the same velocity. More particularly,

$$x_{R/b}(t) = \sin\left(\frac{2\pi t}{T}\right)$$

where T is given.

(b) **Slow catch, fast recovery.** This has been modelled by the function

$$x_{R/b}(t) = \sin\left(\frac{2\pi t}{T}\right) - 0.25\sin\left(\frac{4\pi t}{T}\right).$$

(c) **Fast catch, slow recovery.** This has been modelled by the function

$$x_{R/b}(t) = \sin\left(\frac{2\pi t}{T}\right) + 0.25\sin\left(\frac{4\pi t}{T}\right).$$

(d) **Still rower.** We are interested in comparing the displacement obtained with the fast/slow recovery movements with the one obtained when the rower makes no action, that is,

$$x_{R/b}(t) = 0.$$

We wonder if it could be even better (in terms of achieving a larger displacement of the boat) doing nothing rather than doing a bad movement.

Also, we have tried these four kinds of $x_{R/b}$ in two cases: a boat moving with a certain initial velocity and a boat initially at rest. The distances covered by the boat in each case can be seen in Figure 4.6. As the rower is facing the back part of the boat, the catch movement takes place when $\dot{x}_{R/b} > 0$, while the recovery part is when $\dot{x}_{R/b} < 0$. The slopes of $x_{R/b}$ in each part give the velocity at which each movement is done.

As we can observe in the graphics, there is no difference (qualitatively speaking) between the case of a boat with initial velocity and a boat initially at rest (apart from the last one, as it is obvious). The only difference is the total covered distance, as it should be expected.

However, the kind of rower action has a significative effect. We can see that in case (c) the boat goes forward all the time. However, in case (b) eventually moves backwards!

We remark the amazing fact that (c) eventually moves backwards is a consequence of the quadratic character of the friction term. If we assumed a linear damping term (that is, $m_b\ddot{x}_b = -C_1\dot{x}_b - m_R\ddot{x}_{R/b}$), we would observe no difference between the solutions, qualitatively speaking. In this case the difference would only be the total covered distance (see Figure 4.7).

After these examples, the nowadays accepted opinion among some fixed-rowing coaches that a quick recovery reduces the efficiency of the stroke seems to be correct. It even seems that a rowing technique following case (d) would be better than the one following (b). But in the real problem this recovery phase is combined with the catch one (which includes the oar forces and so the equation that models the complete rowing is not (4.4.3)). Moreover, in a more realistic approach we should take into account that the recovery has to be done with a certain velocity in order to start the next stroke with strength enough. This could be the subject of a future work. Anyway the results of this section are a strong evidence in favour of the fast catch-slow recovery strategy.

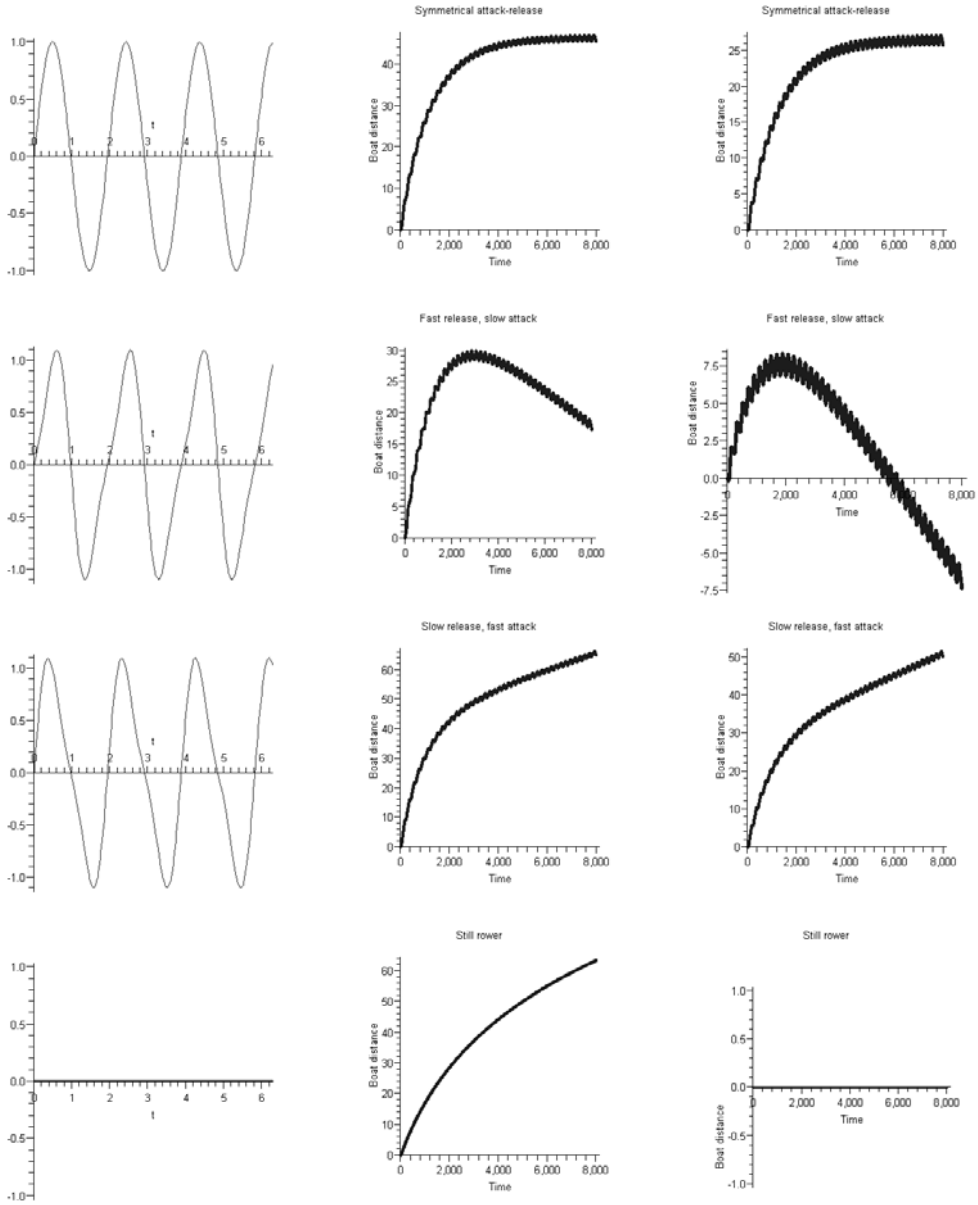


Figure 4.6: (Left) position of the rower's centre of gravity, (centre) displacement of the boat with initial velocity, (right) displacement of the boat initially in rest for the four types of rower movement

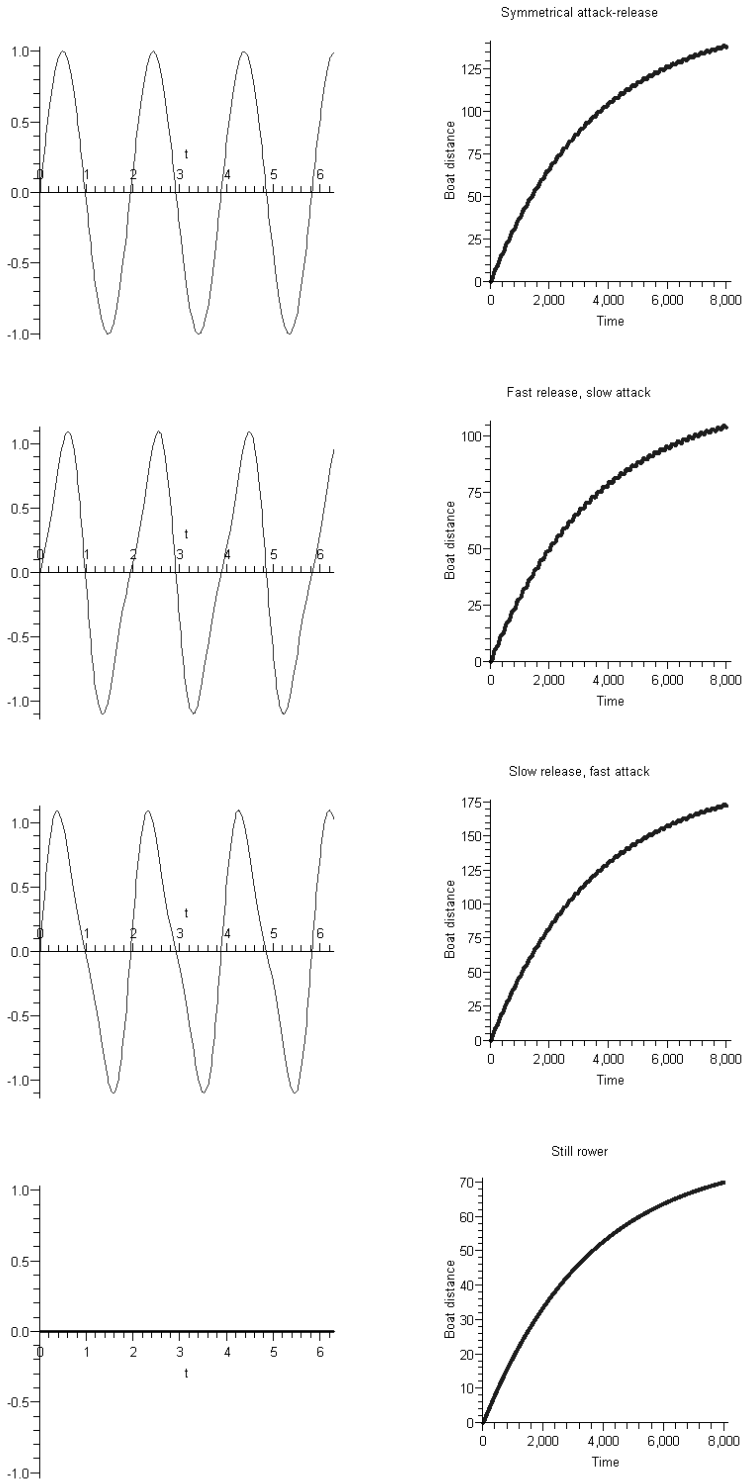


Figure 4.7: (Left) position of the rower's centre of gravity, (right) displacement of the boat with initial velocity for the four kinds of rower movement in the case of a linear damping

4.5 References

1. D. Cabrera, A. Ruina and V. Kleshnev, A simple 1^+ dimensional model of rowing mimics observed forces and motions, *Hum. Mov. Sci.* **25** (2006), 192–220.

2. N. Caplan and T. Gardner, A fluid dynamic investigation of the Big Blade and Macon oar blade designs in rowing propulsion, *J. Sports Sci.* **25 (6)** (2007), 643–650.

3. N. Caplan and T. Gardner, A mathematical model of the oar blade-water interaction in rowing, *J. Sports Sci.* **25 (9)** (2007), 1025–1034.

4. D. L. Pope, On the dynamics of men and boats and oars, *Mechanics and Sport, ASME* (1973), 113–130.

Fine prediction of energy power production in wind farms

by R. Alabern, J. A. Carrillo, J. Escalante, A. Ferreiro, J. Rosado, and A. Ruiz-de-Villa

5.1 Technological need: wind power prediction

Clean energy production systems and environmental policies have as a side effect a much more complicated management scenario for the electric market. The energy produced by wind farms cannot be stored and it has to be immediately absorbed by the electrical national network. Due to the natural oscillations of this source of energy, the network has to establish strict intervals of expected energy production to adapt itself the response of the overall electrical national network.

For instance, if the prediction is that wind will not blow strong enough one day, the carbon-feeded power-plants will have to be ready to start up to avoid any electric shortage or blackouts. This is the reason why wind farms must provide the electrical distributor with their power prediction with 14 hours in advance for each of the 24 hours starting at 0.00 of the $(D+1)$-day, or day to be predicted, numbered from 0.00 to 23.00, i.e., at least before 10.00 of the D-day or prediction day.

This prediction can be changed or tuned at most one hour before each of the intra-day sessions marked at 16.00 and 21.00 of the D-day. Moreover, the prediction can be changed or tuned for the remaining hours at 1.00, 4.00, 8.00 and 12.00 of the day of prediction. There is a tolerance of 20% for avoiding penalization for under/over-production.

In order to assess this prediction, we assume the following inputs:

- The Local Weather Forecast usually giving wind direction θ and speed u, pressure and temperature T in a coarse scale, typically of the order of 10 kilometers.

- The historic data of wind power produced by the specific wind farm during several years of management. This can be complemented by the local wind rose of the wind farm; see a typical wind rose in Figure 5.1 and the historical data of wind speed and direction. These data are collected by a single meteorological station located at the wind farm.

- The historic data of the velocity errors $\mathcal{E}$ committed by the weather forecast prediction with respect to the local measurements.

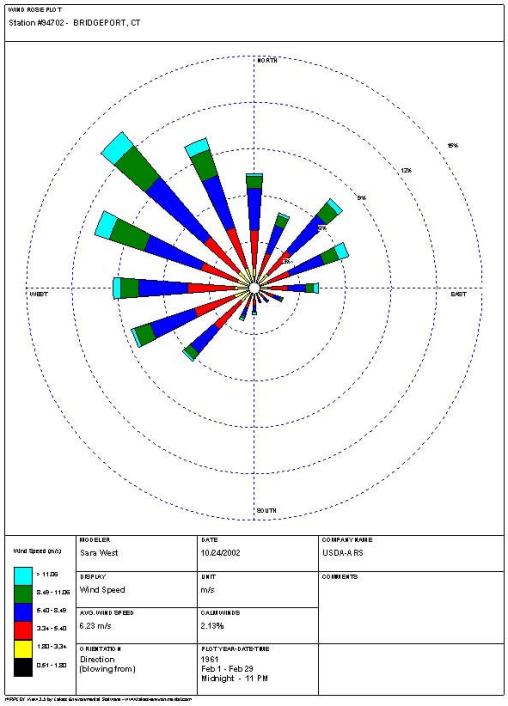


Figure 5.1: Wind rose in Bridgeport, Connecticut, for February 1961

The technological problem proposed by Ambio S.A. is how to use these inputs to obtain the best possible prediction of energy power generated by the wind farm. Another important issue is how to estimate the risk of penalization due to over/under-production.

This report is based on the discussions in the group formed by R. Alabern, J. A. Carrillo, J. Escalante, J. Ferré-Borrull, E. Ferrer, A. Ferreiro, F. Martínez, A. Pérez, J. Rosado, A. Ruiz-de-Villa, and F. Utzet.

5.2 Physical considerations and numerical models

The first important point to focus on is that the energy power obtained from a single wind turbine depends nonlinearly on the wind speed. Assuming the wind blades are optimally directed perpendicular to the wind direction, the energy power equation is

$$\text{Energy} = \frac{1}{2}\rho\, A\, u^3 \tag{5.2.1}$$

where ρ is the air density, A is the area of the rotor and u is the average wind speed through the rotor. This relation is called the Betz law of wind energy; see Figure 5.2. This nonlinear dependence implies that small changes in wind speed produces much larger differences in energy output. Moreover, the wind turbines have a minimum wind speed to operate around 3–4 m/s and they saturate their response between 15 and 20 m/s depending on wind turbine models. Moreover, if the wind speed is larger that 25 m/s they stop operating to avoid structural damages.

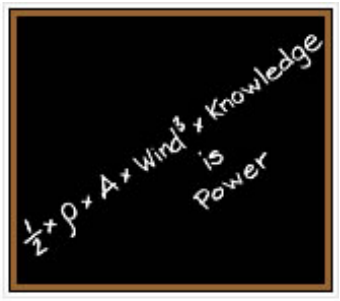


Figure 5.2: Expression of Betz' law by the Danish Wind Industry Association

The local weather forecast service provides the results of pressure, temperature, wind speed and direction for each of the points of a mesh of size 10 kilometers. This mesh is very coarse for being able to capture local wind effects, orography, roughness or any local characteristic of the wind farm.

One possibility is then to downscale the numerical solution of the fluid mechanics equations to the wind farm scale. In order to do that, there are some possibilities depending on the availability of computational resources:

- To perform a numerical interpolation to induce values of the wind speed and direction at the points where the windmills are situated based on

the prediction at the points given by the local weather forecast and the values obtained at the local meteorological weather station of the wind farm.

- Use some highly sophisticated engineering software for computational fluid mechanics such as FLUENT [1], to incorporate local roughness and landscape, turbulent effects, wind wake effects and the overall behavior of the wind farm using the data of the weather forecast as boundary conditions.

The first option is quite inaccurate and then surely not reliable due to the coarseness of the mesh and the sparseness of the interpolating points. Nonetheless, it can give a trend of the expected values. The second option is not realizable in real time due to its heavy numerical cost but it can be done off-line for certain typical conditions of wind in the local wind farm or for a certain number of "normal conditions". For instance, a wind farm situated at the Ebro river will have many days in which the wind blows from northwest (*cierzo*). Off-line simulation results in these specific conditions will surely be quite accurate and can be used inside neural networks; see Section 5.4.

Due to these considerations, we propose as possible answer to this technological need a statistical model which could include as additional information some of the prediction outputs obtained from these numerical computations based on physical models. Let us sketch our strategy depending on the availability of the different inputs:

1. To perform a fine data analysis of the historical measurements of the total energy production of the wind farm in order to cluster them in terms of wind directions and seasons or other relevant parameters, see next section. Assuming no other input is given, then our solution is based on a direct evaluation of these clustering curves at the weather prediction.

2. To find a distribution fitting the historical errors committed by the prediction in the past; see next section. Based on the error distribution, we propose the following procedure:

    2.1. By a Montecarlo type algorithm, we use the clustering curves and error distribution to find expected values for the power production at all needed times. Moreover, we can estimate the probability of penalization; see Subsection 5.3.1.

    2.2. Independently, we can adjust a time series to improve the short-time prediction, i.e., the intra-day sessions; see Subsection 5.3.2.

2.3. Make use of the extreme value theory to fit the tails of the wind speed probability density function to improve the prediction of penalty risk for unusual violent wind speed conditions; see Subsection 5.3.3.

3. Neural networks can be used to tune final predictions based on the different above possibilities; see Section 5.4.

5.3 Data analysis

The main purpose at this stage is to obtain the power curve associated to a wind farm. Figure 5.3 shows its typical energy output.

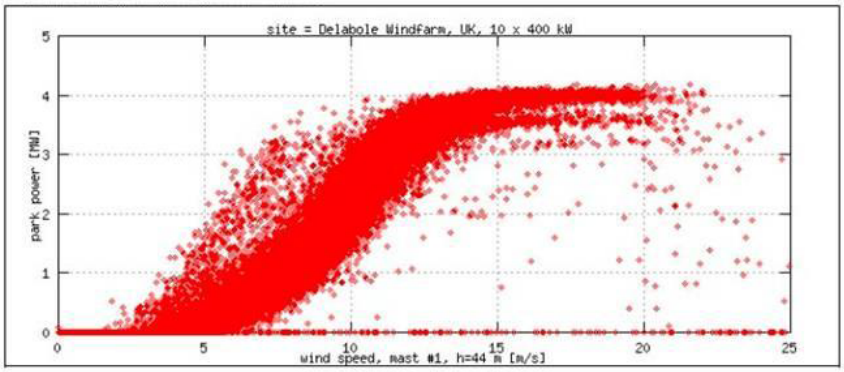


Figure 5.3: Power production in a wind farm

Based on its S-shape, increasing character and the saturation phenomena, it is reasonable to adjust a curve to this cloud of points with one of these two possibilities:

1. Using cubic splines: this method allows us to do a very good interpolation of the data and it is not difficult to code it up.

2. Using the logistic/sigmoidal function:

$$\frac{P_{\max}}{1 + Ce^{-ku}}$$

where $P_{\max}$ is the saturation energy power. This curve is typically used to model the population growth in an environment with limited resources, sales of a product where the total number of sales is limited or patient's response to medication. Least-squares method can be used to estimate the parameters of the sigmoid function. This is usually contained in any commercial software package.

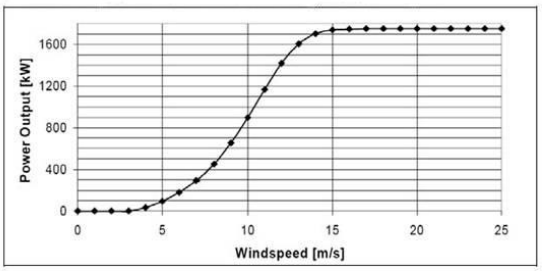


Figure 5.4: Typical shape of the resulting power curve

In Figure 5.3, we observe a certain number of values outside the overall trend which should be removed before adjusting the curve. It is then necessary to do a first study of outliers.

Actually, the curves should be adjusted after a clustering of the data has been performed. For the clustering, it seems natural to consider one of the next two options:

1. The wind direction θ and the temperature T.

2. The wind direction θ and the seasons.

The temperature is important because of air's density. When the air is cooler, it is denser so it produces more energy; see (5.2.1). The difference between the two options is that the first one gives us a better partition of the data. However, the result using this one does not need to be better than the second one, which is actually easier, because we can do the partition in only three blocks (Winter, Spring–Autumn, Summer).

To simplify the exposition, let us take the second option for which we propose a not-hierarchical common algorithm for the wind direction. This one assumes that we already have k clusters and then it distributes the data in a way that minimize the residual variance. The algorithm proposed is called the H-averages and the way to find the first k clusters is using the wind rose, Figure 5.1.

There is usually a predominant wind direction for which we assume not having outliers, so it is natural to think that we have two angles that concentrate all the information about the direction of the wind for this wind farm. Then the directions of the wind are angles of the interval (θ_0, θ_k).

Let us consider $\{(\theta_{i-1}, \theta_i)\}_1^k$, with

$$\theta_i = \theta_0 + i\frac{\theta_k - \theta_0}{k},$$

as a partition of this interval to obtain the first k clusters. Then we take the centroid (average value) c_k^0 of each cluster and call it the initial seed. The next step is to assign the values to each of the centroids in a way that minimizes the residual variance. Once we have all the values in a new sequence of clusters, we take the new centroids c_k^1 and consider them as new seeds. We repeat iteratively this process. It is necessary to give a stopping criteria, e.g., we can limit the number of iterations.

In the data analysis, we can also fit a probability distribution to the historic data of the prediction errors $\mathcal{E}$ with respect to the park measurements. For instance, in [3] it was assumed that such errors for the wind velocity are normally distributed componentwise, thus $\mathcal{E} \sim \mathcal{N}$.

5.3.1 Montecarlo

By sampling the error distribution of the velocity $\mathcal{E}$ obtained in the data analysis and taking into account the wind velocity from the weather forecast, we get a sampling of the expected wind direction and speed. Now, we compute the estimate of the power output by reading the corresponding value from the adjusted curve to the corresponding cluster and wind speed. Moreover, from this sampling we can give the probability of paying a penalization.

These ideas can be used also to plan the contract policy. Wind farm owners may agree in different contracts and rules in the electric market which are usually quite complex. Contracts will have a gain function of the form $\text{Gains}_a(u, \theta)$, where Gains_a is a explicit function for a given market agreement denoted by a. Here, we assume that this function depends only on wind speed and direction.

We can calculate gain expectation for the market agreements given the actual weather forecast. Again, by sampling the error distribution, we will get possible values for the wind velocity $(u_1, \theta_1), (u_2, \theta_2), \ldots, (u_n, \theta_n)$. Then the expected value of the gain is

$$\mathbb{E}\left[\text{Gains}_a(u, \theta)\right] \approx \frac{1}{n} \sum_{i=1}^{n} \text{Gains}_a(u_i, \theta_i),$$

and taking large statistics in order to get a stable value of gains for each possible contract, we can decide the best choice for our objectives using market rules and their complexity.

5.3.2 Short-term prediction

Another important issue is to incorporate the new weather information obtained along the day to improve the predictions in the intra-day sessions.

Moreover, we can see how different the weather forecast information is behaving from the real local data. We want to use this information to correct our prediction.

To estimate this error, we propose to use time series techniques because their computational cost is very low. These techniques are based on statistical models, hence they need the previous statistical study in Section 5.2. The idea is that the errors between the physical prediction and the real data until a given moment will explain (and let us correct) the prediction of the desired time horizon. For instance, if we denote by Y_t the real wind data, by Y_t^p the physical prediction for the moment t and by $e_t = Y_t - Y_t^p$ the error, for a time horizon t_T and in a given moment $t_k < t_T$ we are looking for a functional equation like

$$e_{t_T} = f(e_{t_k}, e_{t_{k-1}}, \dots),$$

or equivalently

$$Y_{t_T} = Y_{t_T}^p + f(e_{t_k}, \dots, e_{t_{k-m}}),$$

using as lags $m \geq 0$ as we need. Usually

$$f(e_{t_k}, \dots, e_{t_{k-m}}) = \beta + \alpha_0 e_{t_k} + \cdots + \alpha_m e_{t_{k-m}} + z_{t_k},$$

where β and α_j, $0 \leq j \leq m$, are parameters which need to be estimated using statistical tools, and z_{t_k} is another error with a certain probability distribution. There are different functional equations. To know which one fits better, it is necessary to use historical data. Once we know the functional equation and we have estimated its parameters, the model is ready to be used. In the future, the only maintenance needed is to review with some frequency the estimated parameters.

5.3.3 Extreme values

As it can be seen in Figure 5.5, the shape of the wind speed-power curve shows maximum efficiency for high wind speed. But just a little range of high speed is desirable since an extreme speed will cause the stopping of the machine for security reasons. The problem arises in estimating the power supplies when extreme wind speed has been forecasted. For a regular speed, a continuous range of possible power production is given depending on several considerations regarding the actual wind farm. On the other hand, when critical values of wind speed are forecasted, the possible supplies of power can vary between maximum efficiency and no production at all. Often, the consequence of a bad prediction is a penalty from the government.

We assume that the areas which are chosen to set a wind farm have rare extreme wind speed, otherwise the farm would be closed most of the

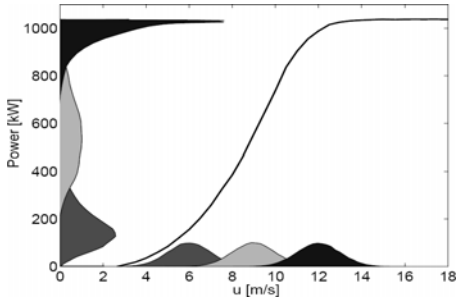


Figure 5.5: Generalized wind speed-power curve and associated pdfs

time. Therefore, the statistical study of this rare events arise as a natural solution. In this direction, many authors agree that the distribution of wind speed follows a Weibull function, regarding the facts that the power curve is zero with no wind and the tail of the pdf. The Weibull function is defined as

$$\underline{G}(x) = \begin{cases} 1 - \exp\left[-\left(\frac{x-\lambda}{\delta}\right)^{\beta}\right] & \text{if } x \geq \lambda \\ 0 & \text{otherwise,} \end{cases}$$

where λ, δ and β are the constants known as location, scale, and shape parameters, respectively, and such that $\delta > 0$ and $\beta > 0$. In Figure 5.6 we have plotted several Weibull distributions. Extreme value theory is a branch

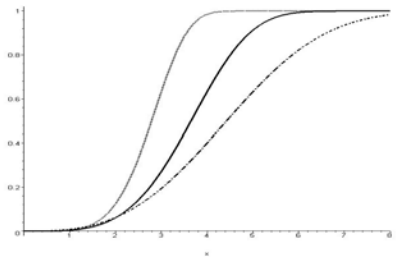


Figure 5.6: Weibull distribution for different scale and shape parameters

of statistics born fifty years ago and used to study the probability of large storms causing substantial flooding along coasts with areas under sea level such as Holland. Thus, it has been deeply studied and applied with success into other areas such as the study of records; hence many existing results might prove be applicable in wind farm studies.

From our point of view, an effort to study the possibilities of including extreme value theory in the model of power prediction of wind farms would be of considerable importance.

5.4 Neural networks

Neural networks (NN) have shown themselves as a powerful tool when people have to deal with an overcoming amount of data for which the way it has to be taken into account is not evident.

The key words to understand NN are architecture and learning. A NN consists of a set of *cells* where an activation function is implemented, which are distributed in several groups (or layers) and interconnected. Some of them get an input value, process it, and send the result to the cells they are connected to, which get it modified by a weight added by the connection link and do the same, till the last set of cells produce the output.

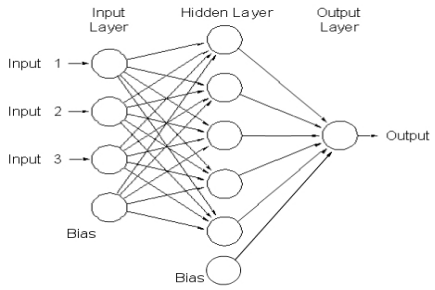


Figure 5.7: Three-layer network with bias

The number of cells in the NN and the way they are separated in layers and interconnected, which determines the flow of the data, is what we call *architecture* of the NN; the process by which the weights added by the links is fixed is known as *learning process*. For a better understanding of NN, the reader can refer, for instance, to [2, 5].

As an example of how NN can be used in prediction, in [4] the authors present a three-layer neural network with four entries, which correspond to velocity and direction of the wind registered in each of the meteorologic stations of the park and one exit corresponding to the power production, for each one of the mills. With this, they achieve short term predictions which improve the results obtained by other ways.

Due to our restrictions, we cannot offer here an appropriate architecture for the particular case of our study, but some general ideas which have to be tested.

The three-layer model is optimal in a forward propagation architecture with linear activation functions in cells belonging to the intermediate layers, so it would be a good starting point. For the input layer we have to decide which data we want to use (the forecast prediction, the values registered in

our station, the results of numerical simulations, a combination of both...), and for the output layer, a sigmoidal function adjusts naturally the power curve of the mills, thus a cell with an activation function of this kind for every mill should be tried. Studying separately each mill shall simplify the learning process and do the model more adaptable; summing the different outputs, we will get a prediction for the total production of the park.

5.5 References

1. FLUENT Flow Modeling Software, ANSYS, Inc., `www.fluent.com`.

2. S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall (1999).

3. M. Lange, *Analysis of the Uncertainty of Wind Power Predictions*, Energy & Meteo Systems, Wuppertal (2003).

4. S. Li, D. C. Wunsch, E. A. O'Hair, M. G. Giesselmann, Using neural networks to estimate wind turbine power generation, *IEEE Transactions on Evolutionary Computation* **16** (2001).

5. K. Mehrotra, C. K. Mohan, S. Ranka, *Elements of Artificial Neural Networks*, MIT (1997).

6. Vindmølleindustrien, Danish Wind Industry Association, June–October 2007, `www.windpower.org`.

Multiple fault diagnosis of rain sensors for the sewer net in Barcelona

by Aureli Alabert, José Ignacio Burgos, and Jordi Saludes

6.1 Introduction

The sewer network of Barcelona uses a set of 23 rain sensors for collecting real-time data of rain intensity. By merging this information with the data collected from the level sensors in pipes, the central control system manages the opening/closing of sewer gates to avoid flooding, while trying simultaneously to maximize flow to the water processing plants.

Rain sensors are liable to fail in intense rain conditions, which is precisely when they are most needed. To make things worse, more than one sensor uses to fail in these episodes. Therefore, a procedure to detect and isolate faults in the rain sensor net is of capital importance, as it is also the ability to reconfigure the net after a fail. Moreover, due to the nature of the failures it is not possible to model the errors of the sensors as a random Gaussian distribution.

A fault-tolerant net is usually built by exploiting the redundancy of the data: Since near sensors tend to deliver correlated data, a way to detect faulty sensors is to provide a set of equations binding readings from close sensors, that the system must fulfil (up to a prescribed admissible error) at any time.

This paper introduces three different approaches to the problem. They can be read and implemented independently. In Section 6.2, we describe the method of robust estimation to detect outliers in a random sample; this method may provide an alternative approach to detect and isolate faulty sensors. Section 6.3 deals with the optimal design of the set of models that help in the detection of the failure of a sensor; this approach can be applied with the initial set of sensors as well as with the subset of working sensors once one or some of them have been discarded. Section 6.4 is devoted to infer models from the geometry of the net.

The conclusions presented here come from discussions held during the GEMT meeting by A. Alabert, J. I. Burgos, O. Farràs, S. Oliva, C. Padró, X. Muñoz, J. C. Naranjo, A. Rodríguez, J. Saludes, and J. Solà-Morales.

6.2 Robust fitting

When modeling the 23 rain sensors of the sewer network of Barcelona, we are interested in detecting faulty sensors. Faulty sensors are sensors that provide a measurement that is uncorrelated with the real data it should provide. Typical examples are obstructed sensors that always report zero rain. Therefore we cannot model such errors as random Gaussian noise in the measurements. The usual least square methods for adjusting a model to a sample are based on the assumption of Gaussian errors and, therefore, are not applicable in this case. In this section we will review the problems that the least square method has in the presence of outliers and how robust estimates can overcome these difficulties.

Given a collection of data $\{x_1, \ldots, x_N\}$, we will make the following assumptions:

1. Most of the data are reasonably correct. This means, for instance, that the data correspond to the phenomenon we are studying plus a random error that can be modeled with a Gaussian distribution. But there is the possibility that some of the data have a significant error due to malfunction of the measure device. These data will be called outliers and clearly cannot be modeled using a Gaussian distribution.

2. There is a good model for the phenomenon we are studying. For instance, there is a parametric model that depends only on a few number of parameters, that can be adjusted to the data. Moreover we have a knowledge of the statistics of this model. In particular we have some knowledge of the distribution of the discrepancy between the model and the correct data.

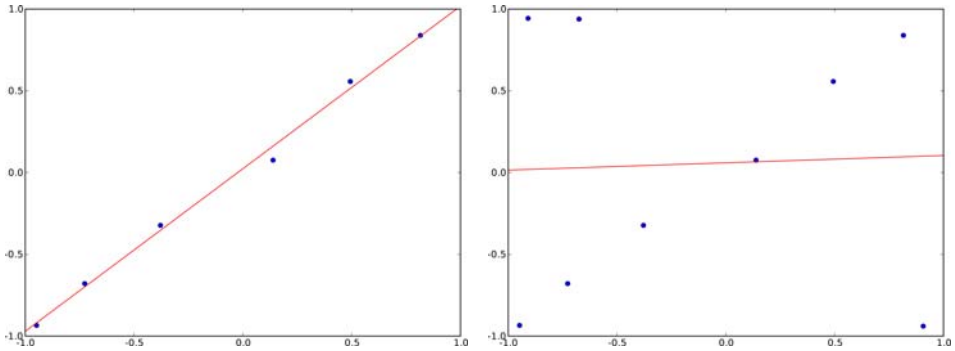
With these assumptions we seek a method to detect the presence of outliers and, what is also important, identify them. Since we are assuming the existence of a good parametric model, we can design a strategy for detecting outliers as follows:

1. First we adjust the model to the data.

2. Using the knowledge of the model statistics, we can design a test that tells us if the data are consistent with the model. Therefore we can detect the presence of outliers.

3. We can now compare the data with the results predicted by the model. In order to identify the outliers, we are now tempted to determine that the outliers are those data that show bigger differences between the experimental result and the predicted result.

Note however that, if the method we use to adjust the model is not robust enough, the presence of even a small number of outliers can ruin completely the fitting of the model, making it unusable to identify the outliers. As an example, we represent in Figure 6.1 the result of adjusting a line to (a) a set points that follow a line plus some random Gaussian error, and (b) the same set with three outliers.



(a) Linear regression without outliers (b) Linear regression with outliers

Figure 6.1: Linear regression

We can see that the outliers have a very strong influence on the result, ruining completely the adjusting process. The reason of the strong influence of the outliers in the result is precisely the fact that we are minimizing the square of the error. Thus the points with a big error have an even bigger weight in the minimizing process. Recall that the least square method is justified by the assumption of Gaussian errors, and the Gaussian distribution decays with the exponential of minus the square of the error, thus making very unlikely the presence of points with errors much higher than the standard deviation. If this assumption is fulfilled then the least squares method produces the best result. Nevertheless, if the errors do not follow a Gaussian distribution, the least squares method may not be the optimal choice.

In the last decades, a lot of effort has been devoted to the design of statistic estimators that are tolerant to the presence of outliers. These kind of estimators are generically known as robust estimators.

A classical example of a robust estimator is the median. Assume that we have a sample $\{y_1, \ldots, y_n\}$ of a random variable Y that can be written as

$$Y = \mu + X,$$

where μ is a parameter and X is a random variable with zero mean. If X follows a Gaussian distribution, then the maximum likelihood estimate for the parameter μ is the arithmetic mean of the sample. By contrast, if n is odd and X follows a Laplace distribution with probability density

$$f(X) = \frac{1}{2b} \exp\left(-\frac{|X|}{b}\right),$$

then the maximum likelihood estimate is the median of the sample. It is very easy to see with an example that the presence of an outlier has a very strong influence on the arithmetic mean, whereas it has a very small influence on the median. The median is therefore a robust estimator. In the same way as we can define the arithmetic mean as the estimator M that minimizes

$$\sum_{i=1}^{n} (y_i - M)^2,$$

when n is odd we can define the median m as the estimator that minimizes

$$\sum_{i=1}^{n} |y_i - m|.$$

This consideration leads us to one of the main methods to design robust estimators: instead of minimizing the sum of the squares of the errors, we minimize the sum of the absolute values. However, there is a problem minimizing the sum of absolute values of the errors, because the resulting function is not differentiable, making it difficult to find the minimum. To solve this problem we *regularize* the function to minimize. To this end we choose a small regularizing parameter $\epsilon > 0$ and define the minimization function as

$$F = \sum_i \sqrt{(e_i^2 + \epsilon^2)},$$

where the sum runs over all the points in the sample, and e_i is the error of the point i. Observe that

$$\sqrt{(e_i^2 + \epsilon^2)} \sim \begin{cases} \epsilon + \dfrac{e_i^2}{2\epsilon} & \text{if } |e_i| \ll \epsilon, \\ |e_i| & \text{if } \epsilon \ll |e_i|. \end{cases}$$

Therefore, for small errors we have the good differentiability properties of the x^2 function, whereas for big errors we have only linear growth, that provides us with the required robustness. For each value of ϵ we obtain a different estimator. Then the robust estimator is the limit when ϵ goes to zero.

We will now develop with some detail a method to solve an overdetermined system of linear equations that uses these ideas of robust estimators. We consider the equation

$$Ax = b,$$

where A is an $n \times m$ matrix with $n > m$ and rank m, while b is a column vector of dimension n and x is the vector of unknowns, of dimension m. We denote by a_i^t the i-th row of the matrix A. The above equation is equivalent to the system of equations

$$a_i^t x - b_i = 0,\ i = 1, \ldots, n,$$

that has more equations than unknowns. We seek an approximate solution of this system of equations. To this end, for some $\epsilon > 0$, we minimize the function

$$F = \sum_{i=1}^{n} \sqrt{(a_i^t x - b_i)^2 + \epsilon^2}.$$

The optimal solution will be obtained by equating to zero the gradient of F with respect to X. Let $D(X)$ be the diagonal matrix whose i-th entry is

$$D_{i,i} = \frac{1}{\sqrt{(a_i^t x - b_i)^2 + \epsilon^2}}.$$

Then the solution of the minimization process is the solution of the equation

$$A^t D(X) A X = A^t D(X) b.$$

This nonlinear equation can be solved iteratively as follows. Let X^0 be any starting solution. For instance, we can choose $X^0 = 0$. We let X^1 be the solution of the system of linear equations

$$A^t D(X^0) A X = A^t D(X^0) b.$$

Then the solution at the n-th iteration will be the solution of the system of linear equations

$$A^t D(X^{n-1}) A X = A^t D(X^{n-1}) b.$$

It is possible to show that this iterative process is always convergent.

Recall that the problem of adjusting a line to a set of points is a particular case of an overdetermined system of equations. Given a set of points $\{(x_i, y_i)\}_{i=1,\ldots,n}$ we seek the best solution to the system of equations

$$x_i a + b = y_i,\ i = 1, \ldots, n,$$

where now the unknowns are a and b.

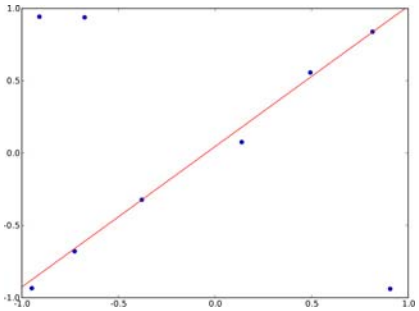


Figure 6.2: Robust regression with outliers

Figure 6.2 represents the result of applying the robust regression method to the set of points containing three outliers. As we can see, the presence of outliers has almost no effect on the result of the regression allowing us to clearly identify the outliers.

In order to apply the above considerations to the rain sensors of the sewer system of Barcelona we need first a good parametric model of the rain in Barcelona with few parameters. The Mediterranean rain is very difficult to model due to its chaotic nature, although averages on time should be susceptible to be modeled with a bi-quadratic model. Therefore we suggest to use historical data to test whether such a model can be used to detect faulty sensors. Note however that the discussion in this section is just a glimpse on the use of robust estimators for the detection of outliers. We recommend to consult the vast literature on the subject before starting a study in this direction.

6.3 Combinatorial optimization

At present, the detection of a faulty sensor is carried out by checking its reading against a predicted outcome. The prediction is produced by a linear model involving three other sensors which are known to be highly correlated with the first one.

The relation between sensors and models of sensors can be summarized in an incidence table. For our purposes, this table will be defined formally as a matrix $X = (x_{ij})$, with

$$x_{ij} = \begin{cases} 1, \text{if sensor } j \text{ appears in the model of sensor } i, \\ 0, \text{otherwise.} \end{cases}$$

Our goal is to obtain automatically an incidence matrix for a given set of sensors, with the essential requirements that:

- The columns of the matrix must be different, so that in case of detecting significant deviations in some models, the responsible sensor can be immediately spotted.

- The correlations among sensors in a model must be as high as possible.

Such an automatic procedure could be used both in the construction of the initial matrix of 23 sensors (off-line computation) and in the construction of new matrices with less sensors (on-line computations), once the failure of a sensor has been asserted, in order to be able to immediately detect new failures. When applied to this on-line setting, the viability of the method will depend on the necessary computation time, which should be checked experimentally and will depend in turn on the machine in which the algorithm runs.

6.3.1 How to create an incidence matrix

Assume we have n sensors. Consider variables x_{ij} with $1 \leq i \leq n$, $1 \leq j \leq n$, defined as above. We have to decide on the value of these n^2 variables x_{ij}.

We impose the following conditions:

1. Each sensor has to be present in its own model. Therefore, we can fix $x_{ii} = 1$ for $1 \leq i \leq n$, and there are only $n^2 - n$ variables left to decide upon.

2. In addition to their own sensor, three more sensors must be present in each model. This can be expressed by saying that the sum of each row of the matrix must be exactly equal to four, and therefore translates into the n conditions

$$\sum_{j=1}^{n} x_{ij} = 4, \quad 1 \leq i \leq n.$$

3. We want all columns to be different. This amounts to say that, when subtracting any two columns j and k, the sum of the squares of the difference column must be greater or equal to 1:

$$\sum_{i=1}^{n} (x_{ij} - x_{ik})^2 \geq 1, \quad 1 \leq j < k \leq n.$$

We have here $n^2/2 - n/2$ conditions.

On the other hand, from all possible incidence matrices satisfying these constraints, we are interested in the one with the highest possible correlation among the sensors taking part in the same model. There are different possibilities to consider, but the simplest one is just requiring that the sum of all correlations among the chosen sensors for all models be the greatest possible. If we denote by $C(i,j)$ the correlation (estimated from historical data) among the sensors i and j, we would like to maximise, by choosing convenient values 0 or 1 of the decision variables x_{ij}, the quantity

$$\sum_{i,j=1}^{n} C(i,j)x_{ij}.$$

Indeed, we are adding here the correlations between sensors i and j only if sensor j appears in the model of sensor i, that is, if $x_{ij} = 1$. If there are significative negative correlations between sensors, then it is better to replace $C(i,j)$ by $|C(i,j)|$, but this is not important in the following discussion. We thus arrive at an optimization problem that can be formulated as:

$$\max \sum_{i,j=1}^{n} C(i,j)x_{ij}$$

subject to:

$$\begin{aligned}
&x_{ii} = 1, \quad 1 \le i \le n,\\
&\sum_{j=1}^{n} x_{ij} = 4, \quad 1 \le i \le n,\\
&\sum_{i=1}^{n} (x_{ij} - x_{ik})^2 \ge 1, \quad 1 \le j < k \le n,\\
&x_{ij} \in \{0,1\}, \quad 1 \le i \le n.
\end{aligned}$$

This is a *binary programming* (also called 0-1 *programming*) problem with a linear objective function and nonlinear constraints.

Conceptually, the simplest way to tackle this problem is to *linearize* it, that is, to transform the problem into a linear one, in the manner explained below, and apply a standard method for solving linear problems with binary variables. For instance, the so-called *branch and bound* method performs well for binary problems, and the algorithm can be found in most optimization books (see, for instance, [4]). The drawback of linearizing is the introduction of many auxiliary 0-1 variables, and also of many new restrictions, so that the dimension of the resulting mathematical programming problem is even higher.

A second way is to study the nonlinear problem directly, exploiting its particular structure. It seems indeed feasible to devise an algorithm with a better performance than linearization plus branch and bound. In case the latter reveals as impractically slow, this possibility deserves closer attention.

6.3.2 Linearization of the optimization problem

The third set of constraints above, where the nonlinearities appear in the form of quadratic functions, are equivalent to

$$2\sum_{i=1}^{n} x_{ij}x_{ik} + 1 \leq \sum_{i=1}^{n}(x_{ij} + x_{ik}), \quad 1 \leq j < k \leq n,$$

taking into account that $x_{ij} = x_{ij}^2$.

We still have the cross products $x_{ij}x_{ik}$, but they can be linearized with the introduction of a new family of decision variables α_{ijk} defined as

$$\alpha_{ijk} = x_{ij}x_{ik}, \quad 1 \leq i \leq n,\ 1 \leq j < k \leq n.$$

Observing again that all variables take values in $\{0, 1\}$, it can be easily seen that for each i and j, the last equality is equivalent to the system of linear inequalities

$$\begin{cases} x_{ij} + x_{ik} \geq 2\alpha_{ijk} \\ x_{ij} + x_{ik} - \alpha_{ijk} \leq 1. \end{cases}$$

Therefore, the nonlinear constraints can be replaced by the set of linear constraints

$$\begin{cases} 2\sum_{i=1}^{n} \alpha_{ij} + 1 \leq \sum_{i=1}^{n}(x_{ij} + x_{ik}), \quad 1 \leq j < k \leq n, \\ x_{ij} + x_{ik} \geq 2\alpha_{ijk}, \quad 1 \leq i \leq n,\ 1 \leq j < k \leq n, \\ x_{ij} + x_{ik} - \alpha_{ijk} \leq 1, \quad 1 \leq i \leq n,\ 1 \leq j < k \leq n. \end{cases}$$

We arrive at a linear binary programming problem with $n^3/2+n^2/2-n$ decision variables and $n^3 - n^2/2 + n/2$ constraints. In the case $n = 23$, this means 6,325 variables and 11,914 constraints.

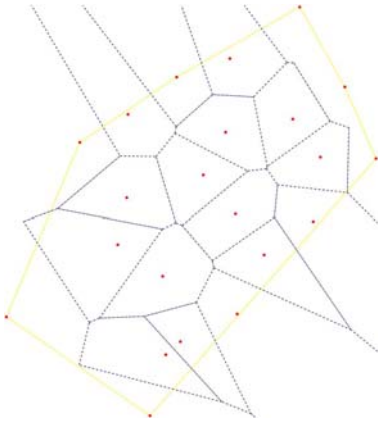


Figure 6.3: Voronoi cells (blue) for the nowadays sensor net (red) inside the convex hull (yellow)

6.4 Triangulations

Before considering the problem of finding faulty sensors, we must consider how we estimate rain intensity at a given point using the readings from the sensor net. In the following R will be the set of the 22 sensors, and for $r \in R$ we will denote by $r(t)$ the reading of rain intensity by r at time t. (The geographical coordinates of one of the sensors were not known at the time of writing this report.)

6.4.1 Piecewise constant rain intensity

Let us first assume the rain intensity is piecewise constant. More precisely, we will assume that rain intensity is constant in a circular patch. There can be more that one patch and they can be moving in time.

With no more information, the best guess we can do on rain intensity at a given point comes from the reading of the nearest sensor. To assign each point to its nearest neighbor, we can use the *Voronoi diagram* $\mathcal{V}(R)$ (see [1, 2]) which gives a partition of the plane consisting of polygonal regions. The Voronoi cell for a sensor r contains the points which are closer to r than to any other $r' \in R$.

Some of these regions are unbounded (this is the case for the ones corresponding to sensors in the convex hull of the net). It is clear that we cannot estimate rain for points far away from the net. In the following we will assume that we are interested only on points inside the convex hull.

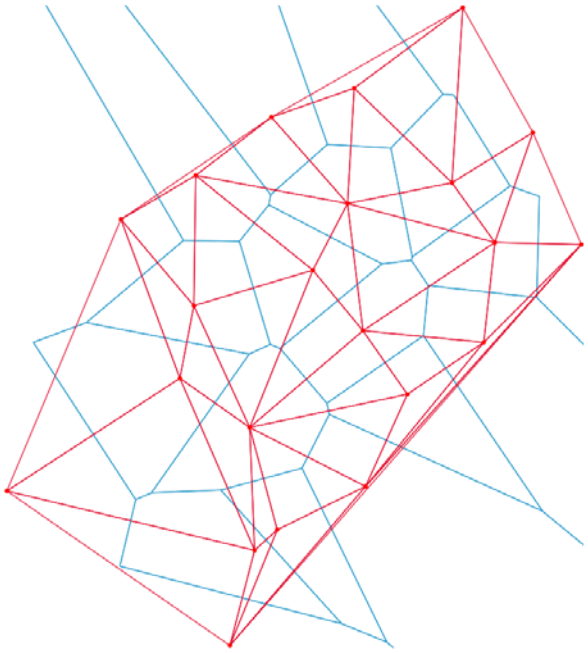


Figure 6.4: Delaunay triangulation for the actual net

6.4.2 Delaunay triangulation

The dual of the *Voronoi diagram* $\mathcal{V}(R)$ is the *Delaunay triangulation* $\mathcal{D}(R)$ (see [1, 2]). Both the Voronoi diagram and the Delaunay triangulation consist of polygons (cells and triangles), delimited by edges and vertices. Being dual means that inside each Voronoi cell there is a vertex of a Delaunay triangle, and that each Voronoi vertex corresponds to a triangle in the Delaunay triangulation. The Delaunay triangulation has remarkable geometrical properties that will help at improving our detection models. For instance:

1. $\{r, s\}$ is an edge of $\mathcal{D}(R)$ if and only if there is a circle incident with r and s not containing any other site of R.

2. The circumscribed circle of a triangle in $\mathcal{D}(R)$ does not contain any other site.

3. The *maximal enclosing radius* of a triangulation T is the maximal radius of circumscribed circles of triangles of T. Then $\mathcal{D}(R)$ minimizes the maximal *enclosing radius* among all triangulations of R; see [1, 2].

6.4.3 Fault detection and isolation

We will describe a set of models for the sensor net. A model is a system of equations, in terms of the sensors readings, that we expect to be satisfied provided that all the sensors that appear in the model are working well. Let us call *Delaunay graph* the pair (R, E), where R is the set of nodes and E the set of edges of triangles in $\mathcal{D}(R)$.

The simplest of all models able to do fault detection consists on comparing two sensors $r, s \in R$, that is: In absence of faults, we state $r(t) = s(t)$ as null hypothesis. As soon as the difference between $r(t)$ and $s(t)$ is large enough to reject this null hypothesis, we know that there is a fault on r or s. We identify the model $r(t) = s(t)$ with the edge $\{r, s\} \in E$ of the Delaunay graph. Thus, this class of models corresponds to subgraphs of the Delaunay graph.

At this point, it seems natural to bind sensors which are near, for one expects them to have similar readings. This is where Delaunay triangulation comes to help: Suppose we are wondering if a sensor r is working fine. Any model $r(t) = s(t)$ will help to decide it in the positive sense if r, s fall in the same rain patch. However, it follows from Property 1 above that if $\{r, s\}$ is not an edge, then each circle containing r and s also contains another $u \in R$. Thus, any rain patch covering r and s also covers u and the information about the goodness of r could also be derived from the model $u(t) = r(t)$. Put in another way: Models relating Delaunay edges can assert performance for rain in cases while any other combination could not.

Increasing the degree of the model graph

A minimal set of models will be a *matching* for the Delaunay graph, i.e., a one to one correspondence between two halves of R. This is fine if we were interested only on *detecting* a fault, but we want also to know which sensor is at fault (namely *isolating* the fault). So if we were to be able to discriminate the faulty one, we will need more than one model for each sensor.

Let us consider these alternatives ordered by node degree:

Matching. Each sensor appears in exactly one model. Acceptable only for detecting faults.

Union of cycles. Each sensor appears in exactly two models. The resulting graph is the disjoint union of cycles. A set of models in this class consists on taking a subset T of the triangles of $\mathcal{D}(R)$ in such a way that, for all $r \in R$, there is exactly one triangle $\Delta_r \in T$ having r as vertex. Then take the models given by the edges of triangles of T.

Higher degree sets. Other configurations with more models per node are also possible. A decision procedure for this kind can be as follows. Given $r \in R$, consider the partition of the neighbors of r, $N_r = H_r \cup I_r$, where $s \in H_r$ if $\{s, r\} \in E$ and the corresponding model holds and $u \in I_r$ if $\{s, r\} \in E$ and the corresponding model fails. If there is a circle C such that $\{r\} \cup H_r \subset C$ but $I_r \cap C = \emptyset$, then r is not faulty.

6.4.4 Piecewise linear rain intensity

A more sophisticated model for rain intensity comes from assuming that it is locally described by a linear function of the position,

$$I(x, y; t) = A(t)x + B(t)y + C(t). \tag{6.4.1}$$

Since the function depends on three parameters, we will need to sample at least three points. Thus, to estimate the rain intensity at a point we will need the closest three sensors to this point. Given a set of three sensors $\Delta = \{r_1, r_2, r_3\} \subset R$, the linear estimation of $I(t)$ at a point p will be

$$\hat{I}(p; t) = \sum_{i=1}^{3} \alpha_i r_i(t), \tag{6.4.2}$$

where $(\alpha_1, \alpha_2, \alpha_3)$ are the barycentric coordinates of p respect to Δ, that is the triple of real values fulfilling

$$p = \sum_{i=1}^{3} \alpha_i r_i; \quad \sum_{i=1}^{3} \alpha_i = 1.$$

Again we want to use triangles with near vertices to minimize the possibility of being only partially covered by a rain patch. For this, we look once more at $\mathcal{D}(R)$, since Property 3 above ensures that triangles $\Delta \in \mathcal{D}(R)$ have the smallest enclosing circle. Therefore, in the following we will consider only configurations derived from Delaunay triangulations.

Models on quadrilaterals

Since three readings are necessary to estimate a local linear function as (6.4.1), it is clear that we will need at least one more sensor to build a model suitable for fault detection. This can be done by joining two adjacent triangles $\Delta_1, \Delta_2 \in \mathcal{D}(R)$ with associated estimating functions $\hat{I}_1$ and $\hat{I}_2$ as

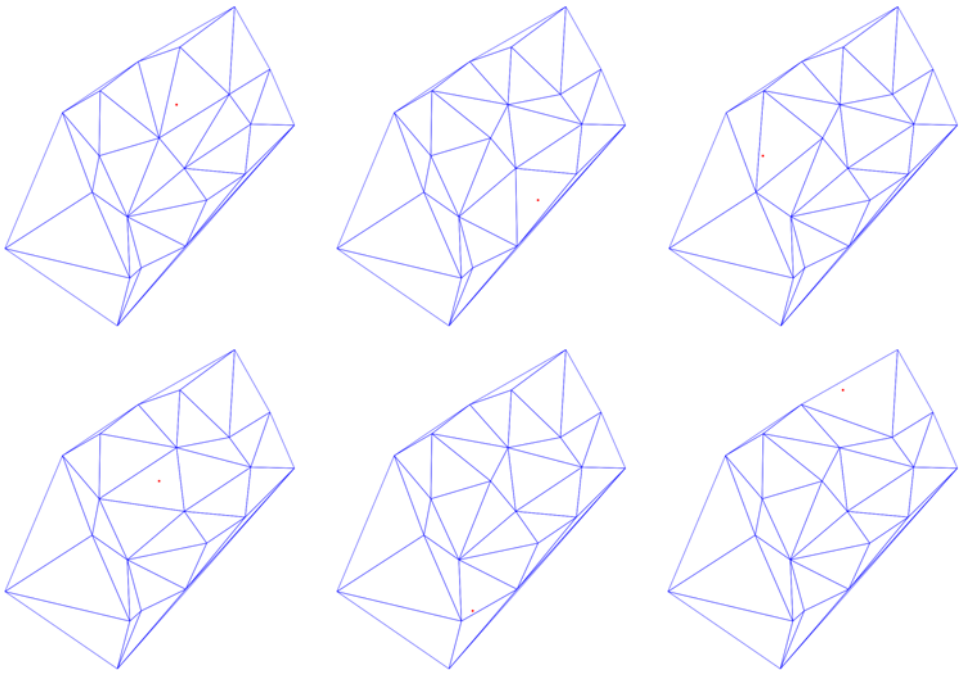


Figure 6.5: Estimating triangles for some sensors in the net

in (6.4.2), to form a quadrilateral. The validity of the join model can be checked by evaluating

$$|\hat{I}_1(r_2;t) - r_2(t)| + |\hat{I}_2(r_1;t) - r_1(t)|,$$

where r_1, r_2 are the vertices of the quadrilateral not in the common diagonal, namely $r_2 \in \Delta_2 \setminus \Delta_1$ and $r_1 \in \Delta_1 \setminus \Delta_2$. Using the aforementioned duality between Voronoi diagrams and Delaunay triangulations, the problem reduces to compute a *matching* for the graph of the vertices and edges of the Voronoi diagram.

Coarsening the net

Another approach to detect and isolate faulting sensors works by comparing the readings of each sensor $r \in R$ against a piecewise linear estimation given by the rest of the net $R \setminus \{r\}$. More precisely, for each $r \in R$, we compute the triangle $\Delta_r \in \mathcal{D}(R \setminus \{r\})$ that contains r and compare $r(t)$ with $\hat{I}_r(r;t)$, where $\hat{I}_r$ is the linear estimation corresponding to Δ_r as in (6.4.1). Figure 6.5 shows the reduced triangulation for some sensors.

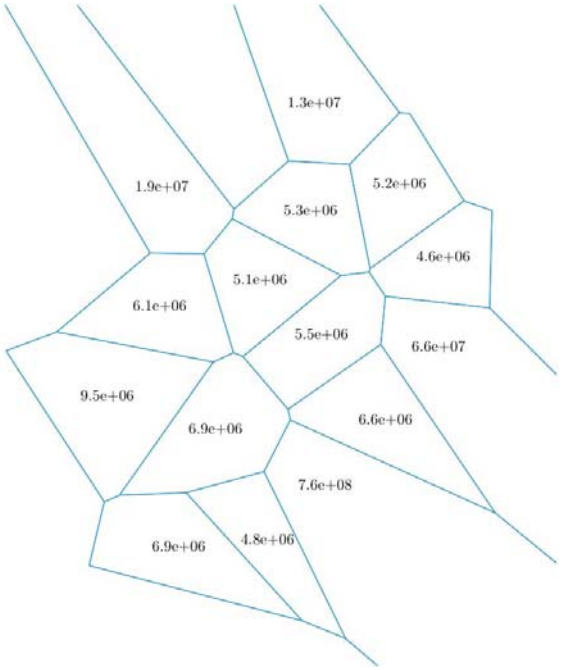


Figure 6.6: Areas for each Voronoi cell

6.4.5 Assessment of the geometry of the actual net

To finish, we will show some of the deficiencies of the sensor net regarding the actual geometry and propose some actions to correct them.

Voronoi diagram and Delaunay triangulation

Notice that some of the sensors are located very near the convex hull (see Figure 6.3), and therefore have extremely large cells. (See the areas in Figure 6.6.)

Improving the sensor net geometry

As mentioned above, the Delaunay triangulation is the dual graph of the Voronoi diagram: Each region corresponds to a vertex of the triangulation (a sensor position) and each triangle corresponds to a Voronoi vertex. In particular, edges of Voronoi are dual of Delaunay edges and the lines supporting those cut orthogonally the ones supporting these. This does not mean that the actual edges cut, nor that the nearest sensor to a given point must be a vertex of the Delaunay triangle containing it. In this case the triangulation will be a *Pitteway triangulation* [3]. It is clear that in the present state the

sensor does not admit such a triangulation, for each triangle edge should cut the corresponding edge of the Voronoi dual and this is not the case, as we can see from the figure. It is advisable to rearrange the net to become Pitteway.

Also, to make the cells more even, there is the Lloyd's algorithm [5]. In short, this algorithm moves the sites of the Voronoi diagram towards the centroids of the corresponding Voronoi cell. Although this algorithm tries to move all sites simultaneously, economic considerations may, in practice, force us to move only some.

Golden sensors

Completing the net with extremely reliable sensors (*golden sensors*) will be an alternative to improving the net geometry. Optimal position for these sensors should be further investigated.

6.5 References

1. Ding-Zhu Du, Frank Hwang (eds.), *Computing in Euclidean Geometry*, Lecture Notes Series on Computing **4**, World Scientific, 1995.

2. Jacob E. Goodman, Joseph O'Rourke (eds.), *Handbook of Discrete and Computational Geometry*, CRC Press, 1997.

3. M. L. V. Pitteway, *Computer Graphics Research in an Academic Environment*, Datafair 1973.

4. H. M. Salkin, K. Mathur, *Foundations of Integer Programming*, North-Holland, 1989.

5. Stuart P. Lloyd, Least squares quantization in PCM, *IEEE Transactions on Information Theory* **28**, no. 2 (1982), 129–137.

Participants

Maria Aguareles (MA1, UPC)

Roc Alabern Palau (DM, UAB)

Aureli Alabert (DM, UAB)

Xavier Alameda i Pineda (CEBSIAM, UPC)

José Ignacio Burgos Gil (DAiG, UB)

José Antonio Carrillo de la Plata (DM, UAB i ICREA)

Carles Comas (AMBIO S.A.)

Joan Escalante Esteve (DM, UAB)

Oriol Farràs (MA4, UPC)

Josep Ferré-Borrull (DEEEA, URV)

Albert Ferreiro Castilla (DM, UAB)

Edgar Ferrer Serrano (DM, UAB)

Enric Fossas (IOC, UPC)

Jaume Franch Bullich (MA4, UPC)

Toni Guillamon (MA1, UPC)

Fernando Martínez (MA2, UPC)

Albert Mas Blesa (DM, UAB)

Àlex de Mingo (Federació Catalana de Rem)

Margarida Mitjana Riera (MA1, UPC)

Xavier Mora (DM, UAB)

Xavier Muñoz (MA4, UPC)

Joan Carles Naranjo del Val (DAiG, UB)

Miquel Noguera Batlle (MA2, UPC)

Sergio Oliva (IME, Univ. Sao Paulo)

Carles Padró (MA4, UPC)

Marta Pellicer Sabadí (DIMA, UdG)

Agustí Pérez Foguet (MA3, UPC)

Vicenç Puig (ESAII, UPC)

Peregrina Quintela (IMAT, Univ. Santiago de Compostela)

Antonio Rodríguez-Ferran (MA3, UPC)

Jesús Rosado Linares (DM, UAB)

Aleix Ruiz de Villa Robert (DM, UAB)

Noemí Ruiz Munzón (DM, UAB)

Jordi Saludes (MA2, UPC)

Joan Solà-Morales (MA1, UPC)

Frederic Utzet (DM, UAB)

Mauricio Wagensberg (AMBIO S.A.)

Sebastià Xambó (MA2, UPC)

Contents